

# Enhanced Performance Delay and Liveness Monitoring in Segment Routing Networks

*draft-gandhi-spring-sr-enhanced-plm-02*

*Rakesh Gandhi - Cisco Systems ([rgandhi@cisco.com](mailto:rgandhi@cisco.com)) - Presenter*

*Clarence Filsfils - Cisco Systems ([cfilsfil@cisco.com](mailto:cfilsfil@cisco.com))*

*Navin Vaghamshi - Reliance ([Navin.Vaghamshi@ril.com](mailto:Navin.Vaghamshi@ril.com))*

*Moses Nagarajah - Telstra ([Moses.Nagarajah@team.telstra.com](mailto:Moses.Nagarajah@team.telstra.com))*

# Agenda

- Requirements and Scope
- Summary
- Next Steps

# Requirements and Scope

## Requirements:

- Performance Delay Monitoring & Liveness Monitoring in SR networks
  - ✓ End-to-end P2P/P2MP SR paths
  - ✓ Applicable to SR-MPLS/SRv6 data planes
- Running single protocol in SR networks
  - ✓ Simplify implementations and reduce development cost
  - ✓ Simplify deployment and reduce operational complexity
- No endpoint dependency
  - ✓ Stateless on endpoint (e.g. endpoint unaware of the protocol)
  - ✓ Higher scale and faster detection interval

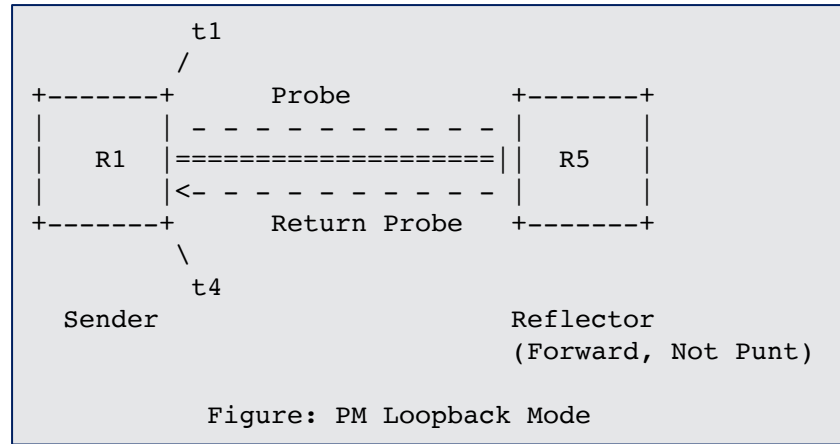
## Scope:

- Example - RFC 5357 (TWAMP Light) defined probe messages
- Example - RFC 8762 (STAMP) defined probe messages

# History of the Draft

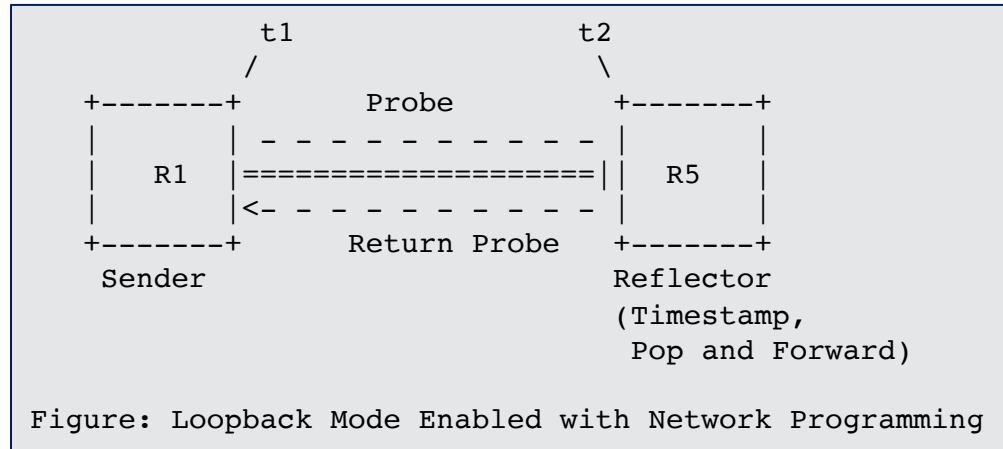
- March 2020
  - Draft was published
- April 2020
  - Presented *version 00* in IETF MPLS WG Virtual Meeting

# PM Probes in Loopback Mode for SR Policy



- Using 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 reflector node out of fast-path
- Round-trip delay ( $t4 - t1$ ) metrics are notified when consecutive M number of probe messages have delay values exceed the configured thresholds
- Liveness failure is notified when consecutive N number of return probe messages are not received at the sender

# Enhanced Performance Delay and Liveness Monitoring of SR Policy



- Using 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
- Reflector node adds the receive timestamp in the payload of the received probe message without punting the message
  - Only adds the receive timestamp if the source address or destination address in the probe message matches the local node address
- One-way delay ( $t2 - t1$ ) metrics are notified when consecutive M number of probe messages have delay values exceed the configured thresholds
- Liveness failure is notified when consecutive N number of return probe messages are not received at the sender

# Example Probe Messages for Timestamp and Forward Function

- Sender adds Transmit Timestamp (t1)
- Reflector adds Receive Timestamp (t2) at fixed offset in payload locally provisioned (consistently in the network)
  - E.g. offset-byte 16 from the start of the payload

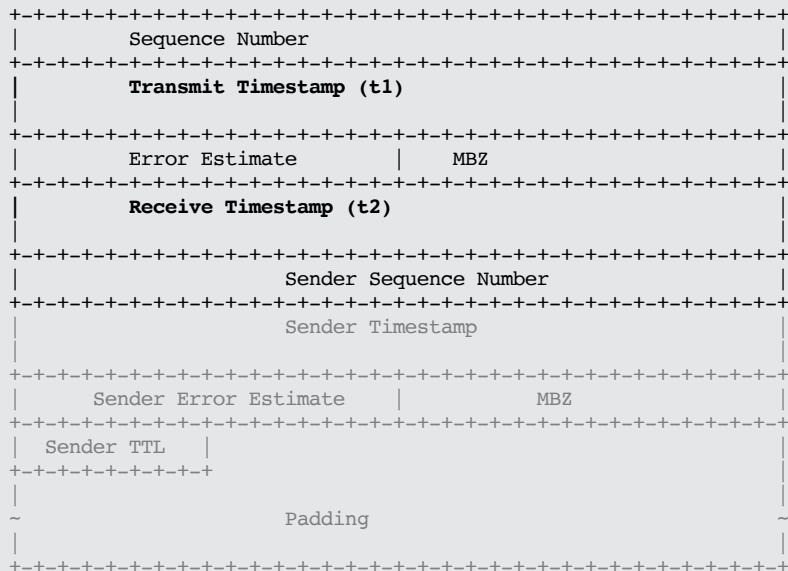


Figure: TWAMP Light Probe Message Format

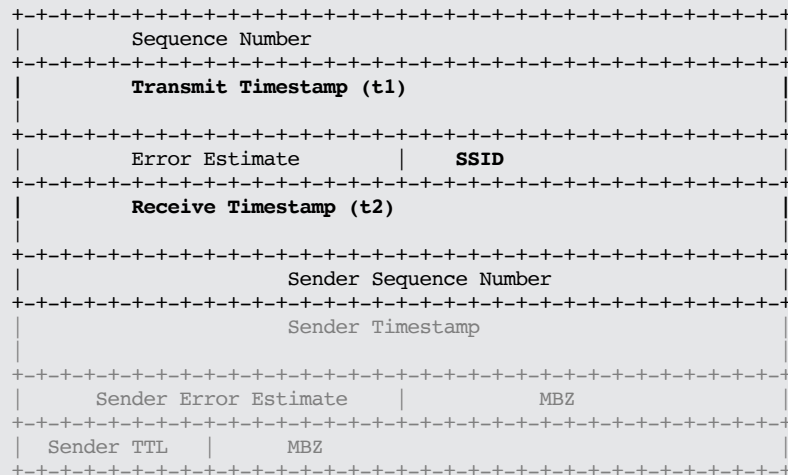


Figure: STAMP Probe Message Format

# SR-MPLS with Timestamp and Forward Function

```

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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     | TC | S |         TTL         |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
.
.
.
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     | TC | S |         TTL         |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          Timestamp Label (TBA1)          | TC | S |         TTL         |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| IP Header                                     |
. Source IP Address = Reflector IPv4 or IPv6 Address .
. Destination IP Address = Sender IPv4 or IPv6 Address .
. Protocol = UDP .
.
+-----+
| UDP Header                                     |
. Source Port = As chosen by Sender .
. Destination Port = As chosen by Sender .
.
+-----+
| Payload                                     |
.
+-----+

```

Example Probe Message with Timestamp Label for SR-MPLS

- Extended Special-purpose label (TBA1) is defined for Timestamp and Forward network programming
- Reverse Path can be IP or SR-MPLS
- Source and Destination Addresses are swapped to represent the Reverse direction path



# SRv6 with Timestamp and Forward Function

```
+-----+
| IP Header |
| . Source IP Address = Sender IPv6 Address |
| . Destination IP Address = Destination 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 = As chosen by Sender |
| . |
+-----+
| Payload |
| . |
+-----+
```

Example Probe Message with Endpoint Function for SRv6

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

# 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.
- PM probe messages 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) if return path is **also** SR-MPLS
  - For IPv6
    - Sweeping flow label in IPv6 header

# Example Provisioning Model

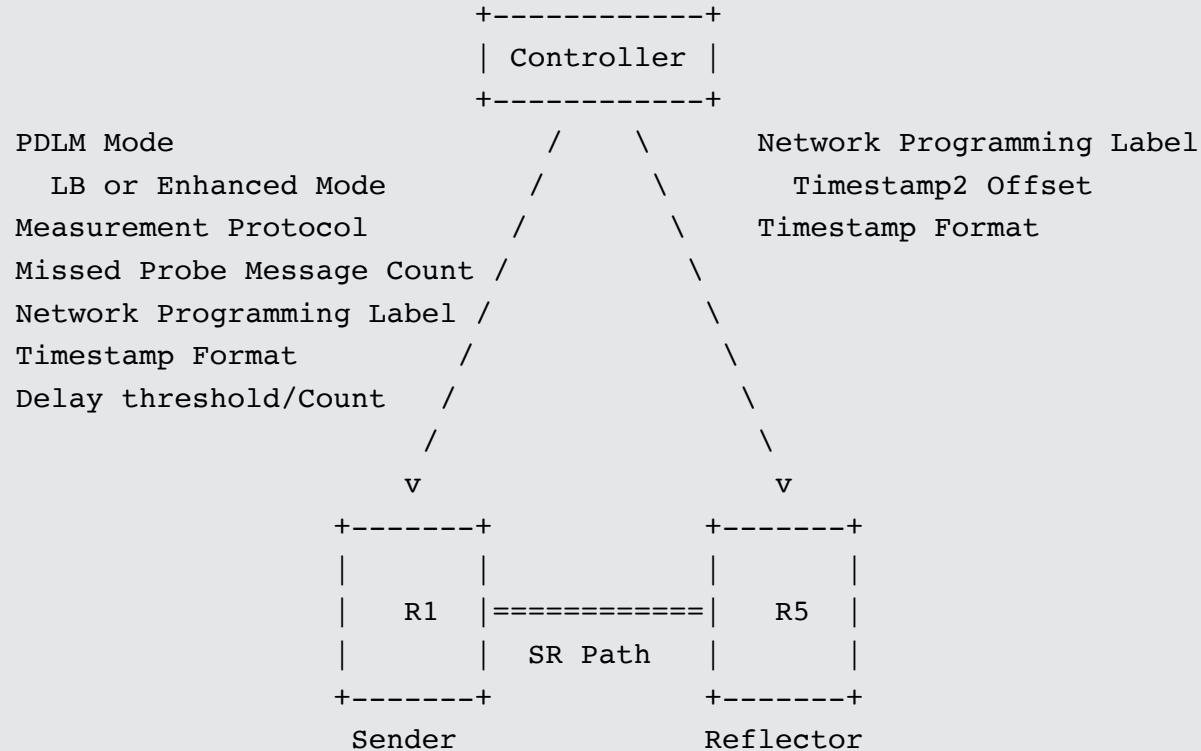


Figure 2: Example Provisioning Model

# Next Steps

- Welcome your comments and suggestions
- Requesting SPRING WG adoption

# Thank you

# Enhanced Loopback Mode for SR-MPLS Policy

