INTERNATIONAL TELECOMMUNICATION UNION
TELECOMMUNICATION STANDARDIZATION SECTOR
STUDY PERIOD 2025-2028

SG11-TD744/GEN
STUDY GROUP 11
Original: English

Question(s): 4/11 Geneva, 17-26 November 2025

TD

Source: Editors

Title: Agreement – draft new Supplement ITU-T Q.Suppl.79 (ex Q.sup.sdwan-srv6) "Implementation of Software-defined wide area networking (SD-WAN) service based on Segment Routing IPv6 (SRv6)" (Geneva,17-26 November 2025)

Contact:	Jiaqi Sun China Telecom China	Tel: + 86 20 38639179 E-mail: sunjiaq@chinatelecom.cn
Contact:	Zhihua Liu China Telecom China	Tel: + 86 20 38639939 E-mail:liuzh41@chinatelecom.cn
Contact:	Junya Huang China Telecom China	Tel: + 86 20 38639837 E-mail:huangjy40@chinatelecom.cn
Contact:	Ying Cheng China Unicom P.R.China	Tel: +861066259394 E-mail:chengying10@chinaunicom.cn

Abstract: This document is the output of draft new Supplement ITU-T Q.Suppl.79 (ex Q.sup.sdwan-srv6) "Implementation of Software-defined wide area networking (SD-WAN) service based on Segment Routing IPv6 (SRv6)" for agreement. This document includes the results of discussion in the Q4/SG11 meeting which was held on 17-26 November 2025.

This document is based on input document SG11-C207-R2 and the results of discussion in the Q4/11 meeting held Geneva 17-26 November 2025.

The following table shows discussion results for input documents.

Document Number	Source	Title	Meeting results
SG11-C207-R2	China telecom, China Unicom	ITU-T Q.sup.sdwan-srv6 -Proposal to advance the whole text and consent	Accepted with modification

Draft new Supplement 79 to ITU-T Q-series Recommendations

Supplement to Recommendation ITU-T Q.3741 - Implementation of Software-defined wide area networking (SD-WAN) service based on Segment Routing IPv6 (SRv6)

Summary

This draft Supplement provides an implementation of SD-WAN service based on SRv6. This draft gives more detailed information flow and concrete information message parameters of SD-WAN services based on SRv6 to better understand the signaling procedures of SD-WAN information flow and illustrate the brought out of SD-WAN services.

Keyword

SD-WAN, SRv6

Table of Contents

Draft new Supplement 79 to ITU-T Q-series Recommendations

Supplement to Recommendation ITU-T Q.3741 - Implementation of Software-defined wide area networking (SD-WAN) service based on Segment Routing IPv6 (SRv6)

1. Scope

This Supplement provides information on implementation of SD-WAN service based on SRv6 and gives concrete information message parameters needed to carry out a SD-WAN service, the scope of this Supplement includes:

- Overview for SD-WAN service based on SRv6;
- Data model for SD-WAN service based on SRv6
- Information flow of SD-WAN services based on SRv6;
- Signalling requirements of SFi over SRv6 network.

2. References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Supplement. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Supplement are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Supplement does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Q.3741]	Recommendation ITU-T Q.3741(2019)“Signalling requirements for SD-WAN service”
[Broadband Forum TR-069]	Broadband Forum TR-069(2013), “CPE WAN Mangement Protocol”
[IETF RFC 6241]	IETF RFC 6241(2011),“Network Configuration Protocol(NETCONF)”
[IETF RFC 8342]	IETF RFC 8342(2018),“Network Management Datastore Architecture”
[IETF RFC 8754]	IETF RFC 8754(2020),“IPv6 Segment Routing Header(SRH)”
[IETF RFC 8986]	IETF RFC 8986(2021),“Segment Routing over IPv6 (SRv6) Network Programming”

3. Definitions

3.1. Terms defined elsewhere

This Supplement uses the following terms defined elsewhere:

3.1.1 software-defined networking [b-ITU-T Y.3300]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

3.1.2 SD-WAN [ITU-T Q.3741]: An ecosystem of hardware (including customer-premises equipment, such as edge devices), software (including controllers) and services that enable

enterprise-grade performance, reliability and security of WAN services in various software-defined manners.

3.1.3 WAN gateway [ITU-T Q.3741]: A gateway that provides WAN access for individuals and enterprises.

3.2. Terms defined in this Supplement

This Supplement defines the following terms:

3.2.1 SD-WAN service based on SRv6: A SD-WAN service in which the overlay network is implemented with its underlay network adopting SRv6 (Segment Routing IPv6).

4. Abbreviations and acronyms

This Supplement uses the following abbreviations and acronyms:

BNG	Broadband Network Gateway
BSID	Binding Segment Identifier
CPE	Customer-Premises Equipment
CWMP	CPE WAN Management Protocol
CLI	Command Line Interface
NETCONF	Network Configuration Protocol
RPC	Remote Procedure Call
SD-WAN	Software-defined wide area networking
SID	Segment Identifier
SRv6	Segment Routing IPv6
SSL	Secure Socket Layer
SSH	Secure Shell
TCP	Transmission Control Protocol
TWAMP	Two-Way Active Measurement Protocol
URL	Uniform Resource Locator
UPF	User Plane Function
vCPE	Virtualized Customer Premises Equipment

5. Conventions

None.

6. Overview for SD-WAN service based on SRv6

6.1. Motivation

Software-defined wide-area networking (SD-WAN) is built on the same principle as software-defined networking (SDN):It abstracts the wide-area network to a set of capabilities that is independent of how those capabilities are provided. SD-WAN service connects customers' data centres, branch offices, and cloud environments, and offers virtual networks over different networks, including MPLS networks, SRv6 and public Internet.

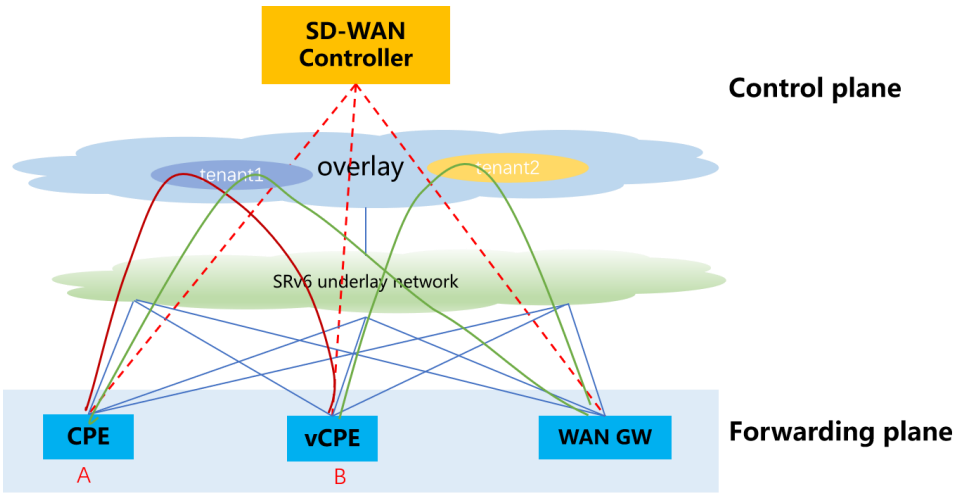

Figure 6-1. overview of SD-WAN service based on SRv6

[ITU-T Q.3741] provides a signalling architecture of SD-WAN service and gives generally information flow and message of signalling interface SFi. SRv6 is one of the protocols used on forwarding plane, which has been widely acknowledged and deployed because of its network programmability and efficiency to enable the creation of interoperable overlays with underlay optimization. As shown in Figure 6-1, the SD-WAN controller as the central management entity, configures a binding SID (BSID) within the SRv6 network via signalling interfaces. This BSID steers traffic from a source node (A) to a destination node (B) along an optimized path, which meets low-latency requirements by utilizing SRv6 segment lists and policy enforcement.

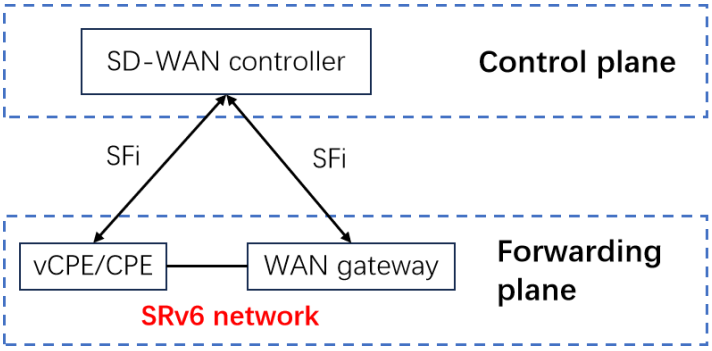

Figure 6-2 signalling architecture of SD-WAN service based on SRv6 network

This Supplement describes implementation of SD-WAN service based on SRv6 and details the required signaling information in SRv6 network scenario, including data model,information flow, information message parameters.It specifies how SD-WAN service based on SRv6 follow the CWMP(CPE WAN Management Protocol)[BBF TR069] standard with SRv6-specific enhancements to support advanced segment routing capabilities. This Supplement maintains compatibility with the established CWMP framework while introducing SRv6-specific extensions. These extensions enable vCPE/CPE devices to register SRv6 capabilities, receive locator configurations, and participate in the SRv6 service. This approach allows service providers to leverage existing CWMP infrastructure while incrementally deploying advanced segment routing capabilities for SD-WAN services.

6.2. SD-WAN service based on SRv6

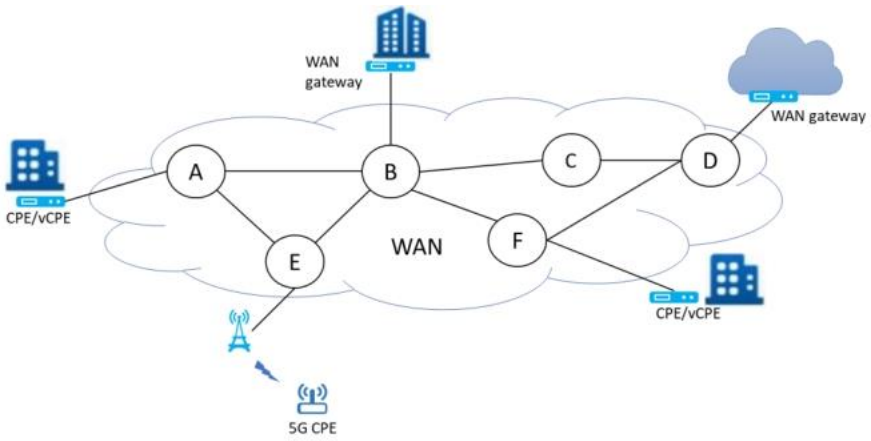

Figure 6-3 scenario of SD-WAN service

In a modern business environment, enterprises often require seamless connectivity between headquarters, multiple branches, and cloud services. By leveraging SD-WAN service based on SRv6, organizations can achieve efficient and intelligent traffic management across their wide area networks, supporting various CPE connection methods, such as fixed IP, broadband dial-up with dynamic IPs, and 5G connections, allowing branches to connect using the most suitable and available access types (Figure 6-3). The SD-WAN controller orchestrates these connections, dynamically adapting to IP changes and ensuring continuous business operations.

In implementing SRv6, fixed IP connections are straightforward, but dynamic IPs require some way to handle the changing IP address. CPE/vCPE is accessed via broadband/5G networks to obtain its Internet Protocol (IPv6) address X/Yassigned by BNG/UPF.Here, X is the IP address prefix obtained by the device, and Y is the length of the IPv6 address prefix. Based on the obtained IPv6 address prefix, the CPE/vCPE generates its SRv6 service SID,which is based on the network programming model defined in [IETF RFC 8986]. Then, by carrying these service SIDs, it advertises the VPN private network routing information to other headquarters/branches WAN gateway/CPE/vCPE, thus realizing SD-WAN service based on SRv6

The following describes the extension method of SRv6 service SID. CPE/vCPE obtains the IPv6 address with the prefix X/Y. Based on the pre-configured number of bits for the SRv6 locator of SRv6 SID on the CPE device, the first extension bit length is calculated in combination with the X/Y address prefix,then the X/Y address prefix is extended using the first extension bit length, thereby deriving the device's SRv6 SID. According to the VPN service type between the branches, CPE obtain the corresponding function field(such as End.DT4, etc.). This function field is used to identify the forwarding operation which the device will execute. By concatenating the previously generated SRv6 SID and the function field, CPE obtains a concatenated identifier. If the number of bits of the concatenated identifier is equal to 128bits, then the concatenated identifier is the service SID of the CPE. If the concatenated identifier is less than 128 bits, the second extension bit is calculated to fill the padding field, so that the concatenated identifier is expanded to 128 bits, thereby obtaining the SRv6 service SID of the CPE.

Based on the IPv6 address obtained by CPE/vCPE and the IPv6 address of the peer CPE/vCPE/WAN gateway, CPE/vCPE uses these addresses to establish a BGP peer connection relationship with the peer CPE/vCPE/WAN gateway, thereby realizing the control plane connection between them based on it.Based on this control plane connection, the peer CPE/vCPE/WAN gateway receives private network routing information carrying CPE/vCPE's SRv6 Service SID and CPE/vCPE receives the service packets sent by the peer CPE/vCPE/WAN gateway, which are transmitted based on its Service SID,thus implementing the SD-WAN service based on SRv6.If the peer CPE/vCPE access through dynamic IPs, the private network routing information carrying SRv6 service SID is advertised by the controller.

7. Data model for SD-WAN services based on SRv6

This clause defines the data model for key entities in SD-WAN services based on SRv6, including attributes of functional components such as vCPE/CPE, WAN gateway,SD-WAN service,statistic collection and dynamic WAN path control. The model supports information exchange and configuration management across signalling interfaces(e.g.,CWMP for vCPE/CPE, NETCONF for WAN gateway) as specified in clauses 8 and 9.

7.1. Data model for vCPE/CPE

The vCPE/CPE data model describes the core attributes of customer premises equipment (physical or virtual) in SD-WAN network, used for its registration, configuration and status reporting via CPE WAN Management Protocol [BBF TR069].

Table 7-1 Data model for vCPE/CPE

Element	Description
DeviceID	Unique identifier of the vCPE/CPE, consisting of Manufacturer, OUI, ProductClass, and SerialNumber .
IPv6Enable	Indicates whether IPv6 is enabled on the device
IPv6Prefix	IPv6 address prefix assigned to the vCPE/CPE (format: X/Y, where Y is the prefix length), used to generate SRv6 Service SID (see clause 6.2).
SRv6Locator	SRv6 locator derived from IPv6Prefix, including locator name, prefix value, and length (e.g., "Locator_CPE1: 2001:0C68:2100::/64").
BGPPeerStatus	Status of BGP peer connections (e.g., "Established", "Connecting")
SupportedServices	List of SRv6-based services supported by the device, e.g., "SRv6 EVPN VPWS", "SRv6 L3VPN".
ConnectionStatus	Status of the connection to the controller, e.g., "Registered", "Configuring", "Disconnected".
HardwareVersion	Hardware version of the device (reported in clause 8.1).
SoftwareVersion	Software/firmware version of the device (reported in clause 8.1).
EventCode	Code indicating the event that triggered the message (e.g., "1 BOOT", "VALUE CHANGE").

7.2. Data model for WAN gateway

The WAN gateway data model describes attributes of dedicated routing devices in the SRv6 network, used for its registration and configuration via NETCONF[IETF RFC 8342] and [IETF RFC 6241].

Table 7-2 Data model for WAN gateway

Element	Description
GatewayID	Unique identifier of the WAN gateway, for example, "GW-PE123" (static, assigned by the controller).
HardwareModel	Hardware model of the gateway, defined by the manufacturer (e.g., "SRv6-GW-5000"). Indicates hardware specifications such as port capacity and supported features.

IPv6LoopbackAddress	Fixed IPv6 address of the loopback interface, used for BGP EVPN peer establishment (clause 6.2).
SRv6Locator	SRv6 locator configured on the gateway, including name, prefix, and length (e.g., "Locator_GW1: 2001:0C68:3000::/64").
BGPNeighborList	List of BGP peers (IPv6 addresses), including CPEs, other gateways, or route reflectors (RRs).
BGPSessionState	Detailed state of BGP sessions
SupportedServices	List of SRv6-based services supported,for example,"SRv6 EVPN VPWS", "SRv6 L3VPN", "TWAMP Light".

7.3. Data model for SD-WAN Service

The SRv6 Service data model defines attributes of SRv6 VPN services.

Table 7-3 Data model for SRv6 VPN Service

Element	Description
ServiceID	Unique identifier of the SRv6 service instance (assigned by the controller).
ServiceType	Type of service carried over SRv6 (e.g., VPWS, L3VPN).
VPNInstance	Name of the associated VPN instance configured on the network device(for VPN services).
RouteDistinguisher	Route Distinguisher (RD) for the SRv6 service (unique per service).
ImportRT	Import Route Target (RT) for SRv6 service routes.
ExportRT	Export Route Target (RT) for SRv6 service routes.
LocatorName	Name of the SRv6 locator bound to the service
ServiceSID	SRv6 SID (Segment Identifier) assigned to the service.
EndSID	SID of the egress node for the SRv6 service
IngressNodeID	ID of the ingress node (CPE/gateway) where the service starts.
EgressNodeID	ID of the egress node (PE/gateway) where the service ends.
TunnelID	ID of the SRv6 tunnel. SRv6 tunnel is a path through the SRv6 network, which is defined by an ordered list of SRv6 SIDs contained in the Segment Routing Header(SRH) of IPv6 packets.
TunnelEncapsulation	Encapsulation type of the SRv6 tunnel.For example, “SRH” denotes Segment Routing Header insertion within an IPv6 packets,or “IPv6”denotes full encapsulation of the original packet within a new IPv6 header bearing an SRH(as defined in [IETF RFC 8754]).
TunnelStatus	Operational status of the SRv6 tunnel.

7.4. Data model for statistic collection

The statistics collection data model defines the attributes required for active performance monitoring in SD-WAN services based on SRv6.

Element	Description

MonitorTaskID	Unique identifier for a monitoring task, assigned by the controller. Used to associate configuration, reports, and updates for the same task.
DeviceID	Unique identifier of the monitored device (e.g., vCPE/CPE). Its composition aligns with the data model in clause 7.1.
LocalIP	The current IPv6 address of the device, used as the source IP for monitoring packets.
PeerIP	The IPv6 address of the peer device in the monitoring task, used as the destination IP for monitoring packets.
MonitorRole	The role of the device in the monitoring task. Enumerated values: "SessionSender" or "SessionReflector".
MonitorCycle	The interval (in seconds) at which the device reports monitoring data to the controller. It is configured based on SLA requirements.
MetricTypes	The list of performance metrics to be collected. Enumerated values include: "Latency"(in milliseconds), "Bandwidth"(in Mbps), and "PacketLossRate"(in percentage),jitter(in milliseconds)
ReportTimestamp	The UTC timestamp when the monitoring data was collected.
OldIP	The previous IPv6 address of the device before an IP change event. This element is used in IP change reports to clear outdated records.
NewIP	The new IPv6 address of the device after an IP change event. The controller updates the monitoring configuration based on this value.
ChangeTimestamp	The UTC timestamp when the IP address change was detected by the device.
ConfigReason	The reason for triggering a monitoring configuration update. Enumerated value: "IPAddressChanged".

7.5. Data model for dynamic WAN path

The dynamic path control data model describes attributes of traffic paths managed by the SD-WAN controller, based on SRv6 segment lists.

Table 7-4 Data model for dynamic WAN path

Element	Description
PathID	Unique identifier of the path, e.g., "Path_CPE1_GW1".
SourceDeviceID	The unique identifier of the path’s source node.This element should be the DeviceID(as defined in clause7.1) for a vCPE/CPE, or the DeviceID(as defined in clause 7.2) for a WAN gateway.
DestinationDeviceID	The unique identifier of the path's destination node. This element should be the DeviceID(as defined in clause 7.1) for a vCPE/CPE, or the GatewayID(as defined in clause 7.2) for a WAN gateway.
SegmentList	SRv6 segment list , e.g., ["2001:0C68:3000::1:End", "2001:0C68:4000::1:End"].

PathMetrics	Real-time performance metrics, including: - Latency: Current delay (e.g., "20 ms"). - Bandwidth: Available bandwidth (e.g., "100 Mbps"). - PacketLossRate: Loss ratio (e.g., "0.1%").
PathPriority	Priority level (e.g., "High" for low-latency paths, "Medium" for default paths).
Status	Path status, e.g., "Active", "Standby", "Down".
TargetBSID	Binding Segment Identifier (BSID) of the dynamic SRv6 path, a 128-bit IPv6 address that binds the SegmentList to the actual underlay forwarding path.It is used to steer business traffic to the specified dynamic path.(e.g., "2001:0C68:5000::1:BSID")

8. Information flow of SD-WAN services based on SRv6

8.1. The information flow of vCPE/CPE registration and configuration

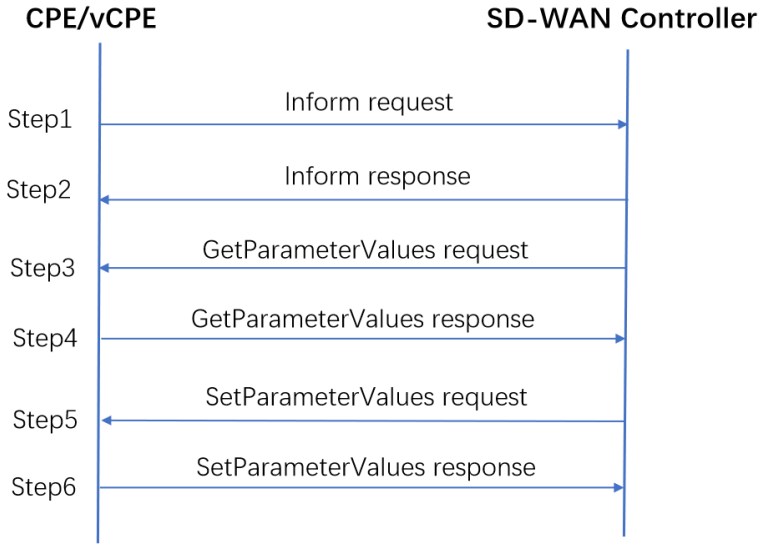

Figure 8-1 Information flow of vCPE/CPE registration and configuration

The registration and configuration process for vCPE/CPE in SD-WAN service based on SRv6 follows CWMP protocols with SRv6-specific enhancements to support advanced segment routing capabilities.The illustrated flow is initiated after the vCPE/CPE establishes a TCP connection and a secure SSL/TLS channel with the SD-WAN controller,as the standard CWMP connection setup procedures. The following steps focus on the SRv6-specific enhancements defined in this Supplement.

Step1: The CPE sends an inform message to the controller, including SRv6 capability parameters.This message contains DeviceID,IPv6Enable,SupportedService ,Hardware Version, Software Version and EventCode which is defined in clause 7.1.This allows the CPE to advertise its readiness for SRv6 configuration.

Step2: The controller processes the inform request, authenticates the device, and returns an Inform response message,including ConnectionStatus.

Step3: SD-WAN Controller sends a GetParameterValues request to query SRv6-specfic parameters essential for service setup. These parameters include device's IPv6Prefix, SRv6locator ,defined in clause 7.1.

Step4:CPE responds with a GetParameterValues reponse, returning the local IPv6Prefix and SRv6locator value to the Controller.

Step5: Based on the collected information in responses and network policies, controller sends SetParameterValues to configure SRv6 to the vCPE/CPE. This includes setting up SRv6 locators and function parameters that define the segment routing infrastructure. Concurrently, it configures the BGP peer connection unsing the CPE's IPv6 address and sets up VPN tunnels(e.g. EVPN VPWS or L3VPN instances).

Step6: The vCPE/CPE applies the received configuration parameters. It generates its SRv6 Service SID based on the obtained IPv6 prefix.It then responds with a SetParameterValuesResponse indicating success or failure for each parameter setting attempt. The CPE/vCPE implements the SRv6 configuration and establishes required protocol sessions.

8.2. The information flow of WAN GATEWAY registration and configuration

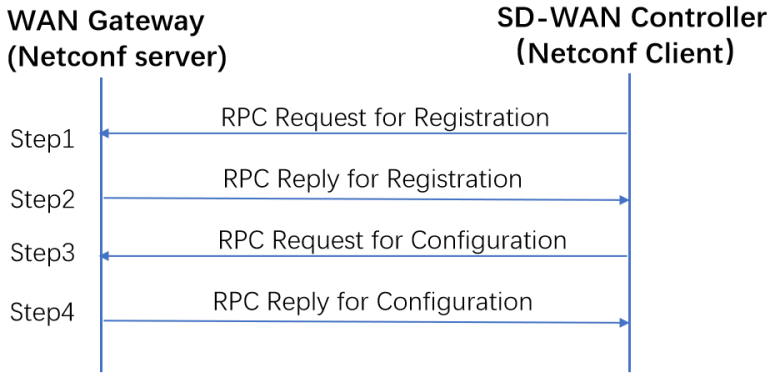

Figure 8-2 Information flow of WAN GATEWAY registration and configuration

The WAN Gateway registration and configuration in SD-WAN service based on SRv6 are implemented using the NETCONF protocol between the Gateway and the SD-WAN Controller. This implementation controls over SRv6 network elements and supports segment routing capabilities.The flow begins with standard NETCONF over SSH connection setup,including SSH handshake and capability exchange.The following steps emphasize the SRv6-enhanced process.

Step1: The SD-WAN controller sends an RPC request for gateway registration. It collects basic identity and capability information ,including SRv6-specific parameters GatewayID, IPv6LoopbackAddress, SRv6Locator and SupportedServices(defined in clause 7.2) to establish the gateway's identity and SRv6 capabilities in the controller's database.

Step2: After parsing the registration RPC request, the WAN gateway returns a RPC reply message to the controller.This reply indicates the registration result,including confirmed SRv6-specific parameters.

Step3: After successful registration, the SD-WAN Controller sends an RPC request for SD-WAN service configuration to the WAN gateway. The RPC request specifies the service type and includes service-specific parameters from SD-WAN service data model which is defined in clause 7.3, such as ServiceID, RouteDistinguisher, ImportRT, ExportRT, LocatorName, ServiceSID,TunnelID and TunnelEncapsulation.

Step4: After processing the service configuration RPC request,the WAN Gateway returns an RPC reply message to the Controller. This reply indicates the service configuration result and SRv6 service runtime information, such as BGPNeighborList,BGPSessionState,TunnelStatus. which is defined in clause 7.2 and 7.3.

8.3. The information flow of statistics collection

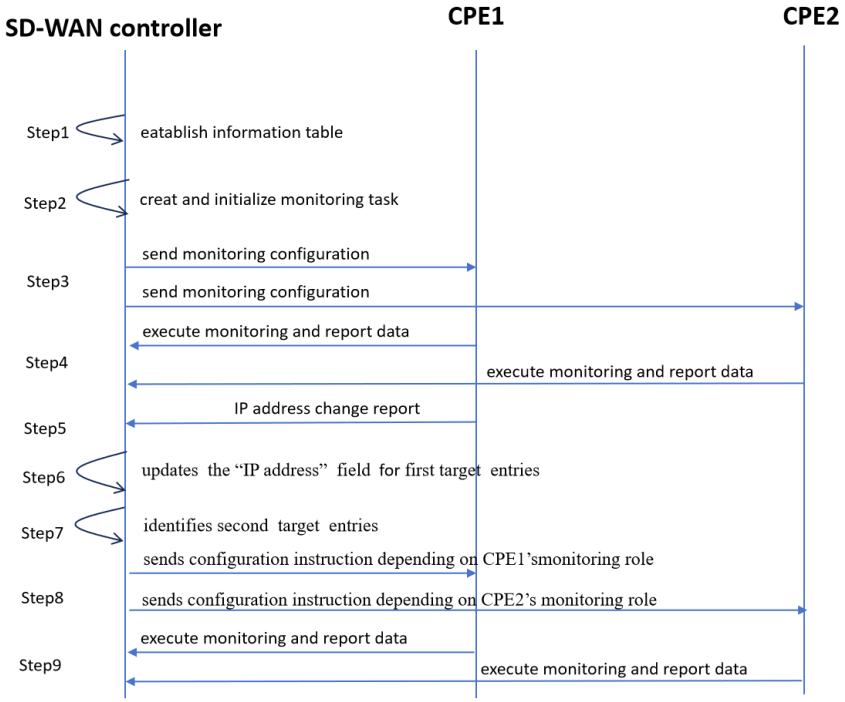

Figure 8-3 Information flow of statistics collection

The statistic collection establishes an active monitoring system where the controller maintains real-time visibility into network performance .A key capablility of this system is to ensure uninterrupted monitoring even when a CPE's IP address changes (e.g., due to broadband reconnection or 5G network handover).This is achieved through a stateful management mechanism where the controller uses a unique MonitorTaskID to track and update the monitoring session, seamlessly associating the collected data before and after the IP change.

Step1: The controller establishes an information table for network active monitoring tasks, where each entry includes a monitoring task identifier for a network active monitoring task, and a device identifier, an IP address, and a monitoring role of a terminal device corresponding to the network active monitoring task..

Step2: The controller creates and initializes network active monitoring tasks in response to a request for generating a network active monitoring task(e.g.,initiated by O&M personnel via the management interface). The controller retrieves the device information library and select devices (session sender and session reflector) matching the monitoring requirements of network active monitoring tasks,then adds an entry for the session sender and an entry for session reflector device in correspondence with the network active monitoring task in information table.

Step3: Base on the information table,the controller sends monitoring configuration instruction to the session sender and reflector device:

- Entry for the session sender device: Includes monitoring task identifier, device identifier (e.g., Serial Number), current IP address , and monitoring role ("session sender"),using the data model defined in clause 7.4;
- Entry for the session reflector device:Includes the same monitoring task identifier, device identifier, current IP address, and monitoring role ("session reflector"), using the data model defined in clause 7.4.

Step4: CPE1 and CPE2 execute the network active monitoring task according to the configuration and report monitoring data to the controller periodically.The controller utilizes these reports to evaluate the health status of the SD-WAN service in real-time.

Step5: When CPE1 with dynamic IP currently executing a network active monitoring task, detects an IP address change event, it initiates an IP address change report to the controller corresponding to the network active monitoring task. The IP address change report information includes device's unique device identifier, new IP address.

Step6: After receiving the IP address change report from CPE1 , the controller extracts the device identifier and new IP address of the CPE1 and searches the information table for the first target entry using CPE1's identifier,then updates the "IP address" field of the first target entry from the old IP address to the new IP address.

Step7: Based on the monitoring task identifier in the first target entry, the controller determines the target network active monitoring task corresponding to the CPE1, and locates the second target entry in the information table that corresponds to the target network active monitoring task,and then determines, from the second target entry, the second terminal device(CPE2 in figure8-3) associated with the target network active monitoring task.

Step8: Based on the monitoring role of the CPE1 in first target entry, the controller sends network monitoring configuration instruction to CPE1/CPE2 to instruct CPE1/CPE2 to execute network active monitoring task based on new IP address. Upon CPE1 and CPE2 receive the new monitoring configuration instruction from the controller ,they execute monitoring based on the new configuration instruction.

- If the device's(CPE1's) role is "session reflector": the controller sends configuration instruction to the peer terminal devices(CPE2 in figure8-3) corresponding to the target network active monitoring task.The instruction requires CPE2 to use CPE1's new IP address as the destination IP when executing the network active monitoring task.
- If the device's(CPE1's) role is "session sender": the controller sends configuration instruction directly to CPE1 corresponding to the target network active monitoring task.The instruction requires CPE1 to use its new IP address as the source IP address when executing the target monitoring task.

Step9: The controller collects and analyzes monitoring reports from CPE1 or CPE2. The monitoring reports include monitoring task identifier, source and destination IP addresses, report time, and monitoring data. The controller integrates the monitoring reports under different IP addresses based on the monitoring task identifier.

8.4. The information flow of dynamic WAN path control

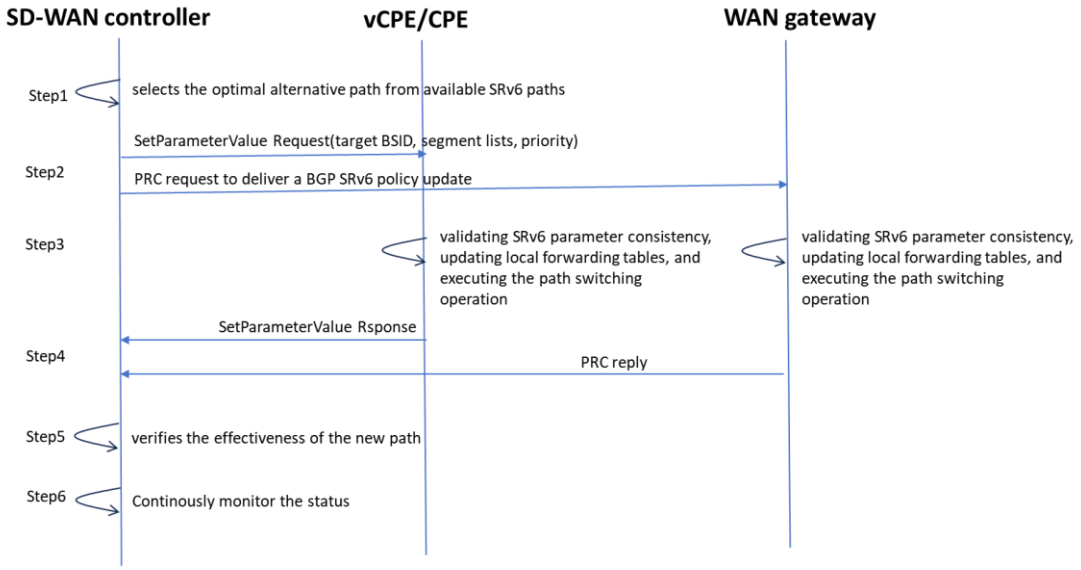

Figure 8-4 Information flow of dynamic WAN path control

Dynamic WAN path control in SRv6-based SD-WAN solutions enables intelligent path selection and switching through SRv6's network programming capabilities.When application performance degradation or network quality deterioration is detected , the controller performs analysis and decision-making based on the SRv6 policy engine.

Step1:Based on the performance analysis and business SLA requirements, the SD-WAN controller selects the optimal alternative path from available SRv6 paths. The controller generates appropriate path control instructions containing SRv6-specific parameters including Target BSID, segment list, and path priority settings. These instructions are formatted according to the target device type and management protocol .

Step2:For vCPE/CPE devices, the controller sends path control requests using the SetParameterValues in CWMP protocol. The request includes SRv6 path parameters such as target BSID, segment list and path priority. For WAN gateway devices, the controller utilizes NETCONF/YANG protocols to deliver BGP SRv6 policy updates containing new candidate paths with Target BSID, segment list and path priority.

Step3:The receiving devices (vCPE/CPE or WAN gateway) process the path control instructions by validating SRv6 parameter consistency, updating local forwarding tables, and executing the path switching operation. The devices ensure proper mapping of business flows to the new SRv6 paths while maintaining existing connections during the transition period.

Step4:Upon completing the path switching operation, the devices send response messages to the SD-WAN controller indicating the execution results. The response includes status information activated path status and current performance metrics of the new path. For vCPE/CPE devices, this uses enhanced CWMP response message, while WAN gateways employ enhanced NETCONF response messages,unsing the data model defind in clause7.5.

Step5:The SD-WAN controller verifies the effectiveness of the new path by monitoring application performance and network metrics. The controller may perform additional optimizations by fine-tuning SRv6 parameters or selecting alternative paths based on continuous performance data collection and analysis.

Step6:The controller maintains updated path information in its database and may initiate further optimizations through additional path adjustments. This continuous optimization process ensures that the SD-WAN service based on SRv6 maintains optimal performance according to the defined business policies and service level agreements.

9. Signalling requirements for SFi over SRv6 network

9.1. Overview

This clause specifies the signalling requirements for implementing SD-WAN service based on SRv6 network. It details the signalling structures, message types, transaction procedures, and parameter definitions for key processes, including vCPE/CPE and WAN Gateway registration, configuration, statistics collection, and dynamic path control. All of the messages consist of the message header and the message body. The message format is described in Figure 9-1.

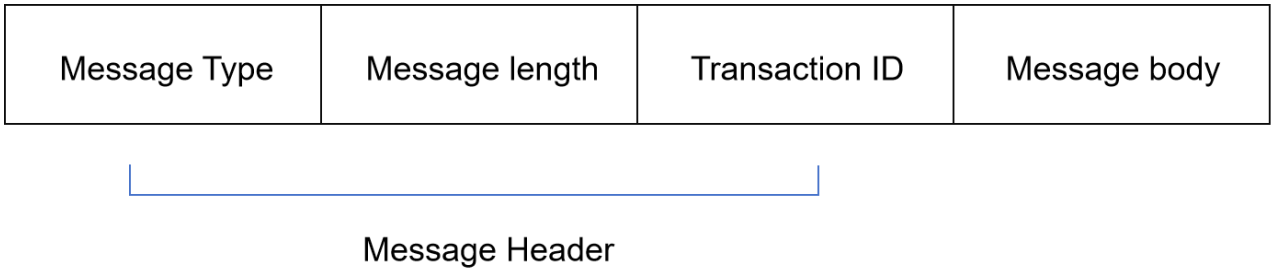

Figure 9-1 Message composition

The message header field contains the following information:
– Message type: uniquely identifies the message;
– Message length: specifies the length of the Message Body in bytes;
– Message transaction ID: generated by the initiator to track message interactions.
The message body field contains specific content of the message.

9.2. Signaling requirements for vCPE/CPE registration and configuration

According to the information flow defined in clause 8.1, when CPE device starts up, it establishes a secure HTTP connection with the SDN Controller based on the SD-WAN Controller IP address that has been configured. The vCPE/CPE registration and configuration process follows the signalling requirements specified below.

9.2.1 vCPE/CPE Inform request (Step1 in Clause 8.1)

The vCPE/CPE Inform request message is defined as CPE_Inform_Req-Message.

The CPE_Inform_Req-Message, indicated by the message type in the message header field, is sent by the vCPE/CPE to the SD-WAN controller to inform about its status and capabilities.

Message format:

```
< CPE_Inform_Req-Message > ::= < Message Header >
                {DeviceID}
                {EventCode}
                {ParameterList}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `DeviceID` uniquely specifies the identifier of the vCPE/CPE,including Manufacturer, OUI, ProductClass, SerialNumber (as specified in the example XML) which is described using the data model in clause 7.1.

（2） `EventCode` uniquely specifies the trigger event (e.g., "1 BOOT" for first startup).

（3） `ParameterList` uniquely specifies device attributes IPv6Enable, SupportedService, HardwareVersion and SoftwareVersion defined in clause7.1.

Note-· The Transaction ID of request and response messages must be identical to ensure correct mapping.

9.2.2 Controller Inform response(step 2 in clause 8.1)

The controller Inform response message is defined as CPE_Inform_Resp-Message.

The CPE_Inform_Resp-Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the vCPE/CPE to respond to the inform request.

Message format:

```
< CPE_Inform_Resp-Message > ::= < Message Header >
                {ConnectionStatus}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `ConnectionStatus` uniquely specifies the authentication and registration status of the connection from the controller to the CPE. A successful response (with ConnectionStatus indicating "Registered") concludes the initial session establishment phase

1. Controller GetParameterValues request(step3 in clause 8.1)

The controller GetParameterValues request message is defined as CPE_GetParam_Req-Message.

The CPE_GetParam_Req-Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the vCPE/CPE to query parameters essential for SRv6 SD-WAN service.

Message format:

```
< CPE_GetParam_Req-Message > ::= < Message Header >
                {ParameterList}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `ParameterList` uniquely specifies a list of parameters essential for SRv6 SD-WAN service setup.The parameters are required by the controller to access device capabilities and initial state before configuration.The list includes IPv6Prefix,SRv6Locator,using the data model in clause 7.1.

2. vCPE/CPE GetParameterValues response(step4 in clause 8.1)

The vCPE/CPE GetParameterValues response message is defined as CPE_GetParam_Resp-Message.

The CPE_GetParam_Resp-Message, indicated by the message type in the message header field, is sent by the vCPE/CPE to the SD-WAN controller to provide the values of the requested parameters.

Message format:

```
< CPE_GetParam_Resp-Message > ::= < Message Header >
                {ParameterList}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `ParameterList` uniquely specifies the values of the requested parameters. This response provides the controller with essential data to proceed with configuration.If a parameter is unavailable, the value may be omitted or set to a default.

9.2.3 Controller SetParameterValues request(step5 in clause 8.1)

The controller SetParameterValues request message is defined as CPE_SetParam_Req-Message.

The CPE_SetParam_Req-Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the vCPE/CPE to configure SRv6-based services.

Message format:

```
< CPE_SetParam_Req-Message > ::= < Message Header >
                {ParameterList}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `ParameterList` uniquely specifies parameters to configure SRv6-based services on the CPE.This includes SRv6 basic configuration and VPN service configuration(EVPN VPWS or L3VPN).SRv6 basic configuration parameters are used to enable SRv6 capability of the device and define the SRv6 Locator, serving as the foundation for VPN service which is described using the date model defined in clause 7.1(including IPv6Enable,IPv6Prefix,SRv6Locator). VPN service parameters are used to implement point-to-point Layer 2 line services or Layer 3 IP interconnection across branches which is described using the data model in clause 7.3(including ServiceType,VPNIntance,RouteDistinguisher,ImportRT,ExportRT,LocatorName,TunnelID, TunnelEncapsulation).The service type is determined by the parameter “ServiceType” in the data model.

3. 9.2.4 CPE SetParameterValues response(step6 in clause 8.1)

The vCPE/CPE SetParameterValues response message is defined as CPE_SetParam_Resp-Message.

The CPE_SetParam_Resp-Message, indicated by the message type in the message header field, is sent by the vCPE/CPE to the SD-WAN controller to confirm the configuration results.

Message format:

```
< CPE_SetParam_Resp-Message > ::= < Message Header >
                {ParameterList}
              Meanings and explanations:
```

The detailed information indicates but not limited to:

（1） `ParameterList` uniquely specifies the result of each parameter configuration operation corresponding to the SetParameterValues request. For successful configuration, the entry includes the parameter name and a status code (e.g., "0" denotes success). For failed configuration, the entry includes the parameter name, an error code (non-zero, with values defined in CWMP to represent reasons like "invalid value", "read-only parameter"), and optionally an error description.This response enables the SD-WAN Controller to verify whether SRv6 basic configurations and VPN service configurations were applied successfully on the CPE, ensuring the correctness of SRv6 SD-WAN service setup.

9.3. Signaling requirements for WAN gateway registration and configuration

WAN gateways, as dedicated routing devices in the SRv6 network (typically with leased line access and fixed IPv4/IPv6 addresses), use NETCONF over SSH for device management (consistent with [b-IETF RFC 6241], [b-IETF RFC 8342]). The WAN gateway registration and configuration process follows the signalling requirements specified below.

Controller 9.3.1 RPC request for WAN gateway registration (Step1 in clause 8.2)

The controller RPC request for WAN gateway registration message is defined as GW_Reg_Req_Message.

The GW_Reg_Req_Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the WAN gateway to collect registration parameters.

Message format:

```
<GW_Reg_Req_Message> ::= <Message Header>
```

```
                    {RegistrationParameters}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `RegistrationParameters` uniquely specifies a list of parameters to collect， including GatewayID, IPv6LoopbackAddress,SRv6Locator,SupportedServices which is described using the data model in clause 7.2.

9.3.2 WAN gateway RPC reply for WAN gateway registration (Step2 in clause 8.2)

The WAN gateway RPC reply for registration message is defined as GW_Reg_Resp-Message.

The GW_Reg_Resp-Message, indicated by the message type in the message header field, is sent by the WAN gateway to the SD-WAN controller to respond to the registration request.

Message format:

```
<GW_Reg_Resp-Message> ::= <Message Header>
                    {ConfiguredParameters}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `ConfiguredParameters` uniquely specifies the confirmed parameters if the registration is successful, which is described using the data model in clause 7.2.

9.3.3 Controller RPC Request for WAN gateway SD-WAN service configuration (Step3 in clause 8.2)

The controller RPC request for WAN gateway service configuration message is defined as GW_Service_Config_Req_Message.

The GW_Service_Config_Req_Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the WAN gateway to configure SD-WAN services.

Message format:

```
<GW_Service_Config_Req_Message> ::= <Message Header>
                    {ServiceType}
                    {ServiceParameters}
```

Meanings and explanations:

The detailed information indicates but not limited to:

（1） `ServiceType` uniquely specifies the type of service carried over SRv6 ,supporting "EVPN VPWS"(Layer 2) or "SRv6 L3VPN"(Layer3).

（2） `ServiceParameters` uniquely specifies service-specific parameters which is described using the data model in clause 7.3,including ServiceID, RouteDistinguisher, ImportRT/ExportRT, LocatorName, serviceSID,TunnelIDand TunnelEncapsulation.

9.3.4 WAN gateway RPC reply for WAN gateway SD-WAN service configuration (Step4 in clause 8.2)

The WAN gateway RPC reply for service configuration message is defined as GW_Service_Config_Resp_Message.

The GW_Service_Config_Resp_Message, indicated by the message type in the message header field, is sent by the WAN gateway to the SD-WAN controller to confirm the service configuration.

Message format:

```
<GW_Service_Config_Resp_Message> ::= <Message Header>
                    {ServiceStatus}
                    {ServiceRuntimeInfo}
```

Meanings and explanations:

The detailed information indicates but not limited to:

(1) ServiceStatus uniquely specifies the status code indicating the result. For example, "0" for success, "301" for invalid ServiceID format.

(2)

1. ServiceRuntimeInfo uniquely specifies the operational status if successful, returns operational status, such as BGPNeighborList/ BGPSessionState and TunnelStatus which is described using the data model in clause 7.2 and 7.3.

9.4. Signaling requirements for statistics collection

The signaling for statistics collection is implemented based on CWMP and the process follows the signaling requirements specified below,unsing the data model defined in clause 7.4.

9.4.1 Monitoring configuration request (Step3 in clause 8.3)

The monitoring configuration request message is defined as Stats_Monitor_Config_Req_Message.

The Stats_Monitor_Config_Req_Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the target device to configure monitoring tasks.

Message format:

```
<Stats_Monitor_Config_Req_Message> ::= <Message Header>
                    {MonitorTaskID}
                    {DeviceID}
                    {LocalIP}
                    {PeerIP}
                    {MonitorRole}
                    {MonitorCycle}
                    {MetricTypes}
```

Meanings and explanations:

The detailed information indicates but not limited to:

(1) MonitorTaskID uniquely specifies a unique identifier for the monitoring task to associate all interactions in the statistics collection process.
(2) DeviceID uniquely specifies the unique identifier of the target device.
(3) LocalIP: uniquely specifies the current IPv6 address of the target device , used as the source IP for monitoring packets .
(4) PeerIP uniquely specifies the IPv6 address of the peer device , used as the destination IP for monitoring packets.
(5) MonitorRole uniquely specifies the monitoring role of the device, with enumerated values: "SessionSender" or "SessionReflector" .
(6) MonitorCycle uniquely specifies the interval for reporting monitoring data, configured by the

controller based on service-level agreement (SLA) requirements.
(7) MetricTypes uniquely specifies the list of performance metrics to be collected, with enumerated values: "Latency", "Bandwidth", and "PacketLossRate".

9.4.2 Monitoring data report (Step4 in clause 8.3)

The monitoring data report message is defined as Stats_Monitor_Data_Resp_Message.

The Stats_Monitor_Data_Resp_Message, indicated by the message type in the message header field, is sent by the CPE/vCPEs to the SD-WAN controller to report monitoring data.

Message format:

```
<Stats_Monitor_Data_Resp_Message> ::= <Message Header>
                    {MonitorTaskID}
                    {ReportTimestamp}
                    {LocalIP}
                    {PeerIP}
                    {MetricTypes}
```

Meanings and explanations:

(1) MonitorTaskID: uniquely specifies the identifier that matches the MonitorTaskID in Clause 9.4.1 to ensure data is mapped to the correct task.
(2) ReportTimestamp: uniquely specifies the UTC timestamp when the data was collected, used for time-series performance analysis.
(3) LocalIP: uniquely specifies the source IPv6 addresses of the monitoring packets, consistent with the configuration in Clause 9.4.1.
(4) PeerIP: uniquely specifies the destination IPv6 addresses of the monitoring packets, consistent with the configuration in Clause 9.4.1.
(5) MetricTypes: uniquely specifies the list of performance metrics,including "Latency"(The real-time network delay), "Bandwidth"(The available network bandwidth), and "PacketLossRate"(The packet loss ratio).

9.4.3 IP address change report (Step5 in clause 8.3)

The IP address change report message is defined as Stats_IP_Change_Report_Message.

The Stats_IP_Change_Report_Message, indicated by the message type in the message header field, is sent by the CPE/vCPEsto the SD-WAN controller to report an IP address change event.

Message format:

```
<Stats_IP_Change_Report_Message> ::= <Message Header>
                    {MonitorTaskID}
                    {DeviceID}
                    {OldIP}
                    {NewIP}
                    {ChangeTimestamp}
                    {EventCode}
```

Meanings and explanations:

(1) DeviceID uniquely specifies the unique identifier of the CPE/vCPEs with the IP change.
(2) MonitorTaskID uniquely specifies the identifier of the monitoring task affected by the IP change, ensuring the controller only updates relevant configurations.

(3) `OldIP` uniquely specifies the IPv6 address of the device before the change, used by the controller to clear outdated records.
(4) `NewIP` uniquely specifies the new IPv6 address of the device.
(5) `ChangeTimestamp` uniquely specifies the UTC timestamp when the IP change occurred.
(6) `EventCode` uniquely specifies a code specifying the type of event that triggered this report.In the context of IP address change, this would typically be a predefined value IP_CHANGE".

9.4.4 Updated monitoring configuration request (Step8 in clause 8.3)

The updated monitoring configuration request message is defined as Stats_Monitor_Update_Req_Message.

The Stats_Monitor_Update_Req_Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the CPE/vCPEs to update monitoring configuration after an IP change.

Message format:

```
<Stats_Monitor_Update_Req_Message> ::= <Message Header>
                  {MonitorTaskID}
                  {DeviceID}
                  {LocalIP}
                  {PeerIP}
                  {MonitorRole}
                  {ConfigReason}
```

Meanings and explanations:

(1) `MonitorTaskID` uniquely specifies the identifier that matches the MonitorTaskID in the IP change report to maintain task continuity.
(2) `DeviceID` uniquely specifies the unique identifier of the CPE/vCPEs for the updated configuration.
(3) `LocalIP` uniquely specifies the updated local IPv6 address (only required for the IP-changed device).
(4) `PeerIP` uniquely specifies the updated peer IPv6 address (only required for the peer device).
(5) `MonitorRole` uniquely specifies the monitoring role, remaining consistent with the original configuration ("SessionSender" or "SessionReflector") to avoid role conflicts.
(6) `ConfigReason` uniquely specifies the reason for the configuration update, with a fixed enumerated value: "IPAddressChanged".

9.5. Signaling requirements for dynamic WAN path control

Dynamic SD-WAN path control signaling supports intelligent path selection and switching for SRv6 network.The dynamic SD-WAN path control process follows the signalling requirements specified below and uses data model defined in clause 7.5.

9.5.1 Dynamic path configuration request (Step2 in clause 8.4)

The dynamic path configuration request message is defined as DC_Path_Config_Req_Message.

The DC_Path_Config_Req_Message, indicated by the message type in the message header field, is sent by the SD-WAN controller to the target device (vCPE/CPE or WAN gateway) to configure dynamic paths.

Message format:

```
<DC_Path_Config_Req_Message> ::= <Message Header>
                    {PathID}
                    {SourceDeviceID}
                    {DestinationDeviceID}
                    {SegmentList}
                    {TargetBSID}
                    {PathPriority}
```

Meanings and explanations:

(1) PathID: uniquely specifies the unique identifier of the path.
(2) SourceDeviceID: uniquely specifies the ID of the path's source device.
(3) DestinationDeviceID: uniquely specifies the ID of the path's destination device.
(4) SegmentList: uniquely specifies the SRv6 segment list.
(5) TargetBSID: uniquely specifies the Binding Segment Identifier (BSID) of the dynamic SRv6 path.
(6) PathPriority: uniquely specifies the priority level of the path.

9.5.2 Dynamic path configuration response (Step4 in clause 8.4)

The dynamic path configuration response message is defined as DC_Path_Config_Resp_Message.

The DC_Path_Config_Resp_Message, indicated by the message type in the message header field, is sent by the device to the SD-WAN controller to confirm the path configuration.

Message format:

```
<DC_Path_Config_Resp_Message> ::= <Message Header>
                    {PathID}
                    {Status}
                    {PathMetrics}
```

Meanings and explanations:

(1) PathID: uniquely specifies the identifier that matches the PathID in the request message to ensure task mapping.
(2) Status: uniquely specifies the result of path configuration indicating whether path switching is completed.
(3) PathMetrics uniquely specifies the current performance metrics of the new path.

Bibliography

[b-ITU-T Y.3300] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking*.

Appendix I

Example message body of different signalling procedures described in clause 9

I.1 vCPE/CPE registration and configuration messages

(1) CPE Inform request (Step 1 in Clause 8.1)

Example Message Body(simplified):

```
<cwmp:InformRequest>
<DeviceId>
  <Manufacturer>ABC Corp</Manufacturer>
  <SerialNumber>SN123456</SerialNumber>
</DeviceId>
<EventCode>1 BOOT</EventCode>
<ParameterList>
  <Name>Device.IPv6enable</Name>
  <Value>true</Value>
  <NameDevice.SupportedServices</Name>
  <Value>SRv6 EVPN VPWS,SRv6 L3VPN</Value>
</ParameterList>
```

(2) Controller Inform response(step 2 in clause 8.1)

Example Message Body:

```
<cwmp:InformResponse>
  <ConnectionStatus>Registered</ConnectionStatus>
</cwmp:InformResponse>
```

(3) Controller GetParameterValues request(step 3 in clause 8.1)

Example Message Body:

```
<cwmp:GetParameterValues>
  <ParameterNames SOAP-ENC:arrayType="xsd:string[4]">
    <string>IPv6Prefix</string>
    <string>SRv6Locator</string>
  </ParameterNames>
</cwmp:GetParameterValues>
```

(4) CPE GetParameterValues response(step 4 in clause 8.1)

Example Message Body:

```
<cwmp:GetParameterValues>
```

<ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[2]">

<ParameterValueStruct>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>IPv6Prefix</Name>

<Value>2001:db8::/32</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>SRv6Locator</Name>

<Value>2001:db8:1::/48</Value>

</ParameterValueStruct>

</ParameterList>

(5) Controller SetParameterValues request(step5 in clause 8.1)

Example Message Body:

<cwmp:SetParameterValues>

<ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[9]">

<ParameterValueStruct>

<Name>IPv6Enable</Name>

<Value>true</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>IPv6Prefix</Name>

<Value>2001:db8::/32</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>SRv6Locator</Name>

<Value>2001:db8:1::/48</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>ServiceType</Name>

<Value>L3VPN</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>VPNInstance</Name>

<Value>Branch_VPN1</Value>

</ParameterValueStruct>

<ParameterValueStruct>
<Name>RouteDistinguisher</Name>
<Value>100:1</Value>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>ImportRT</Name>
<Value>target:100:1</Value>
<ParameterValueStruct>
<Name>ExportRT</Name>
<Value>target:100:1</Value>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>ServiceSID</Name>
<Value>2001:db8:1::100</Value>
</ParameterValueStruct>
</ParameterList>

(6) CPE SetParameterValues response(step6 in clause 8.1)

Example Message Body:
<cwmp:SetParameterValuesResponse>
<ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[9]">
<ParameterValueStruct>
<Name>IPv6Enable</Name>
<Status>0</Status>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>IPv6Prefix</Name>
<Status>0</Status>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>SRv6Locator</Name>
<Status>0</Status>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>ServiceType</Name>
<Status>0</Status>
</ParameterValueStruct>

<ParameterValueStruct>

<Name>VPNInstance</Name>

<Status>0</Status>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>RouteDistinguisher</Name>

<Status>0</Status>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>ImportRT</Name>

<Status>0</Status>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>ServiceSID</Name>

<Status>1004</Status>

<Status>0</Status>

</ParameterValueStruct>

</ParameterList></cwmp:SetParameterValuesResponse>

I.2 WAN gateway registration and configuration messages

(1) Registration Request (Step1 in clause 8.2)

Example Message Body(simplified):

<rpc message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">

<gw_reg_req>

<request-type>Register</request-type>

<registration-params>

<gateway-id>GW-PE-001</gateway-id>

<ipv6-loopback>2001:DB8:FEED::1/128</ipv6-loopback>

<srv6-locator>

<name>Locator_GW1</name>

<prefix>2001:DB8:FEED:1::/64</prefix>

</srv6-locator>

<supported-services>SRv6 EVPN VPWS, SRv6 L3VPN</supported-services>

</registration-params>

</gw_reg_req></rpc>

(2) Registration Response (Step2 in clause 8.2)

Example Message Body(simplified):

<rpc-reply message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">

<gw_reg_resp>

<registration-status>0</registration-status>

<configured-params>

<gateway-id>GW-PE-001</gateway-id>

<ipv6-loopback>2001:DB8:FEED::1/128</ipv6-loopback>

<srv6-locator>Locator_GW1:2001:DB8:FEED:1::/64</srv6-locator>

</configured-params>

<error-details></error-details>

</gw_reg_resp></rpc-reply>

(3) Configuration Request (Step3 in clause 8.2)

Example Message Body(simplified):

<rpc message-id="104" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">

<gw_service_config_req>

<service-type>L3VPN</service-type>

<service-params>

<service-id>SRv6-L3VPN-002</service-id>

<route-distinguisher>65001:200</route-distinguisher>

<import-rt>target:65001:200</import-rt>

<export-rt>target:65001:200</export-rt>

<locator-name>Locator_GW1</locator-name>

<service-sid>2001:DB8:FEED:1::100</service-sid>

</service-params>

</gw_service_config_req></rpc>

(4) Configuration Response (Step4 in clause 8.2)

Example Message Body(simplified):

<rpc-reply message-id="104" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">

<gw_service_config_resp>

<service-status>0</service-status>

<service-runtime-info>

<bgp-session-state>Established</bgp-session-state>

<tunnel-status>Up</tunnel-status>

</service-runtime-info>

<service-error-details></service-error-details>

</w_service_config_resp></rpc-reply>

I.3 Statistics collection messages

(1) Monitoring configuration request

Example Message Body(simplified):

<cwmp:Stats_Monitor_Config_Req>

<ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[7]">

<ParameterValueStruct>

<Name>MonitorTaskID</Name>

<Value>001</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>DeviceID</Name>

<Value>

<SerialNumber>SN123456</SerialNumber>

</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>LocalIP</Name>

<Value>2001:0C68:2100::1/64</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>PeerIP</Name>

<Value>2001:0C68:2200::1/64</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>MonitorRole</Name>

<Value>SessionSender</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>MonitorCycle</Name>

<Value>30</Value>

</ParameterValueStruct>

<ParameterValueStruct>

<Name>MetricTypes</Name>

<Value>Latency,Bandwidth,PacketLossRate</Value>

```
</ParameterValueStruct>
</ParameterList>
</cwmp:Stats_Monitor_Config_Req>
```

(2) Monitoring data report

Example Message Body(simplified):

```
<cwmp:Stats_Monitor_Data_Resp>
 <ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[8]">
  <ParameterValueStruct>
   <Name>MonitorTaskID</Name>
   <Value>001</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
   <Name>ReportTimestamp</Name>
   <Value>2025-11-18T10:30:00Z</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
   <Name>LocalIP</Name>
   <Value>2001:0C68:2100::1/64</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
   <Name>PeerIP</Name>
   <Value>2001:0C68:2200::1/64</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
   <Name>Latency</Name>
   <Value>18</Value> <!-- Unit: ms -->
  </ParameterValueStruct>
  <ParameterValueStruct>
   <Name>Bandwidth</Name>
   <Value>95</Value> <!-- Unit: Mbps -->
  </ParameterValueStruct>
  <ParameterValueStruct>
   <Name>PacketLossRate</Name>
   <Value>0.05</Value> <!-- Unit: % -->
  </ParameterValueStruct>
  <ParameterValueStruct>
```

```
    <Name>DeviceID</Name>
    <Value>
      <SerialNumber>SN123456</SerialNumber>
    </Value>
   </ParameterValueStruct>
  </ParameterList>
</cwmp:Stats_Monitor_Data_Resp>
```

(3) IP address change report

Example Message Body(simplified):

```
<cwmp:Stats_IP_Change_Report>
   <ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[5]">
   <ParameterValueStruct>
    <Name>MonitorTaskID</Name>
    <Value>001</Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>DeviceID</Name>
    <Value>
     <SerialNumber>SN123456</SerialNumber>
    </Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>OldIP</Name>
    <Value>2001:0C68:2100::1/64</Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>NewIP</Name>
    <Value>2001:0C68:2100::2/64</Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>ChangeTimestamp</Name>
    <Value>2025-11-18T11:15:30Z</Value>
   </ParameterValueStruct>
  </ParameterList>
</cwmp:Stats_IP_Change_Report>
```

(4) Updated monitoring configuration request

Example Message Body(simplified):

```
<cwmp:Stats_Monitor_Update_Req>
  <ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[6]">
   <ParameterValueStruct>
    <Name>MonitorTaskID</Name>
    <Value>001</Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>DeviceID</Name>
    <Value>
     <SerialNumber>SN654321</SerialNumber> <!-- Peer device CPE2 -->
    </Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>UpdatedLocalIP</Name>
    <Value></Value> <!-- Empty: no change to CPE2's local IP -->
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>UpdatedPeerIP</Name>
    <Value>2001:0C68:2100::2/64</Value> <!-- CPE1's new IP -->
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>MonitorRole</Name>
    <Value>SessionReflector</Value> <!-- CPE2's role remains unchanged -->
   </ParameterValueStruct>
   <ParameterValueStruct>
    <Name>ConfigReason</Name>
    <Value>IPAddressChanged</Value>
   </ParameterValueStruct>
  </ParameterList>
</cwmp:Stats_Monitor_Update_Req>
```

I.4 Dynamic SD-WAN path control messages

(1) Dynamic path configuration request

For vCPE/CPE Example Message Body(simplified):

```
<cwmp:SetParameterValues>
  <ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[8]">
    <ParameterValueStruct>
    <Name>PathID</Name>
    <Value>Path_CPE1_GW1</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>SourceDevice</Name>
    <Value>
      <SerialNumber>SN123456</SerialNumber>
    </Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>DestinationDevice</Name>
    <Value>GW-PE-001</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>SegmentList</Name>
    <Value>["2001:0C68:3000::1:End", "2001:0C68:4000::1:End"]</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>TargetBSID</Name>
    <Value>2001:0C68:5000::1:BSID</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>PathPriority</Name>
    <Value>High</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>PathMetrics</Name>
    <Value>Latency:20ms, Bandwidth:100Mbps, PacketLossRate:0.1%</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>PathStatus</Name>
    <Value>Active</Value>
  </ParameterValueStruct>
 </ParameterList>
```

```
</cwmp:SetParameterValues>
```

For WAN gateway Example Message Body(simplified):

```
<rpc message-id="201" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <DC_Path_Config_Req>
    <path-id>Path_GW1_CPE2</path-id>
    <source-device>GW-PE-001</source-device>
    <destination-device>
      <serial-number>SN654321</serial-number>
    </destination-device>
    <segment-list>
      <sid>2001:0C68:4000::1:End</sid>
      <sid>2001:0C68:5000::1:End</sid>
    </segment-list>
    <target-bsid>2001:0C68:6000::1:BSID</target-bsid>
    <path-priority>Medium</path-priority>
    <path-metrics>
      <latency>25 ms</latency>
      <bandwidth>80 Mbps</bandwidth>
      <packet-loss-rate>0.2%</packet-loss-rate>
    </path-metrics>
    <path-status>Standby</path-status>
    </dyn-path-config-req>
</rpc>
```

(2) Dynamic path configuration response

For vCPE/CPE Example Message Body(simplified):

```
<cwmp:SetParameterValuesResponse>
  <ParameterList SOAP-ENC:arrayType="cwmp:ParameterValueStruct[4]">
    <ParameterValueStruct>
      <Name>PathID</Name>
      <Value>Path_CPE1_GW1</Value> <!-- Matches request PathID -->
    </ParameterValueStruct>
    <ParameterValueStruct>
      <Name>ConfigStatus</Name>
      <Value>0</Value>
    </ParameterValueStruct>
```

```
   <ParameterValueStruct>
     <Name>ActivePathInfo</Name>
     <Value>SegmentList:["2001:0C68:3000::1:End", "2001:0C68:4000::1:End"],
CurrentLatency:18ms</Value>
   </ParameterValueStruct>
   <ParameterValueStruct>
     <Name>ErrorDetails</Name>
     <Value></Value> <!-- Empty for successful configuration -->
   </ParameterValueStruct>
  </ParameterList>
</cwmp:SetParameterValuesResponse>
```

For WAN gateway Example Message Body(simplified):

```
<rpc-reply message-id="201" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <DC_Path_Config_Resp>
    <path-id>Path_GW1_CPE2</path-id>
    <config-status>0</config-status>
    <active-path-info>
      <segment-list>
        <sid>2001:0C68:4000::1:End</sid>
        <sid>2001:0C68:5000::1:End</sid>
      </segment-list>
      <current-metrics>
        <latency>23 ms</latency>
        <bandwidth>78 Mbps</bandwidth>
        <packet-loss-rate>0.15%</packet-loss-rate>
      </current-metrics>
    </active-path-info>
    <bgp-session-state>Established</bgp-session-state>
    <error-details></error-details> <!-- Empty for successful configuration -->
  </DC_Path_Config_Resp>
</rpc-reply>
```

_____