ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Technical Report

(02/2025)

QSTR.MPM-SRv6

Methods for the performance monitoring of SRv6 networks

Technical Report ITU-T QSTR.MPM-SRv6

Methods for the performance monitoring of SRv6 networks

Summary

With the evolution of Internet Protocol Version 6 Plus (IPv6+), the Segment Routing over IPv6 (SRv6) network programming model can offer a more flexible solution to satisfy the complex needs of transport networks in various contexts. With the SRv6 network programming model, it is possible to support valuable services and features such as the layer 2 and layer 3 virtual private network (VPN), traffic engineering (TE) and fast rerouting.

Performance monitoring (PM) is a fundamental function to be performed in a network. It enables operators to detect issues that may require immediate action relating to the quality of service (QoS) parameters and to collect information that can be used for offline network optimization.

The SRv6 PM specifies requirements based on the data plane and the control plane. It is necessary to study both the data management infrastructure and the methods for measuring and collecting data related to nodes and individual traffic based on the SRv6 network. Both these aspects are considered in this Technical Report.

Keywords

Architecture, performance monitoring, SRv6.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Table of Contents

Technical Report ITU-T QSTR.MPM-SRv6

Methods for the performance monitoring of SRv6 networks

1 Scope

This Technical Report studies the methods for the performance monitoring (PM) of Segment Routing over IPv6 (SRv6) networks.

The scope of this Technical Report includes the following aspects:

- Overview for PM of SRv6 networks
- Standardized work in the related standards development organizations (SDOs) for PM of SRv6 network
- The requirements for PM of SRv6
- The monitoring architecture for performance of SRv6 network
- Prospects for SRv6 network PM testing.

2 References

[IETF RFC 8762]	IETF RFC 8762 (2020), *Simple Two-Way Active Measurement Protocol.*
[IETF RFC 8972]	IETF RFC 8972 (2021), *Simple Two-Way Active Measurement Protocol Optional Extensions*.
[IETF RFC 8986]	IETF RFC 8986 (2021), *Segment Routing over IPv6 (SRv6) Network Programming*.
[IETF RFC 9503]	IETF RFC 9503 (2023), *Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Segment Routing Networks*.

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following terms defined elsewhere:

3.1.1 segment [b-IETF RFC 8402]: An instruction a node executes on the incoming packet (e.g., forward packet according to shortest path to destination, or, forward packet through a specific interface, or, deliver the packet to a given application/service instance).

3.1.2 SID [b-IETF RFC 8402]: A segment identifier.

3.1.3 SRv6 [b-IETF RFC 8402]: The instantiation of SR on the IPv6 data plane.

3.1.4 SRv6 SID [b-IETF RFC 8402]: An IPv6 address explicitly associated with the segment.

3.1.5 SR policy [b-IETF RFC 8402]: An ordered list of segments. The headend of an SR Policy steers packets onto the SR Policy. The list of segments can be specified explicitly in SR-MPLS as a stack of labels and in SRv6 as an ordered list of SRv6 SIDs. Alternatively, the list of segments is computed based on a destination and a set of optimization objectives and constraints (e.g., latency, affinity, SRLG, etc.). The computation can be local or delegated to the PCE server. An SR Policy can be configured by the operator, provisioned via NETCONF [b-IETF RFC 6241] or provisioned via PCEP [b-IETF RFC 5440]. An SR Policy can be used for traffic engineering (TE), Operations, Administration, and Maintenance (OAM), or fast reroute (FRR) reasons.

3.1.6 SRv6 SID function [IETF RFC 8986]: The function part of the SID is an opaque identification of local behavior bound to the SID. It is formally defined in Section 3.1 of this document.

3.1.7 SRv6 endpoint behavior [IETF RFC 8986]: A packet processing behavior executed at an SRv6 Segment Endpoint Node. SRv6 Endpoint behaviors related to traffic engineering and overlay use cases.

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

API	Application Programming Interface
BGP	Border Gateway Protocol
BGP-LS	Border Gateway Protocol Link-State
EVPN	Ethernet Virtual Private Network
gNMI	gRPC Network Management Interface
gRPC	Google Remote Procedure Call
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IGP	Interior Gateway Protocol
IOAM	In-band Operation, Administration and Maintenance
IP	Internet Protocol
IPv6	Internet Protocol Version 6
IPv6+	Internet Protocol Version 6 Plus
JSON	JavaScript Object Notation
L2VPN	Layer 2 Virtual Private Network
L3VPN	Layer 3 Virtual Private Network
MB	Megabytes
MIB	Management Information Base
MPLS	Multiprotocol Label Switching
NETCONF	Network Configuration protocol
NMS	Network Management System
NTP	Network Time Protocol
OAM	Operation, Administration and Maintenance
OSS	Operating Support System
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Protocol

PM	Performance Monitoring
PSID	Path Segment Identifier
PTP	Precision Time Protocol
QoS	Quality of Service
REST	Representational State Transfer
RPC	Remote Procedure Call
SDO	Standard Development Organization
SID	Segment Identifier
SL	Segment List
SDN	Software-Defined Networking
SNMP	Simple Network Management Protocol
SR	Segment Routing
SRH	Segment Routing Header
SR-MPLS	Segment Routing over the MPLS forwarding plane
SRv6	Segment Routing over IPv6
SSID	STAMP Session Identifier
STAMP	Simple Two-way Active Measurement Protocol
Sub-TLV	Sub-Type-Length-Value
TSF	Timestamp and Forward
TLV	Type-Length-Value
TWAMP	Two-Way Active Measurement Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
XML	Extensible Markup Language
YANG	Yet Another Next Generation

5 Conventions

In this Technical Report:

– The keywords "is required to" indicate a requirement which is potentially required.

– The keywords "is recommended" indicate a possible requirement which is recommended but which is not necessary to be fulfilled.

6 Overview

With the evolution of Internet Protocol Version 6 Plus (IPv6+), the SRv6 network programming model can offer a more flexible solution to satisfy the complex needs of transport networks in different contexts. The SRv6 network programming model can support valuable services and features such as the layer 2 virtual private network (L2VPN) and layer 3 virtual private network (L3VPN), traffic engineering (TE) and fast rerouting.

Performance monitoring (PM) is a fundamental function to be performed in a network. It enables operators to detect issues that may require immediate action relating to the quality of service (QoS) parameters and to collect information that can be used for offline network optimization.

Two Internet Engineering Task Force (IETF) drafts have been proposed for the PM data plane subsystem for SRv6 and are currently under discussion at the IETF SPRING WG. These drafts rely on existing methodologies for performance measurement in general Internet Protocol (IP) and multiprotocol label switching (MPLS) networks. They propose an extension of these methodologies to the SRv6 PM case. Both proposed solutions focus on the system architecture and protocol specification; however, the actual system implementation and integration in the network data plane need to be defined and validated in the field. Given this, study on methods for PM of SRv6 networks is of great significance.

a) Objectives of SRv6 network PM

Through real-time monitoring of node status, link latency, forward paths, packet loss, and abnormal paths in an SRv6 network, the network administrators can grasp the running status of the entire network in real time, discover possible risks in the network in a timely manner, and cooperate with service forwarding policies to select the best forward path for service flows, ultimately improving the forwarding capability and reliability of the entire network.

b) Monitoring requirements for SRv6 network performance

   i) **Real time**: The monitoring system is required to collect and analyse the performance data of the SRv6 network in real time to promptly identify and solve problems.

   ii) **Accuracy**: The monitoring system is required to provide accurate performance data so that network administrators can accurately assess the state of the network and make appropriate decisions.

   iii) **Scalability**: The monitoring system is required to adapt to the increasing size and complexity of the network to maintain effective monitoring as the network expands.

   iv) **Usability**: The monitoring system is required to provide a user-friendly user interface and easy-to-use tools that allow network administrators to undertake performance analysis and troubleshooting.

c) Monitoring methods for SRv6 network performance

   i) **Traffic monitoring**: By collecting and analysing traffic data in the network, key performance indicators, such as bandwidth utilization, latency, and packet loss, can be acquired. This helps identify potential performance bottlenecks and points of failure.

   ii) **Path monitoring**: By tracing the forward path of packets on an SRv6 network, the network topology and routing can be explored. This helps diagnose network faults and optimize network configurations.

   iii) **Statistical analysis**: Statistical analysis of the collected performance data can identify anomalous behaviours and trends in the network. This helps predict future performance issues and enables action to be taken in advance.

   iv) **Log analysis**: The log information generated by network devices, such as routers and switches, can be analysed to understand the network device status and configuration changes. This helps trace the root cause of failure and enable appropriate remedial action to be taken.

   v) **Fault simulation and testing**: A comprehensive evaluation of the performance of the SRv6 network can be conducted using fault simulation and performance test tools. This helps identify potential performance issues and validates the effectiveness of the solution.

In summary, SRv6 network PM is a key to ensuring the stable operation of SRv6 networks and meet service requirements. By selecting appropriate monitoring methods and tools that meet requirements of timeliness, accuracy, scalability, and usability, the performance and QoS of the SRv6 network can be effectively improved.

7 Standardization for PM of SRv6 networks

7.1 PM of SRv6 network standardization work in ITU-T

The following Recommendations developed by ITU-T are related to PM in the SRv6 networks.

[b-ITU-T G.1051] specifies a set of IP performance parameters and methods of measurement applicable for assessing the quality of packet transfer on inter-domain paths. It is based on the Two-Way Active Measurement Protocol (TWAMP). It can apply to scenarios that require ultra-high reliability and low latency.

[b-ITU-T Y.1540] defines the parameters that may be used to specify and assess the speed, accuracy, dependability, and availability of IP packet transfers of regional and international IP data communication services. The defined parameters are applied to end-to-end or point-to-point IP services and network portions.

[b-ITU-T Y.1543] specifies a method to continuously measure two-way data latency and loss for a defined observation period. It also specifies detailed requirements and methods for QoS monitoring in trusted IP networks.

7.2 PM of SRv6 network standardization work in IETF

There is a large amount of research on SRv6 network performance measurement in the IETF, as follows:

[b-draft-song-spring-siam-07] describes an active measurement method for SRv6 that can support segment-by-segment and end-to-end measurements on any SRv6 path using existing protocols such as in-band operation, administration, and maintenance (IOAM). A packet containing a segment routing header (SRH) uses a flag bit to indicate that it is an active probing packet and requires segment-by-segment processing. Measurement information, such as the IOAM header and data, is encapsulated in the User Datagram Protocol (UDP) payload, which is indicated by a dedicated port number. The probing packet originates from a segment source node, traverses an arbitrary segment path and terminates at a segment endpoint node, as configured by the segment list (SL) in the SRH. Each segment node on the path parses the UDP header and payload when a flag is detected. In IOAM, the node processes the IOAM option, conforming to the standard procedures defined in the IOAM documents. This method is compatible with other SRv6 active measurement methods and supports multiple applications.

[b-draft-ietf-spring-stamp-srpm-17] describes procedures for performance measurement in segment routing (SR) networks using a Simple Two-Way Active Measurement Protocol (STAMP) defined in [IETF RFC 8762] and its optional extensions defined in [IETF RFC 8972] and further augmented in [IETF RFC 9503]. The procedure is used for links, SR paths, and layer 3 and layer 2 services in SR networks and is applicable to both SR over the MPLS forwarding plane (SR-MPLS) and SRv6 data plane.

[b-IETF RFC 9259] describes how existing IPv6 mechanisms for ping and traceroute can be used in SRv6 network. It specifies the operation, administration and maintenance (OAM) flag (O-flag) in the SRH for performing controllable and predictable flow sampling from segment endpoints. In addition, the document describes how a centralized monitoring system performs a path continuity check between nodes within an SRv6 domain.

8 Potential requirements for PM of SRv6

8.1 Overall requirements for PM of SRv6 networks

Network performance is essential for network operators, as various factors can affect it. In the SRv6 network, measuring and collecting SRv6 node status and traffic can reveal the network status and expose potential problems, thereby ensuring the stability and reliability of the network. By analysing SRv6 network data, it is possible to rationally plan and provide network resources, improve their utilization, and reduce operating costs. When a network failure occurs, the analysis of the collected data can help detect the cause and location of the failure and shorten the recovery time. Through long-term analysis of SRv6 network data, solid support for network planning and design can be provided, enhancing the rationality and foresight of network design.

The monitoring requirements for SRv6 network performance encompass measurement parameters, measurement methods, data types, data collection, data reporting, and data analysis.

a) Measurement parameters

To reflect the actual operating conditions of the SRv6 network, the SRv6 network performance measurement parameters are required to comply with the following guidelines:

i) The parameters are required to be specific and clearly defined.

ii) The measurement of the parameters is required to be repeatable, ensuring that the same measurement results can be obtained multiple times under identical conditions.

b) Measurement methods

Multiple measurement methods are required to be applied to satisfy the measurement requirements of different parameters.

i) Active measurements: The transmission and performance metrics in the network are collected by the injection of specific measurement packets into the network. Active measurements provide accurate network performance data but also impose burdens on the network.

ii) Passive measurements: Information regarding the data packets that pass through the nodes is collected by the deployment of monitoring devices or software on the network nodes. Passive measurements do not impose burdens on the network but may not cover all network paths and traffic.

iii) Hybrid measurements: By combining the advantages of active and passive measurements, accurate performance data can be obtained through active measurement, while actual packet information can be collected through passive measurement. Hybrid measurements improve the measurement accuracy and coverage but may require more resources.

c) Data type

SRv6 PM is required to collect and monitor two types of data.

i) Node data: The SRv6 nodes are the key nodes for resource allocation, data processing, and traffic transmission in SRv6 networks and are crucial to network forwarding performance. Node data mainly focuses on the physical location, type, configuration, health status, quality statistics for each link, and traffic loading on the nodes. Node data can reflect the network topology and connection relationships between nodes, thereby providing basic data for network management and optimization.

ii) Traffic data: Information related to traffic, such as the source address, destination address, transmission protocol, port number, size of data packets, and transmission speed, is included. Traffic data can reflect network data transmission and load conditions, thereby providing a basis for network performance analysis and optimization.

d) Data collection

Data collection is the basis of performance monitoring. It is important to quickly and accurately collect the required network information for network performance analysis and troubleshooting. Therefore, data collection is required to meet the following requirements.

i) Accuracy: The data measured and collected is required to be highly accurate and provide a true reflection for network health and performance. It is required to use reliable measurement methods and tools ensure the credibility and usability of the data.

ii) Timeliness: The data measured and collected is required to be real-time and provide timely feedback on the latest status and network changes. Efficient data transmission and processing techniques are required to ensure timely updating and processing of data.

iii) Completeness: The data measured and collected is required to cover all nodes and traffic in the network. Data loss or omission is required to be avoided. A complete data collection and storage mechanism is required to be established. The collected data is required to be stored in a reliable and scalable storage system to ensure data integrity and traceability.

iv) Scalability: The measurement acquisition system is required to have good scalability and be able to adapt to the continuous expansion and change of the network. Modular, distributed design ideas are required to be used to facilitate system expansion and upgrading.

v) Security: The measurement and collection system is required to be highly secure to prevent unauthorized access and data breaches. Security measures, such as strong password authentication and data encryption, are required to be used to ensure the security of data and systems.

e) Data reporting

Once data collection is complete, the data is required to be reported to specific recipients. The reporting process is required to satisfy the following requirements.

i) Reporting format: The reported data is required to be in a standard format to facilitate interpretation and processing by data recipients.

ii) Reporting method: Performance data is required to be reported to a specified data recipient through reliable protocols, such as Hypertext Transfer Protocol Secure (HTTPS), Simple Network Management Protocol (SNMP) and telemetry.

iii) Reporting frequency: The rational reporting frequency is recommended to be set based on service needs and network scale. Key performance data and exception events are required to be promptly reported to ensure timely handling.

iv) Exception handling: When abnormal performance data are detected, alert messages are required to be generated immediately. Alert messages are recommended to contain details of abnormal data, possible causes and handling suggestions.

f) Data analysis

Data analysis can reflect the operational quality of, and potential problems in, the network, which is important for optimizing the user service path and fault troubleshooting. Therefore, data analysis is required to meet the following requirements:

i) Highlighting key performance indicators for network administrators to understand network performance trends and make decisions.

ii) The virtualization capability for generating a logical network topology related to the physical network by gathering network information and labelling the link status, node status and port status.

iii) When a network failure occurs, it is possible to demarcate the fault based on the collected network information and mark the potential fault location in the logical network topology.

iv) Report generation capability to generate regular performance reports, including trend charts and data tables.

v) Bandwidth demand-forecasting capability to support network administrators in optimizing the network path in a timely manner according to historical data, current traffic type and traffic scale.

8.2 Requirements for PM for the SRv6 data plane

The performance metrics for the SRv6 data plane are similar to those of the traditional IP network. The metrics are required to include throughput, packet loss ratio, transmission latency and jitter. The specific explanations are as follows.

a) **SRv6 packet path throughput** refers to the maximum forwarding capacity for processing SRv6 packets along a forward path composed of SRv6 forwarding nodes.

b) **SRv6 packet transmission latency** refers to the period of transmission from the sender to the receiver of the SRv6 data packet. This typically includes unidirectional and bidirectional latencies.

c) **SRv6 packet latency jitter** refers to the fluctuations in transmission latency in the SRv6 packet network. This represents the stability of the network, which is essential for the stable and efficient transmission of the service.

d) **SRv6 packet loss ratio** refers to the ratio of the number of lost SRv6 packets to the total number of SRv6 packets. It is derived by subtracting the number of packets received by the receiver from the number of packets sent by the sender.

8.3 Requirements for PM for the SRv6 control plane

The following metrics for the SRv6 control plane are recommended to be focused on.

a) **Interfaces and link states.** The changes in interfaces and link states lead not only to the fluctuation of the SRv6 control plane but also the packet loss of the data plan transmission. This is the basic metric used for monitoring.

b) **Neighbour states.** These include the Interior Gateway Protocol (IGP) neighbour states, Border Gateway Protocol (BGP) neighbour states, and Border Gateway Protocol Link-State (BGP-LS) neighbour states. The normalization of neighbour states is directly related to the synchronization of the SRv6 control messages.

c) **Service states.** In the SRv6 virtual private network (VPN) service, Ethernet virtual private network (EVPN) and L3VPN states are recommended to be monitored to ensure that the VPN data plane can work normally. In the policy or TE forwarding service of SRv6, policy paths are recommended to be monitored to ensure that the forwarding policy and the forward path are effective.

d) **Command response.** When generating and configuring the SRv6 path information through the controller, the controller monitors whether the command has been processed successfully.

9 The monitoring architecture for performance of SRv6 networks

This clause introduces a framework for SRv6 PM based on the measurement technique defined in [b-draft-ietf-spring-stamp-srpm-17]. STAMP measures the unidirectional or bidirectional IP performance between any two devices that support this technique in an IP network. It supports the statistics of the measurement data, such as the latency, jitter and packet loss ratio. [IETF RFC 8762] defines the basic STAMP functional features and describes the packet format for collecting and transmitting measurement data. [IETF RFC 8972] introduces the STAMP session identifier (SSID) and defines an optional extension that enhances the basic functions of STAMP, which is more applicable to the SRv6 network.

Figure 9-1 shows a reference scenario for STAMP. In this scenario, two nodes, Session-Sender and Session-Reflector, are defined in the data plane to generate measurement traffic. The Session-Sender sends the test-request packets to the Session-Reflector, while the Session-Reflector replies with test-response packets. These packets carry PM information. The controller undertakes a control-plane monitoring protocol between the STAMP sender and the receiver. All control and management operations are performed through the controller. In addition, the controller uploads the measurement data to the data analyser to analyse the SRv6 network performance. The clocks of the two nodes are required to be synchronized before measurement, to ensure accuracy. During the measurement, STAMP test packets are transmitted over the same path as the real data traffic path, to measure performance metrics such as latency and packet loss experienced by the data traffic.

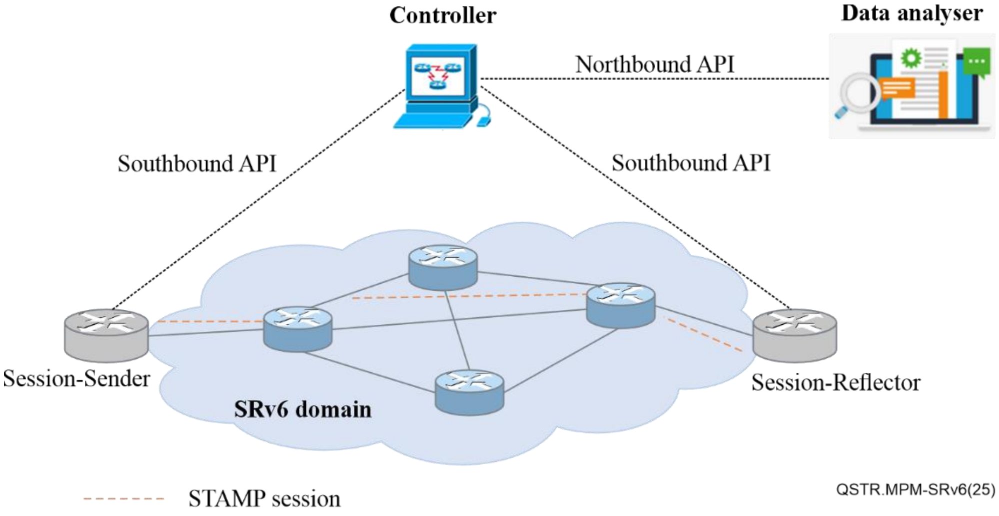


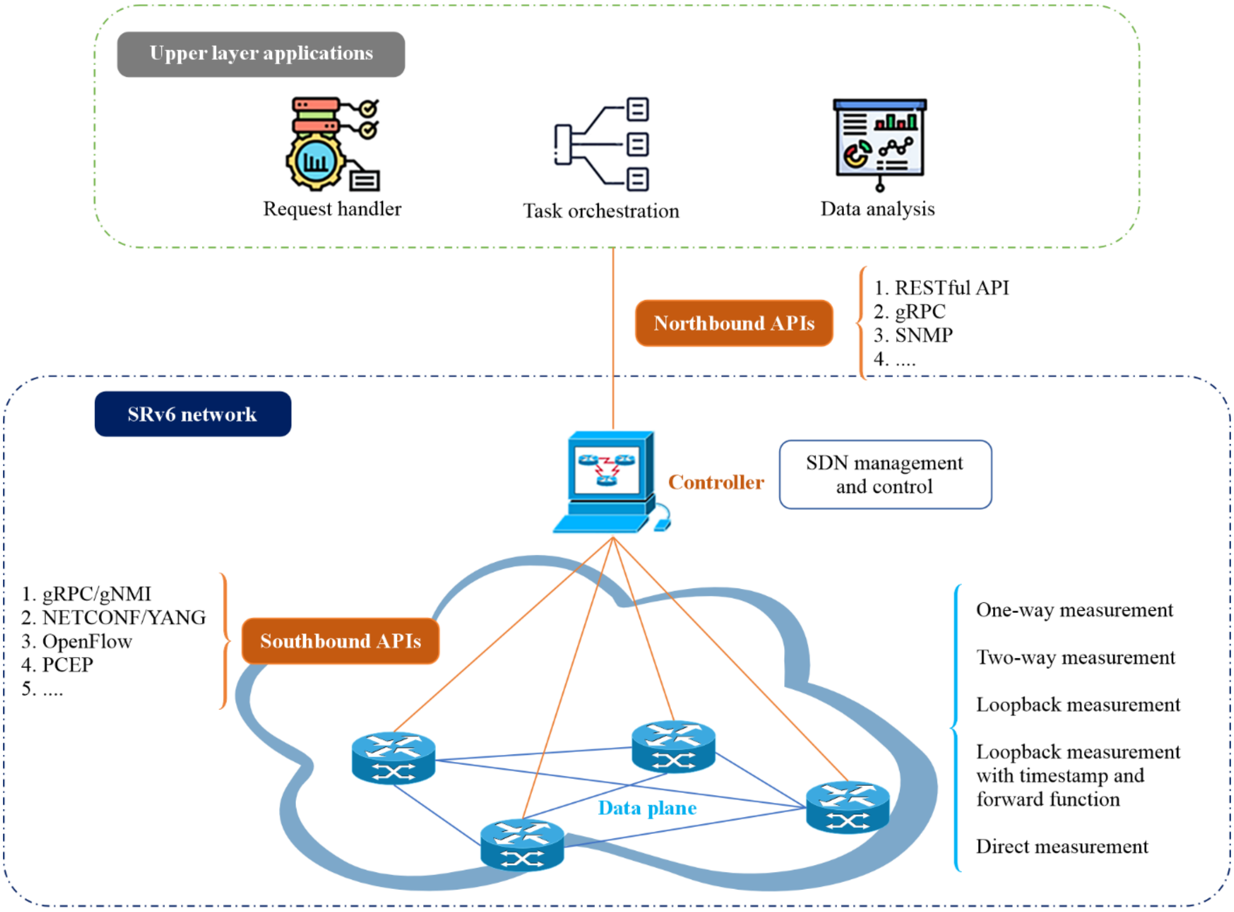
Figure 9-1 – A STAMP reference scenario

9.1 System architecture

9.1.1 System architecture

Figure 9-2 shows a typical architecture for data monitoring and analysis.

During the measurement, the performance data is constantly collected, analysed and processed. Data is reported in the form of a chart or table for understanding network performance.



QSTR.MPM-SRv6(25)

Figure 9-2 – Architecture of the PM system

For PM of the control plane, the protocols are monitored and recorded, including the correctness of the path entries, path information and segment identifier (SID) information. When failure is detected in the control plane, the process of failure detection and recovery is recorded, including the time of failure, time of detection, time of recovery and the effect of the recovery strategy.

For PM of the data plane, the forwarding performance of data packets is monitored in real time, including latency, jitter, packet loss ratio and throughput. In addition, exceptions in the data plane are detected in real time, such as sudden traffic fluctuations and path failures. The functions of each stage in the upper layer are as follows:

a) Request handler: The request handler listens to requests and parses the metadata of the requests.

b) Task orchestration: Task orchestration splits requests into configuration information and distributes them to the controller.

c) Data analysis: Data analysis includes data integration, data processing and data visualization of the data collected from the application programming interface (API).

9.1.2 Test process

As shown in Figure 9-3, there are four stages in the PM of SRv6 devices: initialization and test environment configuration, test execution, end of test, and test result evaluation.

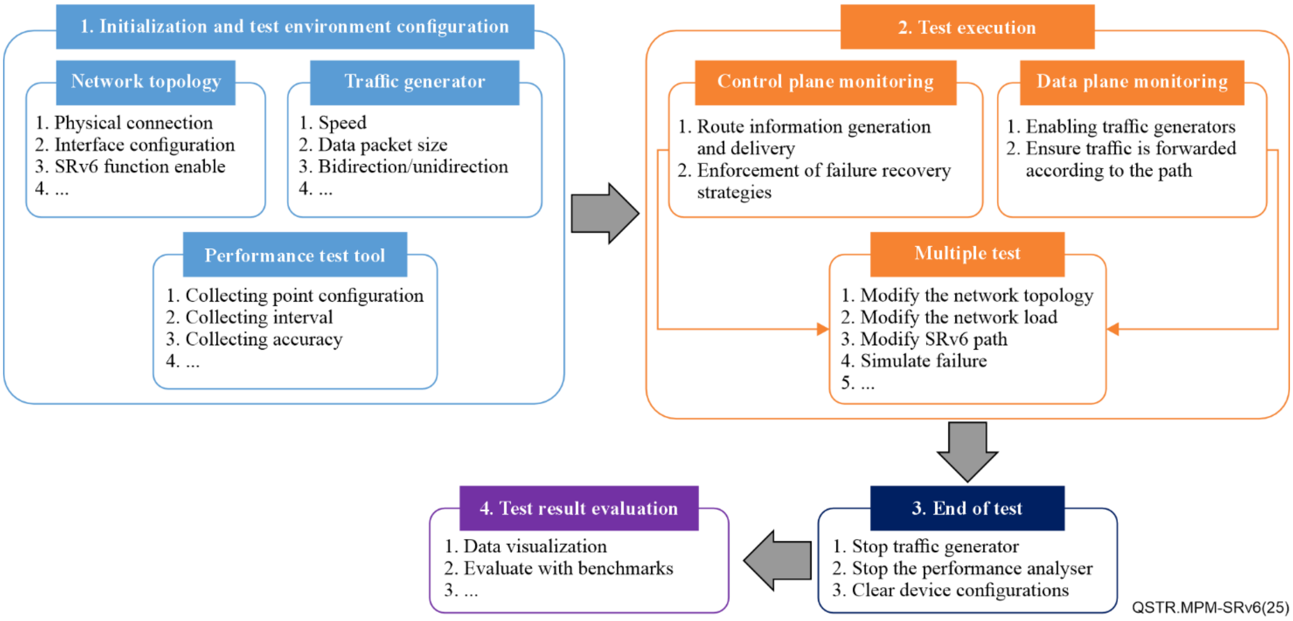

1. Initialization and test environment configuration
Network topology
1. Physical connection
2. Interface configuration
3. SRv6 function enable
4. ...
Traffic generator
1. Speed
2. Data packet size
3. Bidirection/unidirection
4. ...
Performance test tool
1. Collecting point configuration
2. Collecting interval
3. Collecting accuracy
4. ...
2. Test execution
Control plane monitoring
1. Route information generation and delivery
2. Enforcement of failure recovery strategies
Data plane monitoring
1. Enabling traffic generators
2. Ensure traffic is forwarded according to the path
Multiple test
1. Modify the network topology
2. Modify the network load
3. Modify SRv6 path
4. Simulate failure
5. ...
3. End of test
1. Stop traffic generator
2. Stop the performance analyser
3. Clear device configurations
4. Test result evaluation
1. Data visualization
2. Evaluate with benchmarks
3. ...
QSTR.MPM-SRv6(25)

Figure 9-3 – Flow chart of the test and configuration

The four stages are described as follows:

a) Initialization and test environment configuration

The SRv6 network topology is initialized and configured, including the connections between routers and switches, interface configurations and basic SRv6 functions. The parameters of the traffic generator, such as the send rate and size of the data packets, are set by the control centre. In addition, a performance measurement tool is required to be configured to collect the required performance data. A possible process for building the measurement tasks is illustrated in Figure 9-4.

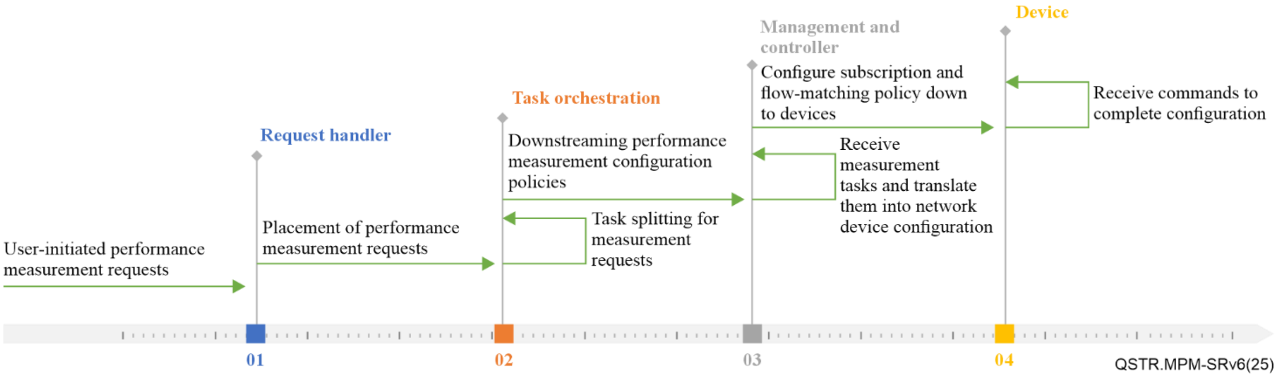

Device
Management and controller
Configure subscription and flow-matching policy down to devices
Receive commands to complete configuration
Task orchestration
Request handler
Downstreaming performance measurement configuration policies
Receive measurement tasks and translate them into network device configuration
Placement of performance measurement requests
Task splitting for measurement requests
User-initiated performance measurement requests
01
02
03
04
QSTR.MPM-SRv6(25)

Figure 9-4 – Flowchart of a measurement task build and release

Figure 9-4 describes the flow of the measurement task. It begins by initiating a service performance measurement request from users. The request is then sent to the request handler. The task orchestration breaks down the measurement request into the performance measurement configuration policies and sends them to the management and controller. The management and controller then decomposes them into network equipment commands and finally passes the commands to devices.

b) Test execution

For the control plane, the SRv6 information is synchronized to the controller in accordance with the routing protocols. The controller calculates the SRv6 path according to the user path constraints, generates the path information and then distributes the preconfigured SRv6 path via the southbound protocol. If a network failure is detected in the control plane, a predefined failure recovery strategy is executed.

For the data plane, the traffic generator is activated to simulate the network traffic transmission. Each test packet is forwarded hop by hop according to the segment routing path specified in the test packet headers before reaching the destination, because the packets are forwarded and processed according to the pre-configured SRv6 paths.

During the measurement, operators can change the test conditions as required, such as the network topology, network loads and SRv6 path configurations, and repeat the tests. In addition, it is recommended to simulate network failure, to monitor the failure recovery performance. A possible process for changing the measurement tasks is illustrated in Figure 9-5.

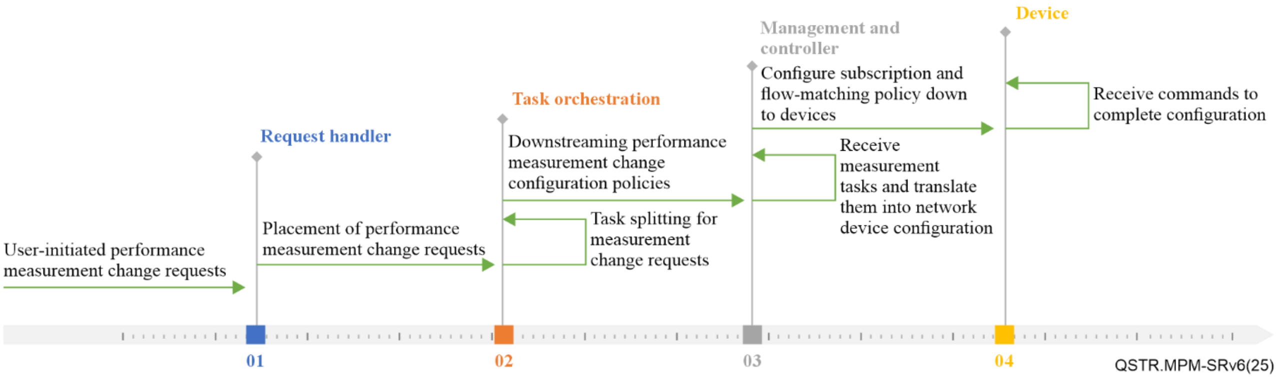

Device
Management and controller
Configure subscription and flow-matching policy down to devices
Receive commands to complete configuration
Task orchestration
Request handler
Downstreaming performance measurement change configuration policies
Receive measurement tasks and translate them into network device configuration
Task splitting for measurement change requests
Placement of performance measurement change requests
User-initiated performance measurement change requests
01
02
03
04
QSTR.MPM-SRv6(25)

Figure 9-5 – Flowchart of changing a measurement task

Figure 9-5 describes the flow of changing a measurement task, beginning with the initiation of the performance measurement change request. The request is then sent to the request handler and passed to the task orchestration. The task orchestration splits the task into performance measurement change configuration policies and sends them to the management and controller. The management and controller then decomposes them into network equipment commands and finally passes the commands to devices.

c) End of test

After a predetermined test time is reached or a specific end condition is met, the traffic generator is stopped and the test ends. Figure 9-6 shows the possible processes involved in ending the measurement task.

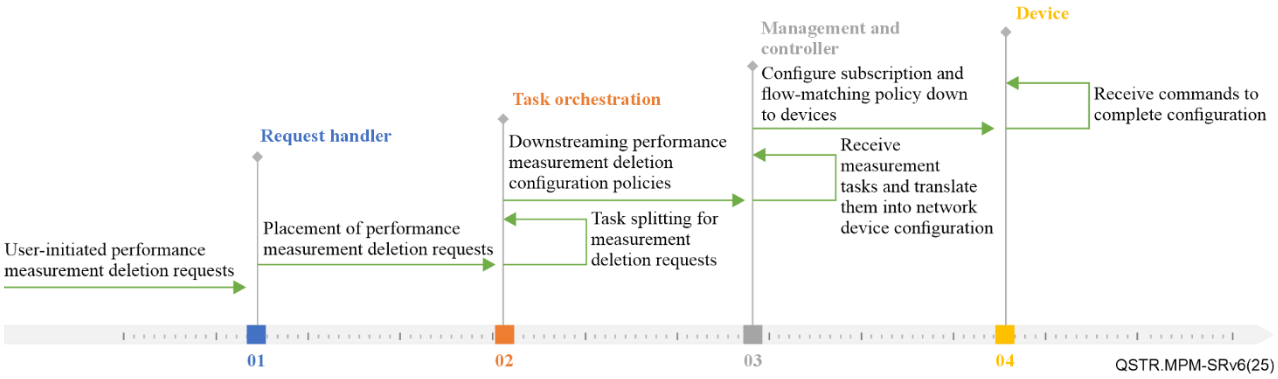

Device
Management and controller
Configure subscription and flow-matching policy down to devices
Receive commands to complete configuration
Task orchestration
Request handler
Downstreaming performance measurement deletion configuration policies
Receive measurement tasks and translate them into network device configuration
Task splitting for measurement deletion requests
Placement of performance measurement deletion requests
User-initiated performance measurement deletion requests
01
02
03
04
QSTR.MPM-SRv6(25)

Figure 9-6 – Flowchart of ending a measurement task

Figure 9-6 describes the flow of the measurement task deletion, beginning with the initiation of the performance measurement deletion request. After the request is received by the request handler, it is passed to task orchestration. The task orchestration then decomposes the task and sends the performance measurement deletion configuration policy to the management and controller. The management and controller transforms it into the configuration commands of the network equipment and finally passes the commands to devices.

d) Result evaluation

The collected performance data is analysed to evaluate the performance of the SRv6 network. For the control plane, the collected performance data is analysed to evaluate the control plane performance metrics introduced in clause 8. For the data plane, all collected performance data is summarized and organized to calculate the data plane performance metrics proposed in clause 8.

9.2 Technologies for the data plane

This clause describes the technologies involved in encoding measurement packets and the methods for monitoring the SRv6 performance in the data plane.

A STAMP session measures the performance metric between two nodes of a given SRv6 path. These two nodes are the Session-Sender and Session-Reflector. Specifically, the Session-Reflector receives packets from the Session-Sender and processes them according to the configuration and the optional control information conveyed by the Session-Sender test packets. Therefore, monitoring the data plane is realized by extending the STAMP packet header with information related to the SRv6 network, such as the IPv6 header and SRv6 SRH containing the SL, to allow the test packet to be transmitted via the expected SRv6 path, and then monitoring the packets at the destination side. Each STAMP session is labelled by a unique 16-bit, non-zero and unsigned integer, which is defined as the SSID.

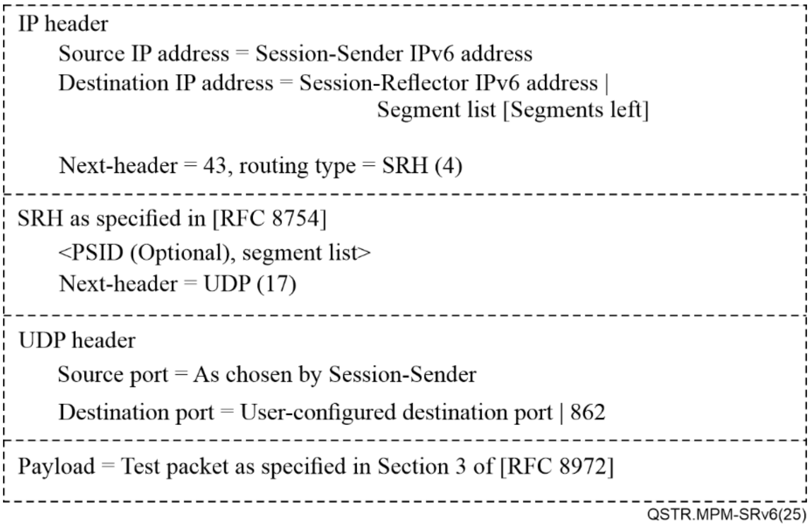
9.2.1 Encoding for measurement packets

For Session-Sender packets, an IPv6 header and an SRv6 SRH carrying the SL are required to be included to realize the transmission in each SL of the SRv6 policy candidate paths [b-IETF RFC 8754], as there are one or various SLs containing many SRv6 SID [IETF RFC 8986] in the candidate paths of the SRv6 policy. These two modes enable the embedding of SRH into extended STAMP test packets: Insert-Mode and Encap-Mode.

In the Insert-Mode, the SRH is inserted after the IPv6 header. Therefore, the test packet comprises three parts: the IPv6 header (including the source address and destination address), the SRH (including the SID list), and the STAMP test part (including the performance data).

In the Encap-Mode, the STAMP test packet is encapsulated in an external IPv6 header with the SRH. Therefore, the test packet comprises four parts: the external IPv6 header (including the source address and destination address), SRH (including the SID list), internal IPv6 header (including the source address and destination address), and the STAMP test part.

Figure 9-7 illustrates an example of a Session-Sender test packet.



IP header

Source IP address = Session-Sender IPv6 address

Destination IP address = Session-Reflector IPv6 address |
Segment list [Segments left]

Next-header = 43, routing type = SRH (4)

SRH as specified in [RFC 8754]

<PSID (Optional), segment list>

Next-header = UDP (17)

UDP header

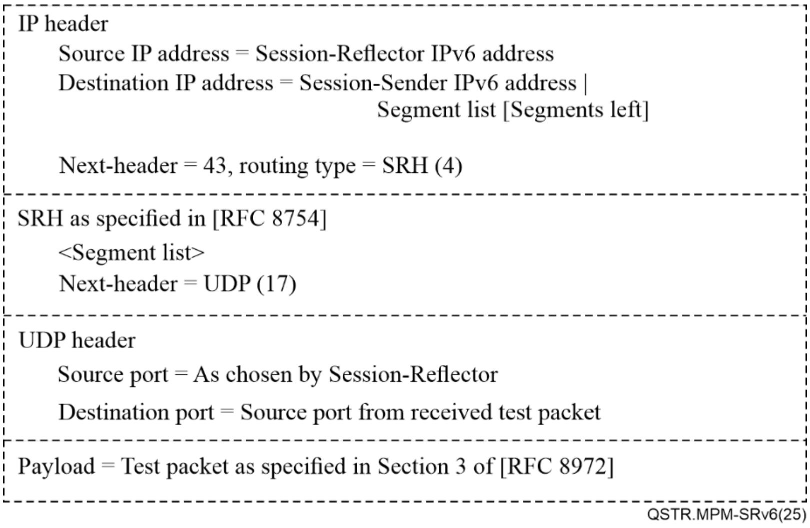
Source port = As chosen by Session-Sender

Destination port = User-configured destination port | 862

Payload = Test packet as specified in Section 3 of [RFC 8972]

Figure 9-7 – Example of a Session-Sender test packet for SRv6 networks

After the Session-Sender test packet is received by the other side of the SRv6 path, the Session-Reflector replies according to the received IP/UDP information. The test packet structure of the Session-Reflector is shown in Figure 9-8.



IP header

Source IP address = Session-Reflector IPv6 address

Destination IP address = Session-Sender IPv6 address |
Segment list [Segments left]

Next-header = 43, routing type = SRH (4)

SRH as specified in [RFC 8754]

<Segment list>

Next-header = UDP (17)

UDP header

Source port = As chosen by Session-Reflector

Destination port = Source port from received test packet

Payload = Test packet as specified in Section 3 of [RFC 8972]

Figure 9-8 – Example of Session-Reflector test packet for SRv6 policy

9.2.2 Measurement methods

9.2.2.1 One-way measurement

In a one-way measurement, the STAMP Session-Sender sends a Session-Sender test packet. After the Session-Reflector receives the test packet, no test packet is generated for replying. In this mode, it is necessary to ensure that the clocks at both ends are identical. A diagram of the one-way measurement mode is shown in Figure 9-9.

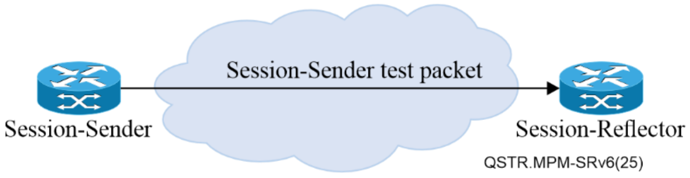


Figure 9-9 – Diagram of the one-way measurement

In this mode, the Session-Sender is recommended to use the "No Reply Requested" flag in the control code sub-type-length-value (Sub-TLV) in the return path type-length-value (TLV) defined in [IETF RFC 9503] when constructing the test packets to request the Session-Reflector not to transmit the reply to the packets, as there is no need for a Session-Reflector to reply to the packets. The SSID field in the received Session-Sender test packets and the local configuration are required to identify the same STAMP session.

9.2.2.2 Two-way (round-trip) measurement

In the two-way measurement, the session-reflector test packets illustrated in Figure 9-8 are sent in the reverse direction from the forward SRv6 path. The reverse path is either the reverse path of the same path where the Session-Sender test packets are transmitted, or a specified reverse path that is appointed by the Session-Sender test packet, as shown in Figure 9-10.

If the reverse path is the reverse path of the same path where the Session-Sender test packets are transmitted, the Session-Sender test packets are required to contain a control code Sub-TLV in the return path TLV, to ensure that the Session-Reflector transmits the reply to the test packets on the same path where the test packets are received [IETF RFC 9503]. If the reverse path is one of the reverse paths specified by the Session-Sender, the Session-Sender test packet is required to contain an SL Sub-TLV in the return path TLV, to ensure that the session-reflector transmits the reply to the test packets on the designated reverse path.

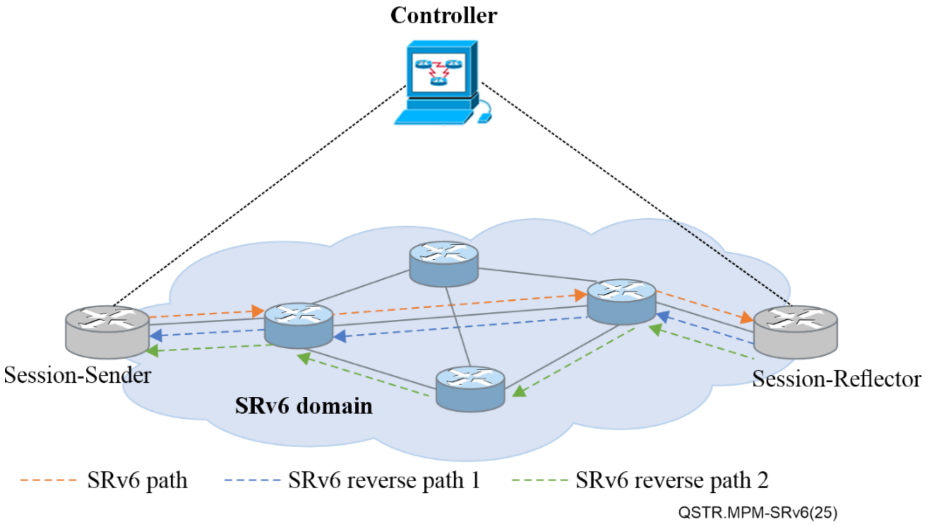


Figure 9-10 – Schematic diagram of the reverse path

In two-way measurement, the performance data is calculated based on the information carried by the test packets.

9.2.2.3 Loopback measurement

In loopback measurement, the received Session-Sender test packets are not punted out of the fast path in the data plane (i.e., to the slow path or control plane). In other words, the Session-Reflector does not perform STAMP functions or generate Session-Reflector test packets but makes the necessary changes to the encapsulation of the received test packets (including IP, SR, and UDP headers) by the loopback function, before returning the packets to the Session-Sender. The Session-Sender identifies the test session of the loopback measurement by SSID after receiving the returned test packets.

To ensure that the test packets initially possess the information of the return path, the SRH carried by the Session-Sender test packets contains either the SL of the forward SRv6 path encoded by the Encap-Mode or the SL of both the forward and the reverse paths encoded by the Insert-Mode. Specifically, if the Encap-Mode is used, an extra internal IPv6 header (after the SRH and before the

UDP header) is added, and the Session-Sender address is set as the destination address. In this case, the Session-Reflector test packet removes the SRH and uses the internal IPv6 return path to forward the packet. If the Insert-Mode is taken to make the Session-Sender test packets contain both forward and reverse SRv6 paths in the SRH; for example, the SL may include the SL of the reverse SR policy [IETF RFC 9503], the binding SID or the node SID of the reverse SR policy. The Session-Reflector retains the SRH and no additional internal IPv6 header is required. Figure 9-11 shows the schema for processing test packets in the Encap-Mode in the loopback measurement.

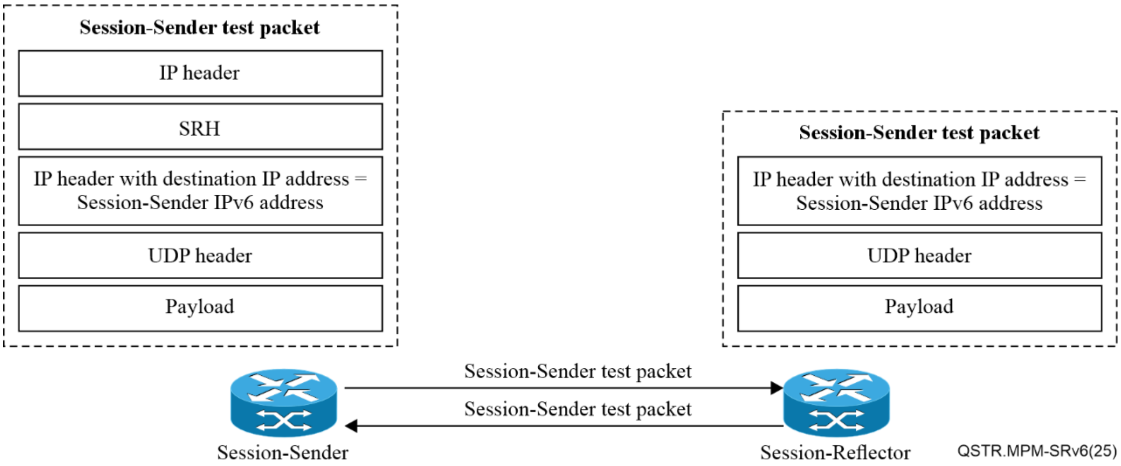

Session-Sender test packet
IP header
SRH
IP header with destination IP address = Session-Sender IPv6 address
UDP header
Payload
Session-Sender test packet
IP header with destination IP address = Session-Sender IPv6 address
UDP header
Payload
Session-Sender test packet
Session-Sender test packet
Session-Sender
Session-Reflector
QSTR.MPM-SRv6(25)

Figure 9-11 – Processing of measurement packets encoded by the Encap-Mode in loopback measurement mode

9.2.2.4 Loopback measurement with timestamp and forwarding function

In this mode, the STAMP Session-Sender initializes a Session-Sender test packet in loopback measurement mode based on SRv6 endpoint behaviours, as defined in [IETF RFC 8986], implementing "timestamp and forward (TSF) network programming functionality" to optimize the Session-Reflector's "operations of sending test packets and generating return test packets" to achieve larger and faster measurement intervals. Specifically, the Session-Sender transmits the timestamp and forwarding endpoint functions (End. TSF), along with the target SID in the SRH, to the Session-Reflector when constructing measurement packets. When the Session-Reflector receives a packet with an End.TSF for the target SID (local SID), it timestamps the test packet at a specific offset and then forwards the packet, using the SRv6 path loopback measurement mode.

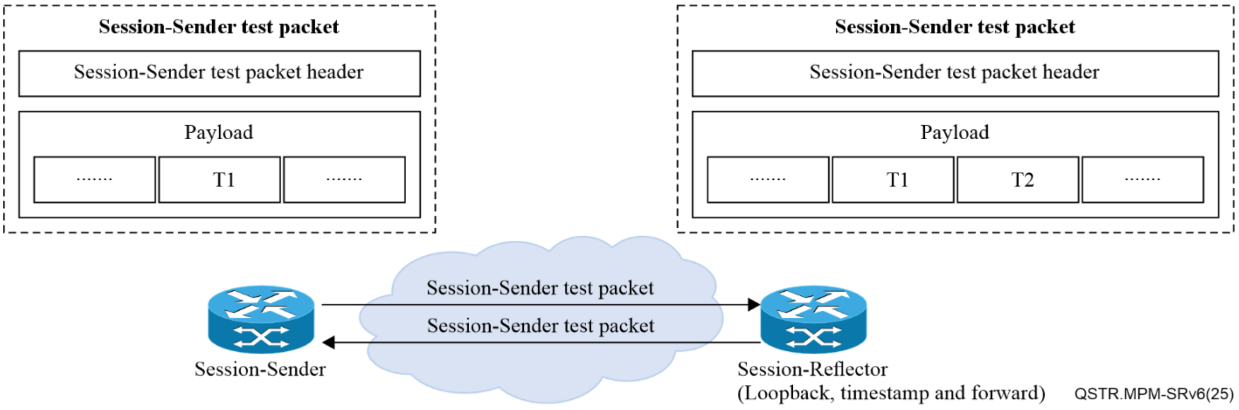
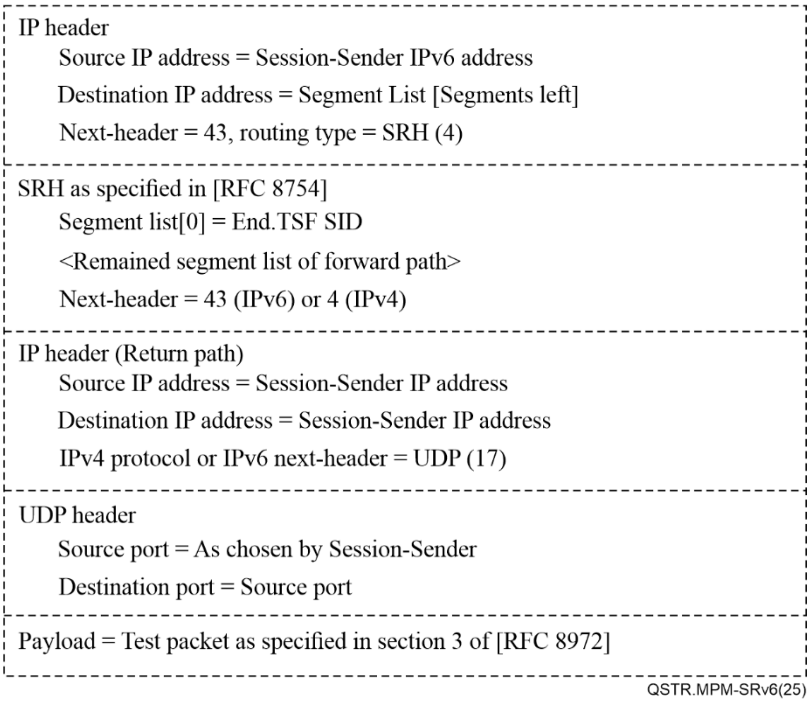

Session-Sender test packet
Session-Sender test packet header
Payload
....
T1
....
Session-Sender test packet
Session-Sender test packet header
Payload
....
T1
T2
....
Session-Sender test packet
Session-Sender test packet
Session-Sender
Session-Reflector
(Loopback, timestamp and forward)
QSTR.MPM-SRv6(25)

Figure 9-12 – Example of test packet process in loopback measurement mode with timestamp and forward function

Figure 9-12 shows an example of the loopback measurement with timestamp and forwarding function. The Session-Sender adds a transmit timestamp (T1) to the payload of the Session-Sender test packet. Because of the network programming function carried by the test packets, the Session-Reflector adds the receive timestamp (T2) at a specific offset in the payload of the received test packets in a fast path in the data plane rather than punting the test packet (e.g., to slow the path or control plane) for processing the STAMP packet.

The test packets constructed by the Session-Sender with the Encap-Mode and the Insert-Mode are illustrated in Figure 9-13 and Figure 9-14, respectively.



IP header
Source IP address = Session-Sender IPv6 address
Destination IP address = Segment List [Segments left]
Next-header = 43, routing type = SRH (4)

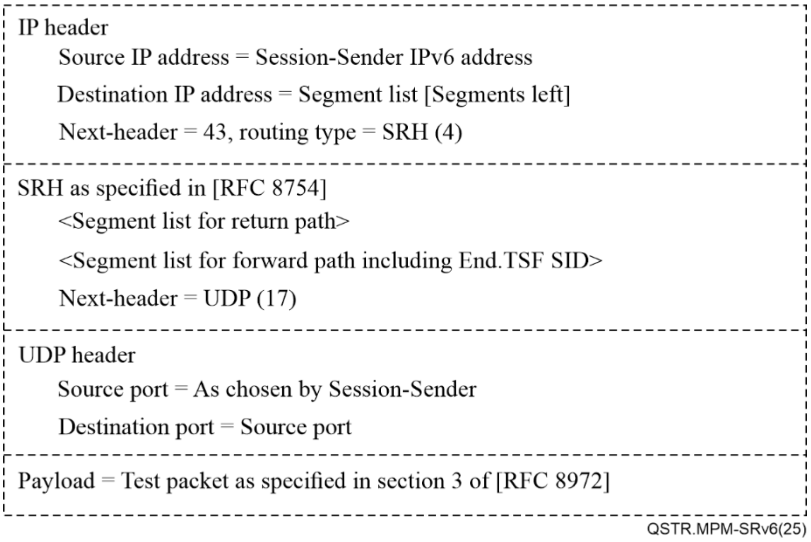
SRH as specified in [RFC 8754]
Segment list[0] = End.TSF SID
<Remained segment list of forward path>
Next-header = 43 (IPv6) or 4 (IPv4)

IP header (Return path)
Source IP address = Session-Sender IP address
Destination IP address = Session-Sender IP address
IPv4 protocol or IPv6 next-header = UDP (17)

UDP header
Source port = As chosen by Session-Sender
Destination port = Source port

Payload = Test packet as specified in section 3 of [RFC 8972]

Figure 9-13 – Example of a test packet constructed by the Encaps-Mode in loopback measurement mode with timestamp and forwarding function



IP header
Source IP address = Session-Sender IPv6 address
Destination IP address = Segment list [Segments left]
Next-header = 43, routing type = SRH (4)

SRH as specified in [RFC 8754]
<Segment list for return path>
<Segment list for forward path including End.TSF SID>
Next-header = UDP (17)

UDP header
Source port = As chosen by Session-Sender
Destination port = Source port

Payload = Test packet as specified in section 3 of [RFC 8972]

Figure 9-14 – Example of a test packet constructed by the Insert-Mode in loopback measurement mode with timestamp and forwarding function

For the timestamp format, [IETF RFC 8762] defines two timestamp formats: the network time protocol (NTP) and IEEE 1588v2 precision time protocol (PTP). By default, the STAMP session in this architecture uses NTP format, as specified in [IETF RFC 8762].

9.2.2.5 Direct measurement

In direct measurement, the STAMP session collects the number of packets from the specific data flow sent and received by the Session-Sender and Session-Reflector, respectively. Test packets sent by the Session-Sender in this mode uses the "direct measurement" TLV (Type 5), as defined in [IETF RFC 8972]. The TLV includes information regarding the counter. In the initial test packet sent by the Session-Sender, the number of transmitted test packets is filled into the Session-Sender Tx counter (S_TxC). After receiving the test packets, the Session-Reflector then fills the number of received test packets into the Session-Reflector Rx counter (R_RxC) and copies the value from the S_TxC field of the received test packet into the session-reflector Tx counter (R_TxC) before transmitting the test packets to the Session-Sender.

In addition, the path segment identifier (PSID), either for the SL or candidate path, can be added to the SL of the STAMP test packets, to measure the incoming packets of the SRv6 path on the Session-Reflector (for incoming traffic counters).

9.3 Technologies for control plane

This clause introduces the configuration, management and reporting of related technologies for monitoring SRv6 performance in the control plane.

In the monitoring framework defined in this Technical Report, the control plane takes the main responsibility for:

a) configuring and managing all STAMP sessions via the SRv6 southbound API;

b) receiving data via the northbound API of the network devices in the lower layer;

c) pushing the testing data to the data analysis platform via the northbound API.

More specifically, the controller distributes a series of configurations to the data plane for the preparation of the STAMP session, including the session parameters, path information, encoding type of data packet and measurement mode information.

SRv6 control plane interacts with the Session-Sender to obtain test data and session state information and pushes them to the data analysis platform in real time. Typically, there are two methods for the SRv6 controller to obtain data: polling and notification. In the polling mode, the controller periodically polls the Session-Sender to collect the data. In notification mode, the Session-Sender "pushes" information to the SRv6 controller by sending a single measurement record or aggregating a group of measurement records in a single notification.

This framework introduces four implementations of the southbound API: Google remote procedure call (gRPC)/gRPC network management interface (gNMI); Network Configuration Protocol (NETCONF)/Yet Another Next Generation (YANG); OpenFlow; and Path Computation Element Protocol (PCEP). For the northbound API, three implementations are introduced in this framework: RESTful API, gRPC and SNMP.

9.3.1 Southbound implementations

9.3.1.1 gRPC/gNMI

gRPC is a technology that realizes the remote procedure call (RPC) API. It provides bidirectional communication with high efficiency. gNMI is the management interface of the gRPC network. This is applicable to configuring and managing network devices. Using gRPC/gNMI, it is possible to manage STAMP sessions in a relatively large-scale network. The implementation is as follows:

a) Establishing sessions: The controller establishes sessions with devices using the gRPC.

b) Pushing configurations: The controller distributes the configuration using gNMI, such as the parameter configuration of the STAMP measurement path, including the SRv6 path and SL.

c) Real-time monitoring: The controller continuously acquires the state information of the devices and the measurement results through the real-time streaming function of the gNMI.

9.3.1.2 NETCONF/YANG

The combination of NETCONF and YANG provides a standardized interface for device configuration and management. The controller interacts with the devices using the NETCONF protocol and defines and verifies the configuration of the STAMP session using the YANG model. The configuration process is as follows.

a) Establishing sessions: The controller establishes secure sessions with devices using NETCONF.

b) Pushing configurations: The controller uses the configuration template defined by the YANG model and pushes the configuration, such as the STAMP session parameters, to devices using NETCONF.

c) Verifying and confirming: The device receives the configuration, verifies it and confirms the application of the configuration to the controller.

d) Monitoring and feedback: The controller obtains the device state information using NETCONF and monitors the STAMP measurement data.

9.3.1.3 OpenFlow

The OpenFlow protocol allows the controller to directly control the forwarding plane of the network devices to achieve centralized management of the traffic flow path. This is applicable to the flexible control and management of STAMP sessions in the SRv6 network. The configuration process is as follows:

a) Flow-table configuration: The controller issues OpenFlow flow-table entries to configure SRv6 paths, including SID lists and SRH.

b) Measurement configuration: Specify the STAMP measurement-related parameters in the flow-table entries to ensure that measurement messages are transmitted according to the configured paths.

c) Monitoring and feedback: With the OpenFlow statistics function, the controller collects the measurement data and monitors the latency and packet loss.

d) Adjustment policy: Based on the measurement results, the controller updates the flow-table entries and adjusts the path to optimize the traffic flow.

9.3.1.4 PCEP

PCEP is the communication protocol between the controller and transponder. This supports the centralized configuration and management of the SRv6 network topology information and SRv6 policy. It is also capable of configuring and managing STAMP sessions and is applicable to more complex scenarios. The configuration process is as follows:

a) Calculate path: The controller calculates the optimal SL for the SRv6 path using PCEP.

b) Send configuration: The calculated path and STAMP measurement parameters are sent to the relevant devices.

c) Measure and monitor: Devices conduct STAMP measurements according to the configured traffic path, whereas the controller obtains measurement data using PCEP.

9.3.2 Northbound implementations

9.3.2.1 RESTful API

RESTful API is a type of interface based on the Hypertext Transfer Protocol (HTTP). It is widely applied in modern network management systems and cloud services. It is suitable for interfacing with upper-tier data analytics applications with standardized interfaces. The configuration process is as follows:

a) Data acquisition: The controller collects STAMP measurement data from SRv6 devices, such as timestamps and counters, and reports them to the network management system (NMS) or operating support system (OSS) through the RESTful API.

b) Data push: The data is, periodically or event-driven, pushed to a designated uniform resource locator (URL) via an HTTP POST request.

c) Formatted data: Measurement data is formatted using JavaScript Object Notation (JSON) or Extensible Markup Language (XML) to ensure the consistency and readability of the data during transmission.

d) Fault handling: Faults and exceptions in API calls are handled to ensure the reliability and integrity of data transmission.

9.3.2.2 gRPC

By combining with telemetry or gNMI, gRPC not only provides southbound API, as described in clause 9.3.1.1, but also provides northbound API. It is more applicable for transmitting measurement data on a larger scale. The process of implementation is as follows:

a) Real-time streaming data: The controller can transmit measurement data by real-time streaming to the upper applications via gNMI or telemetry interfaces, such as NMS or the data analysis application.

b) Data subscription: Upper layer applications can subscribe specific types of measurement data and the controller pushes the relevant data on demand.

c) Efficient transmission: The efficient data serialization and transmission capabilities of the gRPC are utilized to ensure the performance and reliability of large-scale data transmission.

9.3.2.3 SNMP

SNMP is a widely used protocol for network device management. It is mainly used for device monitoring and state reporting, which can be interfaced with controllers to realize the northbound reporting of data. The implementation process is as follows:

a) Management information base (MIB) definition: The controller defines the SRv6 and STAMP measurement-related data in a specific MIB.

b) Data acquisition: The upper-layer NMS acquires measurement data from the controller via the SNMP GET request.

c) Data parsing: The upper-layer application parses the data in the SNMP response for further analysis and processing.

9.4 Data analysis implementation

The controller analyses and processes information using the collected data and control plane information. By using the obtained protocol states, link states and response states of the control commands, it can monitor the main metrics of the control plane, including the interface and link states, neighbour states, service states and command responses. With the data collected from active measurements, it can monitor the main metrics of the data plane, including packet loss ratio, latency, throughput and jitter.

The packet loss ratio can be analysed from the data associated with the counters in the direct measurement. Transmission latency and jitter can be accessed from data related to timestamps in one-way, two-way, loopback, and loopback measurements with timestamp and forwarding functions. Throughput is calculated based on a combination of the transmission latency and data related to the counters.

9.4.1 Latency

Latency is calculated from the timestamp data carried by the test packet.

In the one-way measurement, under the condition that the clocks on both sides are the same, the Session-Sender carries timestamp t1 when sending test packets, whereas the time when the Session-Reflector receives the test packets is t2. Thus, the latency for a unidirectional single trip is: $Latency = t2 - t1$.

In the two-way measurement, under the condition that the clocks on both sides are the same, the Session-Sender carries the transmit timestamp t1, the Session-Reflector carries a receive timestamp t2 and a reply timestamp t3, and the time at which the Session-Sender receives the returned test packet is t4. Thus, the latency of a single cycle is calculated using four timestamps: $Latency = (\mathrm{t4} - \mathrm{t1}) - (\mathrm{t3} - \mathrm{t2})$.

In loopback measurement, the Session-Sender retrieves timestamp t1 from the test packets returned by the Session-Reflector and collects the receive timestamp t2 locally. Both times, t1 and t2, are used to measure the loopback latency; thus, $Latency = \mathrm{t2} - \mathrm{t1}$.

In loopback measurement with timestamp and forwarding function, the Session-Sender retrieves timestamps t1 and t2 from the returned test packets by the Session-Reflector and collects the receive timestamp t3 locally. Thus, the unidirectional latency is $Latency = \mathrm{t2} - \mathrm{t1}$, whereas the loopback latency is $Latency = \mathrm{t3} - \mathrm{t1}$.

9.4.2 Jitter

Jitter is calculated from the absolute value of the latency of the neighbouring cycles. During the measurements, the same measurement mode is required for multiple measurements. After obtaining a series of timestamped data, it is possible to first calculate the latency of each cycle according to the formula described above, and finally calculate the jitter using the values of latency of two neighbouring cycles: $\mathrm{Jitter} = |\mathrm{Latency1} - \mathrm{Latency2}|$.

9.4.3 Packet loss ratio

The packet loss ratio is calculated by reading the value of each counter in the returned packets. If the packet loss ratio is measured for the forward path, the TX counter of the Session-Sender and the RX counter of the Session-Reflector are required to be used. If the packet loss ratio is calculated for the reverse path, the TX counter of the Session-Reflector is required to be read, and the corresponding RX counter is required to be read locally, to calculate the packet loss ratio in the return path.

For the forward path, the TX counter of the Session-Sender is p1 and the RX counter of the Session-Reflector is p2. For the return path, the TX counter of the Session-Reflector is p1 and the corresponding RX counter of the local counter is p2. The packet loss ratio is $\mathrm{Loss} = \frac{\mathrm{p1-p2}}{\mathrm{p1}} \times 100\%$.

9.4.4 Throughput

The throughput is calculated comprehensively by the Session-Sender, by reading the values of each counter in the returned message and combining them with the latency. If the throughput in the forward path is measured, the RX counter of the Session-Reflector is required to be used; if the throughput in the reverse path is calculated, the corresponding RX counter is required to be read locally.

For the forward path, the value of the Session-Reflector Rx counter is p1 and the size of packets that are not processed by the Session-Reflector is xMB. For the reverse path, the value of the local Rx counter is p1. The throughput is: $\mathrm{Throughput} = \frac{\mathrm{p1 \times x}}{\mathrm{Latency}}$.

10 Future work of SRv6 PM

With the large-scale implementation of SRv6, it is important to address the challenge of effectively undertaking PM and ensuring the efficiency and stability of network operations. This clause outlines the future work for SRv6 PM technology.

First, SRv6 PM technology will develop in the direction of intelligence. Advanced technologies, such as artificial intelligence and machine learning, will be utilized to monitor the network states in real time, automatically identify performance bottlenecks, and predict potential network problems. Through intelligent PM, exceptions in the network can be detected and resolved in a timely manner to ensure the continuous and stable operation of the SRv6 network.

Second, SRv6 PM techniques will focus more on end-to-end performance evaluation. In the SRv6 network, end-to-end performance is directly related to user experience and service quality. Therefore, future PM technologies will emphasize comprehensively assessing the end-to-end performance of the network from the users' perspective, including latency, jitter, packet loss rate, and other key indicators. This will help operators understand network conditions more accurately and optimize network resource allocations.

In addition, SRv6 PM technology can be integrated with network security technology. During the network PM, potential security threats are discovered and prevented in a timely manner, to ensure the security and reliability of the network. By integrating the PM and security protection functions, a more robust SRv6 network system can be built.

Finally, the standardization and openness of SRv6 PM technology will become a trend. As SRv6 technology grows in popularity, major vendors and standardization organizations will jointly promote the standardization of PM technology, to achieve interoperability and compatibility between devices from different vendors. Moreover, open-source PM platforms will attract more developers and innovators to jointly promote the continuous progress of SRv6 PM technology.

Bibliography

[b-ITU-T G.1051]	Recommendation ITU-T G.1051 (2023), *Latency measurement and interactivity scoring under real application data traffic patterns.*
[b-ITU-T Q.4061]	Recommendation ITU-T Q.4061 (2019), *Framework of software-defined network controller testing.*
[b-ITU-T Y.1540]	Recommendation ITU-T Y.1540 (2019), *Internet protocol data communication service – IP packet transfer and availability performance parameters.*
[b-ITU-T Y.1543]	Recommendation ITU-T Y.1543 (2018), *Measurements in Internet protocol networks for inter-domain performance assessment.*
[b-IETF RFC 5440]	IETF RFC 5440 (2009), *Path Computation Element (PCE) Communication Protocol (PCEP).*
[b-IETF RFC 6241]	IETF RFC 6241 (2011), *Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Circuit Emulation over Packet (CEP) MIB Using SMIv2*.
[b-IETF RFC 8402]	IETF RFC 8402 (2018), *Segment Routing Architecture.*
[b-IETF RFC 8754]	IETF RFC 8754 (2020), *IPv6 Segment Routing Header (SRH).*
[b-IETF RFC 9259]	IETF RFC 9259 (2022), *Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRv6).*
[b-draft-ietf-spring-stamp-srpm-17]	IETF draft-ietf-spring-stamp-srpm-17 (2024), *Performance Measurement Using Simple Two-Way Active Measurement Protocol (STAMP) for Segment Routing Networks.*
[b-draft-song-spring-siam-07]	IETF draft-song-spring-siam-07 (2024), *SRv6 In-situ Active Measurement.*