VIDEO SERVICES FORUM

Preamble to Video Services Forum (VSF) Technical Recommendation TR-06-4 Part 3

June 20, 2023

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. TR-06-4 Part 1 (Source Adaptation, published in 2022) and TR-06-4 Part 2 (Use of Wireguard VPN in RIST Devices) are part of this series. This document is TR-06-4 Part 3, RIST Relay. The RIST Relay addresses the issue of firewall traversal for RIST devices. The RIST Relay provides a connection service to RIST devices using RIST Advanced Profile. RIST devices connect to the RIST Relay as a rendezvous point, and the RIST Relay either facilitates a direct connection between the RIST devices or relays the traffic between them. The RIST Relay described in this Specification is also capable of supporting group operation (many-to-many communication).

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 3

Reliable Internet Stream Transport (RIST) RIST Relay

Approved June 20, 2023

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

In many common applications, RIST devices are installed behind firewalls. This is typically not an issue for the RIST device starting the communication (the client), as most firewalls allow outgoing traffic from the internal network to the Internet. However, at the receiving site (the server), the firewall needs to be explicitly configured to allow the incoming traffic, a process that may be cumbersome, both technically and administratively. A possible solution to this problem is to have a device on the Internet to facilitate this communication. This Technical Recommendation defines the operational protocol for such a device - the RIST Relay.

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 new or existing standards and recommendations.

RIST devices are typically installed behind firewalls. This is usually not an issue for the device initiating the communication since firewalls are designed to allow outgoing traffic. However, explicit firewall configuration (opening of ports) is required for the device that is being contacted, to allow the incoming RIST traffic to be delivered. This explicit firewall configuration, in some cases, can be cumbersome (or even impossible) for both technical and administrative reasons. A solution to this issue is to have an accessible server on the Internet to facilitate communication, or even relay it from one device to the other. This Technical Recommendation defines the operational protocol for this server – the RIST Relay (RR). The protocol is an extension of RIST Advanced Profile, TR-06-3.

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)
Prinyar Boon (PHABRIX Ltd)	Eric Fankhauser (Evertz)	Ronald Fellman (QVidium)
Michael Firth (Nevion)	Oded Gants (Zixi)	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-06-2:2022, Reliable Internet Stream Transport (RIST) Protocol Specification – Main Profile

VSF TR-06-3:2022, Reliable Internet Stream Transport (RIST) Protocol Specification – Advanced Profile

VSF TR-06-4 Part 2:2023, Use of Wireguard VPN in RIST Devices

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

4 RIST Relay Operation

4.1 Definitions

For the purposes of this Specification, the following definitions apply:

- **RIST Relay (RR):** A server implementing the protocol defined in this Specification, to support communication and connectivity between RIST Endpoints.
- **Endpoint:** A RIST Relay client. An Endpoint uses the RIST Relay to facilitate communication with other Endpoints.
- **Group:** A set of Endpoints engaged in many-to-many communication using the RIST Relay. Group operation is defined in Section 4.3.
- **Bypass Connection:** A direct connection between two Endpoints, facilitated by the RIST Relay.
- **Relay Connection:** A connection between Endpoints whereby the RR relays the packets.

4.2 General Architecture

The general system architecture is shown in Figure 1. The Endpoints shown in the figure may be behind firewalls or directly connected to the Internet and are able to establish connections with the RIST Relay.

Endpoints shall connect to the RR using one of the following methods:

1. A DTLS connection with certificate-based authentication as per TR-06-3 Section 7.2.
2. A DTLS connection with TLS-SRP as per TR-06-3 Section 7.3.
3. A PSK connection, compliant with TR-06-3 Section 8.
   NOTE: Endpoint authentication is provided implicitly by shared knowledge of the passphrase, which has security limitations. Use of this mode is discouraged except for

cases where there are other security measures in place. In the general case, EAP authentication is recommended.

4. A PSK connection with EAP SHA256-SRP6a Authentication, compliant with TR-06-2 Annex D.
5. A Wireguard connection, compliant with TR-06-4 Part 2.

A RIST Relay compliant with this Specification shall support at least one of the above methods. The RIST Relay may support multiple connection methods. If an Endpoint does not have a common connection protocol with a given RR, it will not be able to connect to that RR. Selection of the connection protocol at the Endpoint is outside the scope of this Specification.

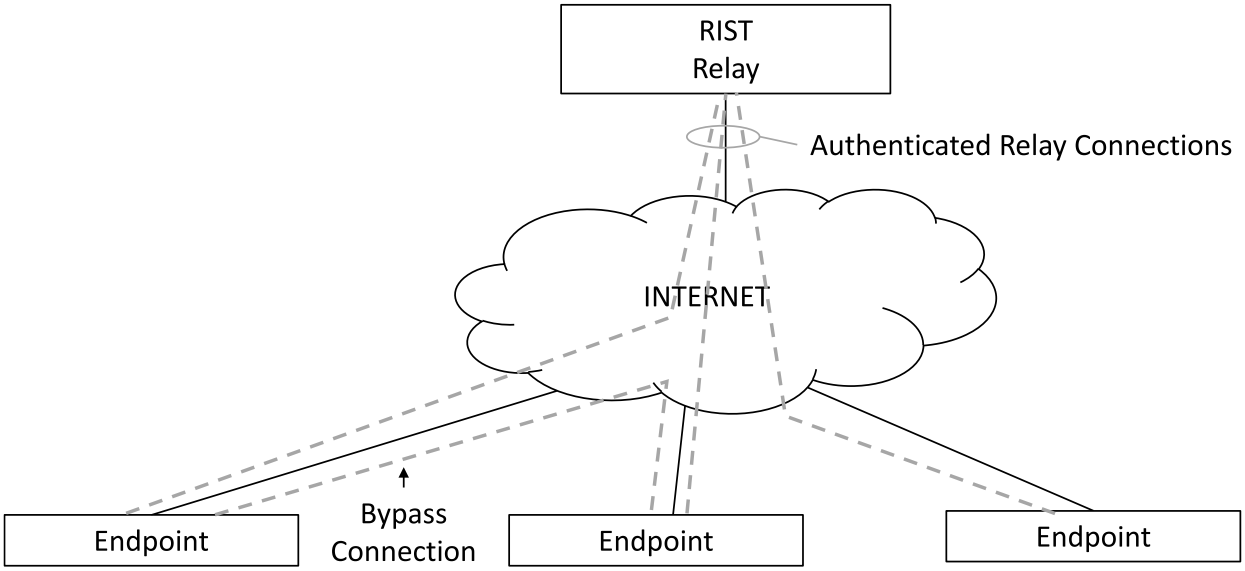

Figure 1: General Architecture

Once connected, the Endpoints shall use RIST Advanced Profile packets as per TR-06-3 Section 5 to communicate with the RIST Relay and with each other.

Endpoints shall use a different SSRC every time they connect to a RIST Relay. This includes successive connections to the same RR.

Each Endpoint shall have a human-readable name, managed by the RR. When the Endpoint connects to the RR, it is identified by its authentication data. Within a given RR, Endpoint names shall be unique, but multiple authentication methods may be associated with a given Endpoint name. As part of the connection process, the RR shall communicate the assigned name to the Endpoint. When an Endpoint requests a connection to another Endpoint, the requested Endpoint shall be identified by its name. Creation and maintenance of the name database in the RR is outside the scope of this Specification.

If an Endpoint has connections to multiple RRs, it shall be prepared to receive either the same or a different name from each RR.

4.3 Group Operation

The RR shall support the concept of Groups. A group is also identified by a human-readable name. Within a given RR, Group names shall be unique. The RR shall provide the following services to a group:

- Every Advanced Profile packet sent to the group shall be replicated to all Endpoints currently connected to the group, except for the one that transmitted the packet. This includes both Tunnel packets (TR-06-3 Section 5.2) and Control packets (TR-06-3 Section 5.3).
- The RR may restrict which Endpoints are allowed to transmit Tunnel packets (TR-06-3 Section 5.2) to the group.

Endpoints connecting to groups where multiple senders are transmitting shall support SSRC multiplexing, as described in TR-06-3 Section 6. In such a case, there is a small probability that that two sending devices use the same SSRC, resulting in an SSRC collision. In such a situation, the RR shall take one of the following actions:

1. Transparently remap the incoming SSRC from one of the senders to resolve the conflict. or
2. Disconnect one of the conflicting senders. As indicated in Section 4.1, if the Endpoint elects to reconnect, it shall do so with a different SSRC.

The mechanisms used to create groups and manage which Endpoint(s) are allowed to transmit tunnel packets to the group are outside the scope of this Specification.

4.4 Multiple Connections Using the Same Credentials

The RR may support multiple connections using the same credentials. It is irrelevant to the RR if these are multiple connections from the same device, or if there are multiple devices using the same credentials. The use cases for this scenario are:

- An Endpoint has multiple independent network connections. It can establish a connection to the RR through each of its network connections for reliability.
- Resource sharing: having multiple devices with the same name. For example, a bank of receivers at a TV station or studio.

If there are multiple idle connections to the RR using the same credentials, and an Endpoint requests a connection to that specific name, the RR shall choose one of the connections at its discretion. In particular, the RR may use the Protocol Support Flags described in Section 5.2.2 and Figure 7 to select a compatible Endpoint.

4.5 Using Bypass or Relay Connections (Informative)

When a device behind a firewall starts sending packets towards a destination on the Internet, the first packet establishes state in the firewall. This state allows response packets to reach the originating device. Firewalls can be generally classified into two types based on the amount of information contained in the state (in addition to the device's internal address):

- **Asymmetric Firewalls** only retain internal/external port information. Any incoming packets, regardless of source IP address, are delivered to the device. Most consumer-grade firewalls are asymmetric.
- **Symmetric Firewalls** retain the destination IP address information in addition to the internal/external port information. Only incoming packets from the original destination are forwarded to the device. Such firewalls provide a higher level of security.

Two devices behind asymmetric firewalls can communicate directly (using UDP) if aided by a server on the Internet, as follows:

1. Both devices start sending UDP packets towards the server on the Internet.
2. The server records the source IP address and source UDP port of these packets from both devices.
3. The server informs the device who desires to start the connection (client device) the source IP address and source UDP port of the other device (server device).
4. The client device starts sending packets to the specified IP address and UDP port.
5. The firewall at the server device side forwards the packets since they match the external UDP port in the state.
6. The server device responds to these UDP packets, and the firewall at the client device side forwards them as well.
7. The server on the Internet is no longer involved in communication.

For the purposes of this Specification, such direct connections are denoted as **Bypass Connections**. Bypass Connections are not available if the server device side is behind a symmetric firewall. The solution, in these cases, is to have the traffic relayed by the server on the Internet. The RIST Relay defined by this Specification supports both Bypass and Relay operation.

One possible security issue with a Bypass connection is Endpoint-to-Endpoint authentication. The client Endpoint is told to directly connect to a server Endpoint. That is a direct connection, facilitated by the RR, but it does not use the RR authentication mechanism. In such a case, the server Endpoint provides authentication data to the RR, and the RR relays this authentication data to the client Endpoint when it is told to connect. This allows the client Endpoint to authenticate the server. This authentication data is the Connection Data described in Section 5.2.2.4 and relayed to the client Endpoint as indicated in Section 5.2.5.2.

5 RIST Relay Protocol

5.1 Protocol Message Encapsulation

RR Protocol messages shall be implemented as TR-06-3 Tunnel Control Messages as per Section 5.3 and Figure 7 of that document. These messages shall be sent using the protected SSRC. Table 1 shows the extensions to TR-06-3 control messages defined in this document.

Table 1: Updated Control Index Values

Control Index	**Message Type**	**Mandatory**
0x0000	NACK Bitmask	
0x0001	NACK Range	
0x0002-0x000F	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	RR Connection Initiation Message	Yes
0x8031	RR Request Denied Response Message	Yes
0x8032	RR Directory List Response Message	RR only
0x8033	RR Connection Request Response Message	Yes
0x8034	RR Connection Incoming Message	Yes
0x8035-0x803F	Reserved for RR-to-Endpoint Control Messages	
0x8040	Endpoint Connection Initiation Response Message	
0x8041	Endpoint Connection Incoming Response Message	Yes
0x8042-0x804F	Reserved for Endpoint-to-RR Control Messages	Yes
0x8050-0xF7FF	Reserved for future control messages	
0xF800-0xFFFF	Reserved for private vendor use	

5.2 Protocol Message Definitions

In all the messages defined in this Section, if a field is marked **Reserved**, it shall be set to zero by the message sender and shall be ignored by the message receiver. All messages are displayed in network byte order.

5.2.1 RR Connection Initiation Message

Direction: RR to Endpoint.
Description: RR shall send this message as soon as the Endpoint authenticates and shall repeat it once per second until a response is received from the Endpoint, or the Endpoint disconnects. This message shall use one of the following formats:

- *RR is ready, your Endpoint name is specified in the Additional Data field*. The Endpoint shall respond to this message with the Endpoint Connection Initiation Response Message described in Section 5.2.2.
- *RR is busy, connect later*. Upon receipt of this message, the Endpoint shall disconnect from the RR and attempt a connection at some later time. The amount of time to wait until the reconnection attempt is left at the discretion of the implementer.
- *Redirect to another RR identified by an IP address and port, which are specified in the Additional Data field*. Upon receipt of this message, the Endpoint shall disconnect from the RR and may connect to the RR identified in the message.
- *Redirect to another RR identified by a hostname, which is specified in the Additional Data Field*. Upon receipt of this message, the Endpoint shall disconnect from the RR and may connect to the RR identified in the message.

The RR Connection Initiation message format is shown in Figure 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Control Index = 0x8030    |            Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Result Code  |  Reserved   |T|       Incoming UDP Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                      Incoming IP Address                       :
:                                                                :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
:                       Additional Data                          :
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: RR Connection Initiation Request Message

The RR shall set the fields in Figure 2 as follows:

- **Result Code (8 bits):** This field shall indicate the connection result, as follows:
  - 0x00: *RR Ready*
  - 0x01: *RR Busy, connect later*
  - 0x02: *Redirect to another RR, specified by IP address*

- 0x03: *Redirect to another RR, specified by hostname*

- **T (1 bit):** This bit shall indicate the IP address type, as follows:
  - T=0: IPv4
  - T=1: IPv6
- **Incoming UDP Port (16 bits):** This field shall report the Endpoint's incoming UDP port as seen by the RR.
- **Incoming IP Address (32 bits if T=0 or 128 bits if T=1):** This field shall report the Endpoint's incoming IP address as seen by the RR.
- **Additional Data (variable):** the size and format of the additional data depends on the Result Code, as described below.

5.2.1.1 Additional Data for Result Code 0x0000 (RR Ready)

The Additional Data format for the RR Ready message is depicted in Figure 3.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Name Length  |                                               |
+-+-+-+-+-+-+-+-+            Endpoint Name                      :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Additional Data Field for RR Ready

The RR shall set the fields in Figure 3 as follows:

- **Name Length (8 bits):** This field shall be set to the size, in bytes of the Endpoint Name field.
- **Endpoint Name (variable):** This field shall be set to the Endpoint name. This field may be null-terminated and shall be 4-byte aligned. If necessary for alignment, this field shall be padded with up to 3 bytes set to zero. Endpoint names shall use UTF-8 formatting.

5.2.1.2 Additional Data for Result Code 0x0001 (RR Busy)

This message has no additional data.

5.2.1.3 Additional Data for Result Code 0x0002 (Redirect by IP Address)

The Additional Data format for the Redirect by IP Address message is depicted in Figure 4.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved          |T|       Redirect UDP Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                      Redirect IP Address                      :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Additional Data Field for Redirect by IP Address

The RR shall set the fields in Figure 4 as follows:

- **T (1 bit):** This bit shall indicate the IP address type, as follows:
  - T=0: IPv4
  - T=1: IPv6
- **Redirect UDP Port (16 bits):** This field shall be set to the UDP port of the next RR to be contacted.
- **Redirect IP Address (32 bits if T=0 or 128 bits if T=1):** This field shall be set to the IP address of the next RR to be contacted.

5.2.1.4 Additional Data for Result Code 0x0003 (Redirect by Host Name)

The Additional Data format for the Redirect by IP Address message is depicted 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Redirect UDP Port       |   Redirect Hostname Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       Redirect Hostname                       :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Additional Data Field for Redirect by Host Name

The RR shall set the fields in Figure 5 as follows:

- **Redirect UDP Port (16 bits):** This field shall be set to the UDP port of the next RR to be contacted.
- **Redirect Hostname Length (16 bits):** This field shall be set to the size of the Redirect Hostname field, in bytes.
- **Redirect Hostname (variable):** This field shall be set to the hostname of the next RR to be contacted. This field may be null-terminated and shall be 4-byte aligned. If necessary for alignment, it shall be padded with up to 3 bytes set to zero. The hostname field shall contain a string suitable for a DNS name query.

5.2.2 Endpoint Connection Initiation Response Message

Direction: Endpoint to RR.

Description: The Endpoint shall send this message to the RR in response to the RR Connection Initiation Message. This message has one of the following formats:

- *Request the list of available Endpoints*. The RR shall respond to this request either with the RR Directory List Response described in Section 5.2.4 or the RR Request Denied Response Message described in Section 5.2.3.
- *Connect to an Endpoint specified by its name*. The RR shall process this message as follows:
    - If the Endpoint is not allowed to communicate with the requested Endpoint, the RR shall respond with the RR Request Denied Response Message described in Section 5.2.3. The policy by which the RR determines whether to allow the connection is outside the scope of this Specification.
    - The RR may request that the Endpoint connect to another RR. In this situation, the RR shall respond with the RR Connection Request Response Message described in Section 5.2.5 with **Response Code** set to 0x02 (Redirect by IP Address) or to 0x03 (Redirect by Hostname).
    - If the Endpoint is allowed to communicate with the requested Endpoint, the RR shall send the RR Connection Incoming Message described in section 5.2.6 to the requested Endpoint. The requested Endpoint shall respond with the Endpoint Connection Incoming Response Message described in Section 5.2.7.
        - If the **Response Code** in the Connection Incoming Response Message is set to 0x00 (Connection Accepted), the RR shall respond to the requesting Endpoint with the RR Connection Request Response Message described in Section 5.2.5. This process is illustrated in Figure 21.
        - If the **Response Code** in the Connection Incoming Response Message is set to 0x01 (Connection Refused), the RR shall respond to the requesting Endpoint with the RR Request Denied Response message described in Section 5.2.3. The RR should set the **Reason Code** to 0x0005 (Connection Refused). This process is illustrated in Figure 22.
- *Endpoint is not ready to communicate yet (connected but idle)*. No response from the RR is required for this message.
- *Endpoint is ready to accept connections*. This message may include optional connection data, described below. No response from the RR is required for this message.

After the initial response, the Endpoint may send this message again at any time in the future. For example, an Endpoint may connect and indicate that it is not ready to communicate, and at some time later it may indicate that it is ready to accept connections.

The RR Connection Initiation Response message 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 = 0x8040    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Request Code |           Protocol Support Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Additional Data                        :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: RR Connection Initiation Response Message

The Endpoint shall set the fields in Figure 6 as follows:

- **Request Code (8 bits):** This field shall identify the type of request, as follows:
  - 0x00: Directory Request
  - 0x01: Connection Request
  - 0x02: Not ready (stay idle)
  - 0x03: Ready to Connect
- **Protocol Support Flags (24 bits):** This field shall identify the protocols supported by the Endpoint, as indicated by the highlighted part of Figure 7. Reserved bits shall be set to zero by the Endpoint and shall be ignored by the RR. The remaining bits shall be set by the Endpoint 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Request Code |B|V|P|D|W|A|C|E|F|S|R|M|T|U|I|    Reserved     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Protocol Support Flags

  - **B (bit 8):** The Endpoint shall set this bit to indicate support for Bypass using IPv4 addressing (for the outer IP address).
  - **V (bit 9):** The Endpoint shall set this bit to indicate support for Bypass using IPv6 addressing (for the outer IP address).
  - **P (bit 10):** The Endpoint shall set this bit to indicate support for PSK.
  - **D (bit 11):** The Endpoint shall set this bit to indicate support for DTLS.
  - **W (bit 12):** The Endpoint shall set this bit to indicate support for Wireguard.
  - **A (bit 13):** The Endpoint shall set this bit to indicate support for ARQ for the modes enabled.
  - **C (bit 14)**: The Endpoint shall set this bit to indicate support for ST 2022-1 FEC for all modes enabled.
  - **E (bit 15):** The Endpoint shall set this bit to indicate support for ST 2022-5 FEC for all modes enabled.
  - **F (bit 16):** The Endpoint shall set this bit to indicate support for IPv4 Mode (Advanced Profile Protocol Type 1).

- **S (bit 17):** The Endpoint shall set this bit to indicate support for IPv6 Mode (Advanced Profile Protocol Type 2).
- **R (bit 18):** The Endpoint shall set this bit to indicate support for Bridge Mode (Advanced Profile Protocol Type 6).
- **M (bit 19):** The Endpoint shall set this bit to indicate support for Main Profile Mode (Advanced Profile Protocol Type 8).
- **T (bit 20):** The Endpoint shall set this bit to indicate support for Media-TS Mode.
- **U (bit 21):** The Endpoint shall set this bit to indicate support for Media-ST2110 Mode.
- **I (bit 22)**: The Endpoint shall set this bit to indicate support for Media-ST2022-6 Mode.

The Endpoint shall set at least one of the P, D, or W bits, and at least one of the F, S, R, M, T, U or I bits. If this condition is not met, the RR shall respond with the RR Request Denied Response Message from Section 5.2.3. In such a case, the RR should set the **Reason Code** field to 0x0006, *Invalid Request*.

- **Additional Data (variable):** the size and format of the additional data depends on the Request Code, as described below.

5.2.2.1 Additional Data for Request Code 0x00 (Directory Request)

This message has no additional data.

5.2.2.2 Additional Data for Request Code 0x01 (Connection Request)

The Additional Data format for the Connection Request message is depicted in Figure 8.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Name Length  |                                               |
+-+-+-+-+-+-+-+-+        Target Endpoint Name                   :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Additional Data Field for Connection Request

The Endpoint shall set the fields in Figure 8 as follows:

- **Name Length (8 bits):** This field shall be set to the size, in bytes of the Target Endpoint Name field.
- **Target Endpoint Name (variable):** This field shall be set to the name of the remote Endpoint to which a connection is requested, in UTF-8 format. This field may be null-terminated and shall be 4-byte aligned. If necessary for alignment, this field shall be padded with up to 3 bytes set to zero.

5.2.2.3 Additional Data for Request Code 0x02 (Not Ready, Stay Idle)

This message has no additional data.

5.2.2.4 Additional Data for Request Code 0x03 (Ready to Connect)

The additional data attached to this message represents connection information to be passed to the connecting Endpoint when a connection is requested to the current Endpoint. The RR shall cache this information for as long as the Endpoint is connected and shall discard it once the Endpoint disconnects. The Additional Data format for the Ready to Connect message is depicted in Figure 9.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Connection Data Length     |      Connection Data Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                       Connection Data                         :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Additional Data Field for Ready to Connect

The Endpoint shall set the fields in Figure 9 as follows:

- **Connection Data Length (16 bits):** This field shall be set to the size, in bytes of the Connection Data field. If no connection data is being presented, this field shall be set to zero.
- **Connection Data Type (16 bits):** This field shall determine the format of the connection data. The following ranges are defined for the data type:
  - 0x0000-0x7FFF: Codes defined in this Specification.
  - 0x8000-0xEFFF: Vendor-defined codes. VSF will add a public repository on Github to track these codes and the corresponding connection data format description for them.
  - 0xF000-0xFFFF: Reserved for testing. This range is intended for development of new Connection Data options; once development is complete, vendors shall provide the VSF with a format description, and the VSF shall assign a code in the 0x8000-0xEFFF range for the format and add it to the Github repository.

  The following codes are defined for the 0x0000-0x7FFF range:
  - 0x0000: No connection data. If Connection Data Length is set to zero, Connection Data Type shall be set to 0x0000 as well.
  - 0x0001: This Endpoint's certificate in PEM format. This may either be a self-signed certificate, or a certificate signed by a CA. A connecting Endpoint, if using DTLS, may use this certificate to authenticate the connection.
  - 0x0002: The CA certificate used to sign this Endpoint's certificate, in PEM format. A connecting Endpoint, if using DTLS, may accept any certificate signed by this CA to authenticate the connection.
  - 0x0003: A concatenation of the Endpoint's certificate and the CA certificate used to sign it, in PEM format. The Endpoint's certificate must be signed by the provided CA. A connecting Endpoint, if using DTLS, may authenticate the

connection by checking the Endpoint's certificate against the provided CA certificate.

When a remote Endpoint requests a connection to this Endpoint and is told to use bypass, the Connection Data information allows the remote Endpoint to authenticate this Endpoint.

5.2.3 RR Request Denied Response Message

Direction: RR to Endpoint.
Description: The RR shall use this message to indicate to the Endpoint that a request conveyed to the RR via the Endpoint Connection Initiation Response Message (Section 5.2.2) has been denied. The message includes enough information for the Endpoint to identify the request that is being denied. The message also includes a reason code to indicate why the request is being denied.

The RR Request Denied Response Message is shown in Figure 10.

```
 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 = 0x8031    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reason Code          | Message Type  |  Request Code |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                       Additional Data                         :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: RR Request Denied Response Message

The RR shall set the fields in Figure 10 as follows:

- **Reason Code (16 bits):** This code shall indicate why the request was denied, as follows:
  - 0x0000: *Unspecified reason* - used when the RR does not wish to provide a specific reason.
  - 0x0001: *Unauthorized* - the Endpoint is not allowed to make this request.
  - 0x0002: *Not found* - the requested Endpoint name exists but is not currently connected to the RR.
  - 0x0003: *Does not exist* - the requested Endpoint name is not in the RR's database.
  - 0x0004: *Out of resources* - the RR is unable to perform the requested operation currently due to lack of resources.
  - 0x0005: *Connection refused* - the selected Endpoint has refused the connection.
  - 0x0006: *Invalid request* - the Endpoint sent an invalid, malformed, or unknown request.
  - 0x0007: *Incompatible protocol* (e.g., Wireguard vs PSK).
  - 0x0008: *Invalid requester IP address* - the requested operation is not allowed for the IP address from which the Endpoint is reaching the RR.

- **Message Type (8 bits):** This field shall indicate what type of control message is being denied, as follows:
  - 0x00: Connection Initiation Response described in Section 5.2.2.
  - All other values are reserved to identify future Endpoint-to-RR messages.
- **Request Code (8 bits):** This field shall be set to the **Request Code** of the Connection Initiation Response message being denied.
- **Additional Data (variable):** If the Connection Initiation Response Message being denied has additional data, the RR shall copy the additional data in this field so the Endpoint can identify the exact message.

5.2.4 RR Directory List Response Message

Direction: RR to Endpoint

Description: The RR shall use this message to provide the directory list to the Endpoint, in response to the Endpoint Connection Initiation Response Message with **Request Code** 0x00, described in Section 5.2.2. The directory, if provided, shall only include entries for Endpoints that are ready to connect. Each entry shall include the following data, using the message format in Figure 11:

- The Endpoint name.
- An indication of whether the name is a group or an individual Endpoint.
- The Protocol Support Flags for the name.
- A mechanism to add future expansion data.

The RR Directory List Response Message is shown in Figure 11.

```
 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 = 0x8032    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Total Number of Directory Entries               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Number of Entries in this Message              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Index of First Entry in this Message             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Directory Entry                        :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                               ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Directory Entry                        :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: RR Directory List Response Message Format

The directory entries shall be divided into consecutive blocks at the discretion of the RR, and each block shall be transmitted in a separate message. The number of entries in each block shall be chosen so that the message does not exceed the MTU and is not fragmented.

The contents of the directory are dynamic, as Endpoints come and go. The RR shall take a snapshot of the directory at the moment it receives the request and build the response based on this snapshot.

The RR shall set the fields in the message shown in Figure 11 as follows:

- **Total Number of Directory Entries (32 bits):** this field shall contain the total number of entries in the directory to be transmitted to the requesting Endpoint.
- **Number of Entries in this Message (32 bits):** this field shall contain the number of directory entries in this message.
- **Index of First Entry in this Message (32 bits):** this field shall contain the index of the first entry, so that the Endpoint knows where the entries fit. This count shall start at zero and shall increment up to the total number of directory entries minus one.
- **Directory Entry (variable):** this shall represent one active Endpoint in the directory, using the format described in Section 5.2.4.1 below.

If the directory is empty, the fields **Total Number of Directory Entries**, **Number of Entries in this Message**, and **Index of First Entry in this Message** shall be set to zero.

Example: assume that the directory has 10 entries, which the RR decided to report in two messages, the first with 4 entries and the second with 6 entries. The message contents are set as follows:

- Message 1:
    - Total Number of Directory Entries: 10
    - Number of Entries in this Message: 4
    - Index of First Entry in this Message: 0
    - Message contains directory entries 0, 1, 2, and 3
- Message 2:
    - Total Number of Directory Entries: 10
    - Number of Entries in this Message: 6
    - Index of First Entry in this Message: 4
    - Message contains directory entries 4, 5, 6, 7, 8, and 9

5.2.4.1 Directory Entry Format

The Directory Entry format is shown in Figure 12.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Directory Entry Length    |A|G|         Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reserved    |            Protocol Support Flags             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Name Length  |                                               |
+-+-+-+-+-+-+-+-+           Endpoint Name                       :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:               Reserved for Future Expansion                   :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Directory Entry Format

The RR shall set the fields in Figure 12 as follows:

- **Directory Entry Length (16 bits):** This field shall contain the total size of the directory entry, in bytes.
- **A (1 bit):** This bit shall indicate whether the directory entry includes an Optional Additional Data field. This field is reserved for use in future versions of this Specification.
    - A=0: the entry does not include optional additional data. RRs compliant with this version of the Specification shall set this bit to zero.
    - A=1: the entry includes optional additional data. Endpoints compliant with this specification shall ignore the Additional Data field if this bit is set to one.
- **G (1 bit):** This bit shall indicate whether the entry is an individual Endpoint or a group.
    - G=0: the entry is an individual Endpoint
    - G=1: the entry is a group name
- **Protocol Support Flags (24 bits):** This field shall contain the Protocol Support Flags for the entry, using the format shown in Figure 7 and described in Section 5.2.2. If there are multiple connections to the RR with the same name, as described in Section 4.4, the RR shall select one of the following strategies to create this field, at its discretion:
    - Use the logical AND of all the entry protocol support fields, if there is at least one common connection protocol supported by all entries, and one common Advanced Profile mode supported by all entries.
      or
    - Use the logical OR of all the protocol support fields.
      or
    - Select the Protocol Support Flags for one of the entries, at the discretion of the RR.
- **Name Length (8 bits):** This field shall contain the size, in bytes, of the Endpoint Name field.

- **Target Endpoint Name (variable):** This field shall contain the name of the Endpoint for this entry, in UTF-8 format. It may be null-terminated and shall be 4-byte aligned. If necessary for alignment, it shall be padded with up to 3 bytes set to zero.
- **Optional Additional Data (variable):** If **A**=1, the entry includes the optional additional data field. This field is intended to be used in future revisions of this Specification. Endpoints compliant with this Specification shall ignore the contents of the field. RRs compliant with this Specification shall set **A**=0 and shall not include the Optional Additional Data.

5.2.5 RR Connection Request Response Message

Direction: RR to Endpoint, in response to an Endpoint Connection Initiation Response Message described in Section 5.2.2 requesting a connection.
Description: This message shall inform the Endpoint how to connect to a requested name. It shall include the following information:

- Remote Endpoint name, for confirmation purposes
- Connection method:
  - *Use the current RR connection for communication*. Upon receiving this message, the Endpoint shall assume that all further messages received in the RR connection are from the remote Endpoint, and all messages sent on the RR connection will be relayed to the remote Endpoint.
  - *Connect to an indicated IP address and UDP port using bypass*. Upon receiving this message, the Endpoint shall disconnect from the RR and shall attempt to directly connect to the remote Endpoint at the specified IP address and UDP port.
  - *Connect to some other RR at an indicated IP address and UDP port*. Upon receiving this message, the Endpoint shall disconnect from the current RR and shall connect to the specified RR. Upon connection to this new RR, the Endpoint shall restart the protocol exchanges described in this Specification for connection to the remote Endpoint.
  - *Connect to some other RR at an indicated hostname*. Upon receiving this message, the Endpoint shall disconnect from the current RR and shall connect to the specified RR. Upon connection to this new RR, the Endpoint shall restart the protocol exchanges described in this Specification for connection to the remote Endpoint.
- The Protocol Support Flags for the remote Endpoint.
- The Connection Data provided by the remote Endpoint, as shown in Figure 9 and described in Section 5.2.2.4.

The RR Connection Request Response Message is shown in Figure 13.

```
 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 = 0x8033    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Response Code |             Protocol Support Flags            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Name Length  |                                               |
+-+-+-+-+-+-+-+-+        Remote Endpoint Name                   :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                       Additional Data                         :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: RR Connection Request Response Message Format

The RR shall set the fields in Figure 13 as follows:

- **Response Code (8 bits):** This code shall indicate the type of connection to be used, as follows:
    - 0x00: *Use the current RR connection for communication*.
    - 0x01: *Use bypass to connect to the requested Endpoint*. The IP address and UDP port shall be specified in the Additional Data, as well as the optional connection data.
    - 0x02: *Redirect to another RR, specified by IP address and port*. The IP address and UDP port shall be specified in the Additional Data.
    - 0x03: *Redirect to another RR, specified by hostname and port*. The hostname and UDP port shall be specified in the Additional Data.
- **Protocol Support Flags (24 bits):** This field shall contain the **Protocol Support Flags** for the remote Endpoint, using the format shown in Figure 7 and described in Section 5.2.2. This field shall be set to zero by the RR and ignored by the Endpoint for Response Codes 0x02 and 0x03.
- **Name Length (8 bits):** This field shall be set to the size, in bytes of the Remote Endpoint Name field.
- **Remote Endpoint Name (variable):** This field shall be set to the name of the remote Endpoint in UTF-8 format to which a connection is requested. This field may be null-terminated and shall be 4-byte aligned. If necessary for alignment, this field shall be padded with up to 3 bytes set to zero.

5.2.5.1 Additional Data for Response Code 0x00 (Use Current RR)

This message has no additional data.

5.2.5.2 Additional Data for Response Code 0x01 (Use Bypass)

The Additional Data field for **Response Code** 0x01 is shown in Figure 14.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved          |T|       Bypass UDP Port         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       Bypass IP Address                       :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Connection Data Length     |     Connection Data Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Connection Data                        :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Additional Data Field for Response Code 0x01

The RR shall set the fields in Figure 14 as follows:

- **T (1 bit):** This bit shall indicate the IP address type, as follows:
  - T=0: IPv4
  - T=1: IPv6
- **Bypass UDP Port (16 bits):** This field shall be set to the UDP port to be used in the bypass connection.
- **Bypass IP Address (32 bits if T=0 or 128 bits if T=1):** This field shall be set to the IP address to be used in the bypass connection.
- **Connection Data Length, Connection Data Type, Connection Data:** These fields shall contain the Connection Data provided by the remote Endpoint, as shown in Figure 9 and described in Section 5.2.2.4. The requesting Endpoint may use this information to authenticate the remote endpoint when it establishes the bypass connection. The requesting Endpoint shall ignore **Connection Data** fields with **Connection Data Type** set to unrecognized Vendor-defined codes (range 0x8000-0xEFFF) and codes reserved for testing (range 0xF000-0xFFFF).
  Note: the purpose of the **Connection Data** field is to allow the requesting Endpoint to optionally authenticate the remote Endpoint. Usage of this mechanism is left to the discretion of the implementer.

Note: An Endpoint advertising multiple bypass protocol support (i.e., PSK, Wireguard, DTLS) needs to be able to dynamically identify the incoming protocol, since only one UDP port is advertised. The details of such identification are left at the discretion of the implementer. Endpoints which are not capable of protocol identification are advised not to advertise support for more than one protocol on a given connection.

5.2.5.3 Additional Data for Response Code 0x02 (Redirect by IP Address)

The Additional Data field for **Response Code** 0x02 is shown in Figure 15.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved          |T|       Redirect UDP Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                      Redirect IP Address                      :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: Additional Data Field for Response Code 0x02

The RR shall set the fields in Figure 15 as follows:

- **T (1 bit):** This bit shall indicate the IP address type, as follows:
  - T=0: IPv4
  - T=1: IPv6
- **Redirect UDP Port (16 bits):** This field shall indicate the UDP port of the next RR to be contacted.
- **Redirect IP Address (32 bits if T=0 or 128 bits if T=1):** This field shall indicate the IP address of the next RR to be contacted.

If the Endpoint receives this response, it shall connect with the indicated RR and restart all negotiation.

5.2.5.4 Additional Data for Response Code 0x03 (Redirect by Hostname)

The Additional Data field for **Response Code** 0x03 is shown in Figure 16.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Redirect UDP Port       |   Redirect Hostname Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       Redirect Hostname                       :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Additional Data Field for Response Code 0x03

The RR shall set the fields in Figure 16 as follows:

- **Redirect UDP Port (16 bits):** This field shall indicate the UDP port of the next RR to be contacted.
- **Redirect Hostname Length (16 bits):** This field shall be set to the size of the Redirect Hostname field, in bytes.
- **Redirect Hostname (variable):** This field shall be set to the hostname of the next RR to be contacted. This field may be null-terminated and shall be 4-byte aligned. If necessary for alignment, it shall be padded with up to 3 bytes set to zero.

If the Endpoint receives this response, it shall connect with the indicated RR and restart all negotiation.

5.2.6 RR Connection Incoming Message

Direction: RR to Endpoint. The RR shall only send this message to Endpoints who have previously sent an Endpoint Connection Initiation Response Message indicating readiness to accept connections.

Description: The RR shall send this message to indicate that some remote Endpoint has requested a connection to the Endpoint receiving the message. This message includes the following information:

- The name of the requesting Endpoint.
- The Protocol Support Flags for the requesting Endpoint.
- The connection method:
    - Using the current RR connection.
      Or
    - Bypass, indicating some IP address and port.

The Endpoint shall respond to this message with the Endpoint Connection Incoming Response Message described in Section 5.2.7.

The RR Connection Incoming Message is shown in Figure 17.

```
 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 = 0x8034    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Connection Code|           Protocol Support Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Name Length  |                                               |
+-+-+-+-+-+-+-+-+       Incoming Endpoint Name                  :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                       Additional Data                         :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: RR Connection Incoming Message Format

The RR shall set the fields in Figure 17 as follows:

- **Connection Code (8 bits):** This code shall indicate the type of incoming connection to be used, as follows:
    - 0x00: Endpoints shall use the current RR connection for communication.
    - 0x01: Requesting Endpoint shall use bypass to connect. The incoming IP address and UDP port shall be specified in the Additional Data.
- **Protocol Support Flags:** This field shall contain the Protocol Support Flags for the connecting Endpoint, using the format shown in Figure 7 and described in Section 5.2.2.

- **Name Length (8 bits):** This field shall contain the size, in bytes, of the Incoming Endpoint Name field.
- **Incoming Endpoint Name (variable):** This field shall contain the Incoming Endpoint Name. It may be null-terminated and shall be 4-byte aligned. If necessary for alignment, it shall be padded with up to 3 bytes set to zero.

5.2.6.1 Additional Data for Connection Code 0x00 (Use Current RR)

This message has no additional data.

5.2.6.2 Additional Data for Connection Code 0x01 (Use Bypass)

The Additional Data field for **Connection Code** 0x01 is shown in Figure 18.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved          |T|         Bypass UDP Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       Bypass IP Address                       :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: Additional Data for Connection Code 0x01

The RR shall set the fields in Figure 18 as follows:

- **T (1 bit):** This bit shall indicate the Bypass IP Address type, as follows:
    - T=0: IPv4
    - T=1: IPv6
- **Bypass UDP Port (16 bits):** This field shall indicate the UDP port for the incoming bypass connection.
- **Bypass IP Address (32 bits if T=0 or 128 bits if T=1):** This field shall indicate the IP address for the incoming bypass connection.

5.2.7 Endpoint Connection Incoming Response Message

Direction: Endpoint to RR, in response to the RR Connection Incoming Message described in Section 5.2.6.
Description: The Endpoint shall send this message to the RR to indicate whether the incoming connection has been accepted.

The Endpoint Connection Incoming Response Message is shown in Figure 19.

```
 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 = 0x8034    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Response Code |                   Reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Name Length  |                                               |
+-+-+-+-+-+-+-+-+      Requesting Endpoint Name                 :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: Endpoint Connection Incoming Response Message Format

The Endpoint shall set the fields in Figure 19 as follows:

- **Response Code (8 bits):** This field shall indicate whether the Endpoint accepts the connection, as follows:
  - 0x00: *Connection is accepted*. After sending this response code, the Endpoint shall behave as follows:
    - If the RR Connection Incoming Message from Section 5.2.6 has Connection Code 0x00 (Use RR), the Endpoint shall assume that all further communication using the RR connection is with the remote endpoint.
    - If the RR Connection Incoming Message from Section 5.2.6 has Connection Code 0x01 (Use Bypass), the Endpoint shall disconnect from the RR after sending the Endpoint Connection Incoming Response Message and wait to be contacted at the same UDP Port by the remote Endpoint.
  - 0x01: *Connection is refused*. After sending this response, the Endpoint shall remain connected to the RR, and in Ready to Connect state.
- **Name Length (8 bits):** This field shall contain the size, in bytes, of the Requesting Endpoint Name field.
- **Requesting Endpoint Name (variable):** This field shall contain the Requesting Endpoint Name. It may be null-terminated and shall be 4-byte aligned. If necessary for alignment, it shall be padded with up to 3 bytes set to zero.

5.3 Protocol Behavior and Timeouts

Several protocol messages described in Section 5.2 require a response. This section describes the behavior of the sending entity (RR or Endpoint) after the message is transmitted, namely, how long to wait for a response, and what action to take if the response is not received. Table 2 lists the per-message behavior and timeouts, as applicable, when the response is not received.

Table 2: Protocol Behavior and Timeouts

Message	**Expected Response**	**Sender/Receiver Behavior**
RR Connection Initiation Message – *RR Ready*	Endpoint Connection Initiation Response Message	RR shall repeat the message once per second until a response is received.
RR Connection Initiation Message – *RR Busy*	No Endpoint response is required	Endpoint shall disconnect. Endpoint may attempt to connect again later. The interval between connection attempts shall be no less than 30 seconds.
RR Connection Initiation Message – *Redirect to Another RR*	No Endpoint response is required	Endpoint shall disconnect and attempt a connection to the specified RR.
Endpoint Connection Initiation Response Message – *Directory Request*	RR Directory List Response Message or RR Request Denied Message	Endpoint should retry after a timeout of 1 second.
Endpoint Connection Initiation Response Message – *Connection Request*	RR Connection Request Response Message or RR Request Denied Message	Endpoint should retry after a timeout of 2 seconds.
Endpoint Connection Initiation Response Message – *Not Ready, Stay Idle*	No RR response is required	RR and Endpoint shall keep the connection open.
Endpoint Connection Initiation Response Message – *Ready to Connect*	No RR response is required	RR and Endpoint shall keep the connection open.
RR Connection Incoming Message	Endpoint Connection Incoming Response Message	RR shall repeat the message once per second until a response is received

5.4 Protocol Message Examples (Informative)

Figure 20 shows an example of an Endpoint connecting to the RR.

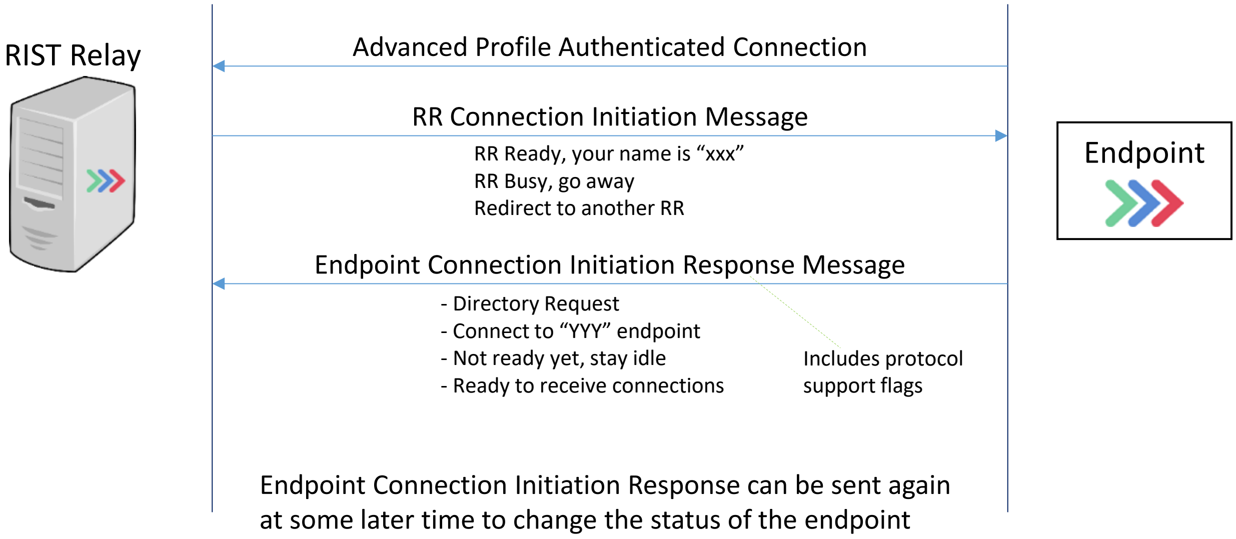
RIST Relay
Endpoint
Advanced Profile Authenticated Connection
RR Connection Initiation Message
RR Ready, your name is "xxx"
RR Busy, go away
Redirect to another RR
Endpoint Connection Initiation Response Message
- Directory Request
- Connect to "YYY" endpoint
- Not ready yet, stay idle
- Ready to receive connections
Includes protocol support flags
Endpoint Connection Initiation Response can be sent again at some later time to change the status of the endpoint

Figure 20: Connecting to the RR

Figure 21shows an example of an Endpoint successfully connecting to another Endpoint.

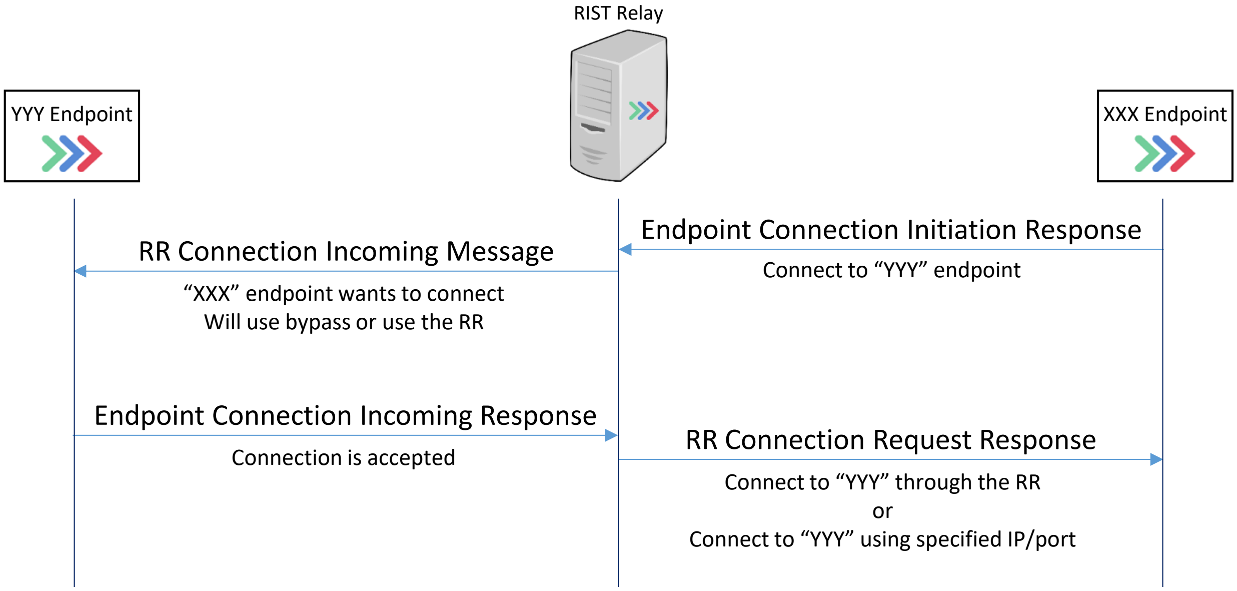
RIST Relay
YYY Endpoint
XXX Endpoint
Endpoint Connection Initiation Response
Connect to "YYY" endpoint
RR Connection Incoming Message
"XXX" endpoint wants to connect
Will use bypass or use the RR
Endpoint Connection Incoming Response
Connection is accepted
RR Connection Request Response
Connect to "YYY" through the RR
or
Connect to "YYY" using specified IP/port

Figure 21: Successful connection between two Endpoints

Figure 22 shows an example where the receiving Endpoint rejects the incoming connection.

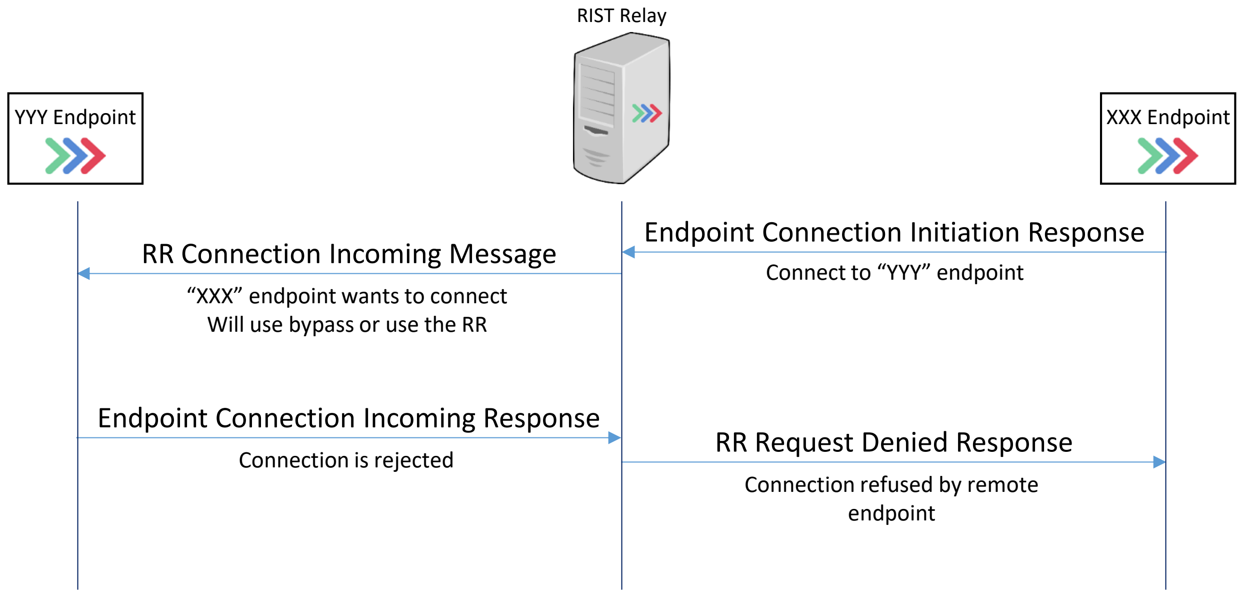
RIST Relay
YYY Endpoint
XXX Endpoint
Endpoint Connection Initiation Response
Connect to "YYY" endpoint
RR Connection Incoming Message
"XXX" endpoint wants to connect
Will use bypass or use the RR
Endpoint Connection Incoming Response
Connection is rejected
RR Request Denied Response
Connection refused by remote endpoint

Figure 22: Destination Endpoint rejects the connection

6 Point-to-Multipoint Streaming Data Transfer (Informative)

From a basic functionality point of view, the RR receives packets from the originating Endpoints (which will be denoted by Senders) and replicates them to the receiving Endpoints (which will be denoted by Receivers). One issue is how to handle retransmissions, and where the ARQ protocol is implemented. There is a continuum of solutions for this issue, splitting the processing load between the RR and the Senders. Two extreme cases are presented below, as well as intermediate cases. The choice of solution is at the discretion of the RR implementer.

6.1 Full Proxy Case

In this case, the RR terminates the ARQ protocol between itself and every device. In other words, there will be independent ARQ sessions between the RR and each of the Senders, and the RR and every Receiver. The advantages of the Full Proxy case are:

- Bandwidth-efficient: retransmissions only go where they need to go. If a packet is lost in the Sender to RR link, it is only retransmitted there, and the Receivers do not even know about it.
- Minimum end-to-end delay: each link can be individually tuned to its needs (NACK delay and number of retries). The data gets to each of the Receivers as fast as possible. There is no correlation between Receivers.
- The load in the Senders is completely independent of the number of receivers.

The only disadvantage of this mode is that there is a higher processing load in the RR, which will need to keep more state per Receiver.

6.2 Transparent Case

In this case RR does as little work as possible and is nothing more than a packet router. More specifically:

- The RR replicates the packets from the Senders to each of the Receivers, unmodified.
- The RR relays any control packets from the Receivers back to the Senders, unmodified.

The consequences of this transparent case are as follows:

- Retransmitted packets from the senders go to every receiver, regardless of whether they asked for it.
- No format or MTU changes are possible. For example, if links to some receivers have smaller MTU than the link to a given sender, the RR is not able to fragment, and these receivers are not able to receive the content. The same applies if the RR is doing outer protocol conversion (i.e., from IPv4 to IPv6) and the outgoing packet exceeds the MTU.
- The use of RTT Echo messages has the following consequences:
    - The Senders receive RTT Echo Requests from all the Receivers and respond to each request.
    - The RTT Echo Responses from the Senders go to all the Receivers. This requires the Receivers to identify and discard the responses that are intended for other receivers by inspecting the Requester SSRC field.
    - When a Sender transmits an RTT Echo Request, there will be a burst of responses, one from each of the receivers. This traffic burst is proportional to the number of Receivers.

The only advantage of the Transparent case is that it simplifies the RR.

There are scalability problems with the Transparent Case when there are many Receivers:

- If the same packet is lost by several Receivers, the Senders can use some algorithm to optimize this and only send one copy of the packet. Let us assume that such an algorithm exists and works perfectly. If there are many receivers over diverse network connections, it is conceivable that almost every packet will be dropped on the way to some Receiver. So, even with a perfect algorithm that only sends one copy of each lost packet, with many receivers, almost every packet will be retransmitted. This means that the bandwidth utilization on all links will be twice (or more) the stream rate.
- The RTT Echo mechanism does not scale in this case. The Senders have to respond to every Receiver (scalability and possibly bandwidth issue at the Senders). If a Sender wants to measure the RTT, it will cause a large network burst when it sends the request. Again, this represents a scalability and possibly bandwidth issue.

6.3 Partial Proxy Cases

As identified above, the Transparent Case has scalability problems with retransmissions and RTT Echo. Partial proxies are possible for each of these cases.

6.3.1 Retransmission Proxy

In this case, the RR buffers the last X seconds of packets received from each Sender (either original or retransmitted). When the RR receives a retransmission request from any Receiver, it checks its buffer, and if the data is there, it satisfies the retransmission request from that Receiver. If the data is not there, the request is passed on to the specific Sender.

Advantage:

- Solves the extra retransmission bandwidth issue.

Disadvantages:

- From the point of view of the Receivers, the RTT will vary depending on whether the transmission comes from the RR or the Sender.
- The implementation complexity in the RR is almost the same as the full ARQ stack.
- This case has the same MTU and format restrictions as the transparent case.

6.3.2 RTT Proxy

In this case, the RR is responsible for the RTT exchange, as follows:

- The RR sends periodic RTT Echo Requests to all the Senders and Receivers.
- The RR caches the last RTT to each of the Senders, and the highest RTT to the Receivers.
- When the RR receives an RTT Echo Request from a Receiver, it responds but delays the response by the value of the last RTT from that specific Sender. The request is not passed to the Sender.
- When the RR receives an RTT Echo Request from a Sender, it responds but delays the response by the value of the highest RTT to the Receivers. The request is not passed to the Receivers.

Advantage:

- Solves the RTT bandwidth problem with large numbers of Receivers.

Disadvantage:

- Increases RR complexity.
- This case has the same MTU and format conversion restrictions as transparent case.

7 ARQ Parameter Selection (Informative)

In RIST, packet loss recovery is done primarily through ARQ (although FEC is also supported as an option). The ARQ algorithm has several tunable parameters, whose values depend on the quality of network and the Round-Trip Time (RTT) between the two devices communicating. In the absence of user input or additional information, the suggested default ARQ parameters from TR-06-1 Appendix B are recommended:

- 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.