7705 Service Aggregation Router Gen 2

Release 25.3.R2

Multiservice ISA and ESA Guide

3HE 21577 AAAA TQZZA 01
Edition: 01
April 2025

Table of contents

List of tables

List of figures

1 Getting started

1.1 About this guide

This guide describes details pertaining to Integrated Services Adapters (ISAs) and Extended Services Appliances (ESAs) and the services they provide. ISA may refer to ISA2 or an ESA-VM unless otherwise specified.

This guide is organized into functional chapters and provides concepts and descriptions of the implementation flow, as well as Command Line Interface (CLI) syntax and command usage.

Unless otherwise indicated, the topics and commands described in this guide apply only to the 7705 SAR Gen 2 platforms listed in Platforms and terminology.

Command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.

Note: Unless otherwise indicated, CLI commands, contexts, and configuration examples in this guide apply for both the classic CLI and the MD-CLI.

The SR OS CLI trees and command descriptions can be found in the following guides:

- *7705 SAR Gen 2 Classic CLI Command Reference Guide*
- *7705 SAR Gen 2 Clear, Monitor, Show, Tools CLI Command Reference Guide* (for both the MD-CLI and classic CLI)
- *7705 SAR Gen 2 MD-CLI Command Reference Guide*

Note: This guide generically covers Release 25.*x*.R*x* content and may contain some content that will be released in later maintenance loads. See the *SR OS R25.x.Rx Software Release Notes*, part number 3HE 21562 000*x* TQZZA, for information about features supported in each load of the Release 25.*x*.R*x* software. For a list of features and CLI commands that are present in SR OS but not supported on the 7705 SAR Gen 2 platforms, see "SR OS Features not Supported on SAR Gen 2" in the *SR OS R25.x.Rx Software Release Notes*.

1.2 Platforms and terminology

Note:
Unless explicitly noted otherwise, this guide uses the terminology defined in the following table to collectively designate the specified platforms.

Table 1: Platforms and terminology

Platform	Collective platform designation
7705 SAR-1	7705 SAR Gen 2

1.3 Conventions

This section describes the general conventions used in this guide.

1.3.1 Precautionary and information messages

The following information symbols are used in the documentation.

DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.

WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.

Caution: Caution indicates that the described activity or situation may reduce your component or system performance.

Note: Note provides additional operational information.

Tip: Tip provides suggestions for use or best practices.

1.3.2 Options or substeps in procedures and sequential workflows

Options in a procedure or a sequential workflow are indicated by a bulleted list. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform one of the listed options to complete the step.

Example: Options in a procedure

1. User must perform this step.
2. This step offers three options. User must perform one option to complete this step.
    - This is one option.
    - This is another option.
    - This is yet another option.

Substeps in a procedure or a sequential workflow are indicated by letters. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform two substeps (a. and b.) to complete the step.

Example: Substeps in a procedure

1. User must perform this step.
2. User must perform all substeps to complete this action.
    a. This is one substep.

b. This is another substep.

2 ISA and ESA hardware

2.1 In this section

This section provides an overview of Nokia's implementation of the virtual ISA.

Note: Virtual ISA MDAs must be configured using the commands described in the *7705 SAR Gen 2 Interface Configuration Guide*.

Note: The following conditions apply to the virtual ISA:

- Virtual ISAs cannot be intermixed within the same ISA group. This limitation applies to all virtual ISA group types.

2.2 Virtual ISA overview

The virtual ISA has no external ports, so all communication passes through the node's virtual datapath, making use of the datapath's queuing and filtering functions like other MDAs.

The actual ingress and egress throughput varies depending on the buffering and processing demands of a specific application.

3 IP tunnels

3.1 IP tunnels overview

This section discusses IP Security (IPsec)tunneling features supported by the virtual tunnel ISA. The virtual tunnel ISA functions as a resource module for the system, providing encapsulation and (for IPsec) encryption functions. The IPsec encryption functions provided by the virtual tunnel ISA are applicable for many applications including mobile backhaul, encrypted SDPs, video wholesale, site-to-site encrypted tunnel, and remote access VPN concentration.

Figure 1: 7705 SAR Gen 2 IPsec implementation architecture shows an example of an IPsec deployment, and the way this would be supported inside a 7705 SAR Gen 2.

Figure 1: 7705 SAR Gen 2 IPsec implementation architecture

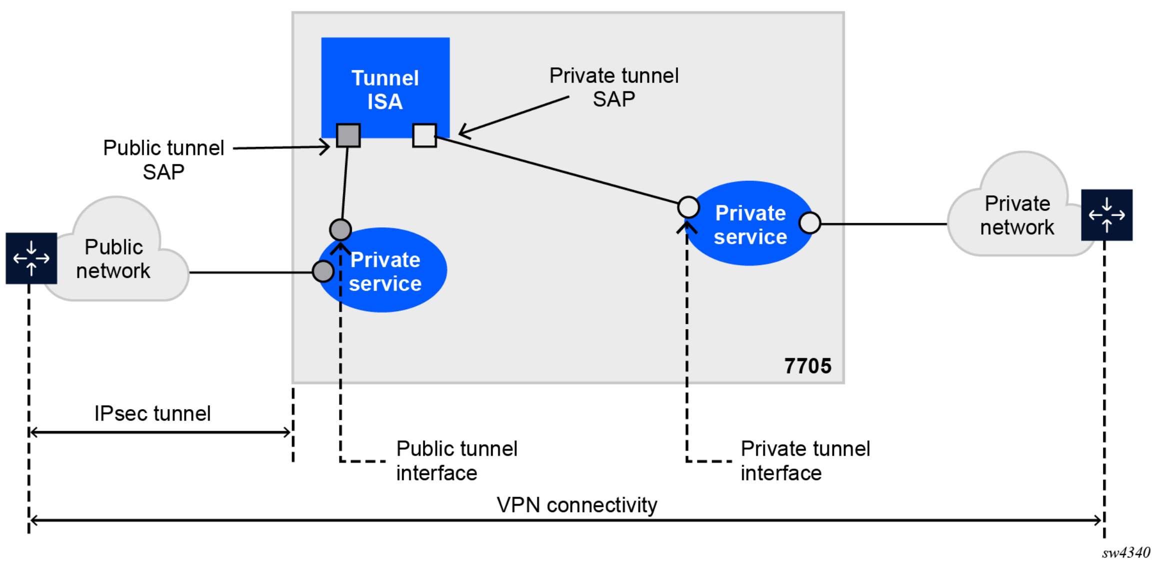


Figure 1: 7705 SAR Gen 2 IPsec implementation architecture, the public network is typically an "insecure network" (for example, the public Internet) over which packets belonging to the private network in the diagram cannot be transmitted natively. Inside the 7705 SAR Gen 2, a public service instance (IES or VPRN) connects to the public network and a private service instance (typically a VPRN) connects to the private network.

The public and private services are typically two different services, and the ISA is the only "bridge" between the two. Traffic from the public network may need to be authenticated and encrypted inside an IPsec tunnel to reach the private network. In this way, the authenticity/confidentiality/integrity of accessing the private network can be enforced.

The ISA provides a variety of encryption features required to establish bidirectional IPsec tunnels including:

control plane

- manual keying
- dynamic keying: IKEv1/v2
- IKEv1 mode: main and aggressive
- authentication: Pre-Shared-Key /xauth with RADIUS support/X.509v3 Certificate/EAP
- Perfect Forward Secrecy (PFS)
- DPD
- NAT-Traversal
- security policy
- DH-Group: 1/2/5/14/15/19/20/21

data plane

- ESP (with authentication) tunnel mode
- authentication algorithm: MD5/SHA1/SHA256/SHA384/SHA512/AES-XCBC
- encryption algorithm: DES/3DES/AES128/AES192/AES256/AES-GCM128/AES-GCM192/AES-GCM256/AES-GMAC128/AES-GMAC192/AES-GMAC256
- anti-replay protection

SR OS uses a configured authentication algorithm for the Pseudorandom Function (PRF).

There are two types of tunnel interfaces and SAPs. See Table 2: Tunnel interfaces and SAPs for more information.

Table 2: Tunnel interfaces and SAPs

Tunnel interface/SAP	Association/configuration
Public tunnel interface	configured in the public service; outgoing tunnel packets have a source IP address in this subnet
Public tunnel SAP	associated with the public tunnel interface; a logical access point to the ISA card in the public service
Private tunnel interface	configured in the private service; can be used to define the subnet for remote access IPsec clients
Private tunnel SAP	associated with the private tunnel interface, a logical access point to the ISA card in the private service

Traffic flows to and through the ISA card as follows:

- **upstream direction**

  The encapsulated (and possibly encrypted) traffic is forwarded to a public tunnel interface if its destination address matches the local or gateway address of an IPsec tunnel or the source address of a GRE or IP-IP tunnel. Inside the ISA card, encrypted traffic is decrypted, the tunnel header is removed, the payload IP packet is delivered to the private service, and from there, the traffic is forwarded again based on the destination address of the payload IP packet.

- **downstream direction**

Unencapsulated/clear traffic belonging to the private service is forwarded into the tunnel by matching a route with the IPsec/GRE/IP-IP tunnel as next-hop. The route can be configured statically, learned by running OSPF on the private tunnel interface (GRE tunnels only), learned by running BGP over the tunnel (IPsec and GRE tunnels only), or learned dynamically during IKE negotiation (IPsec only). After clear traffic is forwarded to the ISA card, it is encrypted if required, encapsulated per the tunnel type, delivered to the public service, and from there, the traffic is forwarded again based on the destination address of the tunnel header.

3.1.1 Tunnel ISAs

A tunnel group is a collection of MS-ISA2s (MDA type **isa2-tunnel**) or ESA-VM (VM type **tunnel**) configured to handle the termination of one or more IPsec tunnels.

The following example displays tunnel group configurations.

Example: MD-CLI

```
[ex:/configure isa]
A:admin@node-2# info
    tunnel-group 1 {
        admin-state enable
        isa-scale-mode tunnel-limit-2k
        primary 1/1
        backup 2/1
        }

    tunnel-group 2 {
        admin-state enable
        multi-active {
            isa 3/1 { }
            isa 3/2 { }
        }

    tunnel-group 3 {
        admin-state enable
        multi-active {
            esa 3 vm 1 { }
            esa 4 vm 1 { }
        }
```

Example: classic CLI

```
A:node-2>config>isa# info
----------------------------------------------
 tunnel-group 1 isa-scale-mode tunnel-limit-2k create
            primary 1/1
            backup 1/2
            no shutdown
        exit

        tunnel-group 2 isa-scale-mode tunnel-limit-2k create
       multi-active
       mda 3/1
       mda 3/2
       no shutdown
        exit

     tunnel-group 3 create
```

```
          multi-active
          esa-vm 3/1
          esa-vm 4/1
          no shutdown
exit
```

An IPsec tunnel belongs to only one tunnel group. There are two types of tunnel groups:

- **single-active tunnel group**

  A single-active tunnel group can have one tunnel-ISA designated as primary and, optionally, one other tunnel-ISA designated as backup. If the primary ISA fails the affected failed tunnels are re-established on the backup (which is effectively a cold standby) if it is not already in use as a backup for another tunnel group.

- **multi-active tunnel group**

  A multi-active tunnel group can have multiple tunnel-ISAs designated as primary. Only one ISA is supported on the 7705 SAR Gen 2.

The **show isa tunnel-group** command allows the user to view information about all configured tunnel groups. This command displays the following information for each tunnel group: group ID, active tunnel-ISA, administrative state, and operational state.

There are three thresholds that are used to monitor memory usage in a tunnel-ISA:

- **max-threshold**

  When the memory usage of an ISA exceeds this threshold, any new IKE states are rejected.

- **high-watermark**

  When the memory usage of an ISA exceed this threshold, a trap is generated.

- **low-watermark**

  When the memory usage of an ISA fall below this threshold, a clear trap is generated.

These three thresholds are fixed, not configurable.

A tunnel-group has an **isa-scale-mode**, which defines the maximum number of all tunnels (all types combined) which can be established on each ISA of the tunnel group. The available tunnel limits vary per platform.

3.1.1.1 Public tunnel SAPs

A VPRN or IES service (the delivery service) must have at least one IP interface associated with a public tunnel SAP to receive and process the following types of packets associated with IPsec tunnels:

- IPsec ESP (IP protocol 50)
- IKE (UDP)

The public tunnel SAP type has the format **tunnel**-*tunnel-group*.**public:***index*, as shown in the following CLI example.

Example: MD-CLI

```
[ex:/configure service]
A:admin@node-2# info
    ies "1" {
        admin-state enable
```

```
        customer "1"
        interface "public" {
            tos-marking-state untrusted
            sap tunnel-1.public:200 {
            }
            ipv4 {
                primary {
                    address 192.168.12.1
                    prefix-length 24
                }
            }
        }
    }
   vprn "2" {
        customer "1"
        bgp-ipvpn {
            mpls {
                admin-state enable
                route-distinguisher "10.1.1.1:65007"
            }
        }
       interface "greTunnel" {
            tunnel true
            ipv4 {
                addresses {
                    address 10.0.0.1 {
                        prefix-length 24
                    }
                }
                dhcp {
                    admin-state enable
                }
            }
            sap tunnel-1.private:210 {
                ip-tunnel "toCel" {
                    admin-state enable
                    delivery-service "service1"
                    remote-ip-address 10.251.12.2
                    backup-remote-ip-address 10.251.12.22
                    local-ip-address 192.168.12.100
                    gre-header {
                        admin-state enable
                    }
                    dest-ip 10.0.0.2 { }
                }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>service# info
----------------------------------------------
        customer 1 create
            description "Default customer"
        exit
        ies 1 customer 1 create
            interface "public" create
                address 192.168.12.1/24
                tos-marking-state untrusted
                sap tunnel-1.public:200 create
                exit
            exit
```

```
            no shutdown
        exit
        vprn 2 customer 1 create
            interface "greTunnel" tunnel create
                address 10.0.0.1/24
                dhcp
                    no shutdown
                exit
                sap tunnel-1.private:210 create
                    ip-tunnel "toCel" create
                        dest-ip 10.0.0.2
                        gre-header
                        source 192.168.12.100
                        remote-ip 10.251.12.2
                        backup-remote-ip 10.251.12.22
                        delivery-service 1
                        no shutdown
                    exit
                exit
            exit
            bgp-ipvpn
                mpls
                    route-distinguisher 10.1.1.1:65007
                    no shutdown
                exit
            no shutdown
        exit
```

3.1.1.2 Private tunnel SAPs

The private service must have an IP interface to an IPsec tunnel to forward IP packets into the tunnel, causing them to be encapsulated (and possibly encrypted) per the tunnel configuration and to receive IP packets from the tunnel after the encapsulation has been removed (and decryption). That IP interface is associated with a private tunnel SAP.

The private tunnel SAP has the format **tunnel**-*tunnel-group*.**private:***index*, as shown in the following CLI example where an IPsec tunnel is configured under the SAP.

Use the following command to see information about an IP tunnel.

```
show ip tunnel
```

Output example

```
===============================================================================
IP Tunnels
===============================================================================
TunnelName                       SapId                          SvcId      Admn
 Local Address                                                  DlvrySvcId Oper
  OperRemoteAddress
-------------------------------------------------------------------------------
tun-1-ipsec-tunnel                 tunnel-1.private:1             201        Up
 192.168.1.2                                                    1201       Up
  192.168.3.2
-------------------------------------------------------------------------------
IP Tunnels: 1
===============================================================================
```

3.1.1.3 IP interface configuration

In the configuration example of the previous section the IP address 10.0.0.1 is the address of the IPsec tunnel endpoint from the perspective of payload IP packets. This address belongs to the address space of the VPRN 1 service and is not exposed to the public IP network carrying the IPsec encapsulated packets. An IP interface associated with a private tunnel SAP does not support unnumbered operation.

It is possible to configure the IP MTU (M) of a private tunnel SAP interface. This sets the maximum payload IP packet size (including IP header) that can be sent into the tunnel, for example, it applies to the packet size before the tunnel encapsulation is added. When a payload IPv4 packet that needs to be forwarded into the tunnel is larger than M bytes the payload packet is IP fragmented (before tunnel encapsulation) if the DF bit is clear, otherwise the packet is discarded. When a payload IPv6 packet that needs to be forwarded into the tunnel is larger than M bytes the packet is discarded if its size is less than 1280 bytes otherwise it is forwarded and encapsulated intact.

3.1.1.4 IP fragmentation and reassembly for IP tunnels

An IPsec tunnel packet that is larger than the IP MTU of some interface in the public network must either be discarded (if the Do Not Fragment (DF) bit is set in the outer IP header) or fragmented. If the tunnel packet is fragmented, then it is up to the destination tunnel endpoint to reassemble the tunnel packet from its fragments. IP reassembly can be enabled for all the IPsec tunnels belonging to a tunnel group. When reassembly is disabled for a tunnel, all received fragments belonging to the tunnel are dropped.

To avoid public network fragmentation of IPsec packets belonging to a particular tunnel, one possible strategy is to fragment IPv4 payload packets larger than a specified size M at entry into the tunnel (before encapsulation and encryption if applicable). The size M is configurable using the **ip-mtu** command under the template, service, or router IPsec tunnel contexts.

If the payload IPv4 packets are all M bytes or less in length then it is guaranteed that all resulting tunnel packets are less than M+N bytes in length, if N is the maximum overhead added by the tunneling protocol. If M+N is less than the smallest interface IP MTU in the public network, fragmentation is avoided. In some cases, some of the IPv4 payload packets entering a tunnel may have their DF bit set. And if needed, the SR OS supports the option (also configurable on a per-tunnel basis) to clear the DF bit in these packets so that they can be fragmented.

The system allows users to configure an **encapsulated-ip-mtu** for a tunnel in the template, service, or router IPsec tunnel contexts. This represents the maximum size of the encapsulated tunnel packet. After encapsulation, If the IPv4 or IPv6 tunnel packet size exceeds the configured **encapsulated-ip-mtu**, the system fragments the packet against the **encapsulated-ip-mtu**.

The following is a description of system behavior about fragmentation:

- **private side**

  If the size, before encapsulation, of the IPv4 or IPv6 packet entering the tunnel is larger than the IP MTU configured for the template, service, or router IPsec tunnel:

  - **IPv4 payload packet**

    If the DF bit is not set in the packet or if the **clear-df-bit** command is configured, the system fragments the packet against the IP MTU configured in the template, service, or router IPsec tunnel context.

    Otherwise, the system drops the packet and sends back an ICMP error Fragmentation required and DF flag set, with the suggested MTU set as the IP MTU.

  - **IPv6 payload packet**

    If the packet size >1280 bytes, the system drops the packet and sends back an ICMPv6 Packet Too Big (PTB) message with the suggested MTU set as the IP MTU.

    If the packet size<=1280 bytes, the system forwards the packet into the tunnel.

- **public side**

  This applies to both ESP and IKE packets, IPv4 and IPv6.

  If the ESP/IKE packet is larger than the **encapsulated-ip-mtu**, the system fragments the packet against the **encapsulated-ip-mtu**; however, when the IPv6 ESP/IKE packet is smaller than 1280 bytes, the system does not fragment it, even if it is larger than the **encapsulated-ip-mtu**.

3.1.1.5 TCP MSS adjustment

The system supports the Transmission Control Protocol (TCP) Maximum Segment Size (MSS) adjustment feature for the following types of tunnels on the ISA:

- IPsec

The intent of TCP MSS adjustment is to avoid IP-level fragmentation for TCP traffic encapsulated in a tunnel by updating the MSS option value in the TCP SYN packet with an appropriate value. This feature is useful when there is tunnel encapsulation that is not known by a TCP host, and the extra tunnel encapsulation overhead may cause IP-level fragmentation.

The system supports TCP MSS adjustment on both the public and private sides.

On the public side, when the ISA receives a tunnel packet (such as ESP), after decryption or decapsulation, if the payload packet is a TCP SYN packet, then the ISA replaces the MSS option with a configured value if the configured MSS value is smaller than the received MSS value or when there is no MSS option:

- If **public-tcp-mss-adjust auto** is configured, then:

  new MSS value =public_side_MTU – tunnel_overhead – TCP fixed header – IP fixed header

  where:

  - public_side_MTU = **encapsulated-ip-mtu**

    If **encapsulated-ip-mtu** is not configured, which means there is no post-encap fragmentation on ISA, then TCP MSS adjust is disabled.

  - TCP fixed header = 20
  - IP fixed header = 20 (Ipv4) or 40 (IPv6)

- If a specific MSS value for the **public-tcp-mss-adjust** is configured, the new MSS value is set to the **public-tcp-mss-adjust** value.

Note:

- The **public-tcp-mss-adjust auto** command only applies to IPsec and IPinIP/GRE tunnels.
- For an IPsec tunnel, the tunnel_overhead is the maximum overhead of the corresponding CHILD_SA.
- For an IPinIP tunnel, the tunnel_overhead is 0.
- For a GRE tunnel, the tunnel_overhead is length of GRE header.

The private side is similar to the public side. The system processes the received TCP SYN packet on the private side if the TCP MSS adjust is enabled. However, there is no **auto** parameter for **private-tcp-mss-adjust** command.

3.1.1.6 MTU propagation

MTU propagation is an optional feature that allows the system to listen for fragmentation-related ICMP error message received from the public side of the tunnel. These error messages include:

- ICMP Destination Unreachable message "fragmentation needed and DF set" (type 3, code 4)
- ICMPv6 Packet Too Big message (type 2)

The suggested MTU value in the ICMP message is used to derive two MTU values:

- Temporary public MTU (TMTU) are determined as follows:
  - The TMTU starts with a configured **encapsulated-ip-mtu** value.
  - If the received MTU is less than 1280 and it is from an ICMPv6 packet, the received value is ignored.
  - If the received MTU is less than 512 and it is from an ICMP packet, the received value is ignored.
  - If the received MTU is greater than or equal to the configured **encapsulated-ip-mtu** value, the received value is ignored.
  - If the received MTU is greater than or equal to the current TMTU, the received value is ignored.
  - If the received MTU is less than the current TMTU, it replaces the current TMTU.
  - To prevent attack and rapid change, there is a damp time of 60 seconds after a new TMTU value is set. Within that time frame, all received MTU values are ignored.
  - TMTU has a lifetime timer (configurable with an aging interval). When the lifetime timer expires, the TMTU’s value is reset to the **encapsulated-ip-mtu** value. The lifetime timer resets whenever a new TMTU value is set.
  - TMTU is a per tunnel value.
- Temporary private MTU (TPMTU) equals TMTU – Tunnel_Encap_Overhead.
  - TPMTU is a per CHILD_SA value.
  - Tunnel_Encap_Overhead is a fixed value for a non-IPsec tunnel-per-tunnel type. For an IPsec tunnel, its value is the maximum overhead based on the value used by the CHILD_SA set using the following command:
    - **MD-CLI**

      ```
      configure ipsec ipsec-transform id
      ```

    - **classic CLI**

      ```
      configure ipsec ipsec-transform
      ```

TMTU and TPMTU are used in the following cases:

- TPMTU is used for fragmenting IP packets received on the private side instead of the configured IP MTU.
- IKEv2 message fragmentation uses TMTU instead of the configured **encapsulated-ip-mtu**.
- IKE IP packet fragmentation uses TMTU instead of the configured **encapsulated-ip-mtu**.

- To derive the TCP MSS value for the TCP MSS adjustment, instead of configured **encapsulated-ip-mtu**.
- ESP packet fragmentation (post-encapsulation fragmentation) does not use TMTU; it only uses the configured **encapsulated-ip-mtu** value.

To enable this feature, configure the **propagate-pmtu-v4** and **propagate-pmtu-v6** commands in the template, service, or router IPsec tunnel contexts.

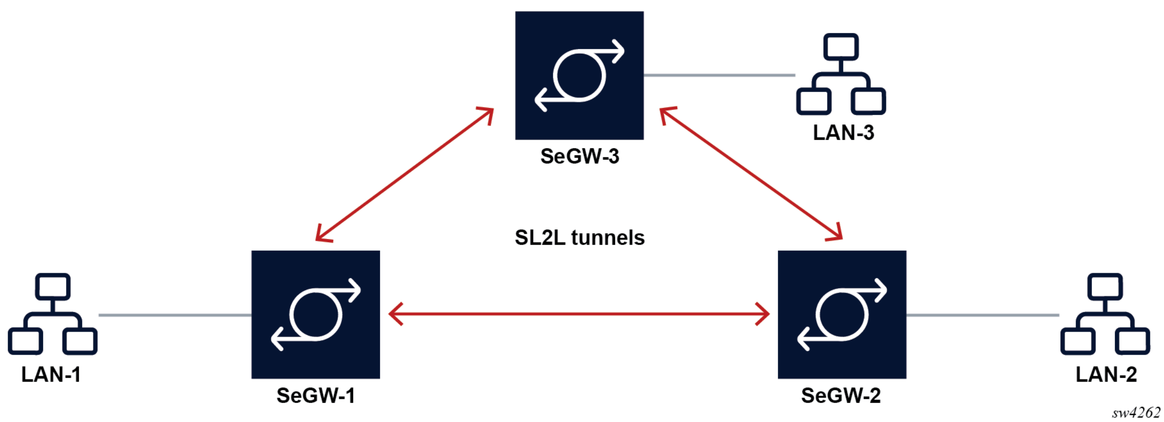
3.1.2 IPsec tunnel types

The types of IPsec tunnels are as follows:

- LAN-to-LAN tunnel (L2L):
  - static LAN-to-LAN (SL2L)
  - dynamic LAN-to-LAN (DL2L)
- remote-access tunnel (RA)

L2L tunnels are typically used for LAN interconnection, while SL2L tunnels are typically used in a mesh topology. The following figure shows SL2L tunnels in a mesh topology.

Figure 2: SL2L tunnels in a mesh topology

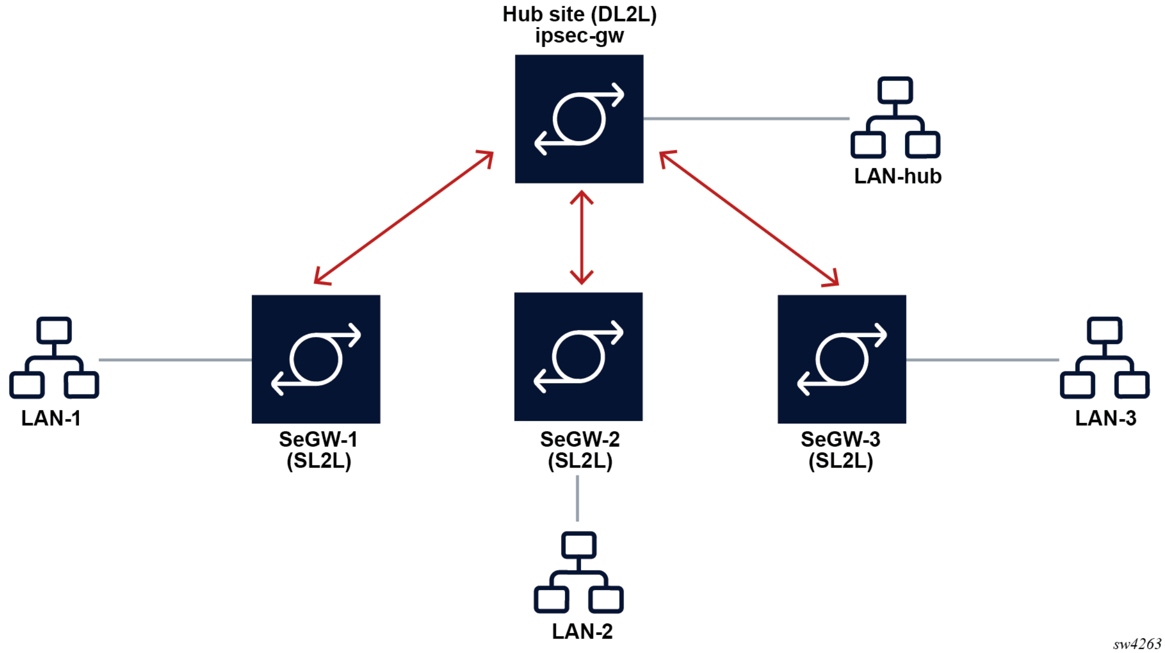


The following features and restrictions apply to SL2L tunnels:

- Per-tunnel configuration is supported, which allows each tunnel to be explicitly configured.
- Each tunnel can be used as the next hop in static routes (directly) or BGP (indirectly).
- SL2L tunnels can act as either the tunnel initiator or responder.
- SL2L tunnels support PSK or certificate-based authentication (or PSK only, in the case of IKEv1).

DL2L tunnels are also used for LAN interconnection. Typically, they are used in a hub-spoke topology where a DL2L tunnel is used on the hub site and the SL2L tunnels are used on the spoke sites. The following figure shows a DL2L tunnel in a hub-spoke topology.

Figure 3: DL2L tunnel in a hub-spoke topology

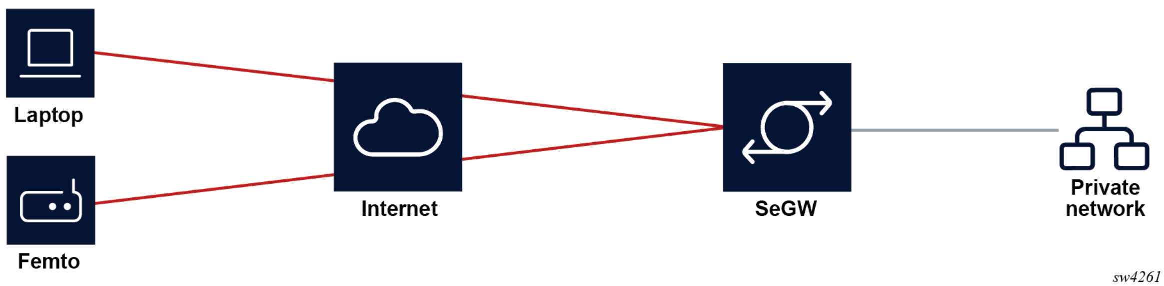


The following features and restrictions apply to DL2L tunnels:

- Per-tunnel configuration is not supported; configuration is per-IPsec gateway only.
- A DL2L tunnel is created dynamically when the IPsec gateway receives an incoming tunnel creation request.
- Reverse routes are created dynamically in a private VRF based on the negotiated TSi to route clear traffic into the tunnel.
- DL2L tunnels can only act as the tunnel responder.
- DL2L tunnels support PSK or certificate-based authentication (or PSK only, in the case of IKEv1)

RA tunnels are typically used by end-user devices to securely access the private network, for example, in the road-warrior VPN use case. The following figure shows an RA tunnel used to access a private network.

Figure 4: RA tunnel used to access a private network



The following features and restrictions apply to RA tunnels:

- Similar to DL2L, configuration for RA tunnels is per-IPsec gateway only.
- An RA tunnel is created dynamically upon receiving a tunnel creation request from the IPsec client.
- The RA tunnel client requests an internal IP address assignment from the SeGW during tunnel creation. These addresses are used as the source address of the private traffic sent by client. Other optional information, such as the DNS server address, can also be returned by the SeGW.
- RA tunnels support the following authentication options:
  - PSK (optionally with RADIUS)
  - certificate-based authentication (optionally with RADIUS)
  - EAP with RADIUS
- RA tunnels support optional RADIUS accounting

3.1.3 Operational conditions

A tunnel group that is in use cannot be deleted. In single-active mode, changes to the primary ISA are allowed only when the tunnel group is in a shutdown state.

Changes can be made to the following:

- enabling or disabling the following configuration

```
configure isa tunnel-group ipsec-responder-only
```

The public interface address can be changed at any time; however, if changed, any static tunnels that were configured to use the public interface address require a configuration changes accordingly. Otherwise, the tunnels are in an operationally down state until their configuration is corrected. The public service cannot be deleted while tunnels are associated.

A tunnel group ID or tag cannot be changed. To remove a tunnel-group instance, it must be in a shutdown state and all IPsec tunnels and IPsec gateways that terminated on the tunnel group must be removed first.

The security policy cannot be changed while an IPsec tunnel is administratively up and using the security policy.

The tunnel local gateway address, peer address, local ID, and public or private service ID parameters cannot be changed while the IPsec gateway or IPsec tunnel is administratively up.

Each IPsec gateway or IPsec tunnel has an administrative state. When the administrative state is down, tunnels cannot be set up.

Each IPsec gateway and IPsec tunnel has an operation state. The operational state can have three possible values:

- **oper-up**

  All configuration and related information are valid and fully ready for tunnel setup.
- **oper-down**

  Some critical configuration information is missing or not ready. Tunnels cannot be set up.
- **limited**

  Not all configuration information is ready to become fully operationally up. When IPsec gateway is in a limited state, it is possible that a new tunnel cannot be established. When the IPsec tunnel is in a limited state, reconnection may fail.

When an IPsec gateway or IPsec tunnel transitions from operationally up to an operationally limited state directly as a result of a hot (non-service affecting) configuration change, established tunnels are not impacted. However, if the IPsec gateway or IPsec tunnel transitions to an operationally down state before it is operationally limited as a result of a service-affecting configuration change, then established tunnels are removed. All operational state transitions are logged.

IPsec gateways or IPsec tunnels can enter the limited state because of the following reasons, among others:

- A Certificate Authority (CA) profile in the configured trust-anchor-profile goes down after the IPsec gateway or IPsec tunnel becomes operationally up.
- An entry in a configured certificate profile goes down after the IPsec gateway or IPsec tunnel becomes operationally up.

3.1.3.1 Dynamic configuration change support for IPsec gateway

All dynamic IPsec tunnels (dynamic LAN-to-LAN tunnels and remote-access tunnels) that terminate on the same IPsec gateway share the same configuration. Use the respective commands in the following contexts to configure an IPsec gateway for an IES or VPRN service:

- **MD-CLI**

```
configure service ies interface sap ipsec-gateway
configure service vprn interface sap ipsec-gateway
```

- **classic CLI**

```
configure service ies interface sap ipsec-gw
configure service vprn interface sap ipsec-gw
```

SR OS provides dynamic configuration change capability to modify specific IPsec gateway configurations without impacting existing tunnels.

The following IPsec gateway configurations are dynamically configurable without shutting down the IPsec gateway:

- Changing the pre-shared key, using the following commands:
  - **MD-CLI**

    ```
    configure service ies interface sap ipsec-gateway pre-shared-key
    configure service vprn interface sap ipsec-gateway pre-shared-key
    ```

  - **classic CLI**

    ```
    configure service ies interface sap ipsec-gw pre-shared-key
    configure service vprn interface sap ipsec-gw pre-shared-key
    ```

- Changing the reference of the IKE policy, using the following commands:
  - **MD-CLI**

    ```
    configure service ies interface sap ipsec-gateway ike-policy
    configure service vprn interface sap ipsec-gateway ike-policy
    ```

- **classic CLI**

```
configure service ies interface sap ipsec-gw ike-policy
configure service vprn interface sap ipsec-gw ike-policy
```

- Changing the reference of the tunnel template, using the following commands:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway default-tunnel-template
configure service vprn interface sap ipsec-gateway default-tunnel-template
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw default-tunnel-template
configure service vprn interface sap ipsec-gw default-tunnel-template
```

- Enabling or changing reference of the RADIUS authentication policy, using the following commands:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway radius authentication-policy
configure service vprn interface sap ipsec-gateway radius authentication-policy
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw radius-authentication-policy
configure service vprn interface sap ipsec-gw radius-authentication-policy
```

- Enabling or changing the reference of the RADIUS accounting policy, using the following commands:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway radius accounting-policy
configure service vprn interface sap ipsec-gateway radius accounting-policy
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw radius-accounting-policy
configure service vprn interface sap ipsec-gw radius-accounting-policy
```

- Enabling, disabling, or changing reference of the TS negotiation, using the following commands:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway ts-list
configure service vprn interface sap ipsec-gateway ts-list
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw ts-negotiation
configure service vprn interface sap ipsec-gw ts-negotiation
```

- Enabling, disabling, or changing reference of the client database, using the command options in the following contexts:

- **MD-CLI**

```
configure service ies interface sap ipsec-gateway client-db
configure service vprn interface sap ipsec-gateway client-db
```

- **classic CLI**

```
configure service ies interface sap ipsec-gw client-db
configure service vprn interface sap ipsec-gw client-db
```

- Changing the certificate configuration, using the commands in the following contexts:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway cert
configure service vprn interface sap ipsec-gateway cert
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw cert
configure service vprn interface sap ipsec-gw cert
```

- Changing DHCPv4-based address assignments, using the commands in the following contexts:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway dhcp-address-assignment dhcpv4
configure service vprn interface sap ipsec-gateway dhcp-address-assignment dhcpv4
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw dhcp
configure service vprn interface sap ipsec-gw dhcp
```

- Changing DHCPv6-based address assignments, using the commands in the following contexts:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway dhcp-address-assignment dhcpv6
configure service vprn interface sap ipsec-gateway dhcp-address-assignment dhcpv6
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw dhcp6
configure service vprn interface sap ipsec-gw dhcp6
```

- Changing local address assignment configuration, using the commands in the following contexts:
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway local address-assignment
configure service vprn interface sap ipsec-gateway local address-assignment
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw local-address-assignment
configure service vprn interface sap ipsec-gw local-address-assignment
```

Existing tunnels are not impacted by dynamic configuration changes. The system uses new configurations for new tunnel negotiations. The system continues to use previous configurations that created the tunnels for ongoing operations (such as rekeying) of the existing tunnel.

3.1.4 OAM interactions

The ISA is IP-addressed by an operator-controlled IP on the public side. That IP address can be used in **ping** and **traceroute** commands and the ISA can either respond or forward the packets to the CPM.

For static LAN-to-LAN tunnels, in multi-active mode, ping requests to public tunnel addresses are not answered if the source address is different from the remote address of the static tunnel.

The private side IP address is visible. The status of the interfaces and the tunnels can be viewed using **show** commands.

Traffic that ingresses or egresses an IES or VPRN service associated with specific IPsec tunnels can be mirrored like other traffic.

Mirroring is allowed per interface (public) or IPsec interface (private) side. A filter mirror is allowed for more specific mirroring.

3.1.5 Redundancy

IPsec supports dead peer detection (DPD).

3.1.6 Statistics collection

Input and output octets and packets per service queue are used for billing end customers who are on a metered service plan. Because multiple tunnels can be configured per interface, the statistics can include multiple tunnels. These can be viewed in the CLI and SNMP.

Reporting (syslog, traps) for authentication failures and other IPsec errors are supported, including errors during IKE processing for session setup and errors during encryption or decryption.

A session log indicates the sort of SA setup when there is a possible negotiation. This includes the setup time, teardown time, and negotiated parameters (such as encryption algorithm) as well as identifying the service a particular session is mapped to, and the user associated with the session.

3.1.7 Security

The ISA module provides security utilities for IPsec-related service entities that are assigned to interfaces and SAPs. These entities (such as card, virtual MS-ISA, and IES or VPRN services) must be enabled in order for the security services to process. The module only listens to requests for security services from configured remote endpoints. In the case of a VPN concentrator application, these remote endpoints could come from anywhere on the Internet. In the cases where a point-to-point tunnel is configured, the module listens only to messages from that endpoint.

3.1.8 IKEv2

IKEv2, defined in RFC 4306, *Internet Key Exchange (IKEv2) Protocol*, is the second version of the Internet Key Exchange Protocol. The main driver of IKEv2 is to simplify and optimize IKEv1. An IKE_SA and a CHILD_SA can be created with only four IKEv2 message exchanges. IKEv2 is supported with the following features:

- static LAN-to-LAN tunnel
- dynamic LAN-to-LAN tunnel
- remote-access tunnel
- pre-shared-key authentication, certificate authentication, EAP (remote-access tunnel only)
- liveness check
- IKE_SA rekey
- CHILD_SA rekey (full Traffic-Selector support including protocol and port range)
- extended ESP sequence number

3.1.8.1 IKEv2 traffic selector and TS-list

The SR OS IKEv2 implementation supports the following traffic selectors:

- IPv4/IPv6 address range
- IP protocol ID
- protocol port range

Port range (including OPAQUE ports) is supported for the following protocols:

- TCP
- UDP
- SCTP
- ICMP
- ICMPv6
- MIPv6

With ICMP and ICMPv6, the system treats the most significant 8 bits of the IKEv2 TS port value as the ICMP message type and the least significant 8 bits as ICMP code.

With MIPv6, the system treats the most significant 8 bits of the IKEv2 TS port value as the mobility header type.

With ICMP, ICMPv6, and MIPv6, the port value in TSi is the value that the tunnel initiator can send, and the port value in TSr is the value that the tunnel responder can send.

The SR OS supports OPAQUE as a TS port selector. An OPAQUE port means that the corresponding CHILD_SA only accepts packets that are supposed to have port information but do not, such as when a packet is a non-initial fragment.

The system allows users to configure a TS-list for each IPsec gateway, applied to both IKEv2 remote access tunnels and dynamic LAN-to-LAN tunnels. Each TS-list contains a local part and a remote part, with each part containing up to 32 entries. Each entry can contain address ranges or subnets, protocols, and port range configurations.

The local part of the TS-list represents the traffic selector for the local system, while the remote part is for the remote peer. If a TS-list is applied on an IPsec gateway, and the system is the tunnel responder, then the local part is TSr and the remote part is TSi.

Combinations of address range, protocol, and port range are not allowed to overlap between entries in the same TS-list.

The system performs traffic selector narrowing as follows.

1. For each TS in the received TSi/TSr, independent address, protocol, and port narrowing is performed. The resulting TS-set is the combination of the address, protocol, and range intersections.
2. The collected TS-set is used as the TSi/TSr.

For a remote access tunnel, TSi narrowing results in an intersection between the following three TSis:

- the TSi received from the client
- the remote part configuration of the TS-list
- a generated TS based on the assigned internal address
  - address (the assigned internal address)
  - protocol (any)
  - port range (any)

The following is an example of a dynamic LAN-to-LAN tunnel.

The configured TS-list local part is as follows:

- Entry 1: 10.10.1.0 → 10.10.1.20, udp, port 100 → 200
- Entry 2: 10.20.1.0 → 10.20.1.20, udp, port 300 → 400

The peer proposes the following TSr:

- Entry 1: 10.10.1.1 → 10.10.1.5, udp, port 110 → 150
- Entry 2: 10.10.1.6 → 10.10.1.10, udp, port 180 → 210
- Entry 3: 10.10.1.15 → 10.10.1.28, udp, port 120 → 160
- Entry 4: 10.20.1.15 → 10.20.1.28, tcp, port 250 → 450

The intersections for the proposed entries are as follows:

- Entry 1: 10.10.1.1 → 10.10.1.5, udp, port 110 → 150
- Entry 2: 10.10.1.6 → 10.10.1.10, udp, port 180 → 200
- Entry 3: 10.10.1.15 → 10.10.1.20, udp, port 120 → 160
- Entry 4: 10.20.1.15 → 10.20.1.20, tcp, port 300 → 400

The resulting TSr system return would be:

- 10.10.1.1 → 10.10.1.5, udp, port 110 → 150
- 10.10.1.6 → 10.10.1.10, udp, port 180 → 200
- 10.10.1.15 → 10.10.1.20, udp, port 120 → 160
- 20.20.1.15 → 20.20.1.20, tcp, port 300 → 400

If more than 32 entries are returned, the system rejects the ts-negotiation and returns TS_UNACCEPTABLE to the peer.

For dynamic LAN-to-LAN tunnels, the system can automatically create a reverse route in a private VRF to route clear traffic into the tunnel. The reverse route is created based on the address range part of the narrowed TSi of the CHILD_SA. If there are multiple TSs in the TSi that have overlapping address ranges, the system creates one or more minimal subnet routes that can cover all address ranges in the TSi. If the auto-created reverse route overlaps with an existing reverse route that points to the same tunnel, the system chooses the route with the larger subnet. If the existing route points to a different tunnel, then CHILD_SA creation fails.

For RADIUS authentication options, such as **psk-radius**, **cert-radius**, or **eap**, the RADIUS server can optionally return a TS-list name via the VSA Alc-IPsec-Ts-Override in the access-accept message, which overrides the TS-list name configured via the CLI.

In the event of a CHILD_SA rekey, if the system is a rekey initiator, it sends the current in-use TS to the peer and expect the peer to return the same TS. If the system is a rekey responder, the system does the same narrowing as was done during CHILD_SA creation.

Configuration of a TS-list can be changed without shutting down the IPsec gateway, although the new TS-list only applies to the subsequent rekey or to the new CHILD_SA creation, and does not affect established CHILD_SAs.

3.1.8.2 IKEv2 fragmentation

In some cases, an IKEv2 message can large, like an IKE_AUTH message with certificate payload. This is likely to cause the IKEv2 packet to be fragmented into a few smaller IP packets. However, in some deployments, there could be devices or network policing, rate limiting or even dropping UDP fragments. In these cases, the SR OS supports fragmenting IKEv2 messages on the protocol level, as specified in RFC 7383, Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation.

This feature is enabled by configuring the an MTU using the following command:

- **MD-CLI**

```
configure ipsec ike-policy ike-version-2 ikev2-fragment mtu
```

- **classic CLI**

```
configure ipsec ike-policy ikev2-fragment
```

The specified MTU is the maximum size of IKEv2 packet.

The system only enables IKEv2 fragmentation for a specific tunnel when the **ikev2-fragment** is configured and the peer also announces its support via sending a IKEV2_FRAGMENTATION_SUPPORTED notification.

3.1.9 SHA2 support

According to RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*, the following SHA2 variants are supported for authentication or pseudo-random functions:

Use HMAC-SHA-256+ algorithms for data origin authentication and integrity verification in IKEv1/2, ESP:

- AUTH_HMAC_SHA2_256_128

- AUTH_HMAC_SHA2_384_192
- AUTH_HMAC_SHA2_512_256

For use of HMAC-SHA-256+ as a PRF in IKEv1/2:

- PRF_HMAC_SHA2_256
- PRF_HMAC_SHA2_384
- PRF_HMAC_SHA2_512

3.1.10 IPsec client lockout

An optional lockout mechanism can be enabled to block malicious clients and prevent them from using invalid credentials to consume system resources, as well as to prevent malicious users from guessing credentials such as a pre-shared key. This mechanism can be enabled by using the **lockout** command.

If the number of failed authentication attempts from a particular IPsec client exceeds a configured threshold during a specified time interval, the client is blocked for a configurable period of time. If a client is blocked, the system drops all IKE packets from the source IP address and port.

The following authentication failures are counted as failed authentication attempts:

- **IKEv1**
  - **psk**: failed to verify the HASH_I payload in main mode
  - **plain-psk-xauth**:
    - failed to verify the HASH_I payload in main mode
    - RADIUS access-reject received
- **IKEv2**
  - **psk**: failed to verify the AUTH payload in the auth-request packet
  - **psk-radius**:
    - failed to verify the AUTH payload in the auth-request packet
    - RADIUS access-reject received
  - **cert**:
    - failed to verify the AUTH payload in the auth-request packet
    - failed to verify the peer’s certification to configured trust-anchors
  - **cert-radius**:
    - failed to verify the AUTH payload in the auth-request packet
    - failed to verify the peer’s certification to configured trust-anchors
    - RADIUS access-reject received
  - **eap**: RADIUS access-reject received

Other failures, such as being unable to assign an address, are not counted.

The AUTH failure counter is reset by either a successful authentication before the client is blocked, the expiration of a block timer, or the expiration of the duration timer.

If multiple IPsec clients behind a NAT device share the same public IP address, a limit for the maximum number of clients or ports behind the same IP address can be configured. If the number of ports exceeds the configured limitation, all ports from that IP address are blocked.

The **clear ipsec lockout** command can also be used to manually clear a lockout state for the specified clients.

3.1.11 IPsec tunnel CHILD_SA rekey

SR OS supports CHILD_SA rekeying for both IKEv1 and IKEv2. The following are the behaviors for the rekey:

- **IKEv1 or IKEv2 CHILD_SA rekey initiator**
  - **outbound**

    The system immediately switches to the new security association (SA) after a new SA is created.
  - **inbound**

    The old SA is kept for three minutes after the new SA is created. Then, it is removed, and upon removal:
    - **IKEv1**

      The system does not send a delete message upon removal.
    - **IKEv2**

      The systems send a delete message upon removal.
- **IKEv1or IKEv2 CHILD_SA rekey responder**
  - **outbound**

    The system keeps using the old SA for 25 seconds after the new SA is created before switching to the new SA. If a delete message of the old SA is received before 25 seconds, the system removes the old SA and starts using new SA.
  - **inbound**

    The old SA is kept for rest of its lifetime. However, if a delete message is received to close the corresponding outbound SA, then the system removes the corresponding inbound SA before its lifetime expires. The system sends a delete message when the old SA lifetime expires.

If the old SA lifetime expires before the 25 seconds or three minutes mentioned above, the old SA is removed upon expiration and the system sends a delete message.

3.1.12 Multiple IKE/ESP transform support

For IPsec tunnels or IPsec gateways, the SR OS allows users to configure up to four IKE transform and four IPsec transform configurations for IKE and ESP traffic.

IKE transform parameters are configured in the **configure ipsec ike-transform** context and referenced in the IKE policy, while IPsec transform parameters are configured in the **configure ipsec ipsec-transform** context and referenced in the tunnel template for dynamic tunnels and in the following context for static tunnels:

- **MD-CLI**

```
configure service vprn interface sap ipsec-tunnel key-exchange dynamic
```

- **classic CLI**

```
configure service vprn interface sap ipsec-tunnel dynamic-keying
```

IKE transform includes the following configurations:

- IKE encryption algorithm
- IKE authentication algorithm
- Diffie-Hellman group
- IKE SA lifetime

IPsec transform includes the following configurations:

- ESP encryption algorithm
- ESP authentication algorithm
- Diffie-Hellman group for CHILD SA rekey with PFS
- CHILD SA lifetime

If multiple IKE and IPsec transform parameters are configured for IPsec gateways and IPsec tunnels, the system uses the configured transforms to negotiate with the peer. This negotiation allows IPsec gateways and IPsec tunnels to support peers with different crypto algorithms.

3.1.13 Reverse routes for dynamic LAN-to-LAN IPsec tunnels

With dynamic LAN-to-LAN IPsec tunnels, one or multiple reverse routes can be automatically created per CHILD_SA in a private service, based on traffic selectors in the negotiated TSi. Use the following command to enable the creation of reverse routes.

```
configure ipsec tunnel-template sp-reverse-route
```

For a specific CHILD_SA, its TSi contains one or multiple address ranges. The system creates one or multiple reverse routes with the largest prefix length to cover all the ranges in the TSi. If a resulting route overlaps with an existing route of the same tunnel, only the route with the smaller prefix length is kept. If a resulting reverse route is a default route, it is ignored if the **ignore-default-route** command option is enabled in the tunnel template.

Use the following command to configure the acceptance of overlapping DL2L reverse routes.

```
configure service vprn ipsec overlapping-reverse-route
```

If a reverse route of an in-setup CHILD_SA overlaps with an existing reverse route from a different tunnel, the system handles it according to following command configurations:

1. If the **overlapping-reverse-route** command is configured as false (disabled):
   a. If the **allow-reverse-route-override-type** command (**allow-reverse-route-override** in classic CLI) is not configured, the in-setup SA fails, resulting in the removal of its tunnel.

   b. If **allow-reverse-route-override-type** is configured to **same-idi** and the in-setup SA belongs to a tunnel that has the same IKEv2 IDi as the corresponding tunnel of the existing reverse route, the system removes the existing tunnel (which includes the existing SA and all others belonging to that tunnel) and creates the in-setup SA; otherwise, the in-setup SA fails.
   c. If **allow-reverse-route-override-type** is configured to **any-idi**, the system removes the existing tunnel (with the existing SA and all other SAs belonging to it) and creates the in-setup SA.
2. If the **overlapping-reverse-route** command is configured as true (enabled), the system creates the in-setup SA and also keeps the existing SA. The system installs all overlapping-but-not-same routes in the route table with their associated metric and preference values configured for the **reverse-route** command in the tunnel template. If there are same routes from different tunnels, the system selects the route to install based on the following rules:
   a. The lower preference route is preferred.
   b. The lower metric route is preferred.
   c. The non-shunting next hop is preferred.
   d. The routes to install are selected from the set of routes where conditions a, b, and c are equal. The quantity of routes to install is the quantity of ECMP next hops specified by the ECMP configuration of the private service (implicitly 1 when ECMP is disabled in the private service). The routes are selected from the set in order of the lowest values returned by the strcmp() function comparing their next-hop strings.

3.2 Using certificates for IPsec tunnel authentication

SR OS supports X.509v3 certificate authentication for IKEv2 tunnel (LAN-to-LAN tunnel and remote-access tunnel). SR OS also supports asymmetric authentication. This means the SR OS and the IKEv2 peer can use different methods to authenticate. For example, one side could use pre-shared key and the other side could use a certificate.

SR OS supports certificate chain verification. For a static LAN-to-LAN tunnel or IPsec gateway, there is a configurable trust-anchor-profile which specifies the expecting CAs that should be present in the certificate chain before reaching the root CA (self-signed CA) configured in the system.

The SR OS's own key and certificate are also configurable per tunnel or IPsec gateway.

When using certificate authentication, the SR OS uses the subject of the configured certificate as its ID by default.

Note: IPsec application is subject to FIPS restrictions; for more information please see the *7705 SAR Gen 2 Basic System Configuration Guide*.

3.2.1 IKEv2 digital signature authentication

RFC 7427 *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)* defines a new IKEv2 AUTH payload method which not only indicates the type of public key, but also the hash algorithm that used to generate the signature; it also includes a new IKEv2 notification: SIGNATURE_HASH_ALGORITHMS, which is used to signal support of RFC 7427 and a list of support hash algorithms to a peer.

RFC 7427 is the default way to perform certificate authentication for IKEv2. The system negotiates its support with the peer as follows:

- **sending**
  - as tunnel initiator, includes SIGNATURE_HASH_ALGORITHMS in the IKE_SA_INIT request.
  - as tunnel responder, includes SIGNATURE_HASH_ALGORITHMS in IKE_SA_INIT response only if the received IKE_SA_INIT request includes it.
  - includes the SHA1/SHA2-256/SHA2-384/SHA2-512 hash algorithms in SIGNATURE_HASH_ALGORITHMS
- **receiving**
  - If the peer does not include SIGNATURE_HASH_ALGORITHMS in the IKE_SA_INIT packet, then it does not support RFC 7427 and the system uses an RSA Digital Signature for the RSA key(value 1), and DSS Digital Signature (value 3) for the DSA key to generate the AUTH payload.

    Note: If the ECDSA key is selected in the cert-profile entry, then the tunnel setup fails in the system.

  - If the peer sends SIGNATURE_HASH_ALGORITHMS, then the system uses RFC 7427 and the strongest hash algorithms that is supported by both sides to generate the AUTH payload. If there is no common hash algorithms supported by both sides, the system falls back to RSA Digital Signature (Auth Method value 1) or DSS Digital Signature (Auth Method value 3).

    Note: If the selected key is an RSA key, there are specific cases that have a short RSA key with long hash algorithm. The system falls back to RSA Digital Signature for RSA key (value 1) even when both sides send SIGNATURE_HASH_ALGORITHMS and there are common hash algorithms.

    To verify the received digital signature of the AUTH payload, the peer must uses one of the algorithms in the SIGNATURE_HASH_ALGORITHMS that the system sends. Otherwise, the tunnel setup fails.

The system continues to use CAs in received cert-request payloads to select the cert-profile entry; if the selected entry is an RSA key, the system needs to decide to whether use PKCS#1-1.5 or RSASS-PSS to generate the signature by using the value set by the following command.

```
configure ipsec cert-profile entry rsa-signature
```

3.3 Trust-anchor profile

SR OS supports multiple trust-anchors per IPsec tunnel or gateway. Users can configure a trust-anchor-profile that includes up to eight CAs. The system builds a certificate chain by using the certificate in the first certificate payload in the received IKEv2 message. If any of configured trust-anchor CAs in the trust-anchor-profile appears in the chain, then authentication is successful. Otherwise, authentication is failed.

SR OS only supports processing of up to 16 hashes for the trust-anchor list from other products. If the remote end is sending more than 16, and a certificate match is in the > 16 range, the tunnel remains down with authentication failure.

3.4 Cert-profile

SR OS supports sending different certificate/chain according to the received IKEv2 certificate-request payload. This is achieved by configuring a cert-profile, which allows up to eight entries. Each entry includes a certificate and a key and, optionally, also a chain of CA certificates.

The system loads the cert and key configured in a cert-profile into memory and builds a chain. Compare-chain is performed for the certificate configured in each entry of the cert-profile upon enabling of the cert-profile. These chains are used in IKEv2 certificate authentication. If a chain computation cannot be completed for a configured certificate, then the corresponding compare-chain is empty or only partially computed.

Because there can be multiple entries configured in the cert-profile, the system needs to pick the cert and key in the correct entry that the other side expects to receive. This is achieved by a lookup of the CAs within one cert-request payload or multiple cert-request payloads in the compare-chain and then picking the first entry that has a cert-request CA appearing in its chain. If there is no such cert, the system picks the first entry in the cert-profile. The first entry shown in the output below, is the first configured entry in the cert-profile. The entry-id of first entry does not have to be 1.

For example, there are three CA listed in certificate-request payload: CA-1, CA-2 and CA-3, and there are two entries configured in the cert-profile as follows:

Example: MD-CLI

```
  cert-profile "cert-profile-1" {
        entry 1 {
            cert "cert-1"
            key "key-1"
        }
        entry 2 {
            cert "cert-2"
            key "key-2"
            send-chain {
                ca-profile ["CA-1" "CA-2"]
            }
        }
    }
```

Example: classic CLI

```
cert-profile “cert-profile-1”
   entry 1
      cert “cert-1”
      key “key-1”
   entry 2
      cert “cert-2”
      key “key-2”
      send-chain
         ca-profile “CA-1”
         ca-profile “CA-2”
```

The system builds two compare-chains: chain-1 for cert-1 and chain-2 for cert-2. Assume CA-2 appears in chain-2, but CA-1 and CA-3 do not appear in either chain-1 or chain-2. Then the system picks entry 2.

After a cert-profile entry is selected, the system generates the AUTH payload by using the configured key in the selected entry. The system also sends the cert in the selected entry as “certificate” payload to the peer.

If a chain is configured in the selected entry, then one certificate payload is needed for each certificate in the configured chain. The first certificate payload in the IKEv2 message is the signing certificate, which is configured by the **cert** command in the chosen cert-profile entry. With the above example, the system sends three certificate payloads: cert-2, CA-1,CA-2.

The following CA chain-related enhancements are supported:

- Enabling a ca-profile triggers a recomputation of the compute-chain in related cert-profiles. The system also generates a new log-1 to indicate a new compute-chain has been generated; the log includes the ca-profile names on the new chain. Another log (log-2), is generated if the send-chain in a cert-profile entry is not in the compute-chain because of this ca-profile change. Another log is generated if the hash calculation for a certificate under a ca-profile has changed.
- When enabling a cert-profile, the system allows the CAs in the send-chain, not in the compute-chain. The system also generates log-2 as above.
- The system allows changes of the configuration of send-chain without disabling the cert-profile.
- When a configure root CA is cross-signed by another CA, multiple overlapping compare-chains for a specific certificate profile entry may occur. Choose one compare-chain by executing the following command to specify the tiebreak CA.

```
configure ipsec cert-profile entry compare-chain-include
```

3.5 IPsec deployment requirements

The following information describes requirements to deploy SR OS IPsec features.

IPsec general

To avoid high CPU loads and some complex cases, the following are the requirements to configure IKEv2 lifetime:

- The IKE_SA lifetime on one side should be approximately twice as large as the other side. The CHILD_SA lifetime on one side should be approximately two or three times larger than the other side.
- With the preceding rule, the lifetime of the side with smaller lifetime should not be too small:
  - IKE_SA: >= 86400 seconds
  - CHILD_SA: >= 3600 seconds
- With first rule, on the side with the smaller lifetime, the IKE_SA lifetime should be at least three times larger than CHILD_SA lifetime.
- The IKE protocol is the control plane of IPsec, therefore, the IKE packet should be treated as high QoS priority in the end-to-end path of the public service.

  On a public interface, a SAP ingress QoS policy should be configured to ensure the IKE packet is treated as high QoS priority.
- The correct system time is required for certificate authentication to work properly.
- The peer’s DPD interval must be larger than 30 seconds and should not send a DPD request if it receives IKE or ESP traffic.

3.6 IKEv2 remote-access tunnel

SR OS supports IKEv2 remote-access tunnel, the difference between a remote-access tunnel and LAN-to-LAN tunnel is remote-access tunnel allows client to request an internal address (and other attributes like DNS address) via IKEv2 configuration payload. The SR OS supports IKEv2 remote-access tunnel with following features:

- authentication methods:
  - pre-shared-key with) or without RADIUS
  - certificate with or without RADIUS
  - EAP or EAP-only with RADIUS
- internal address assignment via IKEv2 configuration payload
- address assignment support:
  - RADIUS server based
  - local address assignment
- RADIUS accounting to report address usage
- RADIUS disconnect message to remove tunnel
- NAT-Traversal support

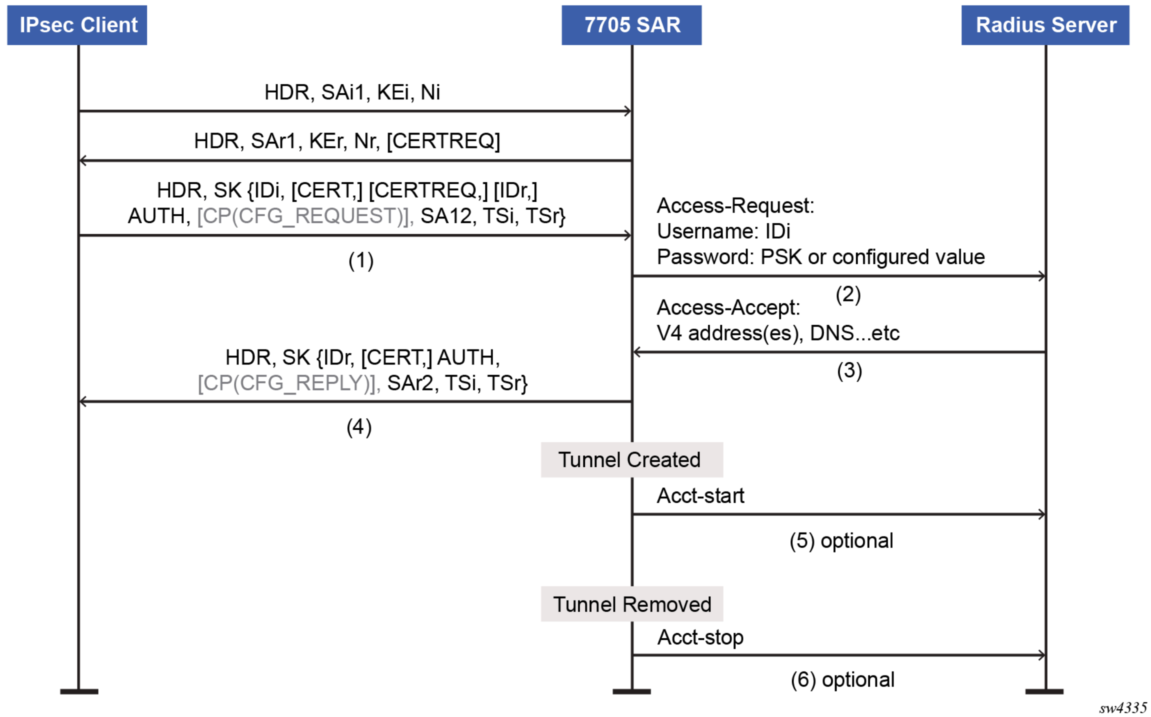
The SR OS only supports address assignments in first CHILD_SA negotiation.

3.6.1 IKEv2 remote access tunnel – RADIUS-based PSK or certificate authentication

If the authentication method of the IKE policy is **psk-radius** or **cert-radius**, then the system authenticates the client using PSK or the certificate as if it is a LAN-to-LAN tunnel. The system also performs a RADIUS authentication or authorization and optionally sends RADIUS accounting messages.

Figure 5: Call flow for RADIUS-based PSK or certificate authentication displays a typical call flow for RADIUS-based PSK or certificate authentication.

Figure 5: Call flow for RADIUS-based PSK or certificate authentication



The Access-Request includes the following attributes:

- Username is IDi.
- User-Password is one of following value's hash according to section 5.2 of RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*:
  - client's PSK if **psk-radius** command option is configured
  - otherwise, a key configured using the **password** command in the RADIUS authentication policy; if a password is not configured and the system does not include the User-Password attribute in access-request.
- Acct-Session-Id represents the IPsec tunnel session.

  The format is: local_gw_ip-remote_ip:remote_port-time_stamp.

  For example: 172.16.100.1-192.168.5.100:500-1365016423.
- Use the following command to configure other RADIUS attributes.
  - **MD-CLI**

    ```
    configure ipsec radius authentication-policy include-radius-attribute
    ```
  - **classic CLI**

    ```
    configure ipsec radius-authentication-policy include-radius-attribute
    ```

  This command configures the following command options.

  - Called-Station-Id (local tunnel address)
  - Calling-station-Id (remote tunnel address:port number)
  - Nas-Identifier (the system name)
  - Nas-Ip-Address (the system IP)
  - Nas-port-id (the public tunnel SAP ID)

If RADIUS authentication is successful, then the RADIUS server sends an access-accept message back; otherwise, an access-reject message is sent back.

The following are supported attributes in access-accept:

- Alc-IPsec-Serv-Id
- Alc-IPsec-Interface
- Framed-IP-Address
- Framed-IP-Netmask
- Alc-Primary-Dns
- Alc-Secondary-Dns
- Alc-IPsec-Tunnel-Template-Id
- Alc-IPsec-SA-Lifetime
- Alc-IPsec-SA-PFS-Group
- Alc-IPsec-SA-Encr-Algorithm
- Alc-IPsec-SA-Auth-Algorithm
- Alc-IPsec-SA-Replay-Window

After the tunnel is successfully created, the system could optionally (depending on the configuration of the RADIUS accounting policy configured in the IPsec gateway), send an accounting-start packet to the RADIUS server, and also send an accounting-stop when the tunnel is removed. The user can use the following command to enable this behavior.

- **MD-CLI**

```
configure ipsec radius accounting-policy update-interval
```

- **classic CLI**

```
configure ipsec radius-accounting-policy update-interval
```

The following are some attributes included in the acct-start/stop and interim-update:

- Acct-status-type
- Acct-session-id (the same as in the access-request)
- Username

Use the following command to configure which RADIUS attributes are included.

- **MD-CLI**

```
configure ipsec radius accounting-policy include-radius-attribute
```

- **classic CLI**

```
configure ipsec radius-accounting-policy include-radius-attribute
```

This command allows the following command options.

- Frame-ip-address: the assigned internal address
- Calling-station-id
- Called-station-id
- Nas-Port-Id
- Nas-Ip-Addr
- Nas-Identifier
- Acct-Session-Time (tunnel session time, only in acct-stop packet)

The system also supports RADIUS disconnect messages to remove an established tunnel, If the following command is enabled in the RADIUS server configuration, then the system accepts the disconnect-request message (RFC 5176, *Dynamic Authorization Extensions to Remote Authentication Dial In User Service* (*RADIUS*)), and tear down the specified remote-access tunnel.

- **MD-CLI**

```
configure router radius server accept-coa
```

- **classic CLI**

```
configure router radius-server server accept-coa
```

For security reasons, the system only accepts a disconnect-request when **accept-coa** is configured and the disconnect-request comes from the corresponding server.

The target tunnel is identified by one of following methods:

- Acct-Session-Id
- Nas-Port-Id + Framed-Ip-Addr(Framed-Ipv6-Prefix) + Alc-IPsec-Serv-Id
- User-Name

By default, the system only returns what the client has requested in the CFG_REQUEST payload. However, this behavior can be overridden using the following command.

```
configure ipsec ike-policy relay-unsolicited-cfg-attribute
```

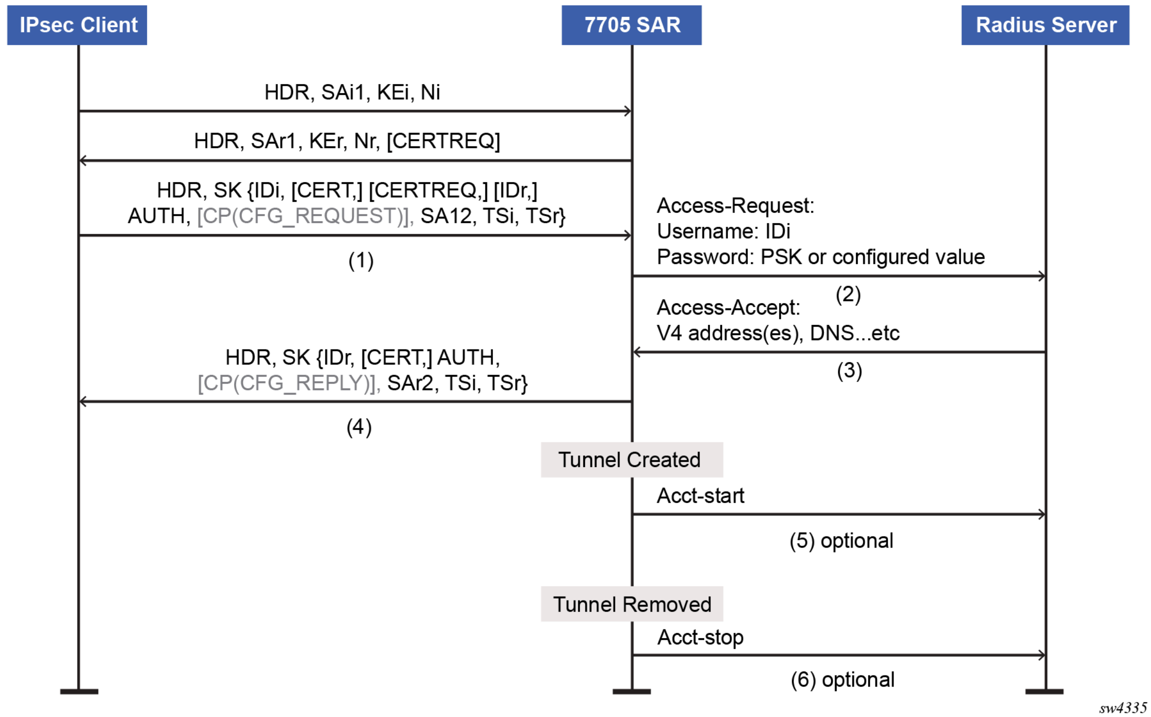
With this configuration, the configured attributes returned from the source (such as the RADIUS server) are returned to the client regardless if the client has requested it in the CFG_REQUEST payload.

3.6.1.1 IKEv2 remote-access tunnel – EAP authentication

The SR OS supports EAP authentication for a IKEv2 remote-access tunnel, in which case, the system acts as an authenticator between an IPsec client and a RADIUS server. It transparently forwards EAP messages between the IKEv2 session and RADIUS session. Thus, the actual EAP authentication occurs between the client and the RADIUS server.

Figure 6: Typical call flow of EAP authentication shows a typical call flow of EAP authentication.

Figure 6: Typical call flow of EAP authentication



Use the following command option to enable EAP authentication.

- **MD-CLI**

```
configure ipsec ike-policy ike-version-2 auth-method eap
```

- **classic CLI**

```
configure ipsec ike-policy auth-method eap
```

When enabled, after the received IKE_AUTH request from the client, the system sends an EAP-Response/ ID with IDi as the value in the access-request to AAA. AAA returns a method request and the system starts passing through between the client and AAA (as shown in Figure 6: Typical call flow of EAP authentication).

The generation of the AUTH payload in the IKE_AUTH response sent by the SR OS (message 4 in flow shown above) is dependent on the following command.

- **MD-CLI**

```
configure ipsec ike-policy ike-version-2 own-auth-method
```

- **classic CLI**

```
configure ipsec ike-policy own-auth-method
```

This command allows the following command options.

psk

The AUTH payload is present and generated by using PSK.

cert

The AUTH payload is present and generated by the configured public and private key pairs as it does in certificate authentication. Any needed certificates are also sent.

eap-only

Neither AUTH nor CERT payload is present.

The RADIUS attributes in authentication and accounting packets are similar to psk-radius and cert-radius with the following differences:

- RADIUS attributes support EAP-Message/Message-Authenticator/State attributes.
- RADIUS attributes support Access-Challenge packet.
- RADIUS attributes support MS-MPPE-Send-Key/ MS-MPPE-Recv-Key in access-accept. These two attributes are required for all EAP methods that generate MSK.

The system provides a method to support EAP and other authentication methods on the same IPsec gateway policy. Use one of the following command options to enable the correct authentication method.

- **MD-CLI**

```
configure ipsec ike-policy ike-version-2 auth-method auto-eap
configure ipsec ike-policy ike-version-2 auth-method auto-eap-radius
```

- **classic CLI**

```
configure ipsec ike-policy auth-method auto-eap
configure ipsec ike-policy auth-method auto-eap-radius
```

With **auto-eap**:

- If there is no AUTH payload in IKE_AUTH request, the system uses EAP to authenticate the client and also uses following command to generate the AUTH payload.
  - **MD-CLI**

  ```
  configure ipsec ike-policy ike-version-2 own-auth-method
  ```

  - **classic CLI**

  ```
  configure ipsec ike-policy own-auth-method
  ```

- If there is an AUTH payload in the IKE_AUTH request, the authorization method is determined by the following command.
  - **MD-CLI**

  ```
  configure ipsec ike-policy ike-version-2 auto-eap-method
  ```

  - **classic CLI**

  ```
  configure ipsec ike-policy auto-eap-method
  ```

  - If the **auto-eap-method** is **psk**, the system proceeds as auth-method: psk

  - If the **auto-eap-method** is **cert**, the system proceeds as auth-method: cert
  - If the **auto-eap-method** is **psk-or-cert**:
    - If the Auth Method field of the AUTH payload is PSK, the system proceeds as auth-method: psk
    - If the Auth Method field of the AUTH payload is RSA or DSS, the system proceeds as auth-method: cert-auth

With **auto-eap-radius**:

- If there is no AUTH payload in an IKE_AUTH request, the system uses EAP to authenticate the client and also uses the following command to generate the AUTH payload.
  - **MD-CLI**

```
configure ipsec ike-policy ike-version-2 own-auth-method
```

  - **classic CLI**

```
configure ipsec ike-policy own-auth-method
```

- If there is an AUTH payload in the IKE_AUTH request, the system uses the following command to generate the AUTH payload.
  - **MD-CLI**

```
configure ipsec ike-policy ike-version-2 auto-eap-own-method
```

  - **classic CLI**

```
configure ipsec ike-policy auto-eap-own-method
```

  - If the **auto-eap-own-method** is **psk**, the system proceeds as auth-method: psk-radius
  - If the **auto-eap-own-method** is **cert**, the system proceeds as auth-method: cert-radius
  - If **auto-eap-own-method** is **psk-or-cert**:
    - If the Auth Method field of the AUTH payload is PSK, the system proceeds as auth-method:psk-radius
    - If the Auth Method field of the AUTH payload is RSA or DSS, the system proceeds as auth-method:cert-radius

3.6.2 IKEv2 remote-access tunnel – authentication without RADIUS

To achieve authentication without RADIUS, the authentication method needs to be configured as **psk** or **cert-auth** and a local address assignment must be configured under the IPsec gateway.

Figure 7: Typical call flow of certificate or PSK authentication without RADIUS shows a typical call flow of certificate or PSK authentication without RADIUS.

Figure 7: Typical call flow of certificate or PSK authentication without RADIUS

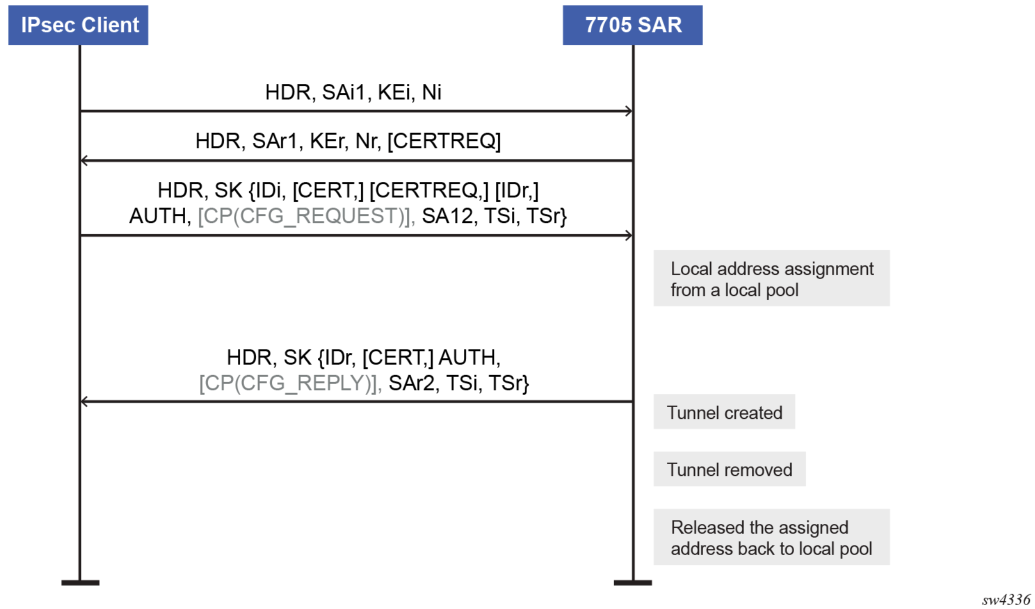
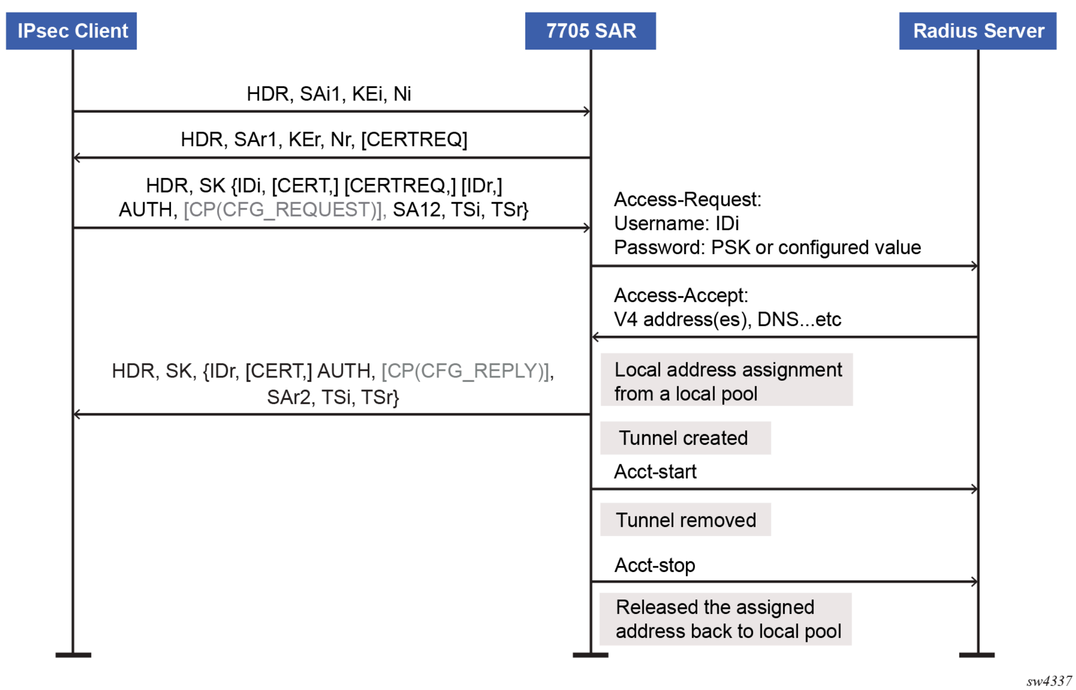


Figure 8: Typical call flow for EAP authentication shows a typical call flow for EAP authentication.

Figure 8: Typical call flow for EAP authentication



In this configuration, the RADIUS authentication and accounting policies in the IPsec gateway context are ignored.

RADIUS disconnect messages are supported in this case. Only the following tunnel identification methods are supported:

- Nas-Port-Id + Framed-Ip-Addr(Framed-Ipv6-Prefix) + Alc-IPsec-Serv-Id
- User-Name

3.6.3 IKEv2 remote-access tunnel – address assignment

The SR OS supports the following methods of address assignment for IKEv2 remote-access tunnels:

- RADIUS
- local address assignment (LAA)
- DHCPv4/v6

For RADIUS-based address assignment, the address information is returned in an access-accept packet. This implies that RADIUS-based address assignment requires using an authentication option with RADIUS, such as **psk-radius**, **cert-radius**, or **eap**.

For LAA, the system gets an address from a pool defined in a local DHCPv4/v6 server. When a tunnel is removed, the assigned address is released back to the pool. If the local DHCPv4/v6 server is shut down, all existing tunnels that have an address from the server are removed. If LAA is shut down, the current established tunnel that used LAA stays up.

For DHCP-based address assignment, the system acts as a DHCP client on behalf of the IPsec client and requests an address from an external DHCP server via the standard DHCP exchange. In this case, the system also acts as a DHCP relay agent, which relays all DHCP packets between the DHCP server and the local DHCP client. DHCP renew and rebind are also supported.

3.6.3.1 DHCPv4 address assignment

The client’s hardware address field (chaddr) in the DHCPv4 header is generated by the SR OS:

- The first 2 bytes of the MAC address are 02:03.
- The remaining 4 bytes are the hash result of IKEv2 IDi.

The following options are included in the DHCPv4 packets sent by the SR OS:

- Option 82 circuit-id (*private-SAP-id* | *private-interface-name*; for example, tunnel-1.private:100 | priv-int)
- Option 82 remote-id (IKEv2 IDi in text format)
- Option 61 client-id is 1 byte that represents the IKEv2 IDi type plus the IKEv2 IDi in text format. The value of the first byte is as follows:
  - ID_IPV4_ADDR = 1
  - ID_DER_ASN1_DN = 2
  - ID_FQDN = 3
  - ID_RFC822_ADDR = 4
  - ID_IPV6_ADDR = 5

3.6.3.2 DHCPv6 address assignment

Because the system performs a DHCP relay function, all DHCPv6 packets sent or received are encapsulated in DHCPv6 relay-forward and relay-reply messages.

The following items are values of key fields and options in DHCPv6 packets sent by the system:

- Hop-count (0)
- Link address (configurable via the CLI)
- Peer-address (auto-generated based on the IKEv2 IDi)
- Option 1 Client Identifier
  - DUID type (2)
  - Enterprise ID (6527)
  - Value is 1 byte that represents the IKEv2 IDi type plus the IKEv2 IDi in text format. The value of the first byte is the same as that of the first byte in Option 61 for DHCPv4.
- Option 16 Vendor Class
  - Enterprise ID (6527)
  - Value (string “SROS IPsec”)
- Option 18 Interface ID (*private-SAP-id* | *private-interface-name*; for example, tunnel-1.private:100 | priv-int)
- Option 37 Remote Identifier
  - Enterprise ID (6527)
  - Value (IKEv2 IDi in text format)

3.6.3.3 DHCPv4/v6 usage notes

- Using a local DHCP server on the same chassis for DHCP-based address assignment is not supported. The DHCP server must be external.
- IPsec DHCP relay uses only the following command **gi-address** configuration found under the IPsec gateway and does not take into account the gateway IP address with a source IP address configuration on any other interfaces.
  -
  - **MD-CLI**

```
configure service ies interface sap ipsec-gateway dhcp-address-assignment dhcpv4 gi-
address
configure service vprn interface sap ipsec-gateway dhcp-address-assignment dhcpv4 gi-
address
```

  - **classic CLI**

```
configure service ies interface sap ipsec-gw dhcp gi-address
configure service vprn interface sap ipsec-gw dhcp gi-address
```

- The following command must be enabled on an interface that has a gateway IP address as the interface address for the interface to use a DHCPv4 address assignment. The system ignores other DHCP or DHCPv6 configurations on the interface, with the exception of the relay-proxy configuration.
  - **MD-CLI**

```
configure service ies interface ipv4 dhcp relay-proxy
configure service vprn interface ipv4 dhcp relay-proxy
```

  - **classic CLI**

```
configure service ies interface dhcp relay-proxy
configure service vprn interface dhcp relay-proxy
```

  - If the DHCP server resides in a private service, and the gateway IP address is an address configured on the corresponding tunnel interface, **relay-proxy** must be enabled on the corresponding private interface.
  - If the DHCP server resides in a routing instance that is different from the private service, then there must be an interface (such as a loopback interface) in the routing instance that has the gateway IP address as the interface address, and gateway IP address must be routable for the DHCP server. Also, the relay proxy must be enabled on the interface in the routing instance.

The biggest difference between the LAA and DHCP-based methods is that LAA uses a local API to get an address from a local pool. There is no DHCP packet exchange for LAA, while a DHCP-based method uses standard DHCP packet exchange to request a packet from an external DHCP server.

Because there are three methods for address assignment, the following is the priority order (descending) of sources to choose if more than one source is configured:

- LAA
- DHCP
- RADIUS

There is no fallback between the different sources.

LAA/DHCP can work with an authentication method that does not involve RADIUS, as well as with an authentication method that involves RADIUS. When using LAA/DHCP with an authentication method that involves RADIUS, the following applies:

- LAA/DHCP only happens after RADIUS is successfully authenticated.
- The address information returned by the RADIUS server is ignored (even if LAA/DHCP is configured but is shut down).
- Non-address-related attributes in access-accept messages such as Alc-IPsec-Serv-Id and Alc-IPsec-Tunnel-Template-Id are still accepted.
- RADIUS accounting is supported in this case, but the Framed-IP-Addr/Framed-IPv6-Prefix reported in the acct-request packet is the LAA/DHCP assigned address, not the address returned by the RADIUS server.
- RADIUS disconnect messages are supported.

3.6.4 IPv6 IPsec support

The SR OS provides the following IPv6 support to IPsec functions:

- IPv6 packets as the ESP tunnel payload
- IPv6 as the ESP tunnel encapsulation

3.6.4.1 IPv6 as payload

IPv6 as payload allows IPv6 packets to be forwarded within an IPsec tunnel. Current support includes the following:

- Tunnel type support includes:
  – static LAN-to-LAN tunnel
  – dynamic LAN-to-LAN tunnel
  – remote-access tunnel (only IKEv2 is supported)
- The prefix length of the IPv6 address on a private interface must be /96 or longer.

3.6.4.2 IPv6 as payload: static LAN-to-LAN tunnel

There are three methods to forward IPv6 traffic into static tunnels on the private side:

- The destination address is a configured destination IP under the tunnel context.
  – The destination IP can be either an IPv6 address or an IPv4 address.
  – In the case of IPv6, it must be either an IPv6 global unicast address or an IPv6 link-local address.
  – In the case of IPv4, it can be used to forward IPv4 traffic into the tunnel.
  – In case of unicast address, dest-ip must be within the prefix configured on the private interface.
  – Up to 16 destination IP addresses can be configured per IPsec tunnel.
- A v6 route with a configured destination IP as the next-hop, this route can be learned from either a static or dynamic from a routing protocol such as BGP.
- An IPv6 static route with an ipsec-tunnel used as the next-hop.

A security policy supports either an IPv4 entry or an IPv6 entry or both for dual-stack.

3.6.4.3 IPv6 as payload: dynamic LAN-to-LAN tunnel

With dynamic LAN-to-LAN tunnels, the system automatically creates a v6 reverse route in the private VPRN based on the received TSi payload with the tunnel as the next hop.

3.6.4.4 IPv6 as payload: remote-access tunnel

The system supports the following IKEv2 IPv6 configuration attributes:

- INTERNAL_IP6_ADDRESS
- INTERNAL_IP6_DNS

The system supports only one internal IPv6 address per tunnel. The following IPv6-related RADIUS attributes are also supported in access-accept:

- Framed-IPv6-Prefix is translated into INTERNAL_IP6_ADDRESS in the configuration payload, which includes two parts. A 16-byte v6 prefix and a one-byte prefix length.
- Alc-Ipv6-Primary-Dns
- Alc-Ipv6-Secondary-Dns

If an internal v6 address has been assigned to the remote-access client, then the Framed-IPv6-Prefix is also included in RADIUS accounting-request packet. The assigned internal v6 address must be within the prefix configured on the corresponding private interface.

If the client request both v4 and v6 address and address source (such as RADIUS or LAA) assign both v4 and v6 address, then both v4 and v6 addresses are assigned to the client via the configuration payload.

3.6.4.5 IPv6 as encapsulation

IPv6 as encapsulation allows IPv4 or IPv6 packets to be forwarded within an IPv6 ESP tunnel, also the IKE protocol can run over IPv6. Current support only includes tunnel type support:

- static LAN-to-LAN tunnel
- dynamic LAN-to-LAN tunnel
- remote-access tunnel (For IKEv1, only v4 over v6 is supported)

For an **ipsec-gw** or **ipsec-tunnel**, only one local gateway address is supported, which could be either an IPv4 or IPv6 address. The SR OS also provides fragmentation and reassembly support for IPv6 ESP/IKE packets.

3.7 Secured interface

A secured interface secures traffic forwarded through a specified IP interface, through one or multiple Secure Interface Tunnels (SI Tunnels) configured under the interface. SI tunnel is conceptually the same as traditional static IPsec tunnels. Some differences are:

- SI tunnels are configured under an IP interface, while static IPsec tunnels are configured under the private tunnel SAP of a tunnel interface.
- With an SI tunnel, the following objects are created automatically with an SI tunnel configuration. There is no need for a separate configuration tunnel configuration:
  – public tunnel SAP
  – public interface
  – private tunnel SAP
  – private tunnel interface
- The public service of SI tunnel is the same service of secured interface, which could be either base router, an IES or an VPRN service.
- The local tunnel address of the SI tunnel must be one of interface addresses of the secure interface. If the secure interface is unnumbered, then it must be one of the interface address of the interface specified by the unnumbered configuration.
- Private service is the same as the public service. The user could also specify a different service.
- On the public side:

  - With a secured interface, by default, all traffic ingress the interface are subject to IPsec processing. If the received traffic is not IPsec traffic (such as ESP and IKE), it is dropped. Use the following commands to change this behavior.

    ```
    configure filter ip-exception
    configure filter ipv6-exception
    ```

    All ingress traffic matching the configured exception filter bypasses IPsec processing and is forwarded through normal routing methods.
  - The system forwards all SI tunnel traffic (after encryption and encapsulation) out through the corresponding secured interface.
  - SSH traffic toward the local system and MPLS/SDP always bypasses IPsec processing.
- On the private side:
  - Like a static IPsec tunnel, traffic is routed into the SI tunnel through a static route or BGP route.
  - When an SI tunnel is operationally down, routes using the next-hop address as the tunnel are unresolved and withdrawn from the route table.
- show, debug, tool, clear, and admin commands that apply to static IPsec tunnels also apply to SI tunnels.
- The following features are not supported with SI tunnels on 7705 SAR Gen 2 (with cellular exit port):
  - destination IP
  - MC-IPsec
  - IPv4 over IPv6
  - IPv6 over IPv6
  - MLDv2 over SI tunnel
- The following features are not supported with SI tunnels on 7705 SAR Gen 2:
  - destination IP
  - MC-IPsec
  - MLDv2 over SI tunnel

3.8 Configuring IPsec with CLI

3.8.1 Provisioning a tunnel ISA

The following example displays a card and ISA configuration.

Example: MD-CLI

```
[ex:/configure]
A:admin@node-2# info
card 1 {
        card-type iom4-e
        mda 1 {
            mda-type me40-1gb-csfp
```

```
        }
        mda 2 {
            mda-type isa2-tunnel
        }
```

Example: classic CLI

```
A:node-2>config# info
----------------------------------------------
...
    card 1
        card-type iom4-e
        mda 1
            mda-type me40-1gb-csfp
            no shutdown
        exit
        mda 2
            mda-type isa2-tunnel
            no shutdown
        exit
        no shutdown
...
----------------------------------------------
```

3.8.2 Configuring a tunnel group

The following example displays a tunnel group configuration in the ISA context. The **multi-active** command specifies that there could be multiple active ISAs in the tunnel group.

Example: MD-CLI

```
[ex:/configure]
A:admin@node-2# info
isa {
        tunnel-group 1 {
            admin-state enable
            isa-scale-mode tunnel-limit-2k
            multi-active {
                isa 1/2 { }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config# info
----------------------------------------------
...
    isa
        tunnel-group 1 isa-scale-mode tunnel-limit-2k create
            multi-active
            mda 1/2
            no shutdown
        exit
    exit
...
----------------------------------------------
```

3.8.3 Configuring router interfaces for IPsec

The following example displays an interface named “internet” that is configured using the network port (1/1/1), which provides network connection on the public side.

Example: MD-CLI

```
[ex:/configure router "Base"]
A:admin@node-2# info
    autonomous-system 123
    interface "internet" {
        port 1/1/1
        ipv4 {
            primary {
                address 10.10.7.118
                prefix-length 24
            }
        }
    }
    interface "system" {
        ipv4 {
            primary {
                address 10.20.1.118
                prefix-length 32
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config# info
----------------------------------------------
...
    router Base
        interface "internet"
            address 10.10.7.118/24
            port 1/1/1
            no shutdown
        exit
        interface "system"
            address 10.20.1.118/32
            no shutdown
        exit
        autonomous-system 123

...
----------------------------------------------
```

3.8.4 Configuring IPsec command options

The following example displays an IPsec configuration.

Example: MD-CLI

```
[ex:/configure ipsec]
```

```
A:admin@node-2# info
    ike-policy 100 {
        ike-transform [100]
        ike-version-2 {
            auth-method eap
        }
    }
    ike-transform 100 {
        dh-group group-14
        ike-auth-algorithm sha-256
        isakmp-lifetime 90000
    }
```

Example: classic CLI

```
A:node-2>config>ipsec# info
----------------------------------------------
        ike-transform 100 create
            dh-group 14
            ike-auth-algorithm sha256
            isakmp-lifetime 90000
        exit
        ike-policy 100 create
            ike-version 2
            auth-method eap
            ike-transform 100
        exit
----------------------------------------------
```

3.8.5 Configuring IPsec in services

The following example displays an IES and VPRN service with IPsec command options configured.

Example: MD-CLI

```
[ex:/configure service]
A:admin@node-2# info
    ies "100" {
        admin-state enable
        customer "1"
        interface "ipsec-public" {
            sap tunnel-1.public:1 {
            }
            ipv4 {
                primary {
                    address 10.10.10.1
                    prefix-length 24
                }
            }
        }
    }

    vprn "200" {
    admin-state enable
    customer "1"
    ipsec {
        security-policy 1 {
            entry 1 {
                local-ip {
                    address 172.16.118.0/24
```

```
                }
                remote-ip {
                    address 172.16.91.0/24
                }
            }
        }
    }
    bgp-ipvpn {
        mpls {
            admin-state enable
            route-distinguisher "1:1"
        }
    }
    interface "corporate-network" {
        ipv4 {
            primary {
                address 172.16.118.118
                prefix-length 24
            }
        }
        sap 1/1/2 {
        }
    }
    interface "ipsec-private" {
        tunnel true
        sap tunnel-1.private:1 {
            ipsec-tunnel "remote-office" {
                key-exchange {
                    dynamic {
                        ike-policy 1
                        ipsec-transform [1]
                        pre-shared-key "0PuXIi3wAoMOkSffbuDDViSy4jnc4N2vRrT1Qw== hash2"
                    }
                }
                tunnel-endpoint {
                    local-gateway-address 10.10.10.118
                    remote-ip-address 10.10.7.91
                    delivery-service "delivery"
                }
                security-policy {
                    id 1
                }
            }
        }
    static-routes {
        route 172.16.91.0/24 route-type unicast {
            ipsec-tunnel "t1" {
                admin-state enable
            }

    }
```

Example: classic CLI

```
A:node-2>config# info
----------------------------------------------
...
    service
        ies 100 name "100" customer 1 create
            interface "ipsec-public" create
                address 10.10.10.1/24
                tos-marking-state untrusted
                sap tunnel-1.public:1 create
```

```
                exit
            exit
            no shutdown
        exit
  vprn 200 customer 1 create
            ipsec
                security-policy 1 create
                    entry 1 create
                        local-ip 172.16.118.0/24
                        remote-ip 172.16.91.0/24
                    exit
                exit
            exit
            route-distinguisher 1:1
            interface "ipsec-private" tunnel create
                sap tunnel-1.private:1 create
                    ipsec-tunnel "remote-office" create
                        security-policy 1
                        local-gateway-address 10.10.10.118 peer 10.10.7.91 delivery-
service-name delivery
                        dynamic-keying
                            ike-policy 1
                            pre-shared-key "humptydumpty"
                            transform 1
                        exit
                        no shutdown
                    exit
                exit
            exit
            interface "corporate-network" create
                address 172.16.118.118/24
                sap 1/1/2 create
                exit
            exit
            static-route-entry 172.16.91.0/24
                ipsec-tunnel "t1"
                    no shutdown
                exit
            exit
            no shutdown
        exit
    exit
...
----------------------------------------------
```

3.8.6 Configuring X.509v3 certificate command options

The following are steps to configure certificate enrollment:

1. Generate a key.
   - **MD-CLI**

     ```
     admin system security pki generate-keypair cf3:/key_plain_rsa2048
     admin system security pki generate-keypair rsa-key-size 2048
     ```

   - **classic CLI**

     ```
     admin certificate gen-keypair cf3:/key_plain_rsa2048 size 2048 type rsa
     ```

2. Generate a certificate request.
   - **MD-CLI**

```
admin system security pki generate-csr key-url cf3:/key_plain_rsa2048 output-url cf3:/
7750-req.cs subject-dn "C=US,ST=CA,CN=7750"
```

   - **classic CLI**

```
admin certificate gen-local-cert-req keypair cf3:/key_plain_rsa2048 subject-
dn "C=US,ST=CA,CN=7750" file 7750_req.cs
```

3. Send the certificate request to CA-1 to sign and get the signed certificate.
4. Import the key.
   - **MD-CLI**

```
admin system security pki import type key input-url cf3:/key_plain_rs2048 output-file
 key1_rsa2048 format der
```

   - **classic CLI**

```
admin certificate import type key input cf3:/key_plain_rsa2048 output key1_rsa2048 format
 der
```

5. Import the signed certificate.
   - **MD-CLI**

```
admin system security pki import type cert input-url cf3:/7750_cert.pem output-file
 7750cert format pem
```

   - **classic CLI**

```
admin certificate import type cert input cf3:/7750_cert.pem output 7750cert format pem
```

The following are steps to configure CA certificate/CRL import.

1. Import the CA certificate.
   - **MD-CLI**

```
admin system security pki import type certificate input-url cf3:/CA_1_cert.pem output-file
 ca_cert format pem
```

   - **classic CLI**

```
admin certificate import type cert input cf3:/CA_1_cert.pem output ca_cert format pem
```

2. Import the CA’s CRL.
   - **MD-CLI**

```
admin system security pki import type certificate input-url cf3:/CA_1_crl.pem output-file
 ca_crl format pem
```

- **classic CLI**

```
admin certificate import type crl input cf3:/CA_1_crl.pem output ca_crl format pem
```

The following example displays a certificate authentication for IKEv2 static LAN-to-LAN tunnel configuration.

Example: MD-CLI

```
[ex:/configure system security pki]
A:admin@node-2# info
    ca-profile "NOKIA-root" {
        admin-state enable
        cert-file "NOKIA_root.cert"
        crl-file "NOKIA_root.crl"
    }

[ex:/configure ipsec]
A:admin@node-2# info
    cert-profile "segw" {
        admin-state enable
        entry 1 {
            cert "segw.cert"
            key "segw.key"
        }
    }
    ike-policy 1 {
        ike-transform [1]
        ike-version-2 {
            auth-method cert
        }
    }
    ike-transform 1 {
    }
    ipsec-transform 1 {
    }
    trust-anchor-profile "nokia" {
        trust-anchor "NOKIA-root" { }
    }

[ex:/configure service vprn "200" interface "ipsec-private" sap tunnel-1.private:1]
A:admin@node-2# info
    ipsec-tunnel "t50" {
        admin-state enable
        key-exchange {
            dynamic {
                ike-policy 1
                ipsec-transform [1]
                cert {
                    cert-profile "segw"
                    trust-anchor-profile "nokia"
                }
            }
        }
        tunnel-endpoint {
            local-gateway-address 192.168.55.30
            remote-ip-address 192.168.33.100
            delivery-service "delivery"
        }
        security-policy {
            id 1
        }
```

```
    }
```

Example: classic CLI

```
A:node-2>config>system>security>pki# info
----------------------------------------------
                ca-profile "NOKIA-root" create
                    cert-file "NOKIA_root.cert"
                    crl-file "NOKIA_root.crl"
                    no shutdown
                exit
----------------------------------------------

A:node-2>config>ipsec# info
----------------------------------------------
        ike-policy 1 create
            ike-version 2
            auth-method cert-auth
            ike-transform 1
        exit
        ipsec-transform 1 create
        exit
        ike-transform 1 create
        exit
       cert-profile "segw" create
            entry 1 create
                cert segw.cert
                key segw.key
            exit
            no shutdown
        exit
        trust-anchor-profile "nokia" create
            trust-anchor "nokia-root"
        exit
----------------------------------------------

A:node-2>config>service>vprn>if>sap
----------------------------------------------
                    ipsec-tunnel "t50" create
                        security-policy 1
                        local-gateway-address 192.168.55.30 peer 192.168.33.100 delivery-
service delivery
                        dynamic-keying
                            ike-policy 1
                            transform 1
                            cert
                                trust-anchor-profile "nokia"
                                cert-profile "segw"
                            exit
                        exit
                        no shutdown
                    exit
```

3.8.7 Configuring and using CMPv2

CMPv2 server information is configured using the commands in the following context.

```
configure system security pki ca-profile cmpv2
```

The following command options are configured in this context:

- **url**

  This command option specifies the HTTP URL of the CMPv2 server. The service specifies the routing instance that the system used to access the CMPv2 server (if omitted, then system uses the base routing instance).

- **service name or service ID**

  This command option is only needed for in-band connections to the server via VPRN services. IES services are not referenced by the service ID as they use the base routing instance.

- **response-signing-cert**

  This command option specifies an imported certificate used to verify the CMP response message if it is protected by signature. If this command is not configured, the CA certificate is used.

- **key-list**

  This command option specifies a list of pre-shared keys used for CMPv2 initial registration message protection.

  If there is no key list defined in the CMPv2 configuration, the system defaults to the CMPv2 transaction input for the command line for authenticating a message without a sender ID. Also, if there is no sender ID in the response message, and there is a key list defined, the system chooses the lexicographical first entry only, and if that fails, it outputs a fail result for the transaction.

All CMPv2 operations are invoked by using the following command:

- **MD-CLI**

  ```
  admin system security pki cmpv2
  ```

- **classic CLI**

  ```
  admin certificate cmpv2
  ```

The system also supports optional commands (such as, **always-set-sender-ir**) to support inter-op with CMPv2 servers.

The following example displays CMPv2 configuration.

Example: MD-CLI

```
[ex:/configure system security pki ca-profile "profile" cmpv2]
A:admin@node-2# info
    response-signing-cert "filename"
    url {
        url-string "http://cmp.example.com/request"
        service-name "foo"
    }
    key-list {
        key "1" {
            password "RGwT0+xs+Zb9708mSFdzMCxTCu8ykxuSpA2mpHzFwzU= hash2"
        }
    }
```

Example: classic CLI

```
A:node-2>config>system>security>pki>ca-profile>cmpv2$ info
----------------------------------------------
                        url "http://cmp.example.com/request" service-name "foo"
                        key-list
```

```
                                key "RGwT0+xs+Zb9708mSFdzMCxTCu8ykxuSpA2mpHzFwzU=" hash2
 reference "1"
                        exit
                        response-signing-cert "filename"
----------------------------------------------
```

3.8.8 Configuring OCSP

OCSP server information is configured using the following command.

```
configure system security pki ca-profile ocsp
```

The **responder-url** command option specifies the HTTP URL of the OCSP responder. The service ID or service name command option specifies the routing instance that system used to access the OCSP responder.

For an IPsec tunnel or IPsec gateway, the user can configure a primary method, a secondary method and a default result using the following commands:

- **MD-CLI**

```
configure ipsec ipsec-transport-mode-profile key-exchange dynamic cert status-verify
configure router interface ipsec ipsec-tunnel key-exchange dynamic cert status-verify
configure service ies interface ipsec ipsec-tunnel key-exchange dynamic cert status-verify
configure service vprn interface ipsec ipsec-tunnel key-exchange dynamic cert status-verify
configure service vprn interface sap ipsec-tunnel key-exchange dynamic cert status-verify
```

- **classic CLI**

```
configure service vprn interface sap ipsec-gw cert status-verify
configure service ies interface sap ipsec-gw cert status-verify
configure service vprn interface ipsec ipsec-tunnel dynamic-keying cert status-verify
configure service vprn interface sap ipsec-tunnel dynamic-keying cert status-verify
configure service ies interface ipsec ipsec-tunnel dynamic-keying cert status-verify
```

The following example, shows OCSP configured as the primary method and CRL as the secondary method.

Example: MD-CLI

```
[ex:/configure service ies "2" interface "ipsec-pub" sap tunnel-1.public:100 ipsec-gateway
 "foo"]
A:admin@node-2# info
    cert {
        status-verify {
            primary ocsp
            secondary crl
        }
    }
```

Example: classic CLI

```
A:node-2>config>service>ies>if>sap>ipsec-gw# info
----------------------------------------------
                        shutdown
                        cert
                            status-verify
```

```
                                primary ocsp secondary crl
                            exit
                        exit
----------------------------------------------
```

3.8.9 Configuring IKEv2 remote-access tunnel

The following are configuration tasks for an IKEv2 remote-access tunnel:

- Create an IKE policy with one of the authentication methods that enabled the remote-access tunnel.
- Configure a tunnel template or IPsec transform. This is the same as configuring a dynamic LAN-to-LAN tunnel.
- Create a RADIUS authentication policy and optionally, a RADIUS accounting policy (a RADIUS server policy and a RADIUS server must be preconfigured).
- Configure a private VPRN service and private tunnel interface with an address on the interface. The internal address assigned to the client must come from the subnet on the private interface.
- Configure a public IES or VPRN service and an IPsec gateway under the public tunnel SAP.
- Configure the RADIUS authentication policy and RADIUS accounting policy (optional) under the IPsec gateway.
- Configure a certificate if any certificate-related authentication method is used.

The following example shows an IKEv2 configuration using cert-radius.

Example: MD-CLI

```
[ex:/configure system security pki]
A:admin@node-2# info
    ca-profile "NOKIA-ROOT" {
        admin-state enable
        cert-file "NOKIA-ROOT.cert"
        crl-file "NOKIA-ROOT.crl"
    }

[ex:/configure aaa]
A:admin@node-2# info
    radius {
        server-policy "femto-aaa" {
            servers {
                router-instance "management"
                server 1 {
                    server-name "svr-1"
                }
            }
        }
    }

[ex:/configure router "Base"]
A:admin@node-2# info
    radius {
        server "svr-1" {
            address 10.10.1.2
            secret "LCa0a2j/xo/5m0U8HTBBNJYdag== hash2"
        }
    }

[ex:/configure ipsec]
```

```
A:admin@node-2# info
    cert-profile "c1" {
        admin-state enable
        entry 1 {
            cert "SeGW2.cert"
            key "SeGW2.key"
        }
    }
    ike-policy 1 {
        ike-transform [1]
        ike-version-2 {
            auth-method cert-radius
        }
    }
    ike-transform 1 {
    }
    ipsec-transform 1 {
    }
    tunnel-template 1 {
        ipsec-transform [1]
    }
    trust-anchor-profile "tap-1" {
        trust-anchor "NOKIA-ROOT" { }
    }
    radius {
        accounting-policy "femto-acct" {
            radius-server-policy "femto-aaa"
            include-radius-attribute {
                calling-station-id true
                framed-ip-addr true
            }
        }
        authentication-policy "femto-auth" {
            radius-server-policy "femto-aaa"
            password "CQsjXp648Zfy3ZJ5NyIsV7gcSkI= hash2"
            include-radius-attribute {
                called-station-id true
                calling-station-id true
            }
        }
    }


[ex:/configure service ies "2"]
A:admin@node-2# info
    admin-state enable
    customer "1"
    interface "pub" {
        sap tunnel-1.public:100 {
            ipsec-gateway "rw" {
                admin-state enable
                default-tunnel-template 1
                ike-policy 1
                cert {
                    cert-profile "c1"
                    trust-anchor-profile "tap-1"
                }
                default-secure-service {
                    service-name "priv"
                    interface "priv"
                }
                radius {
                    accounting-policy "femto-acct"
                    authentication-policy "femto-auth"
```

```
                }
            }
        }
        ipv4 {
            primary {
                address 172.16.100.0
                prefix-length 31
            }
        }
    }

[ex:/configure service vprn "1"]
A:admin@node-2# info
    admin-state enable
    bgp-ipvpn {
        mpls {
            admin-state enable
            route-distinguisher "400:11"
        }
    }
    interface "l1" {
        loopback true
        ipv4 {
            primary {
                address 10.9.9.9
                prefix-length 32
            }
        }
    }
    interface "priv" {
        tunnel true
        ipv4 {
            addresses {
                address 10.20.20.1 {
                    prefix-length 24
                }
            }
        }
        sap tunnel-1.private:200 {
        }
    }
```

Example: classic CLI

```
A:node-2>config>system>security>pki# info
----------------------------------------------
                ca-profile "NOKIA-ROOT" create
                    cert-file "NOKIA-ROOT.cert"
                    crl-file "NOKIA-ROOT.crl"
                    no shutdown
                exit
----------------------------------------------

A:node-2>config>aaa# info
----------------------------------------------
        radius-server-policy "femto-aaa" create
            servers
                router "management"
                server 1 name "svr-1"
            exit
        exit
----------------------------------------------
```

```
A:node-2>config>router# info
----------------------------------------------
        radius-server
            server "svr-
1" address 10.10.10.1 secret "KR35xB3W4aUXtL8o3WzPD." hash2 create
            exit
        exit
----------------------------------------------

A:node-2>config>ipsec# info
----------------------------------------------
        ike-policy 1 create
            ike-version 2
            auth-method cert-radius
            ike-transform 1
        exit
        ipsec-transform 1 create
        exit
        ike-transform 1 create
        exit
        tunnel-template 1 create
            transform 1
        exit
        cert-profile "c1" create
            entry 1 create
                cert SeGW2.cert
                key SeGW2.key
            exit
            no shutdown
        exit
        trust-anchor-profile "tap-1" create
            trust-anchor "NOKIA-ROOT"
        exit
radius-authentication-policy "femto-auth" create
            include-radius-attribute
                calling-station-id
                called-station-id
            exit
            password "DJzlyYKCefyhomnFcFSBuLZovSemMKde" hash2
            radius-server-policy "femto-aaa"
        exit
        radius-accounting-policy "femto-acct" create
            include-radius-attribute
                calling-station-id
                framed-ip-addr
            exit
            radius-server-policy "femto-aaa"
        exit
----------------------------------------------

A:node-2>config>service>ies# info
----------------------------------------------
            interface "pub" create
                address 172.16.100.0/31
                tos-marking-state untrusted
                sap tunnel-1.public:100 create
                    ipsec-gw "rw"
                        cert
                            trust-anchor-profile "tap-1"
                            cert-profile "c1"
                        exit
                        default-secure-service 400 interface "priv"
                        default-tunnel-template 1
                        ike-policy 1
```

```
                        local-gateway-address 172.16.100.1
                        radius-accounting-policy "femto-acct"
                        radius-authentication-policy "femto-auth"
                        no shutdown
                    exit
                exit
            exit
            no shutdown
----------------------------------------------

A:node-2>config>service>vprn# info
----------------------------------------------
            interface "priv" tunnel create
                address 10.20.20.1/24
                sap tunnel-1.private:200 create
                exit
            exit
            interface "l1" create
                address 10.9.9.9/32
                loopback
            exit
            bgp-ipvpn
                mpls
                    route-distinguisher 400:11
                    no shutdown
                exit
            exit
            no shutdown
----------------------------------------------
```

3.8.10 Configuring IKEv2 remote-access tunnel with local address assignment

The following are configuration tasks of IKEv2 remote-access tunnel:

- Create an IKE policy with any authentication method.
- Configure the tunnel template or IPsec transform. This is the same as configuring a dynamic LAN-to-LAN tunnel.
- Configure a private VPRN service and a private tunnel interface with an address on the interface. The internal address assigned to the client must come from the subnet on the private interface.
- Configure a local DHCPv4 or DHCPv6 server with address pool that from which the internal address to be assigned from.
- Configure public IES or VPRN service and IPsec gateway under public tunnel SAP.
- Configure the local address assignment under the IPsec gateway.

The following example shows an IKEv2 remote-access tunnel using cert-auth.

Example: MD-CLI

```
[ex:/configure system security pki]
A:admin@node-2# info
      ca-profile "smallcell-root" {
        admin-state enable
        cert-file "smallcell-root-ca.cert"
        revocation-check crl-optional
    }
```

```
[ex:/configure ipsec]
A:admin@node-2# info
     cert-profile "segw-mlab" {
        admin-state enable
        entry 1 {
            cert "SeGW-MLAB.cert"
            key "SeGW-MLAB.key"
        }
    }
    ike-policy 3 {
        ike-transform [1]
        ike-version-2 {
            auth-method cert
        }
        nat-traversal {
        }
    }
    ike-transform 1 {
    }
    ipsec-transform 1 {
    }
    tunnel-template 1 {
        ipsec-transform [1]
    }
    trust-anchor-profile "sc-root" {
        trust-anchor "smallcell-root" {
        }
    }


[ex:/configure service ies "2"]
A:admin@node-2# info
    admin-state enable
    customer "1"
    interface "pub" {
        sap tunnel-1.public:100 {
            ipsec-gateway "rw" {
                admin-state enable
                default-tunnel-template 1
                ike-policy 3
                cert {
                    cert-profile "segw-mlab"
                    trust-anchor-profile "sc-root"
                    status-verify {
                        default-result good
                    }
                }
                default-secure-service {
                    service-name "priv"
                    interface "priv"
                }
                local {
                    gateway-address 172.16.100.1
                    id {
                        fqdn "segwmobilelab.nokia.com"
                    }
                    address-assignment {
                        admin-state enable
                        ipv6 {
                            router-instance "priv"
                            dhcp-server "d6"
                            pool "1"
                        }
                    }
```

```
                }
            }
        }
        ipv4 {
            primary {
                address 172.16.100.253
                prefix-length 24
            }
        }
    }


[ex:/configure service vprn "3"]
A:admin@node-2# info
    admin-state enable
    bgp-ipvpn {
        mpls {
            admin-state enable
            route-distinguisher "400:1"
        }
    }
    interface "priv" {
        admin-state enable
        tunnel true
        sap tunnel-1.private:200 {
        }
        ipv6 {
            address 2001:db8:beef::101 {
                prefix-length 96
            }
        }
    }
    dhcp-server {
        dhcpv6 "d6" {
            admin-state enable
            pool-selection {
                use-pool-from-client {
                }
            }
            pool "1" {
                options {
                    option dns-server {
                        hex-string 0x20010db8beef00000000000000000808
                    }
                }
                prefix 2001:db8:beef::/96 {
                    failover-control-type access-driven
                }
                exclude-prefix 2001:db8:beef::101/128 { }
            }
        }
    }
```

Example: classic CLI

```
A:node-2>config>system>security>pki# info
----------------------------------------------
                ca-profile "smallcell-root" create
                    cert-file "smallcell-root-ca.cert"
                    revocation-check crl-optional
                    no shutdown
                exit
----------------------------------------------
```

```
A:node-2>config>ipsec# info
----------------------------------------------
        ike-policy 3 create
            ike-version 2
            auth-method cert-auth
            nat-traversal
            ike-transform 1
        exit
        ipsec-transform 1 create
        exit
        ike-transform 1 create
        exit
        cert-profile "segw-mlab" create
            entry 1 create
                cert SeGW-MLAB.cert
                key SeGW-MLAB.key
            exit
            no shutdown
        exit
        trust-anchor-profile "sc-root" create
            trust-anchor "smallcell-root"
        exit
        tunnel-template 1 create
            transform 1
        exit
----------------------------------------------

A:node-2>config>service>ies# info
----------------------------------------------
            interface "pub" create
                address 172.16.100.253/24
                tos-marking-state untrusted
                sap tunnel-1.public:100 create
                    ipsec-gw "rw"
                        default-secure-service 400 interface "priv"
                        default-tunnel-template 1
                        ike-policy 3
                        local-address-assignment
                            ipv6
                                address-source router 400 dhcp-server "d6" pool "1"
                            exit
                            no shutdown
                        exit
                        local-gateway-address 172.16.100.1
                        cert
                            trust-anchor-profile "sc-root"
                            cert-profile "segw-mlab"
                            status-verify
                                default-result good
                            exit
                        exit
                        local-id type fqdn value segwmobilelab.nokia.com
                        no shutdown
                    exit
                exit
            exit
            no shutdown
----------------------------------------------

A:node-2>config>service>vprn# info
----------------------------------------------
            dhcp6
                local-dhcp-server "d6" create
```

```
                    use-pool-from-client
                    pool "1" create
                        options
                            dns-server 2001:db8:::808:808
                        exit
                        exclude-prefix 2001:db8:beef::101/128
                        prefix 2001:db8::beef::/96 failover access-driven pd wan-host
 create
                        exit
                    exit
                    no shutdown
                exit
            exit
            bgp-ipvpn
                mpls
                    route-distinguisher 400:1
                    no shutdown
                exit
            exit
            interface "priv" tunnel create
                ipv6
                    address 2001:db8:beef::101/96
                exit
                sap tunnel-1.private:200 create
                exit
            exit
            no shutdown
```

3.8.11 Configuring secured interfaces

The following example displays a configuration for a secured interface. In this example, a SI tunnel "t1" is configured under interface "toPeer-1" in Base routing instance, along with an exception filter 100 that allows OSPF packets bypass IPsec processing.

Example: MD-CLI

```
[ex:/configure filter]
A:admin@node-2# info
    ip-exception "100" {
        entry 10 {
            match {
                protocol ospf-igp
            }
        }
    }


*[ex:/configure router "Base"]
A:admin@node-2# info
    interface "toPeer-1" {
        ipsec {
            tunnel-group 1
            public-sap 300
            ip-exception "100"
            ipsec-tunnel "t1" {
                private-sap 300
                local-gateway-address-override 192.168.110.20
                remote-gateway-address 172.16.21.1
                key-exchange {
                    dynamic {
                        ike-policy 3
```

```
                    ipsec-transform [2]
                    pre-shared-key "mEg8brQLbDGkSMIqZt7TtTpKaT7xPCY= hash2"
                }
            }
            security-policy {
                id 1
            }
        }
    }
    ipv4 {
        primary {
            address 192.168.110.20
            prefix-length 24
        }
    }
}
ipsec {
    security-policy 1 {
        entry 1 {
            local-ip {
                address 100.0.0.20/32
            }
            remote-ip {
                address 200.1.1.254/32
            }
        }
    }
}
```

Example: classic CLI

```
config>filter# info
----------------------------------------------
        ip-exception 100 create
            entry 10 create
                match protocol ospf-igp
                exit
            exit
        exit
----------------------------------------------
config>router# info
----------------------------------------------
#--------------------------------------------------
echo "IPsec Configuration"
#--------------------------------------------------
        ipsec
            security-policy 1 create
                entry 1 create
                    local-ip 100.0.0.20/32
                    remote-ip 200.1.1.254/32
                exit
            exit
        exit
#--------------------------------------------------
echo "IP Configuration"
#--------------------------------------------------
        interface "toPeer-1"
            address 192.168.110.20/24
            port 1/1/3
            ipsec tunnel-group 1 public-sap 300
                ip-exception 100
                ipsec-tunnel "t1" private-sap 300 create
                    local-gateway-address 192.168.110.20
```

```
                    remote-gateway-address 172.16.21.1
                    security-policy 1
                    dynamic-keying
                        ike-policy 3
                        pre-shared-key "KrbVPnF6Dg13PM/biw6ErD9+g6HZ" hash2
                        transform 2
                    exit
```

3.9 Quantum-safe IPsec

A quantum computer is built on top of quantum mechanics. A key use case for quantum computers is the potential capability to break existing asymmetric cryptography algorithms, such as Diffie-Hellman (DH) exchange, RSA, ECDSA, and so on. For IPsec, this impacts the following areas:

1. Key exchange
2. PKI-based authentication

Standards for quantum-safe algorithms are in development, but in the interim, a phased approach is required in the migration to quantum-safe IPSec. Of the two impacted areas, key exchange is more urgent to address than PKI authentication, because it serves as the security foundation of IPsec protocols.

3.9.1 Secure IKEv2 key exchange via PPK

Traditionally, IKEv2 uses DH or Elliptic-curve DH (ECDH) for all its key derivations during IKE_SA_INIT or IKE_SA/CHILD_SA rekey exchanges.

SR OS supports the ability to mix a Post-quantum Preshared Key (PPK) into the IKEv2 key derivation process, along with traditional DH or ECDH exchanges, as defined in RFC 8784. This feature adds quantum resistance to the IKEv2 protocol until quantum-safe algorithms become available for use. The PPK is a user-provisioned, preshared key that is used to derive the following keys, on top of the key derivation process defined in RFC 7296:

- SK_d - used to derive the CHILD_SA keys and rekey
- SK_pi - used for authentication
- SK_pr - used for authentication

If the PPK has enough entropy, the ESP traffic and IKE traffic after the first IKE_SA rekey are secured against quantum computers.

Note: The IKE traffic before the first IKE_SA rekey is not protected by PPK.

PPK is supported with all IKEv2 authentication methods.

Example: IKEv2 exchange with PPK

```
Initiator                      Responder
-----------------------------------------------------------------
HDR, SAi1, KEi, Ni, N(USE_PPK)  --->
                <--- HDR, SAr1, KEr, Nr, [CERTREQ,] N(USE_PPK)

HDR, SK {IDi, [CERT,] [CERTREQ,]
    [IDr,] AUTH, SAi2,
```

```
    TSi, TSr, N(PPK_IDENTITY, PPK_ID), [N(NO_PPK_AUTH)]}  --->
                           <--  HDR, SK {IDr, [CERT,]
                                AUTH, SAr2,
                                TSi, TSr, N(PPK_IDENTITY)}
```

Both peers signal support for PPK by including the USE_PPK notification in the IKE_SA_INIT exchange. Multiple PPKs can be provisioned, and each PPK is assigned a unique ID. The initiator must also include the ID of the selected PPK in the PPK_IDENTITY notification of IKE_AUTH request message and, optionally, include a NO_PPK_AUTH if using PPK is optional. The responder handles the IKE_AUTH request message in accordance with RFC 8784. The following table describes this handling.

Table 3: IKE authentication with PPK logic for responder

Received USE_PPK	Received NO_PPK_AUTH	Configured with PPK	PPK is Mandatory	Action
No	–	No	–	Standard IKEv2 protocol
No	–	Yes	No	Standard IKEv2 protocol
No	–	Yes	Yes	Abort negotiation
Yes	No	No	–	Abort negotiation
Yes	Yes	No	Yes	Abort negotiation
Yes	Yes	No	No	Standard IKEv2 protocol
Yes	–	Yes	–	Use PPK

3.9.2 Configuring PPK

About this task

Perform the following steps to configure PPK.

Procedure

Step 1. Use commands in the following context to configure a PPK list. The list can contain multiple PPKs, each with a unique ID and value.

```
configure ipsec ppk-list
```

Example

MD-CLI

```
[gl:/configure ipsec]
A:admin@node-2# ppk-list t1 ppk ppk-1 value hex 0x0123456789

[gl:/configure ipsec]
```

```
A:admin@Dut-AD# ppk-list t1 ppk ppk-2 value ascii abcd1234++
```

```
[gl:/configure ipsec]
A:admin@node-2# info
    ppk-list "t1" {
        ppk "ppk-1" {
            value {
                hex "2ySr4tCiD6yVeFnrdVs9v2ME1oRx hash2"
            }
        }
        ppk "ppk-2" {
            value {
                ascii "fHNsiyqqP9VW+Ovb+jEP1PN1zXyH/8ajq64= hash2"
            }
        }
    }
```

Example

classic CLI

```
A:node-2>config>ipsec# ppk-list t1 create ppk-id ppk-1 format hex value 0x123456789
A:node-2>config>ipsec# ppk-list t1 create ppk-id ppk-1 format ascii value abcd1234++
```

```
A:node-2>config>ipsec# info
----------------------------------------------
        ppk-list "t1" create
            ppk-id "ppk-1" format hex value "2ySr4tCiD6yVeFnrdVs9vy4MTT0N" hash2
            ppk-id "ppk-2" format ascii value "fHNsiyqqP9VW+Ovb+jEP1MD9JO9h8kbJMtQ="
 hash2
        exit
```

Step 2. For the IPsec gateway, use the following command to reference the PPK list in the tunnel template.

```
configure ipsec tunnel-template ppk-list
```

Example

MD-CLI

```
(gl:/configure ipsec tunnel-template 1]
A:admin@node-2# info
    ppk-list "t1"
```

Example

classic CLI

```
A:node2>config>ipsec>tnl-temp# info
----------------------------------------------
            ppk-list "t1"
```

Step 3. For the IPsec tunnel, use commands in the following contexts to reference a specific PPK in the PPK list.

- **MD-CLI**

```
configure router interface ipsec ipsec-tunnel key-exchange dynamic ppk
configure service ies interface ipsec-tunnel key-exchange dynamic ppk
```



```
configure service vprn interface sap ipsec-tunnel key-exchange dynamic ppk
configure service vprn interface ipsec-tunnel key-exchange dynamic ppk
```

- **classic CLI**

```
configure router interface ipsec ipsec-tunnel dynamic-keying ppk
configure service ies interface ipsec-tunnel dynamic-keying ppk
configure service vprn interface sap ipsec-tunnel dynamic-keying ppk
configure service vprn interface ipsec-tunnel dynamic-keying ppk
```

Example

MD-CLI

```
[gl:/configure service vprn "400" interface "priv" sap tunnel-1.private:100 ipsec-
tunnel "t1" key-exchange dynamic]
A:admin@node-2# ppk list t1 id ppk-id 1
```

```
[gl:/configure service vprn "400" interface "priv" sap tunnel-1.private:100 ipsec-
tunnel "t1" key-exchange dynamic]
A:admin@node-2# info
    ppk {
        list "t1"
        id "ppk-1"
    }
```

Example

classic CLI

```
A:node-02>config>service>vprn>if>sap>ipsec-tun>dyn$ ppk list "t1" id "ppk-id1"
```

```
A:node-02>config>service>vprn>if>sap>ipsec-tun>dyn$ info
----------------------------------------------
                            ppk list "t1" id "ppk-1"
```

Step 4. Use the following command to configure the mandatory use of PPK under the IKE policy.

- **MD-CLI**

```
configure ipsec ike-policy ike-version-2 ppk-required
```

- **classic CLI**

```
configure ipsec ike-policy ppk-required
```

Example

MD-CLI

```
(gl)[/configure ipsec ike-policy 1 ike-version-2]
A:admin@node-2# info
    ppk-required true
```

Example

classic CLI

```
A:node-2>config>ipsec>ike-policy# info
```

```
-----------------------------------------------
            ike-version 2
            ppk-required
```

4 Network Address Translation

4.1 Terminology

- **deterministic NAT**

  This is a mode of operation where mappings between the inside IP address and the outside IP address and port-range are allocated at the time of configuration. Each IP address host subscriber is permanently mapped to an outside IP and a dedicated port block. This dedicated port block is referred to as deterministic port block. Logging is not needed as the reverse mapping can be obtained using a known formula. The subscriber's ports can be expanded by allocating a dynamic port block in case that all ports in deterministic port block are exhausted. In such case logging for the dynamic port block allocation/de-allocation is required.

- **Large Scale NAT (LSN)**

  This refers to a collection of network address translation techniques used in service provider network implemented on a highly scalable, high performance hardware that facilitates various intra and inter-node redundancy mechanisms. The purpose of LSN semantics is to make delineation between high scale and high performance NAT functions found in service provider networks and enterprise NAT that is usually serving much smaller customer base at smaller speeds. The following NAT techniques can be grouped under the LSN name:

  – Large Scale NAT44 or Carrier Grade NAT (CGN)

  – DS-Lite

  – NAT64

  Each distinct NAT technique is referred to by its corresponding name (Large Scale NAT44 [or CGN], DS-Lite and NAT64) with the understanding that in the context of 7705 SAR Gen 2 platform, they are all part of LSN (and not enterprise based NAT).

  Large Scale NAT44 term can be interchangeably used with the term Carrier Grade NAT (CGN) which in its name implies high reliability, high scale and high performance. These are again typical requirements found in service provider (carrier) network.

- **NAT RADIUS accounting**

  This is the reporting (or logging) of address translation related events (port-block allocation/de-allocation) via RADIUS accounting facility. NAT RADIUS accounting is facilitated via regular RADIUS accounting messages (start/interim-update/stop) as defined in RFC 2866, *RADIUS Accounting,* with NAT specific VSAs.

- **NAT subscriber**

  In NAT terminology, a NAT subscriber is an inside entity whose true identity is hidden from the outside. There are a few types of NAT implementation in 7705 SAR Gen 2 and subscriber definitions for each implementation are defined as follows:

  – **Large Scale NAT44**

    The NAT subscriber is an inside IPv4 address.

- **non-deterministic NAT**

This is a mode of operation where all outside IP address and port block allocations are made dynamically at the time of subscriber instantiation. Logging in such case is required.

- **port block**

  This is collection of ports that is assigned to a subscriber. A deterministic LSN subscriber can have only one deterministic port block that can be extended by multiple dynamic port blocks. Non-deterministic LSN subscriber can be assigned only dynamic port blocks. All port blocks for a LSN subscriber must be allocated from a single outside IP address.

- **port-range**

  This is a collection of ports that can spawn multiple port blocks of the same type. For example, deterministic port-range includes all ports that are reserved for deterministic consumption. Similarly dynamic port-range is a total collection of ports that can be allocated in the form of dynamic port blocks. Other types of port-ranges are well-known ports and static port forwards.

4.2 Network Address Translation (NAT) overview

The 7705 SAR Gen 2 supports Network Address (and port) Translation (NAPT) to provide continuity of legacy IPv4 services during the migration to native IPv6. By equipping the virtual multiservice ISA-BB (MS ISA-BB) in a slot, the 7705 SAR Gen 2 can operate in in the following mode:

- Large Scale NAT

This mode performs source address and port translation as commonly deployed for shared Internet access. The 7705 SAR Gen 2 with NAT is used to provide Internet access to IPv4 Internet resources with a shared pool of IPv4 addresses.

4.2.1 Principles of NAT

Network Address Translation devices modify the IP headers of packets between a host and server, changing some or all of the source address, destination address, source port (TCP/UDP), destination port (TCP/UDP), or ICMP query ID (for ping). The 7705 SAR Gen 2 performs Source Network Address and Port Translation (S-NAPT). S-NAPT devices are commonly deployed in residential gateways and enterprise firewalls to allow multiple hosts to share one or more public IPv4 addresses to access the Internet. The common terms of inside and outside in the context of NAT refer to devices inside the NAT (that is behind or masqueraded by the NAT) and outside the NAT, on the public Internet.

TCP/UDP connections use ports for multiplexing, with 65536 ports available for every IP address. Whenever many hosts are trying to share a single public IP address there is a chance of port collision where two different hosts may use the same source port for a connection. The resultant collision is avoided in S-NAPT devices by translating the source port and tracking this in a stateful manner. All S-NAPT devices are stateful in nature and must monitor connection establishment and traffic to maintain translation mappings. The 7705 SAR Gen 2 NAT implementation does not use the well-known port range (1 to 1023).

In most circumstances, S-NAPT requires the inside host to establish a connection to the public Internet host or server before a mapping and translation occurs. With the initial outbound IP packet, the S-NAPT knows the inside IP, inside port, remote IP, remote port and protocol. With this information the S-NAPT device can select an IP and port combination (referred to as outside IP and outside port) from its pool of addresses and create a unique mapping for this flow of data.

Any traffic returned from the server uses the outside IP and outside port in the destination IP/port fields – matching the unique NAT mapping. The mapping then provides the inside IP and inside port for translation.

The requirement to create a mapping with inside port and IP, outside port and IP and protocol generally prevents new connections to be established from the outside to the inside as may occur when an inside host needs to be a server.

4.2.2 Application compatibility

Applications which operate as servers (such as HTTP, SMTP, and so on) or peer-to-peer applications can have difficulty when operating behind an S-NAPT because traffic from the Internet cannot reach the NAT without a mapping in place.

Different methods can be employed to overcome this such as port forwarding and STUN. The 7705 SAR Gen 2 supports both, following the best-practice RFC for TCP (RFC 5382, *NAT Behavioral Requirements for TCP*) and UDP (RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*). Port Forwarding is supported on the 7705 SAR Gen 2 to allow servers which operate on well-known ports <1024 (such as HTTP and SMTP) to request the appropriate outside port for permanent allocation.

STUN is facilitated by the support of Endpoint-Independent Filtering and Endpoint-Independent Mapping (RFC 4787) in the NAT device, allowing STUN-capable applications to detect the NAT and allow inbound P2P connections for that specific application. Many new SIP clients and IM chat applications are STUN capable.

4.3 Large-Scale NAT

Large-Scale NAT (LSN) functionality represents the most common deployment of S-NAPT in enterprise networks today for internet access.

LSN is typically deployed in a network location with two interfaces, the inside toward the local LANs, and the outside toward the Internet. A Large Scale NAT functions as an IP router and is located between two routed network segments (the ISP network and the Internet).

Traffic can be sent to the LSN function on the 7705 SAR Gen 2 using IP filters (ACL) applied to SAPs or by installing static routes with a next-hop of the NAT application. These two methods allow for increased flexibility in deploying the LSN, especially those environments where IP MPLS VPN are being used in which case the NAT function can be deployed on a single PE and perform NAT for any number of other PE by simply exporting the default route.

The 7705 SAR Gen 2 NAT implementation supports NAT in the base routing instance and VPRN, and through NAT traffic may originate in one VPRN (the inside) and leave through another VPRN or the base routing instance (the outside). This technique can be employed to provide customers of IP MPLS VPN with Internet access by introducing a default static route in the customer VPRN, and NATing it into the Internet routing instance.

As LSN is deployed between two routed segments, the IP addresses allocated to hosts on the inside must be unique to each host within the VPRN.

4.3.1 Port range blocks

The S-NAPT service on the 7705 SAR Gen 2 incorporates a port range block feature to address scalability of a NAT mapping solution. Port range blocks address the issue of logging and NAT subscriber functions by allocating a block of contiguous outside ports to a single NAT subscriber. Instead of logging each NAT mapping, a single log entry is created when the first mapping is created for a NAT subscriber and a final log entry when the last mapping is destroyed. This can substantially reduce the number of log entries.

Port range blocks are configurable as part of outside pool configuration, allowing the operator to specify the number of ports allocated to each NAT subscriber when a mapping is created. When a range is allocated to the NAT subscriber, these ports are used for all outbound dynamic mappings and are assigned in a random manner to minimize the predictability of port allocations (*draft-ietf-tsvwg-port-randomization-05*).

Port range blocks also serve another useful function in a Large Scale NAT environment, and that is to manage the fair allocation of the shared IP resources among different NAT subscribers.

When a NAT subscriber exhausts all ports in their block, further mappings are prohibited. As with any enforcement system, some exceptions are allowed and the NAT application can be configured for reserved ports to allow high-priority applications access to outside port resources while exhausted by low priority applications.

4.3.1.1 Reserved ports and priority sessions

Reserved ports allows an operator to configure a small number of ports to be reserved for designated applications should a port range block be exhausted. Such a scenario may occur when a NAT subscriber is unwittingly subjected to a virus or engaged in extreme cases of P2P file transfers. In these situations, instead of blocking all new mappings indiscriminately, the 7705 SAR Gen 2 NAT application allows operators to nominate a number of reserved ports and then assign a 7705 SAR Gen 2 forwarding class as containing high priority traffic for the NAT application. Whenever traffic reaches the NAT application which matches a priority session forwarding class, reserved ports are consumed to improve the chances of success. Priority sessions could be used by the operator for services such as DNS, web portal, e-mail, VoIP, and so on, to allow these applications even when a NAT subscriber exhausted their ports.

4.3.1.2 Preventing port block starvation

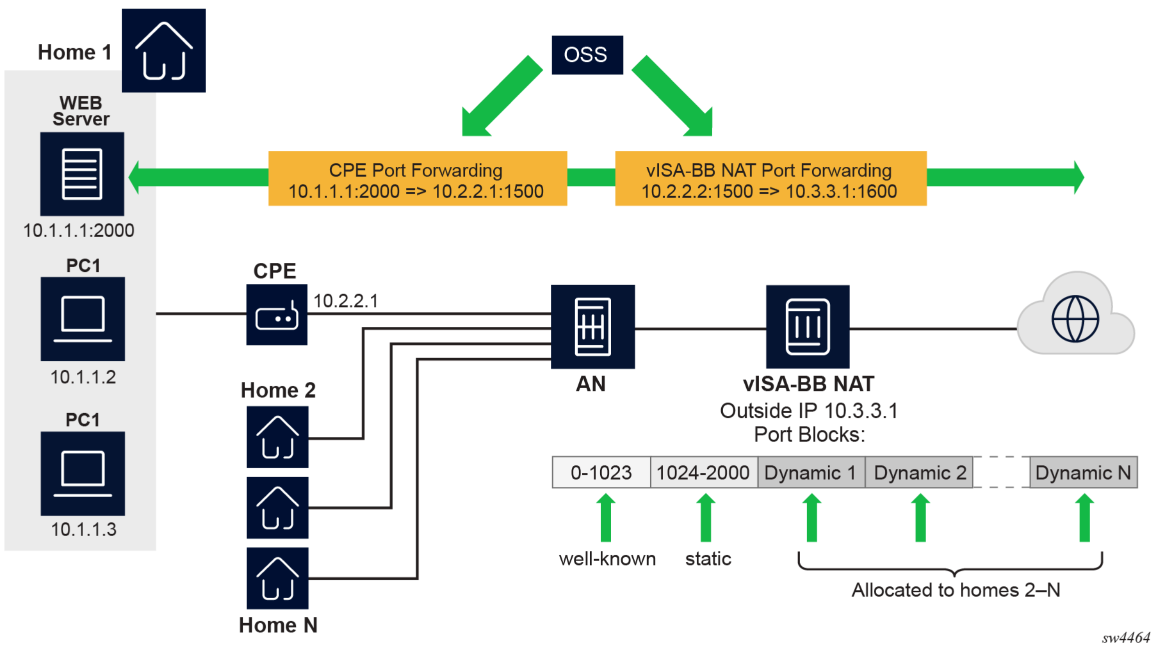
The outside IP address is always shared for the NAT subscriber with a port forward (static or via PCP) and the dynamically allocated port block, insofar as the port from the port forward is in the range >1023. This behavior can lead to starvation of dynamic port blocks for the subscriber. An example for this scenario is shown in Figure 9: Dynamic port block starvation in LSN.

- A static port forward for the WEB server in Home 1 is allocated in the CPE and the vISA-BB NAT application. At the time of static port forward creation, no other dynamic port blocks for Home 1 exist (PCs are powered off).
- Assume that the outside IP address for the newly created static port forward in the vISA-BB is 10.3.3.1.
- Over time dynamic port blocks are allocated for a number of other homes that share the same outside IP address, 10.3.3.1. Eventually those dynamic port block allocations exhaust all dynamic port block range for the address 10.3.3.1.
- After the dynamic port blocks are exhausted for outside IP address 10.3.3.1, a new outside IP address (for example, 10.3.3.2) is allocated for additional homes.

Eventually the PCs in Home 1 come to life and they try to connect to the Internet. Because of the dynamic port block exhaustion for the IP address 10.3.3.1 (that is mandated by static port forward – Web Server), the dynamic port block allocation fails and consequently, the PCs are not able to access the Internet. There is no additional attempt within the vISA-BB NAT to allocate another outside IP address. In the vISA-BB NAT, there is no distinction between the PCs in Home 1 and the Web Server when it comes to source IP address. They both share the same source IP address 10.2.2.1 on the CPE.

The solution for this is to reserve a port block (or blocks) during the static port forward creation for the specific subscriber.

Figure 9: Dynamic port block starvation in LSN



To prevent starvation of dynamic port blocks for the NAT subscribers that use port forwards, a dynamic port block (or blocks) is reserved during the lifetime of the port forward. Those reserved dynamic port blocks are associated with the same NAT subscriber that created the port forward. However, a log would not be generated until the dynamic port block is actually used and mapping within that block are created.

At the time of the port forward creation, the dynamic port block is reserved in the following fashion:

- If the dynamic port block for the NAT subscriber does not exist, then a dynamic port block for the NAT subscriber is reserved. No log for the reserved dynamic port block is generated until the dynamic port block starts being used (mapping created because of the traffic flow).
- If the corresponding dynamic port block already exists, it is reserved even after the last mapping within the last port block had expired.

The reserved dynamic port block (even without any mapping) continues to be associated with the NAT subscriber as long as the port forward for the NAT subscriber is present. The log (syslog or RADIUS) is generated only when there is not active mapping within the dynamic port block and all port forwards for the NAT subscriber are deleted.

Additional considerations with dynamic port block reservation:

- The port block reservation should be triggered only by the first port forward for the NAT subscriber. The subsequent port forwards do not trigger additional dynamic port block reservation.
- Only a single dynamic port block for the NAT subscriber is reserved (that is, no multiple port-block reservations for the NAT subscriber are possible).
- This feature is enabled with the following commands:
  - **MD-CLI**

    ```
    configure router nat outside pool port-forwarding dynamic-block-reservation
    configure service vprn nat outside pool port-forwarding dynamic-block-reservation
    ```

  - **classic CLI**

    ```
    configure router nat outside pool port-forwarding-dyn-block-reservation
    configure service vprn nat outside pool port-forwarding-dyn-block-reservation
    ```

  These commands can be enabled only if the maximum number of configured port blocks per outside IP is greater or equal then the maximum configured number of NAT subscribers per outside IP address. This guarantees that all NAT subscribers (up to the maximum number per outside IP address) configured with port forwards can reserve a dynamic port block.
- If the port-reservation is enabled while the outside pool is operational and NAT subscriber's traffic is already present, the following two cases must be considered:
  - The configured number of NAT subscribers per outside IP is less or equal than the configured number of port blocks per outside IP address (this is permitted) but all dynamic port blocks per outside IP address are occupied at the moment when port reservation is enabled. This leaves existing NAT subscribers with port forwards that do not have any dynamic port blocks allocated (orphaned NAT subscribers), unable to reserve dynamic port blocks. In this case the orphaned NAT subscribers must wait until dynamic port blocks allocated to the NAT subscribers without port forwards are freed.
  - The configured number of NAT subscribers per outside IP is greater than the configured number of port blocks per outside IP address. In addition, all dynamic port blocks per outside IP address are allocated. Before the port reservation is even enabled, the NAT subscriber-limit per outside IP address must be lowered (by configuration) so that it is equal or less than the configured number of port blocks per outside IP address. This action causes random deletion of NAT mappings that do not have any port forwards. Such NAT mappings are deleted until the number of NAT subscriber falls below the newly configured subscriber limit. NAT subscribers with static port forwards are not deleted, regardless of the configured subscriber-limit number. When the number of NAT subscribers is within the newly configured NAT subscriber limit, the port-reservation can take place under the condition that the dynamic port blocks are available. If specific NAT subscribers with port forwards have more than one dynamic port block allocated, the orphaned NAT subscribers must wait for those additional dynamic port blocks to expire and consequently be released.

4.3.2 Association between NAT subscribers and IP addresses in a NAT pool

A NAT subscriber can allocate ports on a single outside IP address or multiple IP addresses in a NAT pool. Nokia recommends that NAT subscribers allocate ports from a single outside IP address. If this IP address runs out of ports, the NAT subscriber runs out of ports. In other words, there is no attempt for a new port to be allocated from a different outside IP address. This method of address allocation to a NAT subscriber is referred to as Paired Address Pooling and is the default behavior in SR OS.

The alternative method of port allocation involves port exhaustion on the originally allocated IP address. An attempt is made to allocate ports from another IP addresses that has free ports available. This results in a NAT subscriber be associated with multiple outside IP addresses. This method is referred to as Arbitrary Address Pooling and can be optionally enabled in SR OS. See RFC 7857, Section 4 for more information.

Arbitrary address pooling may offer more efficient allocations of port-blocks across outside IP address in a NAT pool, but it may negatively affect some applications. For example, an application may require two channels for communication, a control channel and a data channel, each on a different source port from the client perspective on the inside of the NAT. The communication channel may be established on the outside address IP1 and outside port X. If port X is the last free port on the IP1, the SR OS attempts to allocate the next port Y for the data channel from a different outside address, IP2. If the application is robust enough to accept communication from the same client on two different IP addresses, there are no issues. However, some applications may not support this scenario and the communication fails.

Arbitrary address pooling implies the following:

- The NAT subscriber limit per outside IP address loses its meaning because the NAT subscriber can now be associated with multiple IP addresses. Hence, the following command cannot be set:
  - **MD-CLI**

    ```
    configure router nat outside pool large-scale subscriber-limit
    ```

  - **classic CLI**

    ```
    configure router nat outside pool subscriber-limit
    ```

- The number of port blocks configured in a NAT policy using the following command is the aggregate limit that a NAT subscriber can be allocated across multiple outside IP addresses.

  ```
  configure service nat nat-policy block-limit
  ```

- Reserving a port block by SPF configuration (when an SPF is configured before any port blocks are allocated to the NAT subscriber) is not supported. In other words, the following commands are not supported:

  MD-CLI

  ```
  configure router nat outside pool port-forwarding dynamic-block-reservation
  ```

  classic CLI

  ```
  configure router nat outside pool port-forwarding-dyn-block-reservation
  ```

- Arbitrary address pooling is not supported in Layer 2–aware NAT.

Use the following command to show NAT LSN information for the NAT subscriber.

```
show service nat lsn-subscribers subscriber
```

The asterisk (*) next to the IP address field in the output indicates that additional outside IP addresses are associated with this NAT subscriber in this pool.

Output example

```
===============================================================================
NAT LSN subscribers
```

```
===============================================================================
Subscriber                  : [LSN-Host@192.168.1.1]
NAT policy                  : nat-policy-lsn-deterministic
Subscriber ID               : 276824064
-------------------------------------------------------------------------------
Type                        : classic-lsn-sub
Inside router               : "Base"
Inside IP address prefix    : 192.168.1.1/32
ISA NAT group               : 1
ISA NAT group member        : 1
Outside router              : 4
Outside IP address          : 192.0.0.1*
```

Use the detailed version of the command to see additional outside IP addresses and port blocks.

4.3.3 Timeouts

Creating a NAT mapping is only one half of the problem – removing a NAT mapping at the appropriate time maximizes the shared port resource. Having ports mapped when an application is no longer active reduces solution scale and may impact the customer experience should they exhaust their port range block. The NAT application provides timeout configuration for TCP, UDP and ICMP.

TCP state is tracked for all TCP connections, supporting both three-way handshake and simultaneous TCP SYN connections. Separate and configurable timeouts exist for TCP SYN, TCP transition (between SYN and Open), established and time-wait state. Time-wait assassination is supported and enabled by default to quickly remove TCP mappings in the TIME WAIT state.

UDP does not have the concept of connection state and is subject to a simple inactivity timer. Company-sponsored research into applications and NAT behavior suggested some applications, like the BitTorrent Distributed Hash Protocol (DHT) can make a large number of outbound UDP connections that are unsuccessful. Instead of waiting the default five (5) minutes to time these out, the 7705 SAR Gen 2 NAT application supports an udp-initial timeout which defaults to 15 seconds. When the first outbound UDP packet is sent, the 15 second time starts – it is only after subsequent packets (inbound or outbound) that the default UDP timer becomes active, greatly reducing the number of UDP mappings.

4.4 NAT pool addresses and ICMP Echo Request/Reply (ping)

The outside IPv4 addresses in a NAT pool can be configured to answer pings. ICMPv4 Echo Requests are answered with ICMPv4 Echo Replies.

In 1:1 NAT, ICMP Echo Requests are propagated to the host on the inside. The host identified by a NAT binding then answers the ping.

In Network Address Port Translation (NAPT), ICMP Echo Requests are not propagated to the hosts behind the NAT. Instead, the reply is issued by the SR OS from the ESA or ISA.

In Layer 2-aware NAT, use the following command to configure how replies from outside IP addresses are handled:

- **MD-CLI**

```
configure router nat outside pool l2-aware port-block-extension
```

- **classic CLI**

  ```
  configure router nat outside pool port-block-extensions
  ```

In NAPT, the behavior is as follows:

- In Layer 2–aware NAT when **port-block-extensions** is disabled, the reply from an outside IP address is generated only when the IP address has at least one host (binding) behind it.
- In Layer 2–aware NAT when **port-block-extensions** is enabled, the reply from an outside IP address is generated regardless if a binding is present.
- In LSN, the reply from an outside IP address is generated regardless if a binding is present.

For security reasons, the ICMP Echo Reply functionality is disabled by default. Use the following command to enable ICMP Echo Reply functionality.

```
configure router nat outside pool icmp-echo-reply
```

This functionality is on a per-pool basis and it can be configured online while the pool is enabled.

4.5 One-to-one (1:1) NAT

In 1:1 NAT, each source IP address is translated in 1:1 fashion to a corresponding outside IP address. However, the source ports are passed transparently without translation.

The mapping between the inside IP addresses and outside IP addresses in 1:1 NAT supports two modes:

- **dynamic**

  The user can specify the outside IP addresses in the pool, but the exact mapping between the inside IP address and the configured outside IP addresses is performed dynamically by the system in a semi-random fashion.
- **static**

  The mappings between IP addresses are configurable and they can be explicitly set.

The dynamic version of 1:1 NAT is protocol dependent. Only TCP/UDP/ICMP protocols are allowed to traverse such NAT. All other protocols are discarded, with the exception of PPTP with ALG. In this case, only GRE traffic associated with PPTP is allowed through dynamic 1:1 NAT.

The static version of 1:1 NAT is protocol agnostic. This means that all IP based protocols are allowed to traverse static 1:1 NAT.

The following points are applicable to 1:1 NAT:

- Even though source ports are not being translated, the state maintenance for TCP and UDP traffic is still performed.
- Traffic can be initiated from outside toward any statically mapped IPv4 address.
- 1:1 NAT can be supported simultaneously with NAPT (classic non 1:1 NAT) within the same inside routing context. This is accomplished by configuring two separate NAT pools, one for 1:1 NAT and the other for non 1:1 NAPT.

4.5.1 Static 1:1 NAT

In static 1:1 NAT, inside IP addresses are statically mapped to the outside IP addresses. This way, devices on the outside can predictably initiate traffic to the devices on the inside.

The following example shows a static 1:1 NAT configuration.

Example: Static 1:1 NAT configuration (MD-CLI)

```
[gl:/configure router "Base" nat inside large-scale nat44 deterministic]
A:admin@sr-1s# info
    address-map 10.10.0.220 to 10.10.0.220 nat-policy "cgn44" {
        outside-range 192.168.255.206
    }
    address-map 10.10.0.221 to 10.10.0.221 nat-policy "cgn44" {
        outside-range 192.168.255.207
    }
    address-map 10.10.0.222 to 10.10.0.222 nat-policy "cgn44" {
        outside-range 192.168.255.208
    }
    address-map 10.10.0.223 to 10.10.0.223 nat-policy "cgn44" {
        outside-range 192.168.255.209
    }
```

Example: Static 1:1 NAT configuration (classic CLI)

```
A:node-2>config>router>nat>inside>deterministic# info
----------------------------------------------
            address-map 10.10.0.220 to 10.10.0.220 subscriber-type classic-lsn-sub nat-
policy "cgn44" create
                outside-range 192.168.255.206
            exit
            address-map 10.10.0.221 to 10.10.0.221 subscriber-type classic-lsn-sub nat-
policy "cgn44" create
                outside-range 192.168.255.207
            exit
            address-map 10.10.0.222 to 10.10.0.222 subscriber-type classic-lsn-sub nat-
policy "cgn44" create
                outside-range 192.168.255.207
            exit
            address-map 10.10.0.223 to 10.10.0.223 subscriber-type classic-lsn-sub nat-
policy "cgn44" create
                outside-range 192.168.255.206
            exit
----------------------------------------------
```

Static mappings are configured according to the map statements:

- In the MD-CLI, the map statement must be configured by the user, but the following command can be used to produce system-generated maps.

  ```
  tools perform nat deterministic calculate-maps
  ```

  The preceding command outputs a set of system-generated map statements. The map command options can then be copied and pasted into an MD-CLI candidate configuration by the user.
- In classic CLI, the map statement can be configured manually by the user or automatically by the system.

IP addresses from the automatically-generated map statements are sequentially mapped into available outside IP addresses in the pool:

- The first inside IP address is mapped to the first available outside IP address from the pool.
- The second inside IP address is mapped to the second available outside IP address from the pool.

The following mappings apply to the preceding example.

Table 4: Static mappings

Inside IP address	Outside IP address
10.10.0.220	192.168.255.206
10.10.0.221	192.168.255.207
10.10.0.222	192.168.255.208
10.10.0.223	192.168.255.209

4.5.1.1 Protocol-agnostic behavior

Although static 1:1 NAT is protocol agnostic, the state maintenance for TCP and UDP traffic is still required to support ALGs. Therefore, the existing scaling limits related to the number of supported flows still apply.

The following example shows protocol-agnostic behavior in 1:1 NAT is a property of a NAT pool.

Example: Protocol-agnostic behavior configuration (MD-CLI)

```
[ex:/configure router "Base" nat outside]
A:admin@node-2# info
    pool "one-to-one" {
        admin-state enable
        type large-scale
        nat-group 1
              mode one-to-one
        applications {
            agnostic true
        }
        port-forwarding {
            range-start 0
            range-end 0
        }
        port-reservation {
            port-blocks 1
        }
        large-scale {
            subscriber-limit 1
            }
            deterministic {
                port-reservation 65325
            }
        address-range 192.168.2.0 end 192.168.2.10 {
        }
    }
```

Example: Protocol-agnostic behavior configuration (classic CLI)

```
A:node-2>config>router>nat>outside# info
----------------------------------------------
               pool "one-to-one" nat-group 1 type large-scale applications agnostic create
                    no shutdown
                    port-reservation blocks 1
                    port-forwarding-range 0 0
                    subscriber-limit 1
                    deterministic
                        port-reservation 65325
                    exit
                    mode one-to-one
                    address-range 192.168.2.0 192.168.2.10 create
                    exit
                exit
----------------------------------------------
```

The application **agnostic** command is a pool create-time command. This command automatically pre-sets the following pool command options:

- mode is set to one-to-one
- port forwarding range start is set to 0
- port forwarding range end is set to 0
- number of port reservation blocks is set to 1
- the subscriber limit is set to 1
- the deterministic port reservation is set to 65325, which configures the pool to operate in static (or deterministic) mode

When pre-set, these command options cannot be changed while the pool is operating in protocol agnostic mode.

4.5.1.2 Modification of parameters in static 1:1 NAT

Note: This information applies for the classic CLI.

In classic CLI only, command options in the static 1:1 NAT can be changed according to the following rules:

- The deterministic pool must be in a **no shutdown** state when a **prefix** or a **map** command in deterministic NAT is added or removed.
- All configured prefixes referencing the pool via the NAT policy must be deleted (unconfigured) before the pool can be shut down.
- Map statements can be modified only when prefix is shutdown state. All existing map statements must be removed before the new ones are created.

These rules do not apply in MD-CLI.

4.5.1.3 NAT-policy selection

The traffic match criteria used in the selection of specific NAT policies in static 1:1 NAT (the deterministic part of the configuration) must not overlap with traffic match criteria that is used in the selection of a specific NAT policy used in filters or in destination-prefix statement (these are used for traffic diversion to NAT). Otherwise, traffic is dropped in ISA.

A specific NAT policy in this context refers to a non-default NAT policy, or a NAT policy that is directly referenced in a filter, in a destination prefix or a deterministic prefix.

The following example is used to clarify this point.

Example: NAT policy selection (MD-CLI)

```
[ex:/configure router "Base" nat inside large-scale]
A:admin@node-2# info
    nat44 {
        max-subscriber-limit 128
        destination-prefix 192.0.2.0/24 {
            nat-policy "pol-2"
        }
        deterministic {
            prefix-map 10.10.10.0/24 nat-policy "pol-1" {
                map 10.10.10.0 to 10.10.10.255 {
                    first-outside-address 192.168.0.1
                }
            }
        }
    }
```

Example: NAT policy selection (classic CLI)

```
A:node-2>config>router>nat>inside# info
----------------------------------------------
                destination-prefix 192.0.2.0/24 nat-policy "pol-2"
                classic-lsn-max-subscriber-limit 128
                deterministic
                    prefix-map 10.10.10.0/24 subscriber-type classic-lsn-sub nat-policy
 "pol-1" create
                        shutdown
                        map start 10.10.10.0 end 10.10.10.255 to 192.168.0.1
                    exit
                exit
----------------------------------------------
```

In the preceding example:

- Traffic is diverted to NAT using specific **nat-policy** *pol-2*.
- The deterministic (source) prefix 10.10.10.0/24 is configured to be mapped to **nat-policy** pol-1 specifically which points to protocol agnostic 1:1 NAT pool.
- Packets received in the ISA have a source IP of 10.10.10.0/24 and a destination IP of 192.0.2.0/24.
- If no NAT mapping for this traffic exists in the ISA, a NAT policy (and with this, the NAT pool) must be determined to create the mapping. Traffic is diverted to NAT using NAT policy *pol-2*, while the deterministic mapping suggests that the NAT policy *pol-1* should be used (this is a different pool from the one referenced in NAT policy *pol-2*). Because of the specific NAT policy conflict, traffic is dropped in the ISA.

To successfully pass traffic between two subnets through NAT while simultaneously using static 1:1 NAT and regular LSN44, a default (non-specific) NAT policy can be used for regular LSN44.

A specific NAT policy (in a filter, **destination-prefix** command, or in deterministic **prefix-map** command) always takes precedence over a default NAT policy. However, traffic that matches classification criteria (in a filter, **destination-prefix** command, or a deterministic **prefix-map** command) that leads to multiple specific NAT policies, is dropped.

In this case, the four hosts from the prefix 10.10.10.0/24 are mapped in 1:1 fashion to 4 IP addresses from the pool referenced in the specific NAT policy *pol-1*, while all other hosts from the 10.10.10.0/24 network are mapped to the NAPT pool referenced by the default NAT policy *pol-2*. In this way, a NAT policy conflict is avoided.

4.5.1.4 Mapping timeout

Static 1:1 NAT mappings are explicitly configured, and therefore, their lifetime is tied to the configuration.

4.5.1.5 Logging

The logging mechanism for static mapping is the same as in Deterministic NAT. Configuration changes are logged via syslog and enhanced with reverse querying on the system.

4.5.1.6 Restrictions

Static 1:1 NAT is supported only for LSN44. There is no support for DS-Lite/NAT64 or Layer 2–aware NAT.

4.5.2 ICMP

In 1:1 NAT, specific ICMP messages contain an additional IP header embedded in the ICMP header. For example, when the ICMP message is sent to the source because of the inability to deliver datagram to its destination, the ICMP generating node includes the original IP header of the packet plus 64bits of the original datagram. This information helps the source node to match the ICMP message to the process associated with this message.

When these messages are received in the downstream direction (on the outside), 1:1 NAT recognizes them and changes the destination IP address not only in the outside header but also in the ICMP header. In other words, a lookup in the downstream direction is performed in the ISA to determine if the packet is ICMP with a specific type. Depending on the outcome, the destination IP address in the ICMP header is changed (reverted to the original source IP address).

Messages carrying the original IP header within ICMP header are:

- Destination Unreachable Messages (Type 3)
- Time Exceeded Message (Type 11)
- Parameter Problem Message (Type 12)
- Source Quench Message (Type 4)

4.6 LSN – multiple NAT policies per inside routing context

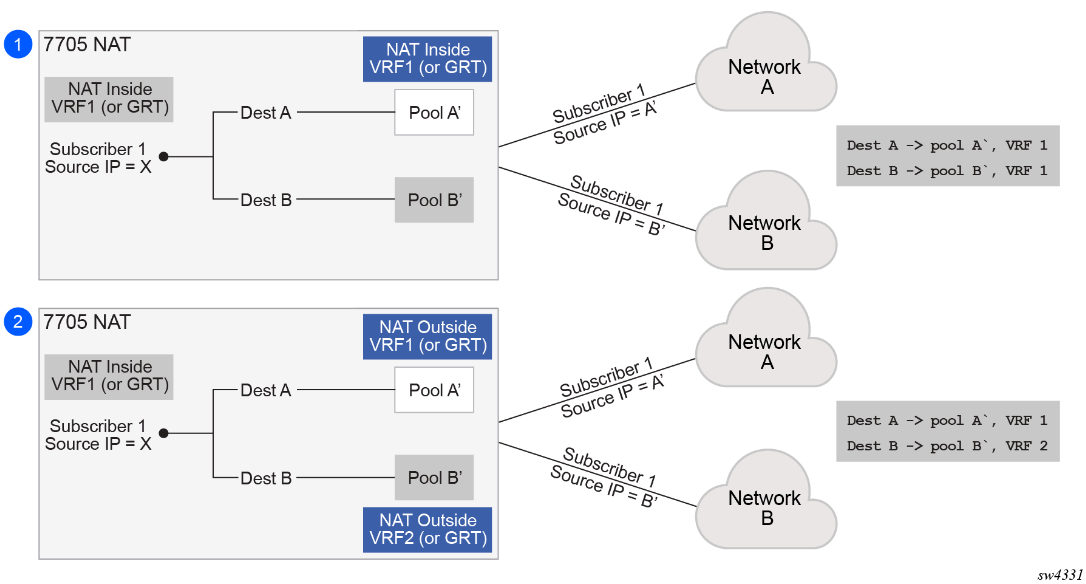
4.6.1 Multiple NAT policies per inside routing context

The selection of the NAT pool and the outside routing context is performed through the NAT policy. Multiple NAT policies can be used within an inside routing context. This feature effectively allows selective mapping of the incoming traffic within an inside routing context to different NAT pools (with different mapping properties, such as port-block size, NAT subscriber-limit per pool, address range, port-forwarding range, deterministic vs non-deterministic behavior, port-block watermarks, and so on) and to different outside routing contexts. NAT policies can be configured:

- via filters as part of the **action nat** command
- via routing with the **destination-prefix** command within the inside routing context

The concept of the NAT pool selection mechanism based on the destination of the traffic via routing is shown in Figure 10: Pool selection based on traffic destination.

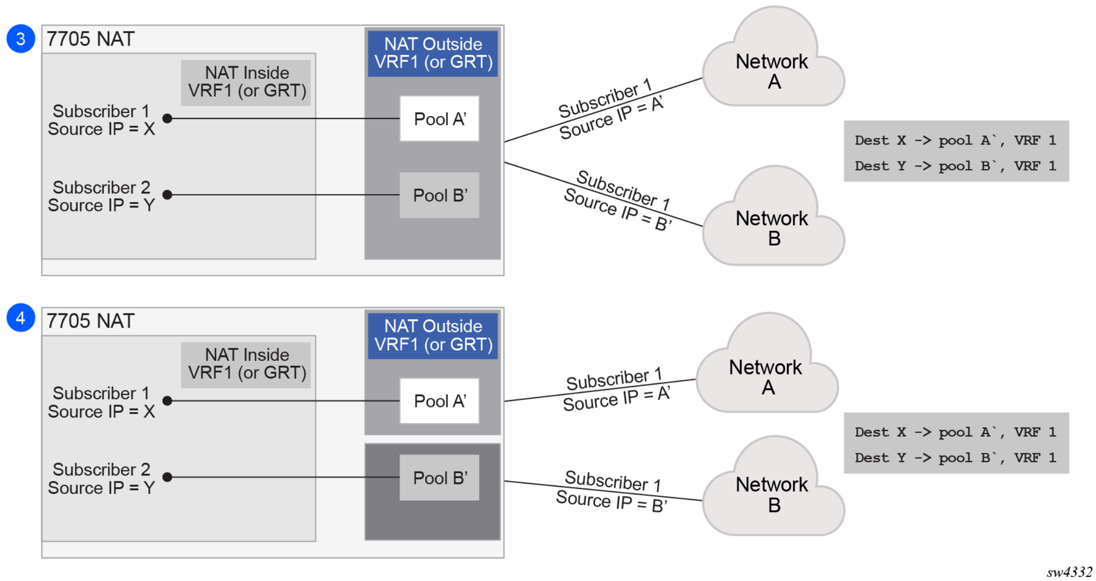
Figure 10: Pool selection based on traffic destination



Diversion of the traffic to NAT based on the source of the traffic is shown in Figure 11: NAT pool selection based on the inside source IP address.

Only the filter-based diversion solution is supported for this case. The filter-based solution can be extended to a five tuple matching criteria.

Figure 11: NAT pool selection based on the inside source IP address



The following considerations must be taken into account when deploying multiple NAT policies per inside routing context:

- The inside IP address can be mapped into multiple outside IP addresses based on the traffic destination. The relationship between the inside IP and the outside IP is 1:N.
- In case where the source IP address is selected as a matching criteria for a NAT policy (or pool) selection, the inside IP address always stays mapped to the same outside IP address (relationship between the inside IP and outside IP address is, in this case, 1:1)
- Static Port Forwards (SPF); each SPF can be created only in one pool. This means that the pool (or NAT policy) must be an input parameter for SPF creation.

4.6.2 Routing approach for NAT diversion

The routing approach relies on upstream traffic being directed (or diverted) to the NAT function based on the following commands:

- **MD-CLI**

```
configure service vprn nat inside large-scale nat44 destination-prefix
configure router nat inside large-scale nat44 destination-prefix
```

- **classic CLI**

```
configure service vprn nat inside destination-prefix
configure router nat inside destination-prefix
```

In other words, the upstream traffic is NAT’d only if it matches a preconfigured destination IP prefix. The **destination-prefix** command creates a static route in the routing table of the inside routing context. This static route diverts all traffic with the destination IP address that matches the created entry, toward the MS-ISA. The NAT function itself is performed when the traffic is in the correct context in the MS-ISA.

The following example displays the configuration of multiple NAT policies per inside routing context with routing based diversion to NAT.

Example: Configuring multiple NAT policies per inside routing context (MD-CLI)

```
[ex:/configure service vprn "66"]
A:admin@node-2# info
    customer "1"
    nat {
        inside {
            large-scale {
                nat44 {
                    destination-prefix 10.20.10.0/24 {
                        nat-policy "policy-1"
                    }
                    destination-prefix 10.30.30.0/24 {
                        nat-policy "policy-1"
                    }
                    destination-prefix 10.40.40.0/24 {
                        nat-policy "policy-2"
                    }
                }
            }
        }
    }

[ex:/configure router "Base"]
A:admin@node-2# info
    nat {
        inside {
            large-scale {
                nat44 {
                    max-subscriber-limit 256
                    destination-prefix 10.20.10.0/24 {
                        nat-policy "policy-1"
                    }
                    destination-prefix 10.30.30.0/24 {
                        nat-policy "policy-1"
                    }
                    destination-prefix 10.40.40.0/24 {
                        nat-policy "policy-2"
                    }
                }
            }
        }
    }
```

Example: Configuring multiple NAT policies per inside routing context (classic CLI)

```
A:node-2>config>service>vprn# info
----------------------------------------------
            shutdown
            nat
                inside
                    destination-prefix 10.20.10.0/24 nat-policy "policy-1"
                    destination-prefix 10.30.30.0/24 nat-policy "policy-1"
                    destination-prefix 10.40.40.0/24 nat-policy "policy-2"
                exit
            exit
----------------------------------------------

A:node-2>config>router# info
...
```

```
#--------------------------------------------------
echo "NAT Configuration"
#--------------------------------------------------
            shutdown
            nat
                inside
                    destination-prefix 10.20.10.0/24 nat-policy "policy-1"
                    destination-prefix 10.30.30.0/24 nat-policy "policy-1"
                    destination-prefix 10.40.40.0/24 nat-policy "policy-2"
                exit
            exit
----------------------------------------------
```

Different destination prefixes can reference a single NAT policy (policy-1 in this case).

In the case where the destination policy does not directly reference the NAT policy, the default NAT policy is used. The default NAT policy is configured directly in the following context:

- **MD-CLI**

```
configure service vprn nat inside large-scale
configure router nat inside large-scale
```

- **classic CLI**

```
configure service vprn nat inside
configure router nat inside
```

After the **destination-prefix** command referencing the NAT policy is configured, an entry in the routing table is created that directs the traffic to the MS-ISA.

4.6.3 Filter-based approach

Use the options under the following context to use a filter-based approach to divert traffic to NAT based on the IP matching criteria.

```
configure filter ip-filter entry match
```

Use the following command to use the filter-based diversion in conjunction with multiple NAT policies..

```
configure filter ip-filter entry action nat [nat-policy nat-policy-name]
```

The association with the NAT policy is made after the filter is applied to the SAP.

4.6.4 Scaling considerations

Each subscriber using multiple policies is counted as one NAT subscriber for the **inside** resources scaling limits (such as the number of subscribers per MS-ISA), and counted as one subscriber per (subscriber and policy combination) for the **outside** limits (**subscriber-limit** subscribers per IP; **port-reservation** port/block reservations per subscriber).

The default NAT policy is counted toward the maximum polices per NAT subscriber.

4.7 Watermarks

Watermarks can be configured to monitor the actual usage of sessions, ports, and port blocks.

For each watermark, a high and a low value must be set. When the high threshold value is crossed in the upward direction, an event is generated (SNMP trap), notifying the user that a NAT resource may be approaching exhaustion. When the low threshold value is crossed in the downward direction, a similar event is generated (clearing the first event), notifying the user that the resource utilization has dropped below the low threshold value.

Watermarks can be defined on the NAT group, pool, and policy level.

- **NAT group**

  Watermarks can be placed to monitor the total number of sessions on an MDA.

- **NAT pool on each NAT group member**

  Watermarks can be placed to monitor the port and port-block occupancy in a pool within each NAT group member.

- **NAT policy**

  In the policy, the user can define watermarks on port usage.

4.8 Port forwards

Port forwards allow devices on the public side of NAT (NAT outside) to initiate sessions toward those devices, usually servers, that are hidden behind NAT (NAT inside). Another term for port forwards is NAT pinhole.

A port forward represents a previously created (before any traffic is received from the inside) mapping between a TCP/UDP port on the outside IP address and a TCP/UDP port on the inside IP address assigned to a device behind the NAT. This mapping can be created statically by configuration using CLI, MIB, YANG, or NETCONF). Port forwards are supported only in NAT pools in Network Address and Port Translation (NAPT) mode. NAT pools in 1:1 mode do not support configured port forwards because, by default, the pools allow traffic from the outside to the inside and this cannot be disabled. Pools in 1:1 mode (whether protocol agnostic) do not perform port translation; therefore the inside and outside always match.

The forwarded ports are allocated from a dedicated port range outside of the port blocks allocated to individual NAT subscribers. There are two ranges dedicated to port forwards in NAT:

- **well-known ports (1 to 1023)**

  This range is always enabled and cannot be disabled in NAT pools that support configured port forwards (non 1:1 NAT pools).

- **ports from the ephemeral port range (1025 to 65535)**

  Port forwards from the ephemeral port space must be explicitly enabled by configuration. They are allocated from a contiguous block of ports where upper and lower limits are defined. Ports reserved for port forwards allocated in the ephemeral port space are also referred to as wildcard ports.

Port forwarding ranges (well-known ports and wildcard ports) are shared by all NAT subscribers on a specific outside IP address. Port blocks that are individually assigned to the NAT subscriber cannot be allocated from the port forwarding range. The wildcard port forwarding range can be configured only when the pool is administratively disabled.

4.8.1 Static port forwards

Use the command options in the following command to manage port forwarding for Large Scale NAT (LSN):

- **MD-CLI, NETCONF, and classic CLI**
  In the MD-CLI, NETCONF, and classic CLI, use the options under the following command to manage NAT Static Port Forwards (SPFs). This command enables large-scale NAT port forwarding actions.

  ```
  tools perform nat port-forwarding-action lsn
  ```

  For the preceding **tools** command, if you do not explicitly configure the following optional fields, the system selects them automatically:

  – port number – number of the source port

  – outside IP – IPv4 address for the outside IP address

  – outside-port number – number of the outside port

  – NAT policy – name of the NAT policy

  If the preceding **tools** command is configured to manage SPFs and preserve SPFs across reboots, you must use the following command to enable persistency of the SPF. With persistency enabled, SPF configuration is stored on the compact flash.

  ```
  configure system persistence nat-port-forwarding
  ```

- **classic CLI**

  In the classic CLI, you can manage SPFs through the preceding **tools** command or the following configuration command. This command creates NAT static port forwards for LSN44.

  ```
  configure service nat port-forwarding lsn
  ```

Example: Manage the SPFs using the tools command (MD-CLI)

```
[/tools perform nat port-forwarding-action]
A:admin@node-2# lsn add router 100 ip 10.2.3.4 protocol udp lifetime infinite outside-port
 888

[/]
*A:node-2# configure system persistence nat-port-forwarding location cf3

[/]
*A:node-2# tools dump persistence nat-port-forwarding
----------------------------------------
Persistence Info
----------------------------------------
Client : nat-fwds
File Info :
Filename : cf3:\nat_fwds.002
File State : CLOSED (Not enough space on disk)
Subsystem Info :
Nbr Of Registrations : 524288
Registrations In Use : 2
Subsystem State : NOK
```

Example: Manage the SPFs using the tools command (classic CLI)

```
*A:node-2# tools perform nat port-forwarding-action lsn create router 100
ip 10.2.3.4 protocol udp lifetime infinite outside-port 888
*A:node-2# configure system persistence nat-port-forwarding location cf3:
*A:node-2# tools dump persistence nat-port-forwarding
----------------------------------------
Persistence Info
----------------------------------------
Client : nat-fwds
File Info :
Filename : cf3:\nat_fwds.002
File State : CLOSED (Not enough space on disk)
Subsystem Info :
Nbr Of Registrations : 524288
Registrations In Use : 2
Subsystem State : NOK
```

Example: Manage the SPFs using the configuration command (classic CLI)

The following command only applies for the classic CLI.

```
*A:node-2>config>service>nat>fwd# lsn router 101 ip 11.11.13.7 protocol udp port 12345
 outside-ip 130.0.255.254 outside-port 3171 nat-policy "pol1_for_2001-pool-0"
```

You can specify a **force** option that is applicable only to LSN pools with flexible port allocations where the dynamic ports in this pool are allocated individually instead of port blocks. The dynamic ports are interleaved with Static Port Forwards (SPFs). This creates increased possibility for a collision between the dynamically-allocated port and the requested SPF during an SPF request.

For instance, if a user requests port X on a public IP address Y, there is a chance that port X is already in use because of the dynamic allocation.

To resolve such conflicts, use the **force** option to ensure that the requested SPF has higher priority, allowing it to preempt an existing dynamically-allocated port. This action overwrites the previous port mapping and deletes all associated sessions.

If you omit the **force** option in such a scenario, the static-port allocation fails. The **force** option can only preempt dynamically-allocated ports and does not affect pre-existing SPFs.

4.9 Modifying active NAT prefix list or NAT classifier via CLI

Table 5: Modifying active NAT prefix list or NAT classifier describes the outcome when the active NAT prefix list or NAT classifier is modified using CLI.

Table 5: Modifying active NAT prefix list or NAT classifier

Action	Outcome	Remarks
	LSN	
CLI – Modifying prefix in the	Changing the prefix in the NAT prefix list internally re-	NAT prefix list is used with multiple NAT policies in Layer 2–aware NAT and for downstream internal subnet in dNAT-only scenario for LSN.

Action	Outcome	Remarks
	LSN	
NAT prefix list	subnets the outside IP address space.	The prefix can be modified (added, removed, remapped) at any time in the NAT prefix list. In the classic CLI, the NAT policy must be first administratively disabled via CLI.
CLI – Removing/ adding NAT policy in the NAT prefix list	Not Applicable	—
CLI – Removing, adding, or replacing the NAT policy in sub-profile	Not Applicable	—
CLI – Removing, adding, or replacing the NAT prefix list under the rtr/nat/inside	Internally re-subnet, no effect on the flows	—

4.10 NAT logging

LSN logging is extremely important to operators who are required by their organizations to track source of suspicious activities.

The 7705 SAR Gen 2 supports several modes of logging for LSN applications. Choosing the right logging model depends on the required scale, simplicity of deployment and granularity of the logged data.

For most purposes logging of allocation or de-allocation of outside port-blocks and outside IP address along with the corresponding LSN subscriber and inside service-id is sufficient.

4.10.1 Syslog, SNMP, local-file logging

The simplest form of NAT logging is via the logging facility in the 7705 SAR Gen 2, commonly called logger. Each port-block allocation or de-allocation event is recorded and send to the system logging facility (logger). Such an event can be:

- recorded in the system memory as part of regular logs

- written to a local file
- sent to an external server by a syslog facility
- sent to a SNMP trap destination

In this mode of logging, all applications in the system share the same logger.

Syslog, SNMP, and local-file logging on LSN and NAT RADIUS-based logging are mutually exclusive.

Use the options under the following context to enable syslog, SNMP, and local-file logging for NAT:

- **MD-CLI**

  ```
  configure log log-events
  ```

- **classic CLI**

  ```
  configure log event-control
  ```

The following output example displays relevant MIB events.

Output example: Relevant MIB events

```
2012 tmnxNatPlBlockAllocationLsn
2013 tmnxNatPlBlockAllocationL2Aw
```

4.10.1.1 Filtering LSN events to system memory

In the following example, a single port-block [1884-1888] is allocated or de-allocated for the inside IP address 10.5.5.5 which is mapped to the outside IP address 198.51.100.1. Consequently, the event is logged in the memory as shown.

Output example: Event log memory output

```
2 2012/07/12 16:40:58.23 WEST MINOR: NAT #2012 Base NAT
"{2} Free 198.51.100.1 [1884-1888] -- vprn10 10.5.5.5 at 2012/07/12 16:40:58"

1 2012/07/12 16:39:55.15 WEST MINOR: NAT #2012 Base NAT
"{1} Map  198.51.100.1 [1884-1888] -- vprn10 10.5.5.5 at 2012/07/12 16:39:55"
```

When the needed LSN events are enabled for logging via the following configuration, they can be logged to memory through standard log ID 99 or be filtered with a custom log ID, such as in this example that follows (**log-id 5**).

Example: Enable LSN events for logging (MD-CLI)

```
[ex:/configure log]
A:admin@node-2# info
    log-events {
        nat event tmnxNatPlL2AwBlockUsageHigh {
            generate false
            throttle false
        }
        nat event tmnxNatIsaMemberSessionUsageHigh {
            generate false
            throttle false
        }
        nat event tmnxNatPlLsnMemberBlockUsageHigh {
```

```
            generate false
            throttle false
        }
        nat event tmnxNatL2AwSubIcmpPortUsageHigh {
            generate false
            throttle false
        }
        nat event tmnxNatL2AwSubUdpPortUsageHigh {
            generate false
            throttle false
        }
        nat event tmnxNatL2AwSubTcpPortUsageHigh {
            generate false
            throttle false
        }
        nat event tmnxNatL2AwSubSessionUsageHigh {
            generate false
            throttle false
        }
        nat event tmnxNatPlBlockAllocationLsn {
            generate true
        }
        nat event tmnxNatResourceProblemDetected {
            generate false
            throttle false
        }
        nat event tmnxNatResourceProblemCause {
            generate false
            throttle false
        }
        nat event tmnxNatPlLsnRedActiveChanged {
            generate false
            throttle false
        }
    }
    filter "1" {
        default-action drop
        named-entry "1" {
            action forward
            match {
                application {
                    eq nat
                }
                event {
                    eq 2012
                }
            }
        }
    }
    log-id "5" {
        filter "1"
        source {
            main true
        }
        destination {
            memory {
            }
        }
    }
```

Example: Enable LSN events for logging (classic CLI)

```
A:node-2>config>log# info
```

```
----------------------------------------------
        filter 1
            default-action drop
            entry 1
                action forward
                match
                    application eq "nat"
                    number eq 2012
                exit
            exit
        exit
        event-control "nat" 2001 suppress
        event-control "nat" 2002 suppress
        event-control "nat" 2003 suppress
        event-control "nat" 2004 suppress
        event-control "nat" 2005 suppress
        event-control "nat" 2006 suppress
        event-control "nat" 2007 suppress
        event-control "nat" 2008 suppress
        event-control "nat" 2009 suppress
        event-control "nat" 2010 suppress
        event-control "nat" 2011 suppress
        event-control "nat" 2012 generate
        event-control "nat" 2014 suppress
        event-control "nat" 2015 suppress
        event-control "nat" 2017 suppress
        syslog 10
        exit
        log-id 5 name "5"
            filter 1
            from main
            to memory
        exit
----------------------------------------------
```

Use the following command to display the log event information.

```
show log event-control "nat"
```

Output example: Log events output

```
=======================================================================
Log Events
=======================================================================
Application
 ID#    Event Name                       P   g/s     Logged     Dropped
-----------------------------------------------------------------------
   2001 tmnxNatPlL2AwBlockUsageHigh      WA  thr          0           0
   2002 tmnxNatIsaMemberSessionUsageHigh WA  thr          0           0
   2003 tmnxNatPlLsnMemberBlockUsageHigh WA  thr          0           0
   2007 tmnxNatL2AwSubIcmpPortUsageHigh  WA  thr          0           0
   2008 tmnxNatL2AwSubUdpPortUsageHigh   WA  thr          0           0
   2009 tmnxNatL2AwSubTcpPortUsageHigh   WA  thr          0           0
   2010 tmnxNatL2AwSubSessionUsageHigh   WA  thr          0           0
   2012 tmnxNatPlBlockAllocationLsn      MI  sup          0           0
   2013 tmnxNatPlBlockAllocationL2Aw     MI  sup          0           0
   2014 tmnxNatResourceProblemDetected   MI  thr          0           0
   2015 tmnxNatResourceProblemCause      MI  thr          0           0
   2016 tmnxNatPlAddrFree                MI  sup          0           0
   2017 tmnxNatPlLsnRedActiveChanged     WA  thr          0           0
   2018 tmnxNatPcpSrvStateChanged        MI  thr          0           0
   2020 tmnxNatMdaActive                 MI  thr          0           0
```

```
   2021 tmnxNatLsnSubBlksFree            MI  sup         0          0
   2022 tmnxNatDetPlcyChanged            MI  thr         0          0
   2023 tmnxNatMdaDetectsLoadSharingErr  MI  thr         0          0
   2024 tmnxNatIsaGrpOperStateChanged    MI  thr         0          0
   2025 tmnxNatIsaGrpIsDegraded          MI  thr         0          0
   2026 tmnxNatLsnSubIcmpPortUsgHigh     WA  thr         0          0
   2027 tmnxNatLsnSubUdpPortUsgHigh      WA  thr         0          0
   2028 tmnxNatLsnSubTcpPortUsgHigh      WA  thr         0          0
   2029 tmnxNatLsnSubSessionUsgHigh      WA  thr         0          0
   2030 tmnxNatInAddrPrefixBlksFree      MI  sup         0          0
   2031 tmnxNatFwd2EntryAdded            MI  sup         0          0
   2032 tmnxNatDetPlcyOperStateChanged   MI  thr         0          0
   2033 tmnxNatDetMapOperStateChanged    MI  thr         0          0
   2034 tmnxNatFwd2OperStateChanged      WA  thr         0          0
=======================================================================
```

The event description is shown in the MIB information that follows.

Output example: Event description output

```
tmnxNatPlL2AwBlockUsageHigh
        The tmnxNatPlL2AwBlockUsageHigh notification is sent when
         the block usage of a Layer-2-Aware NAT address pool
         reaches its high watermark ('true')
         or when it reaches its low watermark again ('false').

tmnxNatIsaMemberSessionUsageHigh
        The tmnxNatIsaMemberSessionUsageHigh notification is sent when
         the session usage of a NAT ISA group member reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatPlLsnMemberBlockUsageHigh
        The tmnxNatPlLsnMemberBlockUsageHigh notification is sent when
         the block usage of a Large Scale NAT address pool
         reaches its high watermark ('true')
         or when it reaches its low watermark again ('false')
         on a particular member MDA of its ISA group.

tmnxNatLsnSubIcmpPortUsageHigh
        The tmnxNatLsnSubIcmpPortUsageHigh notification is sent when
         the ICMP port usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatLsnSubUdpPortUsageHigh
        The tmnxNatLsnSubUdpPortUsageHigh notification is sent when
         the UDP port usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatLsnSubTcpPortUsageHigh
        The tmnxNatLsnSubTcpPortUsageHigh notification is sent when
         the TCP port usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatL2AwSubIcmpPortUsageHigh
        The tmnxNatL2AwSubIcmpPortUsageHigh notification is sent when
         the ICMP port usage of a Layer-2-Aware NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').
```

```
tmnxNatL2AwSubUdpPortUsageHigh
        The tmnxNatL2AwSubUdpPortUsageHigh notification is sent when
         the UDP port usage of a Layer-2-Aware NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatL2AwSubTcpPortUsageHigh
        The tmnxNatL2AwSubTcpPortUsageHigh notification is sent when
         the TCP port usage of a Layer-2-Aware NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatL2AwSubSessionUsageHigh
        The tmnxNatL2AwSubSessionUsageHigh notification is sent when
         the session usage of a Layer-2-Aware NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatLsnSubSessionUsageHigh
        The tmnxNatLsnSubSessionUsageHigh notification is sent when
         the session usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

tmnxNatPlBlockAllocationLsn
        The tmnxNatPlBlockAllocationLsn notification is sent when
         an outside IP address and a range of ports is allocated to
         a NAT subscriber associated with a Large Scale NAT (LSN) pool,
         and when this allocation expires.

tmnxNatPlBlockAllocationL2Aw
        The tmnxNatPlBlockAllocationL2Aw notification is sent when
         an outside IP address and a range of ports is allocated to
         a NAT subscriber associated with a Layer-2-Aware NAT pool,
         and when this allocation expires.

tmnxNatResourceProblemDetected
        The tmnxNatResourceProblemDetected notification is sent when
         the value of the object tmnxNatResourceProblem changes.

tmnxNatResourceProblemCause
        The tmnxNatResourceProblemCause notification is to describe the cause
         of a NAT resource problem.

tmnxNatPlAddrFree
        The tmnxNatPlAddrFree notification is sent when
         a range of outside IP addresses becomes free at once.

tmnxNatPlLsnRedActiveChanged
       The tmnxNatPlLsnRedActiveChanged notification is related to NAT Redundancy
       sent when the value of the object tmnxNatPlLsnRedActive changes. The cause is
       explained in the tmnxNatNotifyDescription which is a printable character
       string.

tmnxNatMdaActive
        The tmnxNatMdaActive notification is sent when
         the value of the object tmnxNatIsaMdaStatOperState changes from
         'primary' to any other value, or the other way around.
         The value 'primary' means that the MDA is active in the group.

tmnxNatLsnSubBlksFree
        The tmnxNatLsnSubBlksFree notification is sent when
         all port blocks allocated to a Large Scale NAT (LSN) subscriber
         are released.
```

```
         The NAT subscriber is identified with its subscriber ID
         tmnxNatNotifyLsnSubId.

         To further facilitate the identification of the NAT subscriber,
         its type tmnxNatNotifySubscriberType,
         inside IP address tmnxNatNotifyInsideAddr
         and inside virtual router instance tmnxNatNotifyInsideVRtrID
         are provided.

         The values of tmnxNatNotifyMdaChassisIndex, tmnxNatNotifyMdaCardSlotNum
         and tmnxNatNotifyMdaSlotNum identify the ISA MDA where the blocks were
         processed.

         All notifications of this type are sequentially numbered with
         the tmnxNatNotifyPlSeqNum.

         The value of tmnxNatNotifyNumber is the numerical identifier of the
         NAT policy used for this allocation; it can be used for correlation
         with the tmnxNatPlBlockAllocationLsn notification; the value zero
         means that this notification can be correlated with all the
         tmnxNatPlBlockAllocationLsn notifications of the subscriber.

tmnxNatDetPlcyChanged
        The tmnxNatDetPlcyChanged notification is sent when
         something changed in the Deterministic NAT map.

         [CAUSE] Such a change may be caused by a modification of the
        tmnxNatDetPlcyTable or the tmnxNatDetMapTable.

     [EFFECT] Traffic flows of one or more given subscribers, subject to NAT, may be
        assigned different outside IP address and/or outside port.

        [RECOVERY] Managers that rely on the offline representation of the
        Deterministic NAT map should get an updated copy.

tmnxNatMdaDetectsLoadSharingErr
        The tmnxNatMdaDetectsLoadSharingErr notification is sent
         periodically at most every 10 seconds while a NAT ISA MDA
         detects that it is receiving packets erroneously, due to
         incorrect load-balancing by the ingress IOM.

         The value of tmnxNatNotifyCounter is the incremental count of
         dropped packets since the previous notification sent by the same MDA.

         [CAUSE] The ingress IOM hardware does not support a particular
         NAT function's load-balancing, for example an IOM-2 does not
         support deterministic NAT.

         [EFFECT] The MDA drops all incorrectly load-balanced traffic.

         [RECOVERY] Upgrade the ingress IOM, or change the configuration.

tmnxNatIsaGrpOperStateChanged
        The tmnxNatIsaGrpOperStateChanged notification is sent when
         the value of the object tmnxNatIsaGrpOperState changes.

tmnxNatIsaGrpIsDegraded
        The tmnxNatIsaGrpIsDegraded notification is sent when
         the value of the object tmnxNatIsaGrpDegraded changes.

tmnxNatLsnSubIcmpPortUsgHigh
      The tmnxNatLsnSubIcmpPortUsgHigh notification is sent when
      the ICMP port usage of a Large Scale NAT subscriber reaches its high watermark
```

```
      ('true') or when it reaches its low watermark again ('false').

      The subscriber is identified with its inside IP address or prefix
      tmnxNatNotifyInsideAddr in the inside virtual router instance
      tmnxNatNotifyInsideVRtrID.

tmnxNatLsnSubUdpPortUsgHigh
        The tmnxNatLsnSubUdpPortUsgHigh notification is sent when
         the UDP port usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

         The subscriber is identified with its inside IP address or prefix
         tmnxNatNotifyInsideAddr in the inside virtual router instance
         tmnxNatNotifyInsideVRtrID.

tmnxNatLsnSubTcpPortUsgHigh
        The tmnxNatLsnSubTcpPortUsgHigh notification is sent when
         the TCP port usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

         The subscriber is identified with its inside IP address or prefix
          tmnxNatNotifyInsideAddr in the inside virtual router instance
          tmnxNatNotifyInsideVRtrID.

tmnxNatLsnSubSessionUsgHigh
        The tmnxNatLsnSubSessionUsgHigh notification is sent when
         the session usage of a Large Scale NAT subscriber reaches its high
         watermark ('true') or when it reaches its low watermark
         again ('false').

         The subscriber is identified with its inside IP address or prefix
         tmnxNatNotifyInsideAddr
         in the inside virtual router instance tmnxNatNotifyInsideVRtrID.

tmnxNatInAddrPrefixBlksFree
        The tmnxNatInAddrPrefixBlksFree notification is sent when
        all port blocks allocated to one or more subscribers
        associated with a particular set of inside addresses
        are released by this system.

        The type of subscriber(s) is indicated by tmnxNatNotifySubscriberType.

        The set of inside IP addresses is associated with the virtual
        router instance indicated by tmnxNatNotifyInsideVRtrID and is of the
         type indicated by tmnxNatNotifyInsideAddrType

        The set of inside IP addresses consists of the address prefix
        indicated with tmnxNatNotifyInsideAddr and tmnxNatNotifyInsideAddrPrefixLen
        unless these objects are empty and zero; if tmnxNatNotifyInsideAddr is empty
        and tmnxNatNotifyInsideAddrPrefixLen is zero, the set contains
        all IP addresses of the indicated type.

        The values of tmnxNatNotifyMdaChassisIndex, tmnxNatNotifyMdaCardSlotNum
        and tmnxNatNotifyMdaSlotNum identify the ISA MDA where the blocks were
        processed.

        All notifications of this type are sequentially numbered with
        the tmnxNatNotifyPlSeqNum.

        This type of notification is typically the consequence of one or
        more configuration changes; the nature of these changes is indicated
        in the tmnxNatNotifyDescription.
```

```
tmnxNatFwd2EntryAdded
        [CAUSE] The tmnxNatFwd2EntryAdded notification is sent when
         a row is added to or removed from the tmnxNatFwd2Table by other means
         than operations on the tmnxNatFwdAction;
         a conceptual row can be added to or removed from the table by operations on
         the tmnxNatFwdAction
         object group or otherwise, by means of the PCP protocol
         or automatically by the system, for example when a subscriber profile is
         changed.
         When the row is added, the value of the object
         tmnxNatNotifyTruthValue is 'true'; when the row is removed,
         it is 'false'.

         [EFFECT] The specified NAT subscriber can start receiving inbound
         traffic flows.
         [RECOVERY] No recovery required; this notification is the result
         of an operator or protocol action.

tmnxNatDetPlcyOperStateChanged
        [CAUSE] The tmnxNatDetPlcyOperStateChanged notification is sent when
         the value of the object tmnxNatDetPlcyOperState changes. The cause is
         explained in the tmnxNatNotifyDescription.
         tmnxNatDetMapOperStateChanged
         [CAUSE] The tmnxNatDetMapOperStateChanged notification is sent when
         the value of the object tmnxNatDetMapOperState changes. The cause is
         explained in the tmnxNatNotifyDescription.

tmnxNatFwd2OperStateChanged
      [CAUSE] The tmnxNatFwd2OperStateChanged notification is sent when
      the value of the object tmnxNatFwd2OperState changes. This
      is related to the state of the ISA MDA where the forwarding entry
      is located, or the availability of resources on that MDA.

      In the case of Layer-2-Aware NAT subscribers, the tmnxNatFwd2OperState
      is 'down' while the subscriber is not instantiated. This would typically
      be a transient situation.

      [EFFECT] The corresponding inward bound packets are dropped while the
      operational status is 'down'.

      [RECOVERY] If the ISA MDA reboots successfully, or another ISA MDA takes over,
      no recovery is required. If more resources become available on the ISA MDA, no
         recovery is required.
```

4.10.1.2 NAT logging to a local file

The following example displays NAT logging to a local file instead of memory.

Example: Enable NAT logging to a local file (MD-CLI)

```
[ex:/configure log]
A:admin@node-2# info
...
file "5" {
        description "nat logging"
        rollover 15
        retention 12
        compact-flash-location {
            primary cf1
        }
```

```
    }
    log-id "5" {
        filter "1"
        source {
            main true
        }
        destination {
            file "5"
        }
    }
```

Example: Enable NAT logging to a local file (classic CLI)

```
A:node-2>config>log# info
----------------------------------------------
        file-id 5
            description "nat logging"
            location cf1:
            rollover 15 retention 12
        exit

        log-id 5
            filter 1
            from main
            to file 5
        exit
```

The events are logged to a local file on the Compact Flash (CF) cf1 in a file under the /log directory.

Note: Logging to the CF represents a single point of failure. Performance (logs per second) of logging onto the CF is limited in comparison to other logging methods (RADIUS, Syslog, and IPFIX). Failure to generate logs because of a failed CF or performance limitation results in dropped NAT traffic. For this reason, local NAT logging in the SR OS is recommended only in a lab environment.

4.10.2 SNMP trap logging

In case of SNMP logging to a remote node, set the log destination to the SNMP destination. Allocation or de-allocation of each port block triggers sending a SNMP trap message to the trap destination.

Example: Configure SNMP trap logging (MD-CLI)

```
[ex:/configure log]
A:admin@node-2# info
...
    filter "1" {
        default-action drop
        named-entry "1" {
            action forward
            match {
                application {
                    eq nat
                }
                event {
                    eq 2012
                }
            }
        }
    }
```

```
    log-id "6" {
        filter "1"
        source {
            main true
        }
        destination {
            snmp {
            }
        }
    }
    snmp-trap-group "6" {
        trap-target "nat" {
            address 192.168.1.10
            port 9001
            version snmpv2c
            notify-community "private"
        }
    }
```

Example: Configure SNMP trap logging (classic CLI)

```
A:node-2>config>log# info
----------------------------------------------
       filter 1
            default-action drop
            entry 1
                action forward
                match
                    application eq "nat"
                    number eq 2012
                exit
            exit
        exit

        snmp-trap-group 6
            trap-target "nat" address 192.168.1.10 port 9001 snmpv2c notify-community
 "private"
        exit
        log-id 6
            filter 1
            from main
            to snmp
        exit
```

The following figure shows an SNMP trap message.

Figure 12: SNMP trap message

```
⊞ Internet Protocol Version 4, Src: 1.1.1.1 (1.1.1.1), Dst: 114.0.1.10 (114.0.1.10)
⊟ User Datagram Protocol, Src Port: snmptrap (162), Dst Port: etlservicemgr (9001)
    Source port: snmptrap (162)
    Destination port: etlservicemgr (9001)
    Length: 358
  ⊞ Checksum: 0x0e2c [correct]
⊟ Simple Network Management Protocol
    version: v2c (1)
    community: private
  ⊟ data: snmpV2-trap (7)
    ⊟ snmpV2-trap
        request-id: 1
        error-status: noError (0)
        error-index: 0
      ⊟ variable-bindings: 14 items
        ⊞ 1.3.6.1.2.1.1.3.0: 19054240
        ⊞ 1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.4.1.6527.3.1.3.65.0.12 (iso.3.6.1.4.1.6527.3.1.3.65.0.12)
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.2.0:
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.4.0:
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.5.0: 50000001
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.8.0: 1894
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.9.0: 1898
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.10.0: 07dc070d00321b002b0000
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.13.0: 1
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.3.0:
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.6.0:
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.7.0: 1a000038
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.11.0:
        ⊞ 1.3.6.1.4.1.6527.3.1.2.65.2.17.0: 5
```

4.10.3 NAT syslog

The follow example displays NAT logs configured to be sent to a syslog remote facility. A separate syslog message is generated for every port-block allocation or de-allocation.

Example: Configure the sending of NAT logs to a syslog remote facility (MD-CLI)

```
[ex:/configure log]
A:admin@node-2# info
...
    filter "1" {
        default-action drop
        named-entry "1" {
            action forward
            match {
                application {
                    eq nat
                }
                event {
                    eq 2012
                }
            }
        }
    }
    log-id "7" {
        filter "1"
        source {
            main true
```

```
        }
        destination {
            syslog "7"
        }
    }
    syslog "7" {
        address 192.168.1.10
    }
```

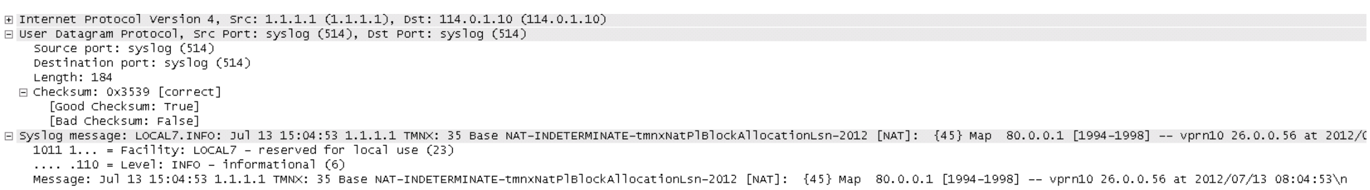
Example: Configure the sending of NAT logs to a syslog remote facility (classic CLI)

```
A:node-2>config>log# info
----------------------------------------------
...
        filter 1
            default-action drop
            entry 1 name "1"
                action forward
                match
                    application eq "nat"
                    number eq 2012
                exit
            exit
        exit
        syslog 7
            address 192.168.1.10
        exit


         log-id 7 name "7"
            filter 1
            from main
            to syslog 7
            no shutdown
        exit
----------------------------------------------
```

The following figure displays a syslog message.

Figure 13: Syslog message



The following example displays the change of configuration for a severity level for this event. Select from the following options:

- **cleared**
- **indeterminate**
- **critical**
- **major**
- **minor**
- **warning**

Example: Change the event severity level (MD-CLI)

```
*[ex:/configure]
A:admin@node-2# log log-events nat event * severity major
```

Example: Change the event severity level (classic CLI)

```
*A:node-2# configure log event-control "nat" 2012 generate major
```

4.10.4 Summarization logs and bulk operations

Bulk operations, such as removing a NAT policy or shutting down a NAT pool, can trigger a cascade of events, such as release of NAT subscribers associated with the NAT policy or a NAT pool. To avoid excessive logging during those operations, summarization logs are used. These logs carry relational information that connects multiple events and are categorized under event log 99 on the CPM. Configurable destinations for these logs include SNMP notification (trap), syslog (sent in syslog format to the syslog collector), memory (sent to memory buffer), local file, and NETCONF.

Tracking NAT subscribers based on the logs becomes more complicated if they were terminated because of bulk operations. A MAP log is generated when NAT resources for the NAT subscriber are allocated; a FREE log is generated when NAT resources for the NAT subscriber are released. Typically, individual MAP logs are paired with corresponding FREE logs to determine the identity and activity duration for the NAT subscriber. However, during bulk operations, individual FREE logs are substituted with a summarized log containing relational information. In such cases, identifying NAT subscriber mappings may necessitate examining multiple logging sources, such as a combination of RADIUS and summarization logs.

To simplify log summarization, a policy ID is added as a connecting option in all logs. The policy ID follows the format: `plcy-id` *XX*

Where: *XX* is a unique number representing a NAT policy and assigned by the router for each inside routing context, as shown in the following example.

```
670 2023/05/31 12:55:00.952 UTC MINOR: NAT #2012 vprn601 NAT
"{986} Map  10.10.10.1 [4001-4279] MDA 5/1 -- 1166016512 classic-lsn-sub
%203 vprn101 192.0.2.1 at 2023/05/31 12:55:00"
```

When an active NAT policy is removed from the configuration within an inside routing context, all NAT subscribers associated with that NAT policy in that context are removed from the system. Instead of generating individual FREE logs for each subscriber, a single summarized log is generated. This summarized log entry contains only the policy ID of the removed NAT policy and the inside service ID. To determine which NAT resources were released, the user must match the policy ID and the service ID in the summarization log with those in all MAP logs that lack a pairing explicit FREE log.

A summarization log is always created on the CPM, regardless of whether RADIUS logging is enabled.

A summarization log is generated on the CPM under the following circumstances:

- **NAT policy removal**

  If there is a single NAT policy for each inside routing context, the summarization log contains the inside service ID (VPRN or Base). To identify the terminated NAT mappings for subscribers, search all individual MAP logs matching the service ID from the summarization log.

When there are multiple NAT policies per inside routing context, the summarization log contains the inside service ID and policy ID. Search individual logs based on policy ID and inside service ID to identify subscribers affected by the NAT policy removal.

- **pool administratively disabled**

  The router sends a summarization log with the outside service ID and all IP address ranges in the pool. Match individual logs based on outside IP address and outside service ID to identify released NAT subscribers.

- **IP address range removal from the pool**

  The summarization log includes the outside service ID and the removed IP address range. Match individual logs based on the outside IP addresses in the range and the outside service ID to identify the released NAT subscribers.

- **Non deterministic source prefix removal**

  The summarization log includes the removed source prefix, policy ID, and inside service ID.

- **Last AFTR address removal**

  The summarization log includes the inside service ID.

Summarization logs are enabled by event controls 2021 (tmnxNatLsnSubBlksFree), 2016 (tmnxNatPlAddrFree), and 2030 (tmnxNatInAddrPrefixBlksFree). These events are suppressed by default. Event control 2021 also reports when all port blocks for a NAT subscriber are freed.

4.11 Histogram

The distribution of the following resources in a NAT pool is tracked in the form of a histogram:

- **Ports and NAT subscribers**

  The distribution of outside ports in a NAT pool is tracked for an aggregate number of NAT subscribers. The output of the following command can reveal the number of NAT subscribers in a pool that are heavy port users, or it can reveal the average number of ports used by most NAT subscribers.

  ```
  show router nat pool histogram
  ```

- **Port blocks and NAT subscribers in a NAT pool**

  The distribution of port blocks is tracked for an aggregate number of NAT subscribers. The output of the **histogram** command can reveal how NAT subscribers are using port blocks in the aggregate.

- **NAT subscribers and IP addresses**

  The distribution of NAT subscribers across IP addresses is tracked. The output of the **histogram** command is used to determine if any substantial imbalances exist.

- **Extended port blocks and outside IP addresses in a NAT pool**

  The distribution of extended port blocks in the NAT pool is tracked in relation to an aggregate number of outside IP addresses. The output of the **histogram** command can reveal how extended port blocks are distributed over IP addresses in an aggregate.

The user can use the displayed information to adjust the port block size per NAT subscriber, the amount of port blocks per NAT subscriber, or see port usage trends over time. Consequently, the user can adjust the configuration as the port demand per NAT subscriber increases or decreases over time. For example, a user may find that the port usage in a pool increased over a period of time. Accordingly, the user can plan to increase the number of ports per port block.

Execute the following show commands to display the **histogram** output.

Ports and NAT subscribers per NAT pool

Use the following command to show ports and subscribers per NAT pool . The output is organized in port buckets with the number of NAT subscribers in each bucket.

```
show router nat pool "pool-1" histogram ports bucket-size 200 num-buckets 10
```

Output example: Ports and NAT subscribers per NAT pool output

```
===============================================================================
Usage histogram NAT pool "pool-1" router "Base"
===============================================================================
Num-ports   Sub-TCP    Sub-UDP    Sub-ICMP
-------------------------------------------------------------------------------
1-199       17170      0          0
200-399     8707       0          0
400-599     2406       0          0
600-799     635        0          0
800-999     322        0          0
1000-1199   0          0          0
1200-1399   0          0          0
1400-1599   0          0          0
1600-1799   0          0          0
1800-       0          0          0
-------------------------------------------------------------------------------
No. of entries: 10
===============================================================================
```

Port blocks and NAT subscribers per NAT pool

Use the following command to show ports and NAT subscribers per NAT pool. The output is organized by the increasing number of port blocks in a NAT pool with the number of NAT subscribers using the number of port blocks indicated in each line.

```
show router nat pool "l2a" histogram port-blocks
```

Output example: Port blocks and NAT subscribers per NAT pool output

```
===============================================================================
Usage histogram NAT pool "l2a" router "Base" port blocks per subscriber
===============================================================================
Num port-blocks      Num subscribers
-------------------------------------------------------------------------------
1                    17398
2                    8550
3                    2352
4                    940
5                    0
6                    0
7                    0
8                    0
9                    0
10                   0
-------------------------------------------------------------------------------
```

```
No. of entries: 10
===============================================================================
```

NAT subscribers and IP addresses per NAT pool (LSN)

Use the following command to show NAT subscribers and IP addresses per NAT pool (LSN). The output is organized in buckets where each bucket shows how the NAT subscribers are spread over the preferred outside IP addresses. For example, the output of the below command shows that each of the 513 IP addresses in the pool have 120 to 129 NAT subscribers. This is a fairly even distribution of NAT subscribers over IP addresses and the favorable output of this command.

```
show router 5 nat pool "demo" histogram subscribers-per-ip bucket-size 10 num-buckets 50
```

Output example: NAT subscribers and IP addresses per NAT pool (LSN) output

```
===============================================================================
Usage histogram NAT pool "demo" router 5 subscribers per IP address
===============================================================================
Num subscribers     Num IP addresses
-------------------------------------------------------------------------------
1-9                 0
10-19               0
20-29               0
30-39               0
40-49               0
50-59               0
60-69               0
70-79               0
80-89               0
90-99               0
100-109             0
110-119             0
120-129             513
130-139             0
140-149             0
150-159             0
160-169             0
170-179             0
180-189             0
190-199             0
200-209             0
210-219             0
220-229             0
230-239             0
240-249             0
250-259             0
260-269             0
270-279             0
280-289             0
290-299             0
300-309             0
310-319             0
320-329             0
330-339             0
340-349             0
350-359             0
360-369             0
370-379             0
380-389             0
390-399             0
400-409             0
```

```
410-419             0
420-429             0
430-439             0
440-449             0
450-459             0
460-469             0
470-479             0
480-489             0
490-                0
```

Extended port blocks in a NAT pool and outside IP addresses

Use the following command to show NAT subscribers and IP addresses per NAT pool (LSN). The output is organized in extended port-block buckets in a NAT pool with the number of outside IP addresses in each bucket.

```
show router nat pool "l2a" histogram extended-port-blocks-per-ip bucket-size 1 num-buckets 10
```

Output example: Extended port blocks in a NAT pool and outside IP addresses output

```
===============================================================================
Usage histogram NAT pool "l2a" router "Base" extended port blocks per IP address
===============================================================================
Num extended-port-blocks      Num IP addresses
-------------------------------------------------------------------------------
-                             -
1-1                           1039
2-2                           6182
3-3                           777
4-4                           194
5-5                           0
6-6                           0
7-7                           0
8-8                           0
9-                            0
-------------------------------------------------------------------------------
No. of entries: 10
===============================================================================
```

The output of each command can be periodically exported to an external destination with the **cron** command.

The following example displays the script, script policy, and CRON configuration

Example: Configure the script, script policy, and CRON (MD-CLI)

```
[ex:/configure system]
A:admin@node-2# info
...
    cron {
        schedule "nat_histogram_schedule" owner "TiMOSCLI" {
            admin-state enable
            interval 600
            script-policy {
                name "dump_nat_histogram"
            }
        }
    }
...
    script-control {
        script "nat_histogram" owner "TiMOSCLI" {
```

```
            admin-state enable
            location "ftp://*:*@138.203.8.62/nat-histogram.txt"
        }
        script-policy "dump_nat_histogram" owner "TiMOSCLI" {
            admin-state enable
            results "ftp://*:*@138.203.8.62/nat_histogram_results.txt"
            script {
                name "nat_histogram"
            }
        }
    }
```

Example: Configure the script, script policy, and CRON (classic CLI)

```
A:node-2>config>system# info
----------------------------------------------
#--------------------------------------------------
echo "System Configuration"
#--------------------------------------------------
...
        script-control
            script "nat_histogram" owner "TiMOSCLI"
                no shutdown
                location "ftp://*:*@138.203.8.62/nat-histogram.text"
            exit
            script-policy "dump_nat_histogram" owner "TiMOSCLI"
                no shutdown
                results "ftp://*:*@130.203.8.62/nat_histogram_results.text"
                script "nat_histogram"
            exit
        exit
        cron
            schedule "nat_histogram_schedule" owner "TiMOSCLI"
                interval 600
                script-policy "dump_nat_histogram"
                no shutdown
            exit
        exit
#--------------------------------------------------
```

The `nat-histogram.txt` file contains the command execution line.

```
show router nat pool "pool-1" histogram ports bucket-size 200 num-buckets 10
```

This command is executed every 10 minutes (600 seconds) and the output of the command is written into a set of files on an external TFP server as displayed in the following example.

Output example: Files on an external TFP server

```
[root@ftp]# ls nat_histogram_results.txt*
    nat_histogram_results.txt_20130117-153548.out
    nat_histogram_results.txt_20130117-153648.out
    nat_histogram_results.txt_20130117-153748.out
    nat_histogram_results.txt_20130117-153848.out
    nat_histogram_results.txt_20130117-153948.out
    nat_histogram_results.txt_20130117-154048.out
    [root@ftp]#
```

4.12 TCP MSS adjustment

4.12.1 Overview

This feature adds support for adjustment of MSS of TCP packets with SYN flag according to access/aggregation network to prevent fragmentation of upstream and downstream TCP packets using ISA-BB.

There are two modes of adjustment operations supported: TCP MSS Adjustment filter on VPRN SAP interfaces and TCP MSS Adjustment for NAT Services.

4.12.2 TCP MSS adjustment filter on VPRN SAP interfaces

About this task

The 7705 SAR Gen 2 supports a configurable filter that adjusts the maximum segment size (MSS) of TCP packets marked with a SYN flag that traverse VPRN SAP interfaces. The MSS adjustment filter prevents upstream and downstream TCP packets from being fragmented.

MSS adjustment is performed by the virtualized integrated BB ISA MDA when an IP filter is enabled with the **action tcp-mss-adjust** command. The filter can be applied on a VPRN SAP interface in the ingress direction, egress direction, or both directions. Both IPv4 and IPv6 filters are supported. For information about the virtualized BB ISA MDA, see the *7705 SAR Gen 2 Interface Configuration Guide*, "Chassis IOM and MDAs".

Perform the following steps to configure a TCP MSS adjustment filter on a VPRN SAP interface:

Procedure

Step 1. Create a NAT group that will be used for MSS adjustment.

The following output is an example of the creation of a NAT group on the virtualized integrated BB ISA MDA in slot 1/6.

```
config
   card 1
        mda 6
            mda-type isa-bb-v
            no shutdown
        exit
        no shutdown
    exit
```

```
configure
    isa
        nat-group 1 create
            active-mda-limit 1
            mda 1/6
            no shutdown
        exit
```

Step 2. Associate the NAT group with a routing instance and configure the MSS value as shown in the following example.

```
config
```

```
    service
        vprn services-id
            mss-adjust-group 1 segment-size 1352
```

Step 3. Create ingress or egress IP filters that perform TCP MSS adjustment.

The following example shows the configuration of IPv4 filters and IPv6 filters that perform TCP MSS adjustment at ingress and egress.

```
configure
    filter
        ip-filter 1 name "1" create
            default-action forward
            description "Ingress"
            entry 1 create
                match protocol tcp
                    tcp-syn true
                exit
                action
                    tcp-mss-adjust
                exit
            exit
        exit
        ip-filter 2 name "2" create
            default-action forward
            description "Egress"
            entry 1 create
                match protocol tcp
                    tcp-syn true
                exit
                action
                    tcp-mss-adjust
                exit
                egress-pbr default-load-balancing
            exit
        exit
        ipv6-filter 1 name "3" create
            default-action forward
            description "Ingress"
            entry 1 create
                match next-header tcp
                    tcp-syn true
                exit
                action
                    tcp-mss-adjust
                exit
            exit
        exit
        ipv6-filter 2 name "4" create
            default-action forward
            description "Egress"
            entry 1 create
                match next-header tcp
                    tcp-syn true
                exit
                action
                    tcp-mss-adjust
                exit
                egress-pbr default-load-balancing
            exit
        exit
    exit
```

Step 4. Apply the filters that perform TCP MSS adjustment to the VPRN SAP interface. The filters can be applied in the ingress direction, egress direction, or both directions. In the following example, the filters are applied in both the ingress and egress directions.

```
config
    service
        vprn service-id
            interface "int1_vprn1" create
                address 10.10.1.1/24
                sap 1/2/3 create
                    ingress
                        filter ip 1
                    exit
                    egress
                        filter ip 2
                    exit
                exit
            exit
        exit
        vprn service-id2
            interface "int1_vprn2" create
                ipv6
                    address 10:1::1/32
                    neighbor 10:1::2 00:02:01:00:00:01
                exit
                sap 1/2/3:1 create
                    ingress
                        filter ipv6 3
                    exit
                    egress
                        filter ipv6 4
                    exit
                exit
            exit
        exit
```

4.12.3 TCP MSS adjustment for NAT services

About this task

This feature provides MSS adjustment for TCP packets to be translated by NAT services.

Procedure

Step 1. Create a NAT group used for NAT services with MSS adjustment.

Example

MD-CLI

```
[ex:/configure isa]
A:admin@node-2# info nat-group 1 {
        redundancy {
            active-mda-limit 2
        }
        mda 1/2 { }
        mda 1/2 { }
    }
```

Example

classic CLI

```
A:node-2>config>isa# info
----------------------------------------------
        nat-group 1 create
            shutdown
            active-mda-limit 1
            mda 1/1
            mda 1/2
        exit
----------------------------------------------
```

Step 2. Create a NAT policy that also adjusts MSS.

Example

MD-CLI

```
[ex:/configure service nat]
A:admin@node-2# info
...
    nat-policy "policy-for-mss-adjust" {
        tcp {
            mss-adjust 1452
        }
    }
```

Example

classic CLI

```
A:node-2>config>service>nat# info
----------------------------------------------
            nat-policy "policy-for-mss-adjust" create
                tcp-mss-adjust 1452
            exit
```

4.13 Configuring NAT

This section provides information to configure NAT using the command line interface.

4.13.1 Large scale NAT configuration

The following example displays a Large Scale NAT configuration.

Example: MD-CLI

```
[ex:/configure]
A:admin@node-2# admin show configuration
configure {
...
    card 3 {
        card-type imm-2pac-fp3
        mda 1 {
            mda-type isa2-bb
```

```
        }
        mda 2 {
            mda-type isa2-bb
        }
    }
    filter {
        ip-filter "123" {
            entry 10 {
                match {
                    src-ip {
                        address 10.0.0.1/8
                    }
                }
                action {
                    nat {
                    }
                }
            }
        }
    }
    isa {
        nat-group 1 {
            admin-state enable
            redundancy {
                active-mda-limit 2
            }
            mda 3/1 { }
            mda 3/2 { }
        }
    }
    service {
        nat {
            nat-policy "ls-outPolicy" {
                pool {
                    router-instance "500"
                    name "nat1-pool"
                }
                timeouts {
                    udp {
                        normal 18000
                        initial 240
                    }
                }
            }
        }
        vprn "500" {
            admin-state enable
            customer "1"
            router-id 10.21.1.2
            nat {
                outside {
                    pool "nat1-pool" {
                        admin-state enable
                        type large-scale
                        nat-group 1
                        port-reservation {
                            ports 200
                        }
                        address-range 10.81.0.0 end 10.81.6.0 {
                        }
                    }
                }
            }
            bgp-ipvpn {
```

```
                mpls {
                    admin-state enable
                    route-distinguisher "500:10"
                    vrf-target {
                        import-community "target:500:1"
                        export-community "target:500:1"
                    }
                }
            }
            interface "ip-192.168.113.1" {
                ipv4 {
                    primary {
                        address 192.168.113.1
                        prefix-length 24
                    }
                    neighbor-discovery {
                        static-neighbor 192.168.113.5 {
                            mac-address 00:00:5e:00:53:00
                        }
                    }
                }
                sap 1/1/1:200 {
                }
            }
        }
        vprn "550" {
            customer "1"
            router-id 10.21.1.2
             nat {
                inside {
                    large-scale {
                        nat-policy "ls-outPolicy"
                    }
                }
            }
            bgp-ipvpn {
                mpls {
                    admin-state enable
                    route-distinguisher "550:10"
                    vrf-target {
                        import-community "target:550:1"
                        export-community "target:550:1"
                    }
                }
            }
            interface "ip-192.168.13.1" {
                ipv4 {
                    primary {
                        address 192.168.13.1
                        prefix-length 8
                    }
                }
                sap 1/2/1:900 {
                    ingress {
                        filter {
                            ip "123"
                        }
                    }
                }
            }
        }
    }
...
```

Example: classic CLI

```
A:node-2# admin display-config
configure
#--------------------------------------------------
echo "Card Configuration"
#--------------------------------------------------
    card 3
        card-type imm-2pac-fp3
        mda 1
            mda-type isa2-bb
        exit
        mda 2
            mda-type isa2-bb
        exit
    exit
#--------------------------------------------------
echo "ISA Configuration"
#--------------------------------------------------
    isa
        nat-group 1 create
            active-mda-limit 2
            mda 3/1
            mda 3/2
            no shutdown
        exit
    exit
#--------------------------------------------------
echo "Filter Configuration"
#--------------------------------------------------
    filter
        ip-filter 123 create
            entry 10 create
                match
                    src-ip 10.0.0.1/8
                exit
                action nat
            exit
        exit
    exit
#--------------------------------------------------
echo "NAT (Declarations) Configuration"
#--------------------------------------------------
    service
        nat
            nat-policy "ls-outPolicy" create
            exit
        exit
    exit
#--------------------------------------------------
echo "Service Configuration"
#--------------------------------------------------
    service
        customer 1 create
            description "Default customer"
        exit
        vprn 500 customer 1 create
            interface "ip-192.168.113.1" create
            exit
            nat
                outside
                    pool "nat1-pool" nat-group 1 type large-scale create
                        port-reservation ports 200
                        address-range 10.81.0.0 10.81.6.0 create
```

```
                        exit
                        no shutdown
                    exit
                exit
            exit
        exit
        vprn 550 customer 1 create
            interface "ip-192.168.13.1" create
            exit
        exit
        nat
            nat-policy "ls-outPolicy" create
                pool "nat1-pool" router 500
                timeouts
                    udp hrs 5
                    udp-initial min 4
                exit
            exit
        exit
        vprn 500 customer 1 create
            router-id 10.21.1.2
            route-distinguisher 500:10
            vrf-target export target:500:1 import target:500:1
            interface "ip-192.168.113.1" create
                address 192.168.113.1/24
                static-arp 192.168.113.5 00-00-5e-00-53-00
                sap 1/1/1:200 create
                exit
            exit
            no shutdown
        exit
        vprn 550 customer 1 create
            router-id 10.21.1.2
            route-distinguisher 550:10
            vrf-target export target:550:1 import target:550:1
            interface "ip-192.168.13.1" create
                address 192.168.13.1/8
                sap 1/2/1:900 create
                    ingress
                        filter ip 123
                    exit
                exit
            exit
            nat
                inside
                    nat-policy "ls-outPolicy"
                exit
            exit
            no shutdown
        exit
    exit
exit all
```

4.13.2 NAT configuration examples

The following examples display configuration information for a VPRN service, router NAT, and a NAT service.

Example: Configure the VPRN service (MD-CLI)

```
[ex:/configure service vprn "100" nat]
```

```
A:admin@node-2# info
    inside {
        l2-aware {
            force-unique-ip-addresses false
        }
        large-scale {
            nat-policy "priv-nat-policy"
            traffic-identification {
                source-prefix-only false
            }
            nat44 {
                destination-prefix 0.0.0.0/0 {
                }
            }
            dual-stack-lite {
                admin-state enable
                subscriber-prefix-length 128
                endpoint 2001:db8:470:fff:190:1:1:1 {
                    tunnel-mtu 1500
                    reassembly false
                    min-first-fragment-size-rx 1280
                }
                endpoint 2001:db8:470:1f00:ffff:190:1:1 {
                    tunnel-mtu 1500
                    reassembly false
                    min-first-fragment-size-rx 1280
                }
            }
            subscriber-identification {
                admin-state disable
                drop-unidentified-traffic false
                attribute {
                    vendor nokia
                    type alc-sub-string
                }
            }
        }
    }
```

Example: Configure the VPRN service (classic CLI)

```
A:node-2>config>service# info detail
----------------------------------------------
        vprn 100 name "100" customer 1 create
            shutdown
            nat
                inside
                    nat-policy "priv-nat-policy"
                    destination-prefix 0.0.0.0/0
                    dual-stack-lite
                        subscriber-prefix-length 128
                        address 2001:db8:470:1f00:ffff:190:1:1
                            tunnel-mtu 1500
                        exit
                        no shutdown
                    exit
                    redundancy
                        no peer
                        no steering-route
                    exit
                    subscriber-identification
                        shutdown
                        no attribute
```

```
                        no description
                        no radius-proxy-server
                    exit
                    l2-aware
                    exit
                exit
                outside
                    no mtu
                exit
```

Example: Configure a router NAT (MD-CLI)

```
[ex:/configure router "Base" nat outside]
A:admin@node-2# info
    pool "privpool" {
        admin-state enable
        type large-scale
        nat-group 3
        address-pooling paired
        icmp-echo-reply false
        mode auto
        applications {
            agnostic false
            flexible-port-allocation false
        }
        port-forwarding {
            dynamic-block-reservation false
            range-start 1
            range-end 1023
        }
        port-reservation {
            port-blocks 128
        }
        large-scale {
            subscriber-limit 65535
            redundancy {
                admin-state disable
            }
        }
        address-range 10.0.0.5 end 10.0.0.6 {
            drain false
        }
    }
    pool "pubpool" {
        admin-state enable
        type large-scale
        nat-group 1
        address-pooling paired
        icmp-echo-reply false
        mode auto
        applications {
            agnostic false
            flexible-port-allocation false
        }
        port-forwarding {
            dynamic-block-reservation false
            range-start 1
            range-end 1023
        }
        port-reservation {
            port-blocks 1
        }
        large-scale {
```

```
            subscriber-limit 65535
            redundancy {
                admin-state disable
            }
        }
        address-range 192.168.8.241 end 192.168.8.247 {
            drain false
        }
    }
    pool "test" {
        type large-scale
        nat-group 1
    }
```

Example: Configure a router NAT (classic CLI)

```
A:node-2>config>router>nat# info detail
----------------------------------------------
            outside
                no mtu
                pool "privpool" nat-group 3 type large-scale create
                    no description
                    port-reservation blocks 128
                    port-forwarding-range 1023
                    redundancy
                        no export
                        no monitor
                    exit
                    subscriber-limit 65535
                    no watermarks
                    mode auto
                    address-range 10.0.0.5 10.0.0.6 create
                        no description
                        no drain
                    exit
                    no shutdown
                exit
                pool "pubpool" nat-group 1 type large-scale create
                    no description
                    port-reservation blocks 1
                    port-forwarding-range 1023
                    redundancy
                        no export
                        no monitor
                    exit
                    subscriber-limit 65535
                    no watermarks
                    mode auto
                    address-range 192.168.8.241 192.168.8.247 create
                        no description
                        no drain
                    exit
                    no shutdown
                exit
            exit
```

Example: Configure a service NAT (MD-CLI)

```
[ex:/configure service nat nat-policy "priv-nat-policy"]
A:admin@node-2# info
    block-limit 4
    filtering endpoint-independent
```

```
    port-forwarding-range-end 1023
    pool {
        router-instance "Base"
        name "privpool"
    }
    alg {
        ftp true
        pptp false
        rtsp true
        sip true
    }
    port-limits {
        forwarding 64
        dynamic-ports 65536
    }
    priority-sessions {
        fc {
            be false
            l2 false
            af false
            l1 false
            h2 false
            ef false
            h1 false
            nc false
        }
    }
    session-limits {
        max 65535
    }
    tcp {
        reset-unknown false
    }
    timeouts {
        icmp-query 60
        sip 120
        subscriber-retention 0
        tcp {
            established 7440
            rst 0
            syn 15
            time-wait 0
            transitory 240
        }
        udp {
            normal 300
            dns 15
            initial 15
        }
    }
    udp {
        inbound-refresh false
    }

[ex:/configure service nat nat-policy "pub-nat-policy"]
A:admin@node-2# info
    block-limit 1
    filtering endpoint-independent
    port-forwarding-range-end 1023
    pool {
        router-instance "Base"
        name "pubpool"
    }
    alg {
```

```
        ftp true
        pptp false
        rtsp false
        sip false
    }
    port-limits {
        dynamic-ports 65536
    }
    priority-sessions {
        fc {
            be false
            l2 false
            af false
            l1 false
            h2 false
            ef false
            h1 false
            nc false
        }
    }
    session-limits {
        max 65535
    }
    tcp {
        reset-unknown false
    }
    timeouts {
        icmp-query 60
        sip 120
        subscriber-retention 0
        tcp {
            established 7440
            rst 0
            syn 15
            time-wait 0
            transitory 240
        }
        udp {
            normal 300
            dns 15
            initial 15
        }
    }
    udp {
        inbound-refresh false
    }
```

Example: Configure a service NAT (classic CLI)

```
A:node-2>config>service>nat# info detail
----------------------------------------------
            nat-policy "priv-nat-policy" create
                alg
                    ftp
                    rtsp
                    sip
                exit
                block-limit 4
                no destination-nat
                no description
                filtering endpoint-independent
                pool "privpool" router Base
                no ipfix-export-policy
```

```
                port-limits
                    forwarding 64
                    no reserved
                    no watermarks
                exit
                priority-sessions
                exit
                session-limits
                    max 65535
                    no reserved
                    no watermarks
                exit
                timeouts
                    icmp-query min 1
                    sip min 2
                    no subscriber-retention
                    tcp-established hrs 2 min 4
                    tcp-syn sec 15
                    no tcp-time-wait
                    tcp-transitory min 4
                    udp min 5
                    udp-initial sec 15
                    udp-dns sec 15
                exit
                no tcp-mss-adjust
                no udp-inbound-refresh
            exit
            nat-policy "pub-nat-policy" create
                alg
                    ftp
                    no rtsp
                    no sip
                exit
                block-limit 1
                no destination-nat
                no description
                filtering endpoint-independent
                pool "pubpool" router Base
                no ipfix-export-policy
                port-limits
                    no forwarding
                    no reserved
                    no watermarks
                exit
                priority-sessions
                exit
                session-limits
                    max 65535
                    no reserved
                    no watermarks
                exit
                timeouts
                    icmp-query min 1
                    sip min 2
                    no subscriber-retention
                    tcp-established hrs 2 min 4
                    tcp-syn sec 15
                    no tcp-time-wait
                    tcp-transitory min 4
                    udp min 5
                    udp-initial sec 15
                    udp-dns sec 15
                exit
                no tcp-mss-adjust
```

```
                    no udp-inbound-refresh
                exit
```

4.14 Expanding a NAT group

Adding or removing an MDA from a NAT group affects all currently active subscribers and may invalidate existing static port forwards and mappings configured in deterministic NAT.

Store configurations offline before removing the configuration as part of the NAT group modification process that is described in the following information. You can restore the configuration to the node after the change is complete.

The procedure to add or remove an MDA from a NAT group is described in the following information.

Adding and removing an MDA from a NAT group in the MD-CLI

In the MD-CLI, use the following steps to add or remove an MDA from a NAT group in the MD-CLI:

1. Administratively disable deterministic prefix policies and delete their mappings. Perform this for every deterministic prefix and their mapping used in a NAT group in which the size is modified. Store the deterministic mapping configuration offline before removing it and then reapply after the change. When the NAT group size is modified and the deterministic mappings reapplied, the commit may fail. If the commit fails, you must create a new mapping. Use the following command to create a new mapping.

   ```
   tools perform nat deterministic calculate-maps
   ```

   Static port forwards configurations created with the **tools** command are automatically deleted during the commitment of the modified NAT group.
2. Commit the changes.
3. Change the active and failed MDA limit.
4. Commit the changes.
5. Re-apply deterministic mappings and static port forwards.

Adding and removing an MDA from a NAT group in the classic CLI

In the classic CLI, use the following steps to add or remove an MDA from a NAT group:

1. Shut down the NAT group.
2. Remove all statically configured large-scale subscribers (such as deterministic, LI, debug, and subscriber aware) in a NAT group that is being modified.
3. A static port forward configuration created via the **tools** command is automatically deleted.
4. Manually delete any static port forward configurations that were created..
5. Shut down and remove the deterministic policies.
6. Delete NAT policy references in all inside routing contexts associated with the NAT group that is being modified.
7. Reconfigure the **active-mda-limit** and **failed-mda-limit** options.
8. Administratively enable the NAT group.

9. Restore previously removed NAT group references in all of the inside routing contexts associated with the modified NAT group.
10. Reapply the subscriber-aware and deterministic subscribers (prefixes and maps), static port forwards, LI, and debug.

5 Standards and protocol support

Note:
The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

5.1 Bidirectional Forwarding Detection (BFD)

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

5.2 Border Gateway Protocol (BGP)

draft-hares-idr-update-attrib-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*

RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP* – helper mode

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* – asplain

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 9494, *Long-Lived Graceful Restart for BGP*

5.3 Bridging and management

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1AX, *Link Aggregation*

IEEE 802.1Q, *Virtual LANs*

5.4 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*

RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*

RFC 7030, *Enrollment over Secure Transport*

RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

5.5 Ethernet VPN (EVPN)

draft-ietf-bess-bgp-srv6-args-00, *SRv6 Argument Signaling for BGP Services*

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*

draft-sr-bess-evpn-vpws-gateway-03, *Ethernet VPN Virtual Private Wire Services Gateway Solution*

RFC 7432, *BGP MPLS-Based Ethernet VPN*

RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*

RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*

RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*

RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*

RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)*

RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*

RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*

RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

5.6 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) Certificate Management Service*

file.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) File Service*

gnmi.proto version 0.8.0, *gRPC Network Management Interface (gNMI) Service Specification*

gnmi_ext.proto version 0.1.0, *gNMI Commit Confirmed Extension*

system.proto version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*

PROTOCOL-HTTP2, *gRPC over HTTP2*

5.7 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-ietf-lsr-igp-ureach-prefix-announce-01, *IGP Unreachable Prefix Announcement* – without U-Flag and UP-Flag

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*

RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*

RFC 2973, *IS-IS Mesh Groups*

RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*

RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*

RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*

RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*

RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*

RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*

RFC 5304, *IS-IS Cryptographic Authentication*

RFC 5305, *IS-IS Extensions for Traffic Engineering TE*

RFC 5306, *Restart Signaling for IS-IS* – helper mode

RFC 5308, *Routing IPv6 with IS-IS*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5310, *IS-IS Generic Cryptographic Authentication*

RFC 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*

RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*

RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3

RFC 7981, *IS-IS Extensions for Advertising Router Information*

RFC 7987, *IS-IS Minimum Remaining Lifetime*

RFC 8202, *IS-IS Multi-Instance* – single topology

RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 8919, *IS-IS Application-Specific Link Attributes*

5.8 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*

RFC 768, *User Datagram Protocol*

RFC 793, *Transmission Control Protocol*

RFC 854, *Telnet Protocol Specifications*

RFC 1350, *The TFTP Protocol (revision 2)*

RFC 2784, *Generic Routing Encapsulation (GRE)*

RFC 3164, *The BSD syslog Protocol*

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

5.9 Internet Protocol (IP) multicast

RFC 1112, *Host Extensions for IP Multicasting*

RFC 2236, *Internet Group Management Protocol, Version 2*

RFC 2365, *Administratively Scoped IP Multicast*

RFC 2375, *IPv6 Multicast Address Assignments*

RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*

RFC 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*

RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*

RFC 3618, *Multicast Source Discovery Protocol (MSDP)*

RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*

RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*

RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*

RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*

RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*

RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

5.10 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*

RFC 792, *Internet Control Message Protocol*

RFC 826, *An Ethernet Address Resolution Protocol*

RFC 1034, *Domain Names - Concepts and Facilities*

RFC 1035, *Domain Names - Implementation and Specification*

RFC 1191, *Path MTU Discovery* – router specification

RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*

RFC 1812, *Requirements for IPv4 Routers*

RFC 1918, *Address Allocation for Private Internets*

RFC 2131, *Dynamic Host Configuration Protocol*; Relay only

RFC 2132, *DHCP Options and BOOTP Vendor Extensions* – DHCP

RFC 2401, *Security Architecture for Internet Protocol*

RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*

RFC 3046, *DHCP Relay Agent Information Option (Option 82)*

RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*

RFC 4884, *Extended ICMP to Support Multi-Part Messages* – ICMPv4 and ICMPv6 Time Exceeded

5.11 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*

RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*

RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*

RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3587, *IPv6 Global Unicast Address Format*

RFC 3596, *DNS Extensions to Support IP version 6*

RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*

RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*

RFC 3971, *SEcure Neighbor Discovery (SEND)*

RFC 4007, *IPv6 Scoped Address Architecture*

RFC 4191, *Default Router Preferences and More-Specific Routes* – Default Router Preference

RFC 4193, *Unique Local IPv6 Unicast Addresses*

RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*

RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*

RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*

RFC 5722, *Handling of Overlapping IPv6 Fragments*

RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6* – IPv6

RFC 5952, *A Recommendation for IPv6 Address Text Representation*

RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*

RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*

RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

5.12 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*

draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*

RFC 2401, *Security Architecture for the Internet Protocol*

RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*

RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*

RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*

RFC 2406, *IP Encapsulating Security Payload (ESP)*

RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*

RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*

RFC 2409, *The Internet Key Exchange (IKE)*

RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*

RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*

RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*

RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*

RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*

RFC 3947, *Negotiation of NAT-Traversal in the IKE*

RFC 3948, *UDP Encapsulation of IPsec ESP Packets*

RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*

RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*

RFC 4301, *Security Architecture for the Internet Protocol*

RFC 4303, *IP Encapsulating Security Payload*

RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*

RFC 4308, *Cryptographic Suites for IPsec*

RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*

RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*

RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*

RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*

RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*

RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*

RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*

RFC 5903, *ECP Groups for IKE and IKEv2*

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*

RFC 6379, *Suite B Cryptographic Suites for IPsec*

RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*

RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*

RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*

RFC 8784, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*

5.13 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*

draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*

RFC 3037, *LDP Applicability*

RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol* – helper mode

RFC 5036, *LDP Specification*

RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*

RFC 5443, *LDP IGP Synchronization*

RFC 5561, *LDP Capabilities*

RFC 5919, *Signaling LDP Label Advertisement Completion*

5.14 Multiprotocol Label Switching (MPLS)

RFC 3031, *Multiprotocol Label Switching Architecture*

RFC 3032, *MPLS Label Stack Encoding*

RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services* – E-LSP

RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*

RFC 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

5.15 Network Address Translation (NAT)

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

5.16 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart* – helper mode

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8920, *OSPF Application-Specific Link Attributes*

5.17 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks.* – MPLS binding SIDs

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

5.18 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*

MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*

MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*

MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*

RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*

RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*

RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*

RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*

RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*

RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*

RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*

RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*

RFC 6073, *Segmented Pseudowire*

RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*

RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*

RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*

RFC 6718, *Pseudowire Redundancy*

RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*

RFC 6870, *Pseudowire Preferential Forwarding Status bit*

RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*

RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*

RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires* – ER-TLV and ER-HOP IPv4 Prefix

RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

5.19 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*

RFC 2597, *Assured Forwarding PHB Group*

RFC 3140, *Per Hop Behavior Identification Codes*

RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

5.20 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*

RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*

RFC 2866, *RADIUS Accounting*

RFC 3162, *RADIUS and IPv6*

RFC 6613, *RADIUS over TCP* – with TLS

RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*

RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*

5.21 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

RFC 2702, *Requirements for Traffic Engineering over MPLS*

RFC 2747, *RSVP Cryptographic Authentication*

RFC 2961, *RSVP Refresh Overhead Reduction Extensions*

RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*

RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*

RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*

RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*

RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

5.22 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*

RFC 2080, *RIPng for IPv6*

RFC 2082, *RIP-2 MD5 Authentication*

RFC 2453, *RIP Version 2*

5.23 Segment Routing (SR)

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF* – node MSD

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS* – node MSD

RFC 8660, *Segment Routing with the MPLS Data Plane*

RFC 8661, *Segment Routing MPLS Interworking with LDP*

RFC 8665, *OSPF Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 9256, *Segment Routing Policy Architecture*

RFC 9350, *IGP Flexible Algorithm*

5.24 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model* – CFB128-AES-192 and CFB128-AES-256

draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*

draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)* draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*

draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*

draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*

draft-ietf-vrrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6* – IPv6

ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*

IANAifType-MIB revision 200505270000Z, *ianaifType*

IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*

IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*

IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*

IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*

RFC 1157, *A Simple Network Management Protocol (SNMP)*

RFC 1212, *Concise MIB Definitions*

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3430, *Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)* – SNMP over UDP over IPv4

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

5.25 Timing

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

RFC 8573, *Message Authentication Code for the Network Time Protocol*

5.26 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP)* – server, unauthenticated mode

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)* – TWAMP

RFC 8762, *Simple Two-Way Active Measurement Protocol* – unauthenticated

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions* – unauthenticated

RFC 9503, *Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Segment Routing Networks* – excluding Sections 3, 4.1.2 and 4.1.3

5.27 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

5.28 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

Customer document and product support

Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback