

Enhanced Performance Measurement and Liveness Monitoring in Segment Routing Networks

draft-gandhi-spring-sr-enhanced-plm-01

Rakesh Gandhi - Cisco Systems (rgandhi@cisco.com) - Presenter

Clarence Filsfils - Cisco Systems (cfilsfil@cisco.com)

Navin Vaghamshi - Reliance (Navin.Vaghamshi@ril.com)

Moses Nagarajah - Telstra (Moses.Nagarajah@team.telstra.com)

Agenda

- Requirements and Scope
- Summary
- Next Steps

Requirements and Scope

Requirements:

- Performance Delay Measurement & Liveness Monitoring in SR networks
 - ✓ End-to-end P2P/P2MP SR paths
 - ✓ Applicable to SR-MPLS/SRv6 data planes
 - ✓ Support ECMP SR paths
- Running single protocol for liveness detection and performance measurement in SR networks
 - ✓ Simplify deployment and reduce operational complexity
- No endpoint dependency
 - ✓ Stateless on endpoint (e.g. endpoint unaware of the probe protocol)
 - ✓ Higher scale and faster detection interval (e.g. packets not punted out of fast-path)

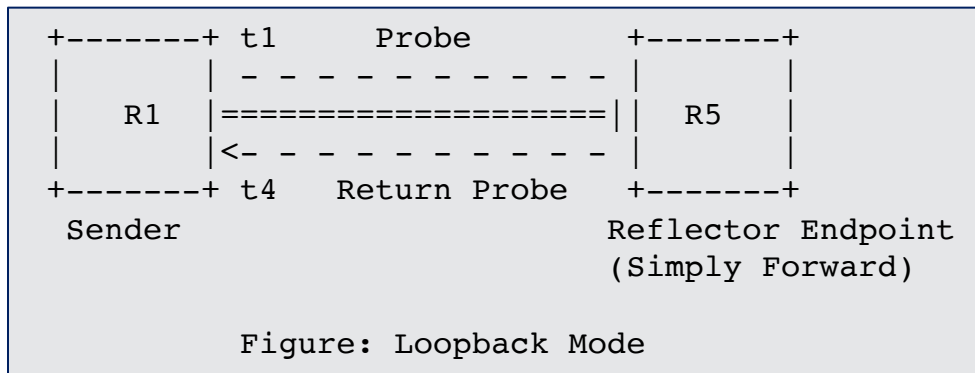
Scope:

- RFC 5357 (TWAMP Light) defined probe messages
- RFC 8762 (STAMP) defined probe messages
- User-configured IP/UDP path for probe messages

History of the Draft

- March 2020
 - Draft was published
- April 2020
 - Presented *version 00* in IETF MPLS WG Virtual Meeting

Liveness Monitoring of SR Policy



- Liveness monitoring for SR Policy uses PM probes (TWAMP Light/STAMP delay measurement messages) in Loopback Mode
- Probe messages sent using Segment List(s) of the SR Policy Candidate Path(s)
- Probe messages are not punted on the remote node (endpoint/reflector) out of fast-path
- Return path can be IP or SR
- Liveness failure is notified when consecutive N number of probe messages are not received back at the sender

TWAMP Light Probe Message

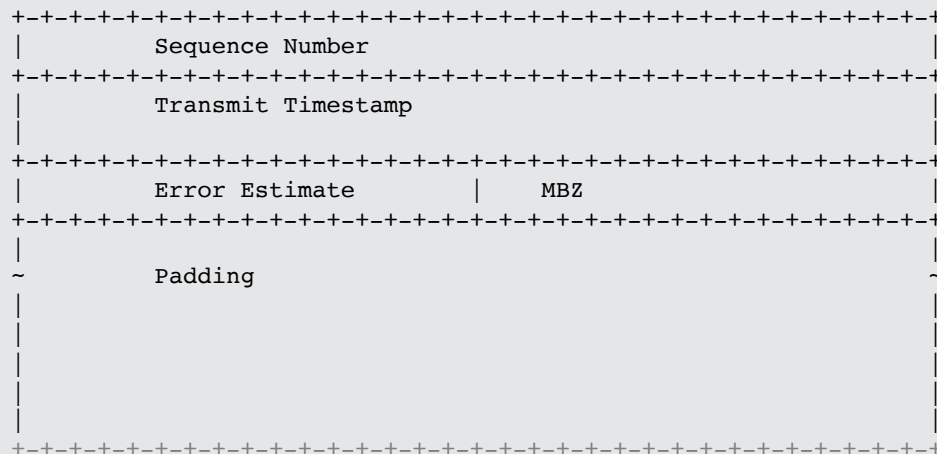


Figure: TWAMP Light Probe Message Format

Loopback Mode

- Sender adds the Transmit Timestamp
- Sender Sequence Number, Sender Timestamp, Sender Error Estimate and Sender TTL in the TWAMP Light messages are not used
 - Reflector does not copy them

STAMP Probe Message

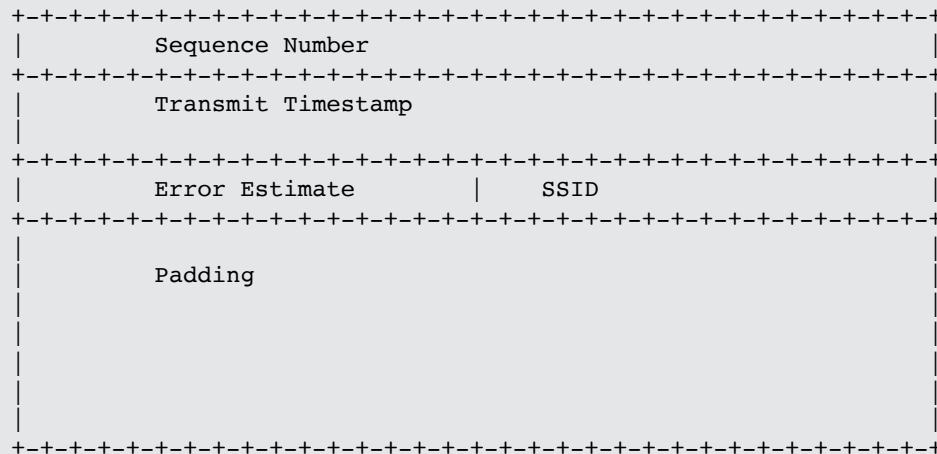
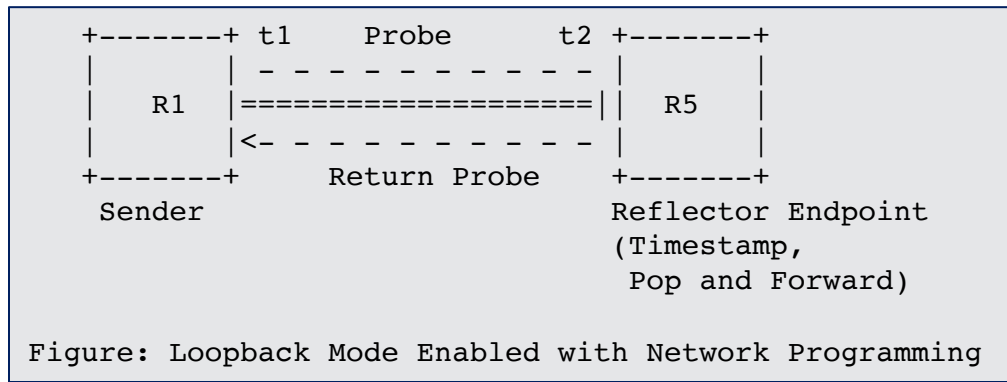


Figure: STAMP Probe Message Format

Loopback Mode

- Sender adds the Transmit Timestamp
- Sender Sequence Number, Sender Timestamp, Sender Error Estimate and Sender TTL in the STAMP messages are not used
 - Reflector does not copy them

Enhanced Liveness Monitoring of SR Policy



- Use PM probes in loopback mode enabled with network programming function
 - The network programming function optimizes the "operations of punt, add receive timestamp and inject the probe packet" on the reflector node
- The endpoint node adds the receive timestamp in the payload of the received TWAMP Light or STAMP probe message without punting the probe message
 - Only add the receive timestamp if the source address in the probe message matches the local node address
- Liveness failure is notified when consecutive N number of probe messages are not received back at the sender
- Delay metrics are notified when consecutive N number of probe messages have delay values exceed the configured thresholds

TWAMP Light Probe Message

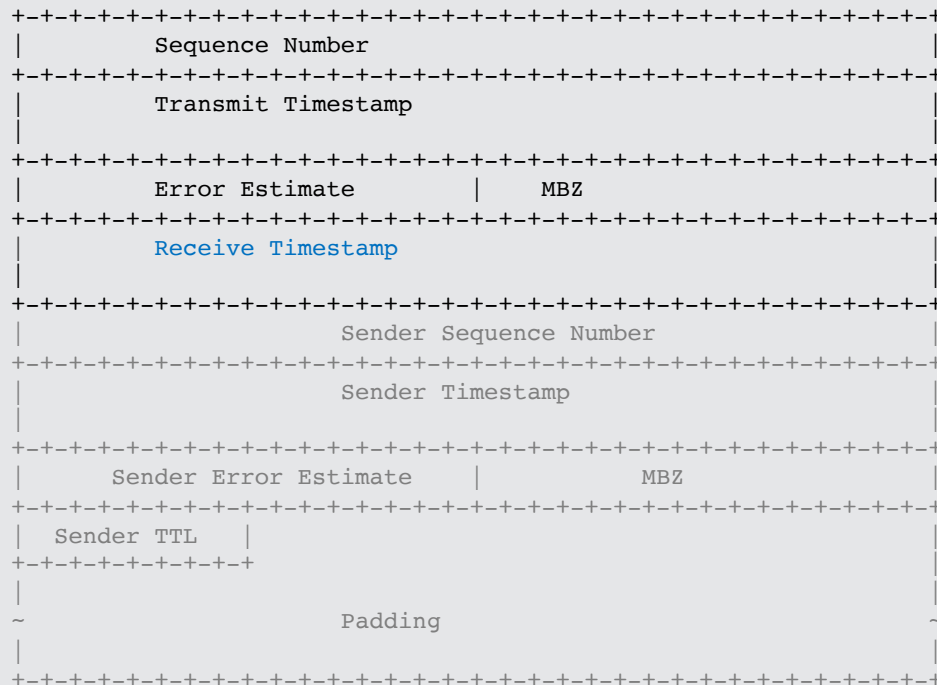


Figure: TWAMP Light Probe Message Format

Enhanced Loopback Mode

- Sender adds the Transmit Timestamp
- Reflector adds the Receive Timestamp at fixed offset locally provisioned (consistently in the network)
 - For TWAMP Light packets, it is at offset-byte 16 from the start of the payload

STAMP Probe Message

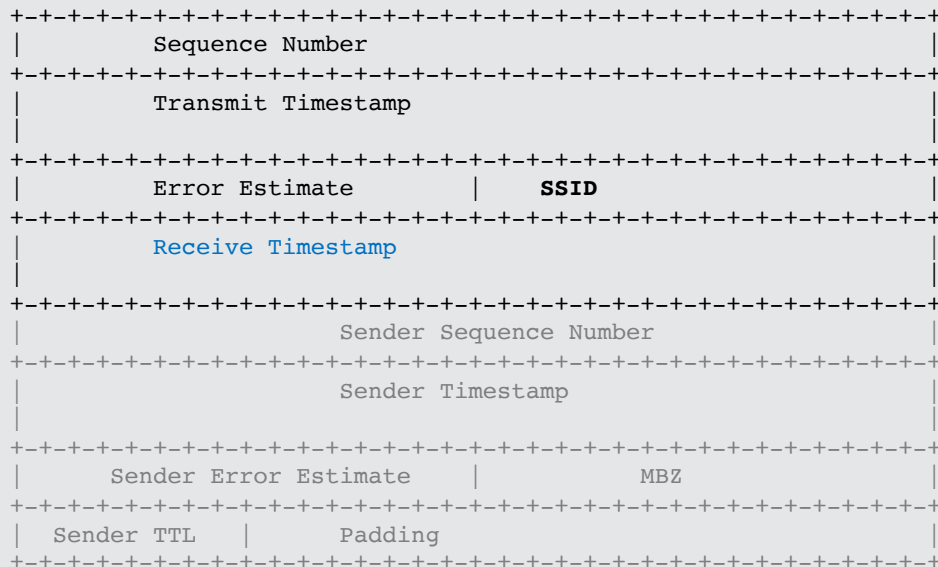


Figure: STAMP Probe Message Format

Enhanced Loopback Mode

- Sender adds the Transmit Timestamp
- Reflector adds the Receive Timestamp at fixed offset locally provisioned (consistently in the network)
 - For STAMP packets, it is at offset-byte 16 from the start of the payload

SR-MPLS with Timestamp Label Example

```
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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          Label(1)          | TC | S |          TTL          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.
.
.
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          Label(n)          | TC | S |          TTL          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          Timestamp Label (TBA1)          | TC | S |          TTL          |
+-----+-----+
| IP Header |
. Source IP Address = Endpoint IPv4 or IPv6 Address .
. Destination IP Address = Sender IPv4 or IPv6 Address .
. Protocol = UDP .
+-----+-----+
| UDP Header |
. Source Port = As chosen by Sender .
. Destination Port = User-configured Port .
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Payload as defined in Section 4.2.1 of RFC 5357 |
| Payload as defined in Section 4.2 of RFC 8762 |
.
+-----+-----+
```

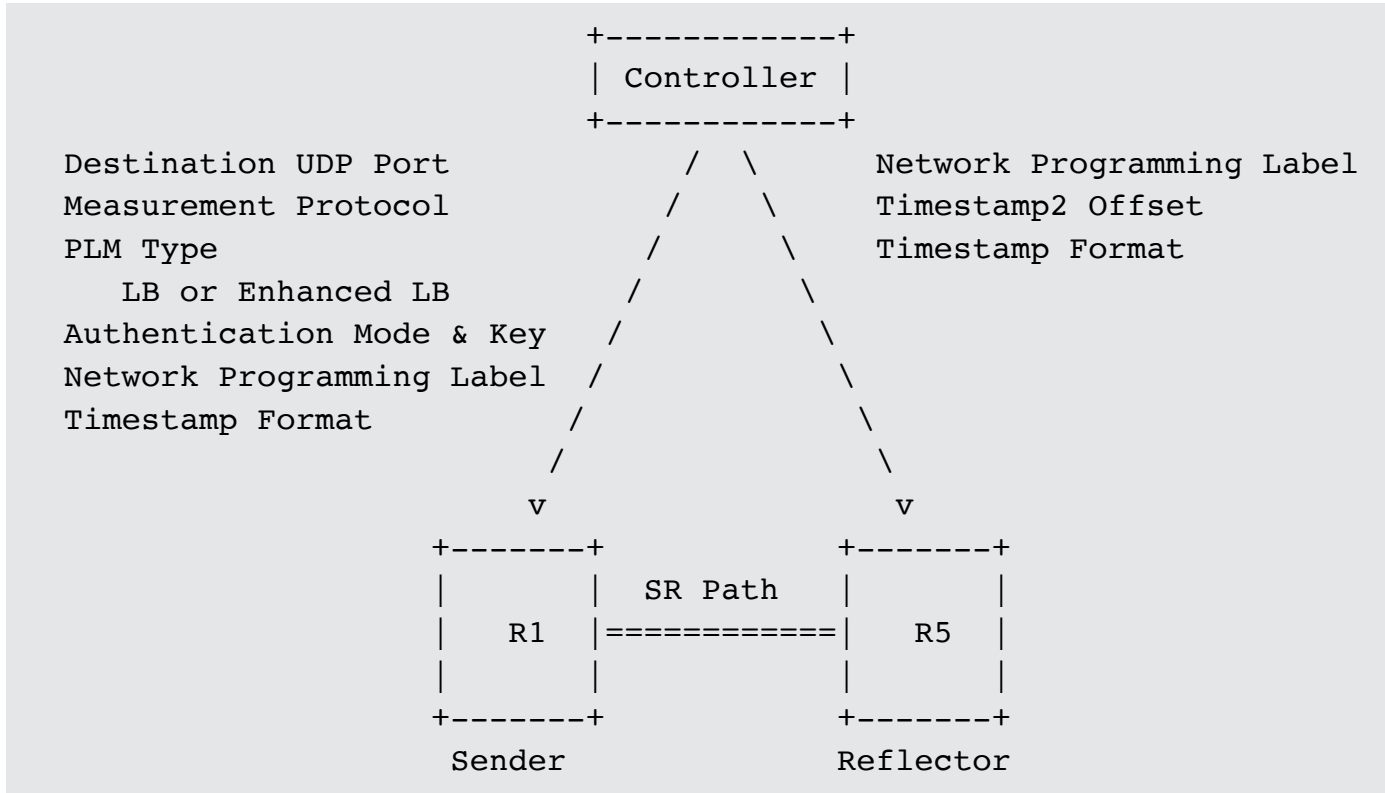
Figure: Example Probe Message for SR-MPLS with Timestamp Label

- Timestamp Label (TBA1) allocated by IANA from Extended Special-Purpose MPLS Label Values
 - Used for Timestamp, Pop and Forward network programming function
 - Timestamp Label (TBA1) is popped by the reflector node
- Source and Destination Addresses are swapped - represent Reverse direction path
- Optionally, Reverse direction SR path label stack can follow the Timestamp Label TBA1

ECMP Support for SR Paths

- SR Paths can have ECMP between the ingress and transit nodes, between transit nodes and between transit and egress nodes.
- Sending PM probe queries that can take advantage of the hashing function in forwarding plane.
- Existing forwarding mechanisms are applicable to PM probe messages. Examples are:
 - For IPv4
 - Sweeping destination address in IPv4 header (e.g. 127/8) when return path is SR-MPLS
 - For IPv6
 - Sweeping flow label in IPv6 header

Example Provisioning Model

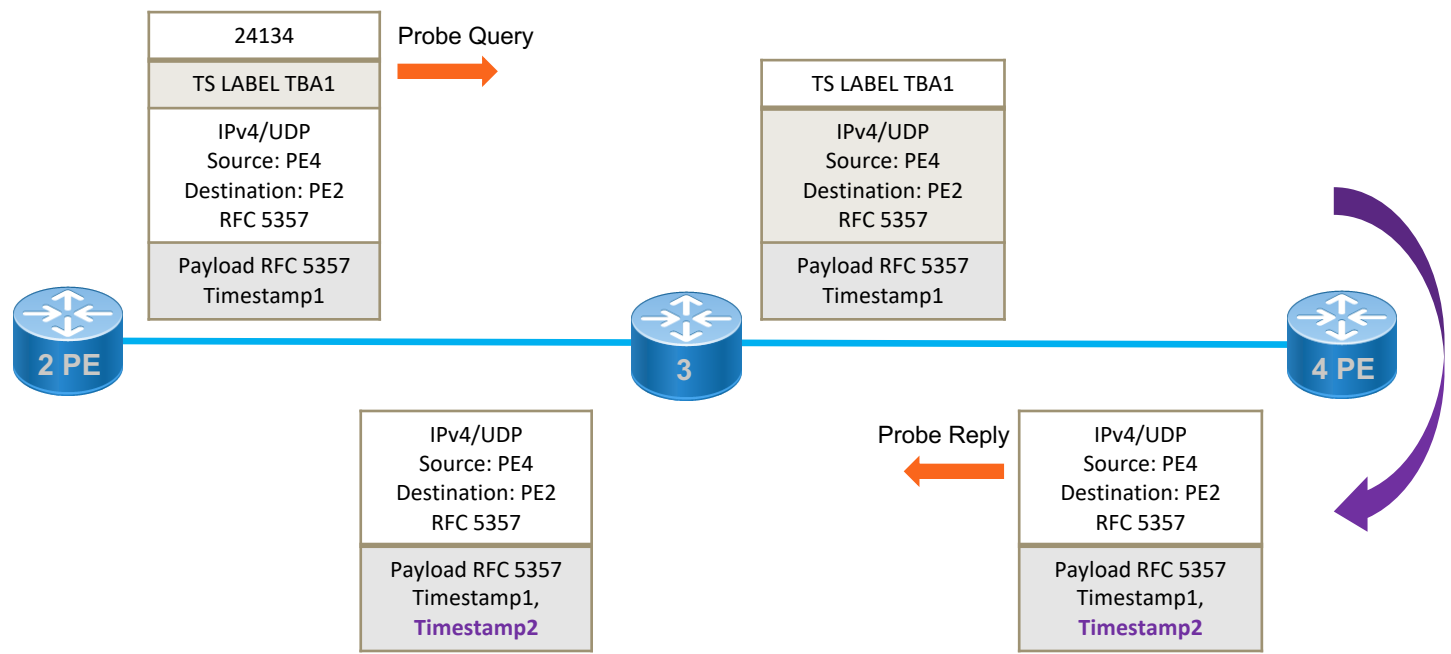


Next Steps

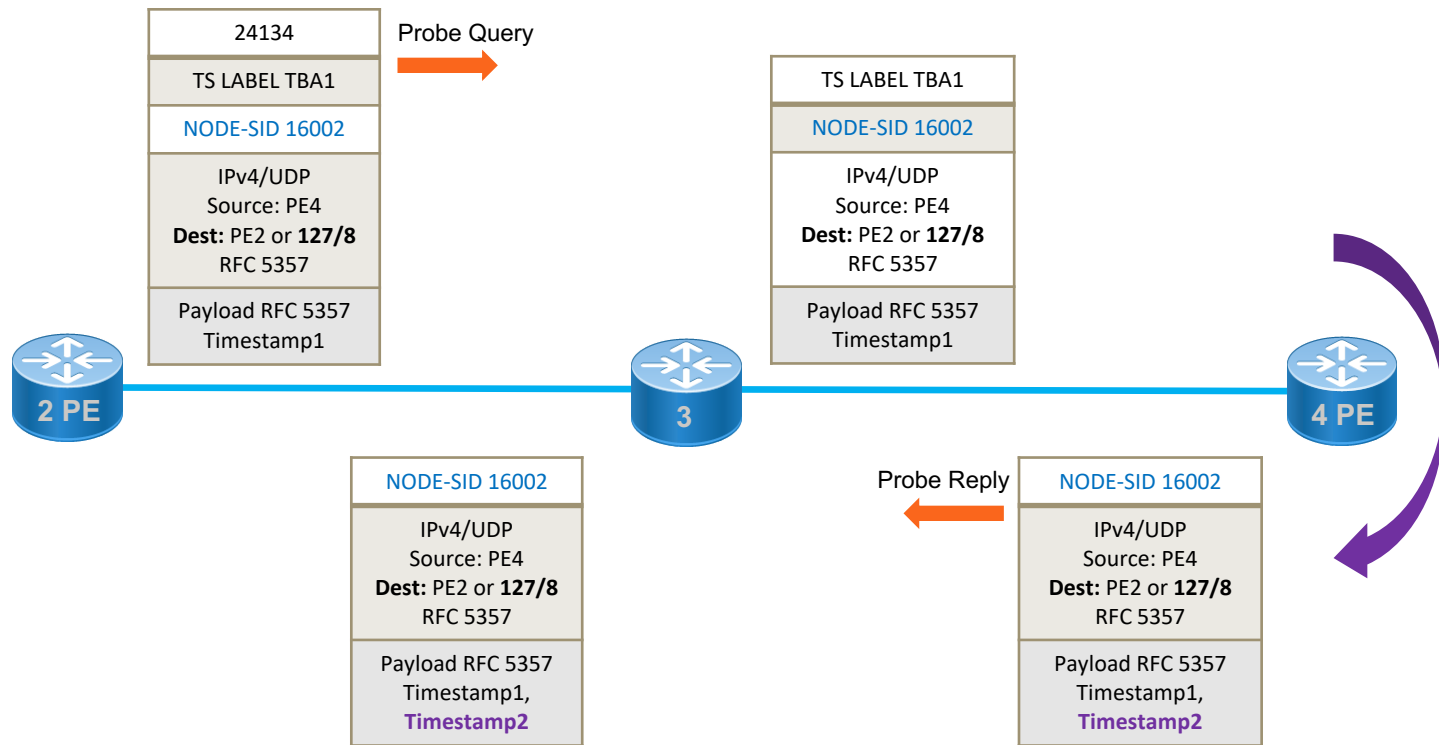
- Welcome your comments and suggestions
- Requesting WG adoption

Thank you

Enhanced Loopback Mode for SR-MPLS Policy - IP/UDP Return Path



Enhanced Loopback Mode for SR-MPLS Policy - SR Return Path



Thank you

Backup

SRv6 with Timestamp and Forward Function Example 1

```
+-----+
| IP Header |
| . Source IP Address = Sender IPv6 Address |
| . Destination IP Address = Next IPv6 Address |
| . |
+-----+
| SRH as specified in RFC 8754 |
| . <Segment List> |
| . END.TSF with Target SID |
| . |
+-----+
| IP Header |
| . Source IP Address = Reflector IPv6 Address |
| . Destination IP Address = Sender IPv6 Address |
| . |
+-----+
| UDP Header |
| . Source Port = As chosen by Sender |
| . Destination Port = User-configured Port |
| . |
+-----+
| Payload as defined in Section 4.2.1 of RFC 5357 |
| Payload as defined in Section 4.2 of RFC 8762 |
| . |
+-----+
```

Example 1: Probe Message for SRv6 with Endpoint Function **IP** Return Path

- Endpoint Function END.TSF is defined for Timestamp and Forward and carried for the Reflector node SID
- Reverse Path is IP
- Reflector node removes SRH
- Inner IPv6 header is required
 - Source and Destination Addresses are swapped to represent the Reverse direction path in the inner IPv6 header

SRv6 with Timestamp and Forward Function Example 2

```

| IP Header
. Source IP Address = Sender IPv6 Address
. Destination IP Address = Next IPv6 Address
.
+-----+
| SRH as specified in RFC 8754
. <Segment List>
. END.TSF with Target SID
.
+-----+
| IP Header (Optional)
. Source IP Address = Reflector IPv6 Address
. Destination IP Address = Sender IPv6 Address
.
+-----+
| UDP Header
. Source Port = As chosen by Sender
. Destination Port = User-configured Port
.
+-----+
| Payload as defined in Section 4.2.1 of RFC 5357 |
| Payload as defined in Section 4.2 of RFC 8762
.
+-----+

```

- Endpoint Function END.TSF is defined for Timestamp and Forward and carried for the Reflector node SID
- Reverse direction SR path carried in SRH
- Reflector node does not remove the SRH
- Inner IPv6 header is optional and not required
 - Source and Destination Addresses are swapped to represent the Reverse direction path in the inner IPv6 header

Example 2: Probe Message for SRv6 with Endpoint Function **SR** Return Path

Thank you