

SteelConnect EX SD-WAN Troubleshooting Guide

Version 16.1R2

December 2019

Table of Contents

1	About SteelConnect EX SD-WAN Troubleshooting	7
1.1	Introduction	7
1.2	Purpose	7
1.3	Audience	7
1.4	Document Conventions	7
1.5	Related Documentation	7
1.6	Technical Support	8
2	SteelConnect EX SD-WAN Stages	9
2.1	SteelConnect EX SD-WAN Stages	9
3	Troubleshooting IKE/IPSec	10
3.1	IKE Session History	10
3.2	IKE Security Association	10
3.3	IPSec Tunnels	11
3.3.1	Overall IPSEC Statistics	11
3.4	Troubleshooting IPSec Stage 1 and Stage 2	12
3.5	Troubleshooting IPSec Stage 3- Branch to Controller	14
3.6	Troubleshooting IPSec Stage 3- Branch to Branch	15
3.7	Certificate Based Authentication	16
3.7.1	Certificate Information	16
3.7.2	Certificate Management Protocol	18
3.7.3	Certificate Management Protocol	18
4	Troubleshooting Routing Protocols	20
4.1	Troubleshooting Routing Protocols	20
4.1.1	Routing Protocol Command Examples	20
4.1.1.1	CLI Command to view Routing Daemon Routing Table (RIB)	21
4.1.1.2	CLI Command to view Routing Daemon Routing Table (RIB) for a routing-instance	22
4.1.1.3	CLI Command to view Routes Received by BGP	22
4.1.1.4	CLI Command to view Routes Received by BGP from one Particular Neighbor	23
4.1.1.5	CLI Command to view VPN IPv4 (L3 VPN) Routes Received by BGP from one Particular Neighbor	24
4.1.1.6	CLI Command to view Routes Advertised by BGP to one Particular Neighbor	25

4.1.1.7 CLI Command to view VPN IPv4 (L3 VPN) Routes sent by BGP to one Particular Neighbor	26
4.2 Troubleshooting BGP Commands	26
4.2.1 BGP Command Examples.....	27
4.2.1.1 CLI Command to view BGP Summary Information for all Routing-Instances	27
4.2.1.2 CLI Command to view BGP Summary Information for a Specific Routing-Instance	28
4.2.1.3 CLI Command to view Information for BGP Peer-Groups of all Routing-Instances	28
4.2.1.4 CLI Command to view Information for BGP Peer-Groups of a Specific Routing-Instance	29
4.2.1.5 CLI Command to view all BGP Neighbors of all Routing-Instances.....	30
4.2.1.6 CLI Command to view all BGP Neighbors of a Specific Routing-Instance	31
4.2.1.7 CLI Command to view a Specific BGP Neighbor of a Specific Routing-Instance	32
4.3 Troubleshooting OSPF Commands	32
4.3.1 OSPF Command Examples	32
4.3.1.1 CLI Command to view OSPF Interfaces	33
4.3.1.2 CLI Command to view OSPF Neighbors.....	34
4.3.1.3 CLI Command to view OSPF Protocol Overview.....	35
4.3.1.4 CLI Command to view OSPF LSA Database.....	36
4.4 Inspecting Routing State in vstated Daemon.....	38
4.4.1 Routing State Command Examples.....	38
4.4.1.1 Example 1.....	39
4.4.1.2 CLI Command to list all Routes of all Routing-Instances	39
4.4.1.3 CLI Command to List all Routes of a Specific Routing-Instance.....	40
4.4.1.4 CLI Command to List a Specific Route of all Routing-Instances.....	40
4.4.1.5 CLI Command to List a Specific Route of a Specific Routing-Instance	40
4.4.2 Inspecting Routing State in vsmd Daemon	41
4.4.2.1 Routing State Command Examples.....	41
5 Troubleshooting SD-WAN Datapath.....	43
5.1 Inspecting vsm Control Plane State	43
5.2 Inspecting vsm Data Plane State	48
5.3 Inspecting vsm Data Plane Statistics	52
6 Troubleshooting SD-WAN Branches.....	61
6.1 Branch Staging.....	61
6.1.1 Controller Configuration	61
6.1.2 Controller Show Commands	61
6.1.3 Branch Life-cycle.....	61

6.2	Stage Three Debugging on SD-WAN Appliance	62
6.2.1	Low-level vty Commands	63
6.3	Stage-3 Debugging on Controller	65
6.3.1	High Level Summary Commands	65
6.3.1.1	Examples	65
6.3.2	Branch Specific Commands	66
6.3.2.1	Examples	66
6.4	Debugging NAT Connectivity Issues	67
6.5	Stage Three Debugging on Branch	68
6.6	Troubleshooting SteelConnect EX FlexVNF Deployment Failure	70
7	Troubleshooting CG-NAT	72
7.1	CGNAT Summary	72
7.2	CGNAT Per Pool Statistics	72
7.3	Inspecting CGNAT State in vsmd	72
7.3.1	Configured Tenants and Summary	73
7.3.2	CGNAT ACLs for Traffic Matching	74
7.3.3	CGNAT ACL Counters	75
7.3.4	CGNAT ACL History	75
7.3.5	CGNAT Pool Config	76
7.3.6	CGNAT Pool Config for Tenant	76
7.3.7	CGNAT Rule Config	77
7.3.8	CGNAT Rule Config for Tenant	77
7.3.9	Active Counters	78
7.3.10	Internal Counters	78
7.3.11	CGNAT Resources State	79
7.3.12	CGNAT Pool Counters	79
7.3.13	Static NAT Mappings	80
7.3.14	CGNAT Pool with Source-Port	81
7.3.15	CGNAT Pool with Source-IP	81
8	Troubleshooting LTE Configuration	85
9	Troubleshooting Link Bandwidth	89
9.1	Configuring Speed Test Server	89
9.1.1	Using GUI	89

9.1.2	Using REST API	90
9.1.3	Using CLI	90
9.2	Viewing Client Speed Test Capability	91
9.3	Initiating Speed Test From Client	91
9.3.1	Using GUI	92
9.3.2	Using REST API	93
9.3.3	Using CLI	95
9.4	Initiating Speed Test From SD-WAN Branch	98
9.4.1	Using GUI	98
9.4.2	Using REST API	100
9.4.3	Using CLI	103

This section includes:

- [About SteelConnect EX SD-WAN Troubleshooting](#)(see page 7)
- [SteelConnect EX SD-WAN Stages](#)(see page 9)
- [Troubleshooting IKE/IPSec](#)(see page 10)
- [Troubleshooting Routing Protocols](#)(see page 20)
- [Troubleshooting SD-WAN Datapath](#)(see page 43)
- [Troubleshooting SD-WAN Branches](#)(see page 61)
- [Troubleshooting CG-NAT](#)(see page 72)
- [Troubleshooting LTE Configuration](#)(see page 85)
- [Troubleshooting Link Bandwidth](#)(see page 89)

1 About SteelConnect EX SD-WAN Troubleshooting

This section covers:

1.1 Introduction

This guide explains the possible troubleshooting solutions to problems that may occur when using SteelConnect EX Director.

1.2 Purpose

The FAQs and Troubleshooting Guide provides Frequently Asked Questions and troubleshooting information intended for those who use SteelConnect EX SD-WAN. These users must have basic knowledge of system level in UNIX and systems' transport and communication protocols.

1.3 Audience

This document is for experienced network and system administrators who are responsible for configuring and managing public and private cloud infrastructure. It is presumed that admins are aware of virtualization concepts, technologies, and setup of network devices.

1.4 Document Conventions

Convention	Description
Bold	Represents UI elements.
Monospace	Text/code that is entered by you.
CLI Outputs	Text/code that is displayed on your screen.
	Notes contain incidental information about the subject and call attention to exceptions.
	Caution indicates actions that can cause loss of data.
	Tips provide great shortcuts, hints, and recommended settings/configurable values.

1.5 Related Documentation

These related documentation are available from Riverbed.

- [SteelConnect EX Director Installation and Basic Configuration Guide](#)
- [SteelConnect EX FlexVNF Installation and Basic Configuration Guide](#)
- [SteelConnect EX Analytics Guide](#)
- [SteelConnect EX FlexVNF Basic SD-WAN Configuration Guide](#)
- [SteelConnect EX FlexVNF Advanced SD-WAN Configuration Guide](#)
- [SteelConnect EX FlexVNF Security Configuration Guide](#)
- [SteelConnect EX FlexVNF Network and System Configuration Guide](#)
- [SteelConnect Analytics and Monitoring Guide](#)
- [SteelConnect EX Director Monitor Guide](#)
- [SteelConnect EX Director Troubleshooting Guide](#)
- [SteelConnect EX Analytics Troubleshooting Guide](#)
- [SteelConnect EX Alarms Guide](#)
- [SteelConnect Entitlement and Licensing Guide](#)

1.6 Technical Support

Send an email to support@riverbed.com or go to support.riverbed.com .

2 SteelConnect EX SD-WAN Stages

This section covers:

2.1 SteelConnect EX SD-WAN Stages

SteelConnect EX SD-WAN has three staging phases. When a SD-WAN branch device is powered up, it automatically goes through three stages before it becomes completely operational. A branch device comes pre-loaded with staging server configuration. In IPSec configuration profile, staging server IP Address is given as remote IP address.

This table explains these three stages.

Stage One	Stage Two	Stage Three
<p>This is the pre-staging phase.</p> <ul style="list-style-type: none"> • IKE session starts between a branch and a staging server. • After the IKE session comes up, staging server assigns an IP address to the branch. • The SteelConnect EX Director IP address is notified to the branch. • A notification is generated to SteelConnect EX Director that the branch device has come up. 	<p>This is the staging phase.</p> <ul style="list-style-type: none"> • SteelConnect EX Director pushes the stage two configuration to branch device, through the staging server. • The controller IP address, in IPSec profile, is given as a remote IP. • The branch device is rebooted. • After the branch device comes up with stage two configuration, it establishes the IKE session with controller. • The controller assigns an IP address to the branch device and generates a notification to SteelConnect EX Director. 	<ul style="list-style-type: none"> • SteelConnect EX Director pushes the stage three configuration to branch device, over the IKE session, and reboots the branch device. • The branch device is fully operational and is a part of customer SD-WAN network. • IKE & IPSec session are created between branch and controller. • VXLAN and ESP sessions are created between branch to branch. • Branch to branch ESP is maintained using lightweight DH key pair proprietary protocol.

For stage one and two, IKE is over VNI interfaces, and for stage three, IKE is over loop back interfaces (TVI interfaces).

3 Troubleshooting IKE/IPSec

This section covers:

3.1 IKE Session History

Troubleshooting IKE involves debugging the session history.

Run the `show orgs org-services <org-name> ipsec vpn-profile branch-cntrl1 ike history` CLI command to debug IKE flaps. The system maintains IKE session history information and reason for flap.

```

versa@PoP3-Ten1-Branch2-cli> show orgs org-services Costco ipsec vpn-profile branch-cntrl1 ike history
Local Gateway: 10.3.12.1           Remote Gateway: 10.1.1.121
Last Known State                   : Active (Rekey)
Last State Timestamp               : 2015-12-21T06:25:5518.101768-00:00
Event History:
0. Event                          : IKE Rekey
Timestamp                        : 2015-12-21T06:25:5526.560768-00:00
Role                            : initiator
Inbound SPI : 0x3fd02bbfd83d0002
Outbound SPI : 0xe28dedee106e0002
1. Event                          : IKE Rekey
Timestamp                        : 2015-12-20T23:00:24534.391488-00:00
Role                            : initiator
Inbound SPI : 0xaf42d13b41ae0002
Outbound SPI : 0x32dd83255e370002
2. Event                          : IKE Rekey
Timestamp                        : 2015-12-20T15:34:53165.19972-00:00
Role                            : initiator
Inbound SPI : 0x598148d4b880002
Outbound SPI : 0x8d396252e73a0002
3. Event                          : IKE Rekey
Timestamp                        : 2015-12-20T08:09:21060.162088-00:00
Role                            : initiator
Inbound SPI : 0xdd149fd165df0002
Outbound SPI : 0xcd3cd3f7e85d0002

```

3.2 IKE Security Association

Run the `show orgs org-services <org-name> ipsec vpn-profile branch-cntrl1 ike security-associations brief` CLI command to display the IKE security association.

Example:

```

versa@PoP3-Ten1-Branch2-cli> show orgs org-services Costco ipsec vpn-profile branch-cntrl1 ike security-
associations brief
REMOTE
GATEWAY          PROFILE NAME    LOCAL SPI          REMOTE SPI          TRANSFORM
-----
10.1.1.121 branch-cntrl1 0x3fd02bbfd83d0002 0xe28dedee106e0002 aes-cbc

```

3.3 IPSec Tunnels

Fetch the IPsec tunnel information to troubleshoot IPsec.

Run the `show orgs org-services <org-name> ipsec vpn-profile branch-cntrl1 security- associations brief` CLI command to display the IPSEC tunnel information.

Example:

```

versa@PoP3-Ten1-Branch2-cli> show orgs org-services Costco ipsec vpn-profile branch-cntrl1 security-
associations brief
Remote Gateway    Transform Inbound SPI Bytes/sec    Outbound SPI Bytes/sec    Tunnel Status    Up Time
-----
10.3.11.1         aes-cbc    0x20aebb9      0            0x20b5bba      0            UP
1071 sec

```

3.3.1 Overall IPSEC Statistics

Follow these steps to find total number of IKE and IPSEC sessions. This gives the number of phase 1 failure, rekeys and other related information.

1. Run the `vsh connect vsmdshell` command to login to vsmd.
2. Run the `show ipsec stats` CLI command at the vsmd command prompt.

```
vsm-vcsn0> show ipsec stats
----- IPSec Control Plane Stats from PM -----
      IKE SA Active           : 4
      IPSec SA Active         : 4
      P1 Done                  : 30
      P1 failed                : 4
      P1 rekeyed               : 19
      IKE SA                   : 7
      IKE SA Initiated         : 7
      IKE SA Responded         : 0
      IKE Attempts             : 11
      IKE Attempts Initiated: 11
      IKE Attempts Responded: 0
      IKE Packets in           : 98
      IKE Packets out          : 162
      IKE Octets in            : 16296
      IKE Octets out           : 31472
      IKE Retransmits          : 57
      IKE Discarded Packets : 0
      IKE Init Failures        : 4
      Init NO responses        : 4
      resp failures            : 0
      Engine Active Flows      : 4
      Transforms               : 4
      Fast Path Packets In     : 104
      Fast Path Packets Fwd    : 0
```

3. Run the `show ipsec stats <tenant-id>` CLI command to view the tenant specific information.

```
show ipsec stats <tenant-id>
tenant-id - 0 for all tenants
```

3.4 Troubleshooting IPSec Stage 1 and Stage 2

Follow these steps to diagnose problems in Stage 1 or Stage 2 communication:

1. Run the `show orgs org-services <org-name> ipsec vpn-profile branch-cntrl1 security-associations brief` CLI command to check if the IPSec session between branch and controller is up.

```
versa@PoP3-Ten1-Branch2-cli> show orgs org-services Costco ipsec vpn-profile branch-cntrl1 security-associations brief
```

Remote Gateway	Transform	Inbound SPI	Bytes/sec	Outbound SPI	Bytes/sec	Tunnel Status	Up Time
10.3.11.1	aes-cbc	0x20aebb9	0	0x20b5bba	0	UP	1071 sec

2. If the IPSec session between branch and controller is not up, then:
 - a. Check the IPSec configuration to ensure that local and remote authentication parameters are matching.
Local IP and Remote IP belong to VNI interfaces.
 - b. Run the `show ipsec config 0 all` CLI command to check if the applied configuration is present in the backend. The **Loaded** parameter in the command output must display **Yes**.

Example:

```
vsm-vcsn0> show ipsec config 0 all
##### Tenant 2 config #####
-----
VPN Name - branch-cntrl1, OBJ ID 1, VPN ID 1
      VPN Type - Branch-SD-WAN
      VRF ID - mgmt1(18)
      Tunnel VRF ID - mgmt1(18)
      VSN ID - 0
      Loaded - Yes
      Local - 10.3.12.1
----- Local Identity -----
      Auth Type - Pre Shared Key
      ID Type - EMAIL
      ID - br2_west_email@Provider.com
      PSK - 1234
----- IPSec Datapath Configuration -----
      Anti Replay : Enabled
      Mode : Tunnel
      PFS : 0
      Transform : 1
      Lifetime : 25000 seconds, 0 mbytes
----- IKE Control Path Configuration -----
      DH Group : 19
      Transform : 1
      Lifetime : 27000
      Version : 2
      DPD Timeout : 10
----- ## Flows 0 -----
      Flow 0 SRC any DST any
-----
```

- c. Ping from local IP to the remote IP (which are specified in IPSec profile) and ensure the remote IP is reachable.
If this step fails, then this is a datapath issue. Use data plane diagnostics to figure out the problem.
3. If the IPSec session between branch and controller is up, then follow these:
 - a. Run the `show interfaces brief` CLI command and check the output. The TVI interface must have an associated IP address.
Interface `tv i-0/3.0` is the IPSec tunnel interface and the IP address assigned by staging server/controller.

```
admin@PoP3-Ten2-Branch5-cli> show interfaces brief
```

NAME	IP	MAC	OPER	ADMIN	TNT

tvi-0/3	n/a	up	up		
tvi-0/3.0	[10.3.1.113/24]	n/a	up		up
1	mgmt				
vni-0/0	52:0a:30:be:05:02		up		up
vni-0/0.0	[113.1.1.5/24]	52:0a:30:be:05:02	up		up
1	grt-vrf				
vni-0/1		52:0a:30:be:05:03	down		down
vni-0/2		52:0a:30:be:ce:04	down		down

- b. Run the `vsh connect infmgr shell` command to connect to **inmgr**.
 - c. Run the `show p2mp <mgmt. -routing-instance>` CLI command and check the output for a valid vnf-manager address (SteelConnect EX Director IP Address).
4. If **Step 3** is verified, then ping from SteelConnect EX Director to the branch device.
If the ping from SteelConnect EX Director to the branch device fails, check if the route table has proper route installed for the SteelConnect EX Director IP address in branch device.
 5. Run the **ssh** command to access branch device from SteelConnect EX Director.

3.5 Troubleshooting IPsec Stage 3- Branch to Controller

Follow these steps to diagnose problems in Stage 3 communication:

1. Run the `show orgs org-services <org-name> ipsec vpn-profile branch-cntrl1 security-associations br` CLI command to check if the IPsec session between branch and controller is up.

```
versa@PoP3-Ten1-Branch2-cli> show orgs org-services Costco ipsec vpn-profile branch-cntrl1 security-associations br
```

Remote Gateway	Transform	Inbound SPI	Bytes/sec	Outbound SPI	Bytes/sec	Tunnel Status	Up Time
10.3.11.1	aes-cbc	0x20aebb9	0	0x20b5bba	0	UP	
1071 sec	>>>> First one is between branch and controller.						
10.3.13.1	aes-cbc	0x20adbba	0	0x20adbba	0	UP	
1113 sec	>>>> Later entries are for branch tobranch.						
10.3.14.1	aes-cbc	0x20adbba	0	0x20adbba	0	UP	
339 sec							
10.1.1.121	aes-cbc	0x20069de	0	0x2000a36	0	UP	
9728 sec							
[ok] [2015-12-21 09:30:00]							

2. Check the IPsec configuration to ensure if local and remote authentication parameters are matching. Local IP and Remote IP belong to VXLAN TVI interfaces.
3. Run the `show ipsec config 0 all` CLI command to check if the applied configuration is present in the backend. The **Loaded** parameter in the command output must display **Yes**.

Example:

```
vsm-vcsn0> show ipsec config 0 all
##### Tenant 2 config#####
-----
VPN Name - branch-cntrl1, OBJ ID 1, VPN ID 1
VPN Type - Branch-SD-WAN
VRF ID - mgmt1(18)
Tunnel VRF ID - mgmt1(18)
VSN ID - 0
Loaded - Yes <<<< == It should be YES
Local - 10.3.12.1
----- Local Identity -----
Auth Type - Pre Shared Key
ID Type - EMAIL
ID - br2_west_email@Provider.com
PSK - 1234
----- IPSec Datapath Configuration -----
Anti Replay : Enabled
Mode : Tunnel
PFS : 0
Transform : 1
Lifetime : 25000 seconds, 0 mbytes
----- IKE Control Path Configuration -----
DH Group : 19
Transform : 1
Lifetime : 27000
Version : 2
DPD Timeout : 10
----- ## Flows 0 -----
Flow 0 SRC any DST any
-----
-----
```

4. If **Step 3** is verified, ping from local IP to remote IP, specified in IPSec profile and ensure remote IP is reachable.
If this step fails, this is a datapath issue. Use data plane diagnostics to figure out the problem.
5. If the vxlan-ping between branch to controller works fine then perform an esp-ping between branch to controller.

A branch with configuration after completion of staging is referred as Stage 3 branch.

3.6 Troubleshooting IPSec Stage 3- Branch to Branch

Follow these steps to diagnose problems in stage 3 communication between two branches:

1. Run the show orgs org-services <org-name> ipsec vpn-profile branch-cntrl1 branch-2- branch security-associations br CLI command to check if the IPSec session between branch to all other branches are up.

First entry is for controller and rest are for branches. If there are five branches, there will be six entries; one for controller and five for five branches. All the IPsec sessions need to be up.

Example:

```
versa@PoP3-Ten1-Branch2-cli> show orgs org-services Costco ipsec vpn-profile branch-cntrl1 branch-2-branch security-associations br
```

Remote Gateway	Transform	Inbound SPI	Bytes/sec	Outbound SPI	Bytes/sec	Tunnel Status	Up Time
10.3.11.1	aes-cbc	0x20aebb9	0	0x20b5bba	0	UP	1071 sec
>>>> First one is between branch and controller.							
10.3.13.1	aes-cbc	0x20adbbb	0	0x20adbba	0	UP	1113 sec
>>>> Later entries are for branch to branch.							
10.3.14.1	aes-cbc	0x20adbba	0	0x20adbba	0	UP	339 sec
10.1.1.121	aes-cbc	0x20069de	0	0x2000a36	0	UP	9728 sec

[ok] [2015-12-21 09:30:00]

DHKEY Pair protocol is used to exchange the IPSEC keys for branch to branch communication.

For every remote branch, one PTVI-ESP interface is created and DHKEY pair protocol generates and refreshes the IPSEC key pairs periodically between any 2 branches. This SPI is associated with the PTVI-ESP interface corresponding to the remote branch.

2. ESP ping between two branches.
3. If **Step 2** fails then enable IPSEC debugging logs.
If the *packet dropped due to invalid SPI index* message is displayed then ensure correct SPI index is associated with right PTVI index.

3.7 Certificate Based Authentication

This section lists the commands that are used to debug IPsec problems in SD-WAN network.

3.7.1 Certificate Information

Run the `show orgs org-services Provider security crypto-cfg pki` CLI command to get certificate information.

Example:

```

admin@manju-vm1-14-cli> show orgs org-services Provider security crypto-cfg pki
security crypto-cfg pki private-keys versa-ctrl1-key
algo          RSA
modulus      1024
pub-key      "AwEAAbSdwDYhEARVISErUJeNeZVWO/bY6DJYWCdLnCxm2tqYgRgk6WF9xs/XakyVmh9eeHb/
\nZoCO2LasKPIrJQ+KiNuGGbe1SYnox16qLai9mabkoGW1B1Iz+58h7tBzo+GRODTL6eXN9dlw\
ninKIROUZ4sYZeIXTMjyYEHZA2jy9pZuZ"

security crypto-cfg pki certificates versa-ctrl1-cert
priv-key     versa-ctrl1-key
ca           versa-ctrl1-cert
CA-CERT      NO
not-before   Mar-3-2016
not-after    Nov-17-2016
pub-key      "AwEAAbSdwDYhEARVISErUJeNeZVWO/bY6DJYWCdLnCxm2tqYgRgk6WF9xs/XakyVmh9eeHb/
\nZoCO2LasKPIrJQ+KiNuGGbe1SYnox16qLai9mabkoGW1B1Iz+58h7tBzo+GRODTL6eXN9dlw\
ninKIROUZ4sYZeIXTMjyYEHZA2jy9pZuZ"
cert-data
"MIIDMzCCAhuGAWIBAgIIBQJwQtYCHZEwDQYJKoZIhvcNAQEFBQAwFjEUMBIGA1UEAwLVmVy\nc2FUZXN0Q0EwHhcNMTYwMzAzMjAxMTA4
WhcNMTYxMTE3MTgxMTM3WjBzMScwJQYDVQQDDDB52\
nZXJzYS1jdHJsMS52ZXJzYS1uZXR3b3Jrcy5jb20xETAPBgNVBAsMCHNvZnR3YXJlMQ4wDAYD\nVQQKDAV2ZXJzYTELMAkGA1UEBwwCU0Mx
CzAJBgNVBAGMAkNBMBQswCQYDVQQGEwJVUzCBnzAN\
nBgkqhkiG9w0BAQEFAAOBjQAwGyKCGYEAATJ3ANiEQCtUhIstQ14151VY79tjOm1hYIMucLGba\n2piBGCTpYX3Gz9dqTJWah154dv9mgI7Y
tqwo8is1D4qI24YZt7VJiejHXqotqL2ZpuSgZaUH\
nUjP7nyHu0HOj4ZE4NMvp5c312XCKcohE5Rnixh14hdMyPJgQfMDaPL2lm5kAwEAAaOBqzCB\nnqDAAdBgNVHQ4EFgQUGJzicSi+eD8NgdR8
Em6YQYU9wfsWDAyDVR0TAAQH/BAIwADAfBgNVHSME\
nGDAWgBQA5Ca3UiqJzkJapVxtFD0Jf43BezAOBgNVHQ8BAf8EBAMCBeAwHQYDVR01BBYwFAYI\nnKwYBBQUHAWIGCCsGAQUFBwMEMckGA1Ud
EQQiMCCBHNZ1cnNhLWN0cmwwQHZ1cnNhLW51dHdv\
ncmtzLmNvbTANBgkqhkiG9w0BAQUFAAOCAQEajlg59j90di7XOqDN2Y9KkNmFEcNrJvn+Cwp8\n\nxJxnW/
AtZQkQ9JFTY5gf9oYdmPnuzXOI8FzNZ+xestAwWC8nV0klWf7jHA2ZKbsKhHN9JZ9S\
nza+/38A+KFMIyF3sF61Orgh9kLUF+SXRX5FlwWLuST0IzfrJlhur4qGchVIPKpHia9fSKukt\
nEURkh14oQZIRQIBDWxv5eiYIKHalTkGj6SlgiKRvSEcz+Se541ow2M7pr/
OQpesw2yJWtOd1\nnLiW5JHPHe4m71Bysyd9Ly3yBpukU5tsjKrZN+jma0IfuwL1/1HA4IIIPwYvpK5OdMh88L/
pt\np6eEhd1no8+AJreKvg=="

security crypto-cfg pki ca-chains versa-ca
ca-chain-certificates 21437e7376b2747332bb51107198e436
ca-name          "CommonName: VersaTestCA, orgUnit: , organization:
" ca-cert-data
"MIIDETCCAfmGAWIBAgIiYRn5dAOLT70wDQYJKoZIhvcNAQEFBQAwFjEUMBIGA1UEAwLVmVy\nc2FUZXN0Q0EwHhcNMUxMTE3MTgxMTM3
WhcNMUxMTE3MTgxMTM3WjBwMRQwEgYDVQQDDAtW\
nZXJzYVRlc3RDQTCCASiwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAAJk59iQdxcFyQuLU\n\nalsI2Pba71U11G0AGZ4NTLTx61r7E3aV
aYAb97y7a5kUMq2jff8ovg493Dm7UsHHlsILpNHG1\
nqLKbG7gPotUXb/
D0mFqFcRp1KaKSp+4BA4BdgDhUG08YttvLTT3w8TCzaqrFE+if6+JkUt1W\n\nnkNMEeOnppDGCBJ3Dh2TJcyICjDnWoxoIqIozVv964mPYZUy
+O1CM0Q5yJ4oZeyeI1dTQMxw6\
nhTa9qpeJ54vMxG91Yg1DWB/BWrS1p1lv7uwBYU5bgE1E46h50ILXLIQe0d891N/
i6PAWd9\n\nnhLLEG9TuwInDaks4OpZ32M132ZGZLN5RgPN17t8CAwEAaAnjMGEWHQYDVR0OBBYEFADkJrds\
nKonOQ1qlve0UPQ1/jcF7MA8GA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAUAQOmt1Iqic5C\n\nnWqVV7RQ9CX+NwXswDgYDVR0PAQH/
BAQDAgGGMA0GCSqGSIsb3DQEBBQUAA4IBAQBZFPBJwVQM\
niY2RylCADC5Vf3kkGzBXqaQirtE59bMQytzgf/28H7g8n+GTz0RDoBclgPbPunyWLuNK5Qx8\n\nnh55fi7Tm21k5Kckzh/
xFQKYEn6011QhgXhvdv11qlnxCOJxd+Q7ELAWe1t+m174mgDcB/L5Q\

```

```
ncURVAjJiSjDh0l0eFaj6fW69CP4F2KLpn100iGisXylwgrLXD2+Oub4w4de2dou58GgVUu9S\nkSKTxNRJXvosk3dGfg9tR+Ovhl4pslXg
/8Axw5ZrgwtfRUlA7DfmTuDxzfqdvtCJ7U87jNGh\
ne6mawICTy0VxnhWHnCsKc3Akra6duJCilCe2q5F1FXd"
```

3.7.2 Certificate Management Protocol

Run the `show ipsec config 0 all` CLI command to get the IPsec configuration.

Example:

```
vsm-vcsn0> show ipsec config 0 all
##### Tenant 1 config#####
-----
VPN Name - gw1, OBJ ID 1, VPN ID 1
    VPN Type - Controller-SD-WAN
    DHKey: Profile = []
    VRF ID - mgmt(10)
    Tunnel VRF ID - mgmt(10)
    VSN ID - 0
    Loaded - Yes
    Crypto offload - Enabled (i.e. use if any hw-accel available)
    Local - 10.10.1.1
    ----- Local Identity -----
        Auth Type - Certificate
        Certificate-Authority - versa-ca
        Certificate present
        Private key present
        ID Type - EMAIL
        ID - versa-ctrl1@versa-networks.com
    ----- IPsec Datapath Configuration -----
        Anti Replay : Enabled
        Mode : Tunnel
        PFS : 0
        Transform : 1
        Lifetime : 25000 seconds, 0 mbytes
    ----- IKE Control Path Configuration -----
        DH Group : 19
        Transform : 1
        Lifetime : 28000
        Version : 2
        DPD Timeout : 30
    ----- ## Flows 0 -----
    -- General AddrRange NetMask #Subnets 0 VSN 0---
        VSN 0 - Range
    -- Mgmt AddrRange NetMask #Subnets 0 VSN 0---
        VSN 0 - Range
    --RAC AuthType 2, EAP Type 0, #Clients 0 --
```

3.7.3 Certificate Management Protocol

Run the `show certd csr summary` CLI command to get summary of certificate.

```

certd> show certd csr summary
Specify <tenant-id>
certd> show certd csr summary 1

```

CSR-Name	Request-State
provider-br1-cert	Sign-Done

```

certd> show certd csr stats
Specify <tenant-id> <csr-name>
certd> show certd csr stats 1 provider-br1-cert
CERTD CSR (provider-br1-cert) stats for Tenant: 1

```

```

CERTD srvr: provider-ca

```

Cert check success (IR)	: 1
Interface down (IR)	: 0
Interface addr unavailable (IR)	: 0
Auth-key fail (IR)	: 0
Auth-cert fail (IR)	: 0
Prv-key gen fail (IR)	: 0
CSR gen fail (IR)	: 0
CSR enroll sent (IR)	: 0
Switch ns fail (IR)	: 0
Revert ns fail (IR)	: 0
Cert sign success (IR)	: 0
CA cert sign success (IR)	: 0
Cert sign rejected (IR)	: 0
Cert sign done (IR)	: 0
Cert sign failure (IR)	: 0
Cert check success (KUR)	: 0
Interface down (KUR)	: 0
Interface addr unavailable (KUR)	: 0
Auth-key fail (KUR)	: 0
Auth-cert fail (KUR)	: 0
Prv-key gen fail (KUR)	: 0
CSR gen fail (KUR)	: 0
CSR enroll sent (KUR)	: 0
Switch ns fail (KUR)	: 0
Revert ns fail (KUR)	: 0
Cert sign success (KUR)	: 0
CA cert sign success (KUR)	: 0
Cert sign rejected (KUR)	: 0
Cert sign done (KUR)	: 0
Cert sign failure (KUR)	: 0

4 Troubleshooting Routing Protocols

This section covers:

4.1 Troubleshooting Routing Protocols

Run these CLI commands to troubleshoot routing protocol related issues:

- `show route`
- `show route routing-instance lan-vrf`
- `show route advertising-protocol bgp`
- `run show route receive-protocol bgp`
- `run show route receive-protocol bgp neighbor-address <IP address>`
- `show route table l3vpn.ipv4.unicast receive-protocol bgp neighbor-address <IP address>`
- `run show route advertising-protocol bgp neighbor-address <IP address>`
- `show route table l3vpn.ipv4.unicast advertising-protocol bgp neighbor-address <IP address>`

4.1.1 Routing Protocol Command Examples

This section contains examples of CLI commands that are used to troubleshoot routing protocol related issues.

4.1.1.1 CLI Command to view Routing Daemon Routing Table (RIB)

```

versa@rams-test2-cli> show route
Routes for Routing instance : global
Codes: E1 - OSPF external type 1, E2 - OSPF external type 2
IA - inter area, iA - intra area,
L1 - IS-IS level-1, L2 - IS-IS level-2
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
* - candidate default

```

Prot	Dest Address/Mask	Next-hop	Age	Interface name
OSPF	0.0.0.0/0	192.168.201.15	00:02:31	vni-0/0
OSPF	0.0.0.0/0	192.168.202.15	00:02:31	vni-0/1
OSPF	1.2.0.0/16	192.168.201.15	00:02:31	vni-0/0
OSPF	1.2.0.0/16	192.168.202.15	00:02:31	vni-0/1
static	1.2.3.0/24	0.0.0.0	00:02:58	Indirect
OSPF	1.2.3.0/24	192.168.201.15	00:02:31	vni-0/0
OSPF	1.2.3.0/24	192.168.202.15	00:02:31	vni-0/1
OSPF	1.10.11.12/32	192.168.201.15	00:02:31	vni-0/0
OSPF	1.10.11.12/32	192.168.202.15	00:02:31	vni-0/1
OSPF	9.10.11.12/32	192.168.201.15	00:02:31	vni-0/0
OSPF	9.10.11.12/32	192.168.202.15	00:02:31	vni-0/1
OSPF	9.10.12.0/24	192.168.201.15	00:02:31	vni-0/0
OSPF	9.10.12.0/24	192.168.202.15	00:02:31	vni-0/1
OSPF	192.168.201.0/24	0.0.0.0	00:02:37	vni-0/0
conn	192.168.201.0/24	192.168.201.14	00:03:08	vni-0/0
local	192.168.201.14/32	0.0.0.0	00:03:08	directly connected
OSPF	192.168.202.0/24	0.0.0.0	00:02:37	vni-0/1
conn	192.168.202.0/24	192.168.202.14	00:03:08	vni-0/1
local	192.168.202.14/32	0.0.0.0	00:03:08	directly connected
OSPF	192.168.203.0/24	0.0.0.0	00:02:37	vni-0/2
conn	192.168.203.0/24	192.168.203.7	00:03:08	vni-0/2
local	192.168.203.7/32	0.0.0.0	00:03:08	directly connected
OSPF	192.168.204.0/24	0.0.0.0	00:02:37	vni-0/3
conn	192.168.204.0/24	192.168.204.7	00:03:08	vni-0/3
local	192.168.204.7/32	0.0.0.0	00:03:08	directly connected

```

[ok] [2014-12-08 10:22:51]

```

4.1.1.2 CLI Command to view Routing Daemon Routing Table (RIB) for a routing-instance

```
admin@rams-test2-cli> show route routing-instance lan-vrf
Routes for Routing instance : lan-vrf AFI: ipv4
Codes: E1 - OSPF external type 1, E2 - OSPF external type 2
IA - inter area, iA - intra area,
L1 - IS-IS level-1, L2 - IS-IS level-2
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
+ - Active Route
Prot  Type  Dest Address/Mask      Next-hop    Age         Interface name
----  -
conn  N/A    +1.1.1.0/24            0.0.0.0     19:51:10    tvi-0/41.0
local N/A    +1.1.1.2/32            0.0.0.0     19:51:10    directly connected
BGP   N/A    +192.168.17.0/24       21.21.22.2  19:50:26    Indirect
BGP   N/A    +192.168.18.0/24       21.21.22.2  19:50:26    Indirect
BGP   N/A    +192.168.45.0/24       21.21.22.2  19:50:26    Indirect
CLI Command to view routes advertised by BGP
versa@rams-test2-cli> show route advertising-protocol bgp
Routes for Routing instance : global
Prefix/Mask      Next-hop      MED Lclpref AS path
-----
1.2.3.0/24       192.168.202.14 0 0    5
1.2.3.0/24       192.168.201.14 0 0    5
[ok] [2014-12-08 10:23:04]
```

4.1.1.3 CLI Command to view Routes Received by BGP

```
versa@rams-test2-cli(config)% run show route receive-protocol bgp
Routes for Routing instance : global
Prefix/Mask      Next-hop      MED Lclpref AS path
-----
0.0.0.0/0        192.168.202.15 0 100   6 6 6 6
1.2.0.0/16       192.168.202.15 0 100   6 6 6 6
1.2.3.0/24       192.168.202.15 0 100   6 6 6 6
1.10.11.12/32    192.168.202.15 0 100   6 6 6 6
9.10.11.12/32    192.168.202.15 0 100   6 6 6 6
9.10.12.0/24     192.168.202.15 0 100   6 6 6 6
128.0.0.0/1
0.0.0.0/0        192.168.201.15 0 100   6 6 6 6
1.2.0.0/16       192.168.201.15 0 100   6 6 6 6
1.2.3.0/24       192.168.201.15 0 100   6 6 6 6
1.10.11.12/32    192.168.201.15 0 100   6 6 6 6
9.10.11.12/32    192.168.201.15 0 100   6 6 6 6
9.10.12.0/24     192.168.201.15 0 100   6 6 6 6
128.0.0.0/1
[ok] [2014-12-08 10:24:45]
```

4.1.1.4 CLI Command to view Routes Received by BGP from one Particular Neighbor

```

versa@rams-test2-cli(config)% run show route receive-protocol bgp neighbor-address 192.168.201.15
Routes for Routing instance : global
Prefix/Mask      Next-hop      MED Lc1pref AS path
-----
0.0.0.0/0        192.168.201.15 0 100    6 6 6 6
1.2.0.0/16       192.168.201.15 0 100    6 6 6 6
1.2.3.0/24       192.168.201.15 0 100    6 6 6 6
1.10.11.12/32    192.168.201.15 0 100    6 6 6 6
9.10.11.12/32    192.168.201.15 0 100    6 6 6 6
9.10.12.0/24     192.168.201.15 0 100    6 6 6 6
128.0.0.0/1
[ok] [2014-12-08 10:25:11]

```

4.1.1.5 CLI Command to view VPN IPv4 (L3 VPN) Routes Received by BGP from one Particular Neighbor

```
admin@rams-test2-cli> show route table l3vpn.ipv4.unicast receive-protocol bgp neighbor-address 21.21.20.2
Routes for Routing instance : global AFI: ipv4
% No entries found.
Routes for Routing instance : grt-vrf AFI: ipv4
% No entries found.
Routes for Routing instance : lan-vrf AFI: ipv4
% No entries found.
Routes for Routing instance : log-vrf AFI: ipv4
% No entries found.
Routes for Routing instance : mgmt2 AFI: ipv4
Routing entry for 192.168.17.0/24
Route Distinguisher : 1L:2
Peer Address      : 21.21.20.2
VPN Label         : 25600
Community         : [ 808:808 1105:1105 ]
Extended community : [ target:3L:4 ]
Next-hop          : 21.21.22.2
MED               : 0
Local Preference  : 300
Origin            : Igp
AS Path           : N/A
Routing entry for 192.168.18.0/24
Route Distinguisher : 1L:2
Peer Address      : 21.21.20.2
VPN Label         : 25600
Community         : [ 808:808 1105:1105 ]
Extended community : [ target:3L:4 ]
Next-hop          : 21.21.22.2
MED               : 0
Local Preference  : 300
Origin            : Igp
AS Path           : N/A
```

4.1.1.6 CLI Command to view Routes Advertised by BGP to one Particular Neighbor

```

versa@rams-test2-cli(config)% run show route advertising-protocol bgp neighbor-address 192.168.201.15
Routes for Routing instance : global
Prefix/Mask      Next-hop      MED Lclpref AS path
-----
1.2.3.0/24       192.168.201.14 0 0      5
[ok] [2014-12-08 10:25:34]
[edit]
versa@rams-test2-cli(config)% run show route advertising-protocol bgp neighbor-address 192.168.202.15
Routes for Routing instance : global
Prefix/Mask      Next-hop      MED Lclpref AS path
-----
1.2.3.0/24       192.168.202.14 0 0      5
[ok] [2014-12-08 10:25:42]
versa@rams-test2-cli(config)%

```

4.1.1.7 CLI Command to view VPN IPv4 (L3 VPN) Routes sent by BGP to one Particular Neighbor

```
admin@rams-test2-cli> show route table l3vpn.ipv4.unicast advertising-protocol bgp neighbor-address 21.21.20.2
Routes for Routing instance : global AFI: ipv4
% No entries found.
Routes for Routing instance : grt-vrf AFI: ipv4
% No entries found.
Routes for Routing instance : lan-vrf AFI: ipv4
% No entries found.
Routes for Routing instance : log-vrf AFI: ipv4
% No entries found.
Routes for Routing instance : mgmt2 AFI: ipv4
Routing entry for 192.168.19.0/24
Route Distinguisher: 1L:2
Peer Address      : 21.21.20.2
VPN Label         : 25600
Community         : [ N/A ]
Extended community : [ N/A ]
Next-hop          : 21.21.23.2
MED               : 0
Local Preference  : 300
Origin            : Igp
AS Path           : N/A
Routing entry for 192.168.20.0/24
Route Distinguisher : 1L:2
Peer Address        : 21.21.20.2
VPN Label           : 25600
Community           : [ N/A ]
Extended community  : [ N/A ]
Next-hop            : 21.21.23.2
MED                 : 0
Local Preference    : 300
Origin              : Igp
AS Path             : N/A
```

4.2 Troubleshooting BGP Commands

Run these CLI commands to troubleshoot BGP related issues:

- show bgp summary
- show bgp summary instance global
- show bgp group brief
- show bgp group brief global
- show bgp neighbor detail
- show bgp neighbor detail global
- show bgp neighbor detail global 192.168.202.15

4.2.1 BGP Command Examples

This section contains examples of CLI commands, that are used to troubleshoot BGP related issues.

4.2.1.1 CLI Command to view BGP Summary Information for all Routing-Instances

```

versa@rams-test2-cli> show bgp summary
routing-instance: global
BGP instance 2
router-id : 192.168.202.14
ipv4 unicast statistics:
iBGP routes in      : 0
eBGP routes in : 8
  Eligible routes : 10
  Ineligible routes : 0
  Active routes   : 9
  Advertised routes : 4
  Suppressed routes : 0
ipv6 unicast statistics:
iBGP routes in      : 0
eBGP routes in      : 0
  Eligible routes   : 0
  Ineligible routes : 0
  Active routes     : 0
  Advertised routes : 0
  Suppressed routes : 0
Neighbor      V  AS      MsgRcvd  MsgSent    Up/Down      State/PfxRcd  PfxSent
192.168.201.15 4  6       334      334        02:28:47     02:28:47      8
192.168.202.15 4  6       334      334        02:28:32     02:28:32      0

```

4.2.1.2 CLI Command to view BGP Summary Information for a Specific Routing-Instance

```

versa@rams-test2-cli> show bgp summary instance global
routing-instance: global
BGP instance 2
router-id : 192.168.202.14
ipv4 unicast statistics:
iBGP routes in : 0
eBGP routes in : 8
  Eligible routes : 10
  Ineligible routes : 0
  Active routes : 9
  Advertised routes : 4
  Suppressed routes : 0
ipv6 unicast statistics:
iBGP routes in : 0
eBGP routes in : 0
  Eligible routes : 0
  Ineligible routes : 0
  Active routes : 0
  Advertised routes : 0
  Suppressed routes : 0
Neighbor      V    AS MsgRcvd   MsgSent Up/Down State/PfxRcd PfxSent
192.168.201.15  4    6   336      336      336      02:29:44      8      2
192.168.202.15  4    6   336      336      336      02:29:29      0
2
[ok][2014-12-09 13:07:44]

```

4.2.1.3 CLI Command to view Information for BGP Peer-Groups of all Routing-Instances

```

versa@rams-test2-cli> show bgp group brief ?
Possible completions:
<routing-instance:string, max: 32 chars>
versa@rams-test2-cli> show bgp group brief
routing-instance: global
BGP instance: 2
Group Type: External          Local AS: 5
Name: g1
Options: < PEER-AS >
peer-as: 6
Total peers: 2      Established: 2
192.168.201.15+60719
192.168.202.15+54389

```

4.2.1.4 CLI Command to view Information for BGP Peer-Groups of a Specific Routing-Instance

```
versa@rams-test2-cli> show bgp group brief global
routing-instance: global
BGP instance: 2
Group Type: External          Local AS: 5
Name: g1
Options: < PEER-AS >
peer-as: 6
Total peers: 2      Established: 2
192.168.201.15+60719
192.168.202.15+54389
[ok] [2014-12-09 13:09:59]
```

4.2.1.5 CLI Command to view all BGP Neighbors of all Routing-Instances

```

versa@rams-test2-cli> show bgp neighbor detail
routing-instance: global
BGP instance: 2
Peer: 192.168.201.15+179 AS 6 Local: 192.168.201.14+60719 AS 5
Type: External State: Established Up for: 02:24:29
Last State: Openconfirm Last Event: RecvKeepAlive
Holdtime: 90
Peer ID: 192.168.201.156 Local ID: 192.168.202.14
Keepalive Interval: 30
BFD: disabled, down
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh functionality
Peer supports Restarter functionality
Peer supports 4 byte AS extension (peer-as 6)
Input messages: Total 325 Updates 2 Refreshes 0
Output messages: Total 335 Updates 2 Refreshes 0
routing-instance: global
BGP instance: 2
Peer: 192.168.202.15+179 AS 6 Local: 192.168.202.14+54389 AS 5
Type: External State: Established Up for: 02:24:14
Last State: Openconfirm Last Event: RecvKeepAlive
routing-instance: global
BGP instance: 2
Peer: 192.168.201.15+179 AS 6 Local: 192.168.201.14+60719 AS 5
Type: External State: Established Up for: 02:24:29
Last State: Openconfirm Last Event: RecvKeepAlive
Holdtime: 90
Peer ID: 192.168.201.156 Local ID: 192.168.202.14
Keepalive Interval: 30
BFD: disabled, down
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh functionality
Peer supports Restarter functionality
Peer supports 4 byte AS extension (peer-as 6)
Input messages: Total 325 Updates 2 Refreshes 0
Output messages: Total 335 Updates 2 Refreshes 0
routing-instance: global
BGP instance: 2
Peer: 192.168.202.15+179 AS 6 Local: 192.168.202.14+54389 AS 5
Type: External State: Established Up for: 02:24:14
Last State: Openconfirm Last Event: RecvKeepAlive
Holdtime: 90
Peer ID: 192.168.201.156 Local ID: 192.168.202.14
Keepalive Interval: 30
BFD: disabled, down
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh functionality

```

```

Peer supports Restarter functionality
Peer supports 4 byte AS extension (peer-as 6)
Input messages: Total 325 Updates 2 Refreshes 0
Output messages: Total 335 Updates 1 Refreshes 0

```

4.2.1.6 CLI Command to view all BGP Neighbors of a Specific Routing-Instance

```

versa@rams-test2-cli> show bgp neighbor detail global
routing-instance: global
BGP instance: 2
Peer: 192.168.201.15+179 AS 6 Local: 192.168.201.14+60719 AS 5
Type: External State: Established Up for: 02:25:51
Last State: Openconfirm Last Event: RecvKeepAlive
Holdtime: 90
Peer ID: 192.168.201.156 Local ID: 192.168.202.14
Keepalive Interval: 30
BFD: disabled, down
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh functionality
Peer supports Restarter functionality
Peer supports 4 byte AS extension (peer-as 6)
Input messages: Total 328 Updates 2 Refreshes 0
Output messages: Total 338 Updates 2 Refreshes 0
routing-instance: global
BGP instance: 2
Peer: 192.168.202.15+179 AS 6 Local: 192.168.202.14+54389 AS 5
Type: External State: Established Up for: 02:25:36
Last State: Openconfirm Last Event: RecvKeepAlive
Holdtime: 90
Peer ID: 192.168.201.156 Local ID: 192.168.202.14
Keepalive Interval: 30
BFD: disabled, down
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh functionality
Peer supports Restarter functionality
Peer supports 4 byte AS extension (peer-as 6)
Input messages: Total 328 Updates 2 Refreshes 0
Output messages: Total 339 Updates 1 Refreshes 0
[ok][2014-12-09 13:03:52]

```

4.2.1.7 CLI Command to view a Specific BGP Neighbor of a Specific Routing-Instance

```

versa@rams-test2-cli> show bgp neighbor detail global 192.168.202.15
routing-instance: global
BGP instance: 2
Peer: 192.168.202.15+179 AS 6 Local: 192.168.202.14+54389 AS 5
Type: External State: Established Up for: 02:26:46
Last State: Openconfirm Last Event: RecvKeepAlive
Holdtime: 90
Peer ID: 192.168.201.156 Local ID: 192.168.202.14
Keepalive Interval: 30
BFD: disabled, down
NLRI advertised by peer: inet-unicast
NLRI for this session: inet-unicast
Peer supports Refresh functionality
Peer supports Restarter functionality
Peer supports 4 byte AS extension (peer-as 6)
Input messages: Total 330 Updates 2 Refreshes 0
Output messages: Total 341 Updates 1 Refreshes 0
[ok] [2014-12-09 13:05:01]

```

4.3 Troubleshooting OSPF Commands

Run these CLI commands to troubleshoot OSPF related issues:

- show ospf interface brief
- show ospf neighbor detail
- show ospf neighbor brief
- show ospf overview brief
- show ospf database
- show ospf database external
- show ospf database router
- show ospf database summary
- show ospf database instance 1

4.3.1 OSPF Command Examples

This section contains examples of CLI commands, that are used to troubleshoot OSPF related issues.

4.3.1.1 CLI Command to view OSPF Interfaces

```

versa@rams-test2-cli> show ospf interface brief
Codes for operation state (State):
BDR - backup designated router; DR - designated router;
ODR - other designated router; PTP - point to point
dwn - down; lbk - loopback; wtg - waiting
Interface      Area      DR ID      BDR ID      State
-----
vni-0/0        0.0.0.0    192.168.201.15  192.168.201.14  BDR
vni-0/1        0.0.0.0    192.168.202.15  192.168.202.14  BDR
vni-0/2        0.0.0.1    192.168.203.8   192.168.203.7   BDR
vni-0/3        0.0.0.1    192.168.204.8   192.168.204.7   BDR
[ok] [2014-12-08 10:20:22]

```

4.3.1.2 CLI Command to view OSPF Neighbors

```

versa@rams-test2-cli> show ospf neighbor detail
State codes: atmpmt - attempt, exchg - exchange, exst - exchange start,
              load - loading, 2-way - two-way, full - full
Op codes: gdown - going down, gup - going up
Intf address Interface State Neighbor ID Pri Op
-----
192.168.201.15 vni-0/0 full 192.168.201.156 128 up
Area: 0.0.0.0
Admin state: up
Relationship state with neighbor: full
Options: 0x52; Re-transmission queue length: 0
Number of neighbor relationship state changes or error: 5
Permanence: dynamic; Hellos suppressed: no; Requested LSAs: 0
Dead timer due in: 00:00:35 (hrs:mins:secs:n/a)
Hitless restart status: not helping
Remaining hitless restart interval: none
Hitless restart result: none
192.168.202.15 vni-0/1 full 192.168.201.156 128 up
Area: 0.0.0.0
Admin state: up
Relationship state with neighbor: full
Options: 0x52; Re-transmission queue length: 0
Number of neighbor relationship state changes or error: 6
Permanence: dynamic; Hellos suppressed: no; Requested LSAs: 0
Dead timer due in: 00:00:34 (hrs:mins:secs:n/a)
Hitless restart status: not helping
Remaining hitless restart interval: none
Hitless restart result: none
192.168.203.8 vni-0/2 full 192.168.203.8 1 up
Area: 0.0.0.1
Admin state: up
Relationship state with neighbor: full
Options: 0x42; Re-transmission queue length: 0
Number of neighbor relationship state changes or error: 6
Permanence: dynamic; Hellos suppressed: no; Requested LSAs: 0
Dead timer due in: 00:00:32 (hrs:mins:secs:n/a)
Hitless restart status: not helping
Remaining hitless restart interval: none
Hitless restart result: none
192.168.204.8 vni-0/3 full 192.168.203.8 1 up
Area: 0.0.0.1
Admin state: up
Relationship state with neighbor: full
Options: 0x42; Re-transmission queue length: 0
Number of neighbor relationship state changes or error: 6
Permanence: dynamic; Hellos suppressed: no; Requested LSAs: 0
Dead timer due in: 00:00:32 (hrs:mins:secs:n/a)
Hitless restart status: not helping
[ok][2014-12-08 10:21:09]

```

Example

```

versa@rams-test2-cli> show ospf neighbor brief
State codes: atmp - attempt, exchg - exchange, exst - exchange start,
              load - loading, 2-way - two-way, full - full
Op codes: gdown - going down, gup - going up
Intf address Interface State Neighbor ID Pri Op
-----
192.168.201.15 vni-0/0 full 192.168.201.156 128 up
192.168.202.15 vni-0/1 full 192.168.201.156 128 up
192.168.203.8 vni-0/2 full 192.168.203.8 1 up
192.168.204.8 vni-0/3 full 192.168.203.8 1 up
[ok] [2014-12-08 10:21:15]

```

4.3.1.3 CLI Command to view OSPF Protocol Overview

```

versa@rams-test2-cli> show ospf overview brief
Router ID    Version Admin Op status Instance
-----
192.168.201.155 2 en up 1
[ok] [2014-12-08 10:21:38]
versa@rams-test2-cli> show ospf database ?
Possible completions:
area - OSPF area ID
area-opq - Opaque area-scope link-state advertisements
asbr - Summary AS boundary router link-state advertisements
external - External link-state advertisements
instance - OSPF instance ID
network - Network link-state advertisements
nssa - Not-so-stubby area link-state advertisements
router - Router link-state advertisements
summary - Summary link-state advertisements

```

4.3.1.4 CLI Command to view OSPF LSA Database

```

versa@rams-test2-cli> show ospf database
OSPF Router with ID (192.168.201.155) Instance ID (1)
Area 0.0.0.0;
Type          LSA ID          ADV Router    Sequence Age(secs)  Checksum
-----
Router        192.168.201.155    192.168.201.155  0x80000004      83      0x0000CEC7
Router        192.168.201.156    192.168.201.156  0x80000369      89      0x00002EDC
Network       192.168.201.15     192.168.201.156  0x80000001      89      0x00002267
Network       192.168.202.15     192.168.201.156  0x80000001      97      0x00001771
Summary       192.168.203.0      192.168.201.155  0x80000001      92      0x0000CB8C
Summary       192.168.204.0      192.168.201.155  0x80000001      92      0x0000C096
AS-Summary    192.168.203.8      192.168.201.155  0x80000001      86      0x00006DE1
A-Opaque      1.0.0.1            192.168.201.155  0x80000001      96      0x00000738
Area 0.0.0.1;
Type          LSA ID          ADV Router    Sequence Age(secs)  Checksum
-----
Router        192.168.201.155    192.168.201.155  0x80000003      91      0x00005F4C
Router        192.168.203.8      192.168.203.8    0x8000185C      92      0x00009EBD
Network       192.168.203.8      192.168.203.8    0x80000001      97      0x00009F34
Network       192.168.204.8      192.168.203.8    0x80000001      97      0x0000943E
Summary       192.168.201.0      192.168.201.155  0x80000001      92      0x0000E178
Summary       192.168.202.0      192.168.201.155  0x80000001      92      0x0000D682
AS-Summary    192.168.201.156    192.168.201.155  0x80000001      86      0x0000B507
A-Opaque      1.0.0.1            192.168.201.155  0x80000001      96      0x00000738
A-Opaque      1.0.0.1            192.168.203.8    0x800000F0      1102    0x00006D76
[ok] [2014-12-08 10:21:46]

```

Example

```

versa@rams-test2-cli> show ospf database area 0.0.0.0
OSPF Router with ID (192.168.201.155) Instance ID (1)
Area 0.0.0.0;
Type          LSA ID          ADV Router    Sequence Age(secs)  Checksum
-----
Router        192.168.201.155    192.168.201.155  0x80000004      96      0x0000CEC7
Router        192.168.201.156    192.168.201.156  0x80000369      102     0x00002EDC
Network       192.168.201.15     192.168.201.156  0x80000001      102     0x00002267
Network       192.168.202.15     192.168.201.156  0x80000001      110     0x00001771
Summary       192.168.203.0      192.168.201.155  0x80000001      105     0x0000CB8C
Summary       192.168.204.0      192.168.201.155  0x80000001      105     0x0000C096
AS-Summary    192.168.203.8      192.168.201.155  0x80000001      99      0x00006DE1
A-Opaque      1.0.0.1            192.168.201.155  0x80000001      109     0x00000738
[ok] [2014-12-08 10:21:59]

```

Example

```

versa@rams-test2-cli> show ospf database external
OSPF Router with ID (192.168.201.155) Instance ID (1)

```

Type	LSA ID	ADV Router	Sequence	Age(secs)	Checksum
External	0.0.0.0	192.168.201.156	0x80000094	902	0x00009DB1
External	1.2.0.0	192.168.201.156	0x80000094	152	0x000078D3
External	1.2.3.0	192.168.201.156	0x80000093	2777	0x000059F0
External	1.10.11.12	192.168.201.156	0x80000093	2027	0x00002806
External	9.10.11.12	192.168.201.156	0x80000093	1653	0x0000BF66
External	9.10.12.0	192.168.201.156	0x80000093	1278	0x00002D04
External	128.0.0.0	192.168.201.156	0x80000052	2403	0x00001D73

```

[ok] [2014-12-08 10:22:11]

```

Example

```

versa@rams-test2-cli> show ospf database router
OSPF Router with ID (192.168.201.155) Instance ID (1)
Area 0.0.0.0;

```

Type	LSA ID	ADV Router	Sequence	Age(secs)	Checksum
Router	192.168.201.155	192.168.201.155	0x80000004	125	0x0000CEC7
Router	192.168.201.156	192.168.201.156	0x80000369	131	0x00002EDC

```

Area 0.0.0.1;

```

Type	LSA ID	ADV Router	Sequence	Age(secs)	Checksum
Router	192.168.201.155	192.168.201.155	0x80000003	133	0x00005F4C
Router	192.168.203.8	192.168.203.8	0x8000185C	134	0x00009EBD

```

[ok] [2014-12-08 10:22:28]

```

Example

```

versa@rams-test2-cli> show ospf database summary
OSPF Router with ID (192.168.201.155) Instance ID (1)
Area 0.0.0.0;

```

Type	LSA ID	ADV Router	Sequence	Age(secs)	Checksum
Summary	192.168.203.0	192.168.201.155	0x80000001	137	0x0000CB8C
Summary	192.168.204.0	192.168.201.155	0x80000001	137	0x0000C096

```

Area 0.0.0.1;

```

Type	LSA ID	ADV Router	Sequence	Age(secs)	Checksum
Summary	192.168.201.0	192.168.201.155	0x80000001	137	0x0000E178
Summary	192.168.202.0	192.168.201.155	0x80000001	137	0x0000D682

```

[ok] [2014-12-08 10:22:31]

```

Example

```

versa@rams-test2-cli> show ospf database instance 1
OSPF Router with ID (192.168.201.155) Instance ID (1)
Area 0.0.0.0;

```

Type	LSA ID	ADV Router	Sequence	Age (secs)	Checksum
Router	192.168.201.155	192.168.201.155	0x80000004	142	0x0000CEC7
Router	192.168.201.156	192.168.201.156	0x80000369	148	0x00002EDC
Network	192.168.201.15	192.168.201.156	0x80000001	148	0x00002267
Network	192.168.202.15	192.168.201.156	0x80000001	156	0x00001771
Summary	192.168.203.0	192.168.201.155	0x80000001	151	0x0000CB8C
Summary	192.168.204.0	192.168.201.155	0x80000001	151	0x0000C096
AS-Summary	192.168.203.8	192.168.201.155	0x80000001	145	0x00006DE1
A-Opaque	1.0.0.1	192.168.201.155	0x80000001	155	0x00000738

```

Area 0.0.0.1;

```

Type	LSA ID	ADV Router	Sequence	Age (secs)	Checksum
Router	192.168.201.155	192.168.201.155	0x80000003	150	0x00005F4C
Router	192.168.203.8	192.168.203.8	0x8000185C	151	0x00009EBD
Network	192.168.203.8	192.168.203.8	0x80000001	156	0x00009F34
Network	192.168.204.8	192.168.203.8	0x80000001	156	0x0000943E
Summary	192.168.201.0	192.168.201.155	0x80000001	151	0x0000E178
Summary	192.168.202.0	192.168.201.155	0x80000001	151	0x0000D682
AS-Summary	192.168.201.156	192.168.201.155	0x80000001	145	0x0000B507
A-Opaque	1.0.0.1	192.168.201.155	0x80000001	155	0x00000738
A-Opaque	1.0.0.1	192.168.203.8	0x800000F0	1161	0x00006D76

4.4 Inspecting Routing State in vstated Daemon

Run these CLI commands to inspect routing state in vstated daemon:

- show
- show route-table
- show route-table routing-instance global
- show route
- show route routing-instance global
- show route destination <specific route>
- show route routing-instance global destination <specific route>

4.4.1 Routing State Command Examples

This section contains examples of CLI commands, that are used to troubleshoot routing state in vstated daemon related issues.

4.4.1.1 Example 1

```
vstated> show
  route          show route [routing-instance instance-name] [destination prefix]
  route-table    show route-table [routing-instance instance-name]
```

CLI Command to list all the VRFs and their corresponding routing table index

```
vstated> show route-table
```

Routing Instance	RTT Index
global	0
fabric	255

CLI Command to list a specific VRF and its corresponding routing table index

```
vstated> show route-table routing-instance global
```

Routing Instance	RTT Index
global	0

4.4.1.2 CLI Command to list all Routes of all Routing-Instances

```
vstated> show route
```

Route Table **for** : global

Destination/Mask	NextHop	IfIndex
0.0.0.0/0	192.168.201.15	257
	192.168.202.15	258
128.0.0.0/1	192.168.201.15	257
1.2.0.0/16	192.168.201.15	257
	200.15.1.0/24	192.168.202.14
	1.2.3.0/24	0.0.0.0
9.10.12.0/24	192.168.201.15	257
8.8.8.8/30	192.168.201.15	257
1.10.11.12/32	192.168.201.15	257
9.10.11.12/32	192.168.201.15	257

Route Table **for** : fabric

Destination/Mask	NextHop	IfIndex
------------------	---------	---------

4.4.1.3 CLI Command to List all Routes of a Specific Routing-Instance

```
vstated> show route routing-instance global
Route Table for : global
+-----+-----+-----+
| Destination/Mask | NextHop | IfIndex |
+-----+-----+-----+
| 0.0.0.0/0 | 192.168.201.15 | 257 |
| | 192.168.202.15 | 258 |
| 128.0.0.0/1 | 192.168.201.15 | 257 |
| 1.2.0.0/16 | 192.168.201.15 | 257 |
| 200.15.1.0/24 | 192.168.202.14 | 258 |
| 1.2.3.0/24 | 0.0.0.0 | 0 |
| 9.10.12.0/24 | 192.168.201.15 | 257 |
| 8.8.8.8/30 | 192.168.201.15 | 257 |
| 1.10.11.12/32 | 192.168.201.15 | 257 |
| 9.10.11.12/32 | 192.168.201.15 | 257 |
+-----+-----+-----+
```

4.4.1.4 CLI Command to List a Specific Route of all Routing-Instances

```
vstated> show route destination 8.8.8.8/30
Route Table for : global
+-----+-----+-----+
| Destination/Mask | NextHop | IfIndex |
+-----+-----+-----+
| 8.8.8.8/30 | 192.168.201.15 | 257 |
+-----+-----+-----+
Route Table for : fabric
+-----+-----+-----+
| Destination/Mask | NextHop | IfIndex |
+-----+-----+-----+
+-----+-----+-----+
```

4.4.1.5 CLI Command to List a Specific Route of a Specific Routing-Instance

```
vstated> show route routing-instance global destination 8.8.8.8/30
Route Table for : global
+-----+-----+-----+
| Destination/Mask | NextHop | IfIndex |
+-----+-----+-----+
| 8.8.8.8/30 | 192.168.201.15 | 257 |
+-----+-----+-----+
```

4.4.2 Inspecting Routing State in vsmd Daemon

Run these CLI commands to inspect routing state in vsmd daemon:

- `show vsm route-table`
- `show vsm route-table name global`
- `show vnet route table 0`

4.4.2.1 Routing State Command Examples

This section contains examples of CLI commands that are used to troubleshoot routing state in vsmd daemon.

CLI Command to list the name and routing-table index of all routing-instances

```
vsm-vcsn> show vsm route-table
Name                               table-id
fabric                             255
global                             0
CLI Command to list the name and routing-table index of a specific routing-instance
vsm-vcsn> show vsm route-table name global
Name                               table-id
global                             0
```

CLI Command to list all the routes of a specific routing table

```
vsm-vcn> show vnet route table 0
```

Routing tables

Internet:

Destination	Gateway	Flags	Refs	Use	Mtu	Netif	Expire
default	192.168.201.15	UG	0	0		1500	vni-0/0 =>
default	192.168.202.15	UG	0	0		1500	vni-0/1
1.2.0.0/16	192.168.201.15	UG	0	0		1500	vni-0/0
1.2.3.0/24	127.0.0.125	UB	0	0		16384	lo0
1.10.11.12/32	192.168.201.15	UG	0	0		1500	vni-0/0
8.8.8.8/30	192.168.201.15	UG	0	0		1500	vni-0/0
9.10.11.12/32	192.168.201.15	UG	0	0		1500	vni-0/0
9.10.12.0/24	192.168.201.15	UG	0	0		1500	vni-0/0
127.0.0.125	link#1	UH	0	0		16384	lo0
128.0.0.0/1	192.168.201.15	UG	0	0		1500	vni-0/0
192.168.201.0/24	link#257	U	0	0		1500	vni-0/0
192.168.201.14	link#257	UHS	1	0		16384	lo0
192.168.202.0/24	link#258	U	0	0		1500	vni-0/1
192.168.202.14	link#258	UHS	1	0		16384	lo0
200.15.1.0/24	192.168.202.14	UG	0	0		16384	lo0

Internet6:

Destination	Gateway	Flags	Refs	Use	Mtu	Netif	Expire
::1	link#1	UH	0	0	16384	lo0	
fe80::/64	link#1	U	0	0	16384	lo0	
fe80::1	link#1	UHS	0	0	16384	lo0	
fe80::/64	link#257	U	0	0	1500		vni-0/0
fe80::5054:ff:fec0:dd9e	link#257	UHS	0	0	16384	lo0	=>
fe80::5054:ff:fec0:dd9e	link#257	UHS	0	0	16384	lo0	
fe80::/64	link#258	U	0	0	1500		vni-0/1
fe80::5054:ff:feel:ebbf	link#258	UHS	0	0	16384	lo0	=>
fe80::5054:ff:feel:ebbf	link#258	UHS	0	0	16384	lo0	
ff01::/32	::1	U	0	0	16384	lo0	
ff01::/32	fe80::5054:ff:fec0:dd9e	U	0	0	1500		vni-0/0
ff01::/32	fe80::5054:ff:feel:ebbf	U	0	0	1500		vni-0/1
ff02::/32	::1	U	0	0	16384	lo0	
ff02::/32	fe80::5054:ff:fec0:dd9e	U	0	0	1500		vni-0/0
ff02::/32	fe80::5054:ff:feel:ebbf	U	0	0	1500		vni-0/1

5 Troubleshooting SD-WAN Datapath

This section covers:

5.1 Inspecting vsmControl Plane State

Follow these steps to debug datapath issues during stage three of SD-WAN zero touch provisioning:

1. Load and commit the branch configuration in SteelConnect EX Director using netconf.
2. Create VLAN and esp ptvi's from configuration depending on whether controller and hub are configured.
3. Run the `show vsm route-table tunnel-local-site name RT_provider` CLI command on vsm control plane to check the local site objects.

Example

```
vsm-vcsn0> show vsm route-table tunnel-local-site name RT_provider
```

```

Name                               table-id
RT_provider                         8
Local site: 0
  Neighbour IP                     : 20.21.20.3
  Site type                         : SD-WAN
  Site Name                        : br101
  Branch ID                        : 101
  Tenant ID                        : 1
  Neighbour mgmt VRF id            : 8
  Neighbour num transport IPs      : 2
    WAN lcl vrf-id                 : 10
    WAN lcl link ifindex           : 1058
    WAN lcl link name              : vni-0/1.0
    WAN lcl circuit name           : wan-link1
    WAN lcl link id : 1
    WAN lcl link shaping rate : 0
    WAN lcl link address           : 192.168.101.3
    WAN rmt transport domain: 1
    WAN lcl vrf-id                 : 10
    WAN lcl link ifindex           : 1060
    WAN lcl link name              : vni-0/2.0
    WAN lcl circuit name           : wan-link2
    WAN lcl link id : 2
    WAN lcl link shaping rate : 0
    WAN lcl link address           : 192.168.101.103
    WAN rmt transport domain: 1
Local site: 1
  Neighbour IP                     : 10.11.10.2
  Site type                         : SD-WAN
  Site Name                        : br101
  Branch ID                        : 101
  Tenant ID                        : 1
  Neighbour mgmt VRF id            : 8
  Neighbour num transport IPs      : 2
    WAN lcl vrf-id                 : 10
    WAN lcl link ifindex           : 1058
    WAN lcl link name              : vni-0/1.0
    WAN lcl circuit name           : wan-link1
    WAN lcl link id : 1
    WAN lcl link shaping rate : 0
    WAN lcl link address           : 192.168.101.3
    WAN rmt transport domain: 1
    WAN lcl vrf-id                 : 10
    WAN lcl link ifindex           : 1060
    WAN lcl link name              : vni-0/2.0
    WAN lcl circuit name           : wan-link2
    WAN lcl link id : 2
    WAN lcl link shaping rate : 0
    WAN lcl link address           : 192.168.101.103
    WAN rmt transport domain: 1

```

4. Run the `show vsm tunnel-remote-endpoint RT_provider` CLI command on vsm control plane to check the remote site objects either learnt via BGP/configuration.

```

vsm-vcsn0> show vsm tunnel-remote-endpoint RT_provider
Neighbour Endpoint: 0      Neighbour IP      : 10.11.12.2
  Site type              : SD-WAN
  Site ID                : 66:00
  Site Name              : br102
  Branch ID              : 102
  Tenant ID              : 1
  Neighbour mgmt VRF id   : 8
  Neighbour num transport IPs : 2
  Neighbour SA            : 66:00:01:00:6c:90:56:6b:f4:fd:d0:3b:c0:35:
1f:a5:d9:e0:f9:c4:2c:47:95:ed:5d:d7:63:3f:1c:a8:de:8b:bc:7d:35:ee:7f:45:92:21:cb:97:5f:96:b1:87:
1a:bf:2d:b1
  Neighbour SA len       : 68
  Neighbour SPI          : 0x00010066
  Neighbour Ptvi Intf    : ptvi-0/70
  WAN rmt circuit name   : wan-link1
  WAN rmt link id        : 1
  WAN rmt link shaping rate : 0
  WAN rmt link address   : 192.168.101.4
  WAN rmt link nat port  : 4790
  WAN rmt transport domain : 1
  WAN rmt circuit name   : wan-link2
  WAN rmt link id        : 2
  WAN rmt link shaping rate : 0
  WAN rmt link address   : 192.168.101.104
  WAN rmt link nat port  : 4790
  WAN rmt transport domain : 1
Neighbour Endpoint: 1
  Neighbour IP          : 20.21.22.3
  Site type              : SD-WAN
  Site ID                : 66:00
  Site Name              : br102
  Branch ID              : 102
  Tenant ID              : 1
  Neighbour mgmt VRF id   : 8
  Neighbour num transport IPs : 2
  Neighbour SA            : 66:00:01:00:6c:90:56:6b:f4:fd:d0:3b:c0:35:
1f:a5:d9:e0:f9:c4:2c:47:95:ed:5d:d7:63:3f:1c:a8:de:8b:bc:7d:35:ee:7f:45:92:21:cb:97:5f:96:b1:87:
1a:bf:2d:b1
  Neighbour SA len       : 68
  Neighbour SPI          : 0x00010066
  Neighbour Ptvi Intf    : ptvi-0/69
  WAN rmt circuit name   : wan-link1
  WAN rmt link id        : 1
  WAN rmt link shaping rate : 0
  WAN rmt link address   : 192.168.101.4
  WAN rmt link nat port  : 4790
  WAN rmt transport domain : 1
  WAN rmt circuit name   : wan-link2
  WAN rmt link id        : 2
  WAN rmt link shaping rate : 0
  WAN rmt link address   : 192.168.101.104
  WAN rmt link nat port  : 4790

```

```

WAN rmt transport domain      : 1
Neighbour Endpoint: 2
Neighbour IP                  : 10.11.11.3
Site type                     : SD-WAN
Site ID                       : 01:00
Site Name                     : cntrlr1
Branch ID                     : 1
Tenant ID                     : 1
Neighbour mgmt VRF id        : 8
Neighbour num transport IPs   : 2
Neighbour SA                   :
Neighbour SA len              : 0
Neighbour SPI                  : 0x00000000
Neighbour Ptvi Intf           : ptvi-0/54
WAN rmt circuit name          : wan-link1
WAN rmt link id               : 1
WAN rmt link shaping rate     : 0
WAN rmt link address          : 192.168.101.2
WAN rmt link nat port         : 4790
WAN rmt transport domain      : 1
WAN rmt circuit name          : wan-link2
WAN rmt link id               : 2
WAN rmt link shaping rate     : 0
WAN rmt link address          : 192.168.101.102
WAN rmt link nat port         : 4790
WAN rmt transport domain      : 1
Neighbour Endpoint: 3
Neighbour IP                  : 20.21.21.3
Site type                     : SD-WAN
Site ID                       : 01:00
Site Name                     : cntrlr1
Branch ID                     : 1
Tenant ID                     : 1
Neighbour mgmt VRF id        : 8
Neighbour num transport IPs   : 2
Neighbour SA                   :
Neighbour SA len              : 0
Neighbour SPI                  : 0x00000000
Neighbour Ptvi Intf           : ptvi2
WAN rmt circuit name          : wan-link1
WAN rmt link id               : 1
WAN rmt link shaping rate     : 0
WAN rmt link address          : 192.168.101.2
WAN rmt link nat port         : 4790
WAN rmt transport domain      : 1
WAN rmt circuit name          : wan-link2
WAN rmt link id               : 2
WAN rmt link shaping rate     : 0
WAN rmt link address          : 192.168.101.102
WAN rmt link nat port         : 4790
WAN rmt transport domain      : 1

```

5.2 Inspecting vsmData Plane State

Follow these steps to ensure the existence of routes to all remote branches, for connectivity:

Without the establishment of IKE and IPSec tunnel, ptvi-esp towards controller will be in down state and route to controller will not be present.

1. Run the `show vnet route summary` CLI command to know the routes in the core FIB and customer FIB for a given tenant.

Example

```
vsm-vcsn0> show vnet route summary
Id  Routing Instance  Count
0   Default          5
8   RT_provider       3
10  grt-vrf           8
12  mgmt1             5
14  mgmt11            5
16  rt1               5
18  rt11              5
1023 fabric         1
Total: 37
vsm-vcsn0> show vnet route table 12
Routing tables
Internet:
Destination      Gateway           GW Idx Flags Refs   Use Mtu Netif Expire Labels Next-FIB
10.10.11.3/32    10.10.11.3       1041 UG      0    0  1400 ptvi-0/56 n/a 65  n/a
10.10.12.2/32    10.10.12.2       1062 UG      0    0  1400 ptvi-0/64 n/a 65  n/a
20.20.21.3/32    20.20.21.3       1044 UG      0    0  1400 ptvi1 n/a 65  n/a
20.20.22.3/32    20.20.22.3       1061 UG      0    0  1400 ptvi-0/63 n/a 65  n/a
127.0.0.125     link#13           13 UHO     0    0  16384 lo12 n/a n/a 0
vsm-vcsn0> show vnet route table 16
Routing tables
Internet:
Destination      Gateway           GW Idx Flags Refs   Use Mtu Netif Expire Labels Next-FIB
127.0.0.125     link#17           17 UHO     0    0  16384 lo16 n/a n/a 0
192.168.150.0/24 link#1055         1055 U      0    0  1500 vni-0/0.0 n/a n/a 0
192.168.150.3    link#1055         0 UHSO    0    0  16384 lo16 n/a n/a n/a
192.168.150.255 link#1055         1055 UHb   0    0  1500 vni-0/0.0 n/a n/a 0
192.168.151.0/24 20.20.22.3       1061 UG      0    0  1400 ptvi-0/63 n/a 25472 n/a
```

2. Run the `show vsm mp|s-label-table` CLI command to check the Incoming Label table in the data path to ensure the correct distribution of labels.

Example

```
vsm-vcsn0> show vsm mpls-label-table
```

```
MPLS Label Table:
```

```
Number of label entries: 6
```

Label	NH type	FIB	Core FIB	Proto
75	VRF-table-label	14	14	IPv4
17	Next-proto	0	0	Ether
25472	VRF-table-label	16	12	IPv4
66	VRF-table-label	8	8	IPv4
25600	VRF-table-label	18	14	IPv4
65	VRF-table-label	12	12	IPv4

- Run the `show vsf tunnel branch-table local` CLI command to check if the branch table is programmed correctly and all the configured local site info and learnt remote branch info is present.

```
vsm-vcsn0> show vsf tunnel branch-table local
Control thread
=====
<Br ID,C-FIB> | Branch Name | CT PTVI | ET PTVI | Tnt ID |
=====
< 101, 8> | br101 | 0 | 0 | 1 |
< 101, 12> | br101 | 0 | 0 | 2 |
< 101, 14> | br101 | 0 | 0 | 3 |
Example
vsm-vcsn0> show vsf tunnel branch-table
Legend:
CT - Clear Text
ET - Encrypted Text
C-FIB - Core-facing FIB
Control thread
=====
<Br ID,C-FIB> | Branch Name | CT PTVI | ET PTVI | Tnt ID |
=====
< 1, 8> | cntrlr1 | 1030 | 1031 | 1 |
< 1, 14> | cntrlr1 | 1048 | 1051 | 3 |
< 102, 12> | br102 | 1062 | 1061 | 2 |
< 102, 14> | br102 | 1064 | 1063 | 3 |
< 1, 12> | cntrlr1 | 1041 | 1044 | 2 |
Worker thread index 0
=====
<Br ID,C-FIB> | Branch Name | CT PTVI | ET PTVI | Tnt ID |
=====
< 1, 8> | cntrlr1 | 1030 | 1031 | 1 |
< 1, 14> | cntrlr1 | 1048 | 1051 | 3 |
< 102, 12> | br102 | 1062 | 1061 | 2 |
< 102, 14> | br102 | 1064 | 1063 | 3 |
< 1, 12> | cntrlr1 | 1041 | 1044 | 2 |
Worker thread index 1
=====
<Br ID,C-FIB> | Branch Name | CT PTVI | ET PTVI | Tnt ID |
=====
< 1, 8> | cntrlr1 | 1030 | 1031 | 1 |
< 1, 14> | cntrlr1 | 1048 | 1051 | 3 |
< 102, 12> | br102 | 1062 | 1061 | 2 |
< 102, 14> | br102 | 1064 | 1063 | 3 |
< 1, 12> | cntrlr1 | 1041 | 1044 | 2 |
```

- Run the `show vsf tunnel access-circuits 1063 brief` CLI command to check the forwarding plane state for any of the given sites and the corresponding network paths between the local and remote site. Pick the Cleartext/Ciphertext *ptviifindex* from the table output of **Step 3**.

Example

```
vsm-vcsn0> show vsf tunnel access-circuits 1063 brief
Encap chain info (in order of imposition):
  Number of Encaps : 4
  Encap 0          : NSH
  Encap 1          : MPLS-over-GRE
  Encap 2          : IPSec-ESP
  Encap 3          : VXLAN
  Transport encap  : IP + UDP
Tunnel check for branch/ackt/route required : TRUE
SPI Ctxt: 7f79b9f3e600
Out SPI : 0x51db000b
In SPI  : 0x001b0066
Legend:
  ED: Endpoint Dependent NAT enabled
  I: SLA in INIT state
  U: SLA in UP state
  D: SLA in DOWN state
Access Circuits to Neighbor IP: 20.20.220.
Control Thread:
```

ID	Src IP	Dest IP	NAT-P	VRF	Cap	IfIndex	Pipe	RT	Gen
17(1,1)	192.168.101.3	192.168.101.4	4790	10	(P,E)	1058	65535		
0	I								
18(1,2)	192.168.101.3	192.168.101.104	4790	10	(P,E)	1058	65535		
0	I								
33(2,1)	192.168.101.103	192.168.101.4	4790	10	(P,E)	1060	65535		
0	I								
34(2,2)	192.168.101.103	192.168.101.104	4790	10	(P,E)	1060	65535		
0	I								

- Run the `show vsf session all detail` CLI command to check session state for transit packets from client (behind Branch1) to server (behind Branch2).

The output displays packet drops information if there are any packet drops in the session. `infra.nn`

Example

```
vsm-vcsn0> show vsf session all detail
Session ID: 2000003 (NFP), Tenant ID: 2, Owner WT: 1
Protocol - Layer-3: 102, Layer-4: 6
Src Address: 192.168.150.4, Port: 46633
Dst Address: 192.168.151.2, Port: 45789
Session Start Timestamp: 7916319
Session Last Active Timestamp: 7924808
Session Idle Timeout: 524288 Session Hard Timeout: 0
Session FDT key: 0x9E00
Session First-Packet Mask: 0 Session Close Mask: 0
Session Flags: 0x8088
Session Egress-VRFs: [ 16, 16 ]
##Session Provider Zone: [0]
##Session filter gen-num: [22], my-ip-tbl gen-num: [1064] route-gen-num: [33]
##Session WAN Access circuit : [ Rx: 0x00 - Tx: 0x11 Encap: 0x0 ]
##Session NHIDs: [ 0, 4 ]
Forward Flow: (VRF ID: 16)
Service Chain: 2 4 19 27
Pkt-In Interest Mask: 0x8
Pkt-Out Interest Mask: 0
Data Interest Mask: 0x8
Total Packets Count: 156006, Dropped Packets Count: 0
Total Bytes Count: 208644684, Dropped Bytes Count: 0
NFP-offload:N[N], RT gen:33[33], MTU:1400[1500], NH-Ready:N[Y] Src-intf route-lkup: 0
Ingress Interface: vni-0/0.0, Egress Interface: ptvi-0/71
QOS Gen ID: 0, Shaping TC/Q: 3/0, Shaping Color: 0
FC/PLP: 12/0
Reverse Flow: (VRF ID: 16)
Service Chain: 2 4 27 19
Pkt-In Interest Mask: 0x4
Pkt-Out Interest Mask: 0
Data Interest Mask: 0x4
Total Packets Count: 10920, Dropped Packets Count: 0
Total Bytes Count: 567848, Dropped Bytes Count: 0
NFP-offload:N[N], RT gen:33[33], MTU:1500[1400], NH-Ready:Y[N] Src-intf route-lkup: 0
Ingress Interface: ptvi-0/71, Egress Interface: vni-0/0.0
QOS Gen ID: 0, Shaping TC/Q: 3/0, Shaping Color: 0
FC/PLP: 12/0
```

5.3 Inspecting vsm Data Plane Statistics

Follow these steps to check any drops in the datapath:

1. Run the `show vsm statistics port` CLI command. Note that IP multicast is available in Releases 20.2 and later.

Example

```

vsm-vcsn0> show vsm statistics port
Interface   : vni-0/0 (port: 0)
Successfully received packets      : 3006
Successfully transmitted packets   : 17
Successfully received bytes        : 0
Successfully transmitted bytes     : 0
Erroneous received packets        : 0
Failed transmitted packets        : 0
Multicast received packets        : 0
RX mbuf allocation failures       : 0
Interface   : vni-0/1 (port: 1)
Successfully received packets      : 13269
Successfully transmitted packets   : 6730
Successfully received bytes        : 0
Successfully transmitted bytes     : 0
Erroneous received packets        : 0
Failed transmitted packets        : 0
Multicast received packets        : 0
RX mbuf allocation failures       : 0
vsm-vcsn0> show vsm statistics datapath
# Forwarded - Filter Lookup        : 8714
# Sent - ARP to CT                 : 3
# Passed - Host-bound rate limit   : 8714
# Injected - Into VUNET            : 9180
# Packets Dropped - Tuple Extract Failure : 105
# Packets Dropped - IPv6           : 105
vsm-vcsn0> show vsm statistics dropped
DPDK ERROR STATISTICS
~~~~~
DATAPATH ERROR STATISTICS
~~~~~
# Packets Dropped - Tuple Extract Failure : 105
# Packets Dropped - IPv6                 : 105
THRM ERROR STATISTICS
~~~~~
Poller PID : 32183
# Drop Packets TX                        : 4
NFP ERROR STATISTICS
~~~~~

VSF ERROR STATISTICS
~~~~~
# Packets Dropped - Tuple extract failure : 14
# Number of calls to icmp_error          : 41
VUNET ERROR STATISTICS
~~~~~
# VN_MOD_IP_ERR_ARP_NO_RESOLV_CNT       : 4
# VN_MOD_IP_ERR_NO_ROUTE_CNT            : 41
# VN_MOD_ETH_ERR_BAD_TYPE_CNT           : 9141

```

2. Run the `show vsm statistics infmgr` CLI command to check statics to/from infmgr to vsm control thread.

Example

```
vsm-vcsn0> show vsm statistics infmgr
```

```
-----
| Inf          Disc    Phy-State Tap-Tx      Tap-Rx  Tun-Tx  Stats-Req  Stats-Resp Stats-Clr
|
-----
| vni-0/0      1       2         16          0        0        0          0          0      |
| vni-0/1      1       2          8          2        0        0          0          0      |
| vni-0/2      1       2          8          1        0        0          0          0      |
-----
```

```
Control packet stats
```

```
TAP TX (to infmgr) packets:    1483
SD-WAN VBP TX (to infmgr) packets: 7540
TAP RX (from infmgr) packets:   32
TUN RX (from infmgr) packets:  1509
SD-WAN VBP RX (from infmgr) packets: 7546
```

```
Misc Stats:
```

```
VSN Slot IP Msg : 0
```

```
Error Stats:
```

```
Send failed due to socketerror :      0
Failed to send Ctrl pkt to infmgr:      0
Failed to send TUN Ctrl pkt to WT (Bad FIB): 0
Failed to send VBP Ctrl pkt to WT (Bad FIB): 0
```

```
vsm-vcsn0> show vsf nfp stats
```

```
# Max allowed sessions          : 1000000
# Session Active                 : 1
# Session Created                : 1
# Session Closed                 : 0
# Session Active (NAT)           : 0
# Session Created (NAT)          : 0
# Session Closed (NAT)           : 0
# Flows Offloaded                : 0
# VS_NFP_S_ETHER_IN              : 245154
# VS_NFP_S_IPV4_IN_PRE           : 245154
# VS_NFP_S_IPV4_IN_POST          : 245154
# VS_NFP_S_IPV4_OUT_POST         : 245156
# VS_NFP_S_ETHER_OUT             : 16232
# Interface transmit count       : 16232
# Sent to tvi interface         : 228922
```

```
vsm-vcsn0> show vsf tunnel stats
```

```
-----
Tunnel encap stats
-----
Tunnel Encap Processing successful          10616
Tunnel Encap Processing dropped:            9
Tunnel IP-UDP transport encap forwarded:    10616
Tunnel MPLSoGRE encap forwarded:           10616
Tunnel VXLAN-GPE encap forwarded:          10616
Tunnel IPSec-ESP encap forwarded:          10605
```

```

Tunnel IPSec-ESP encap scheduled:          10605
Tunnel Encap Pre-processing dropped:        9
Tunnel Encap Send completed                10616
Tunnel Switching Gateway dropped:          9
-----
                        Tunnel decap stats
-----
Tunnel Decap Processing successful          10608
Tunnel IP transport decap forwarded:        10608
Tunnel MPLSoGRE decap forwarded:           10608
Tunnel VXLAN-GPE decap forwarded:          10608
Tunnel IPSec-ESP decap forwarded:           10602
Tunnel IPSec-ESP decap scheduled:           10602
Tunnel Decap inner packet reinjected:      10608
vsm-vcsn0> show vsf tunnel access-circuits 1063 detail
Encap chain info (in order of imposition):
    Number of Encaps : 4
    Encap 0          : NSH
    Encap 1          : MPLS-over-GRE
    Encap 2          : IPSec-ESP
    Encap 3          : VXLAN
    Transport encap  : IP + UDP
Tunnel check for branch/ackt/route required : TRUE
SPI Ctxt: 7f79b9f3e600
Out SPI : 0x51db000b
In SPI  : 0x001b0066
Legend:
    ED: Endpoint Dependent NAT enabled
    I:  SLA in INIT state
    U:  SLA in UP state
    D:  SLA in DOWN state
Access Circuits to Neighbor IP: 20.20.220.3
Control Thread:
  ID   Src IP      Dest IP      NAT-P   VRF   Cap   IfIndex Pipe   RT Gen Flags Rx Pkts   Tx
  Pkts Rx Bytes Tx Bytes
-----
17(1,1) 192.168.101.3 192.168.101.4 4790    10   (P,E)  1058    65535    0      I
0      0          0          0
18(1,2) 192.168.101.3 192.168.101.104 4790    10   (P,E)  1058    65535    0      I
0      0          0          0
33(2,1) 192.168.101.103 192.168.101.4 4790    10   (P,E)  1060    65535    0      I
0      0          0          0
34(2,2) 192.168.101.103 192.168.101.104 4790    10   (P,E)  1060    65535    0      I
0      0          0          0
Worker Thread 0:
Default valid access-circuit id 17
  ID   Src IP      Dest IP      NAT-P   VRF   Cap   IfIndex Pipe   RT Gen Flags Rx Pkts Tx Pkts Rx
  BytesTx Bytes
-----
17(1,1) 192.168.101.3 192.168.101.4 4790    10   (P,E)  1058    65535    0      I
0      0          0          0

```

```

18(1,2) 192.168.101.3 192.168.101.104 4790 10 (P,E) 1058 65535 0 I 0
0 0 0
33(2,1) 192.168.101.103 192.168.101.4 4790 10 (P,E) 1060 65535 0 I 0
0 0 0
34(2,2) 192.168.101.103 192.168.101.104 4790 10 (P,E) 1060 65535 0 I 0
0 0 0
Worker Thread 1:
Default valid access-circuit id 17
ID Src IP Dest IP NAT-P VRF Cap IfIndex Pipe RT Gen Flags Rx Pkts Tx Pkts Rx
Bytes Tx Bytes
-----
17(1,1) 192.168.101.3 192.168.101.4 4790 10 (P,E) 1058 65535 0 I 0
0 0 0
18(1,2) 192.168.101.3 192.168.101.104 4790 10 (P,E) 1058 65535 0 I 0
0 0 0
33(2,1) 192.168.101.103 192.168.101.4 4790 10 (P,E) 1060 65535 0 I 0
0 0 0
34(2,2) 192.168.101.103 192.168.101.104 4790 10 (P,E) 1060 65535 0 I 0
0 0 0

```

- Run the `show vsf tunnel nat-info 1063` CLI command to check if the remote endpoint is behind NAT and to check the translated ip/port issue.

Example

```
vsm-vcsn0> show vsf tunnel nat-info 1063
```

```
ED - Endpoint Dependent
```

```
EIM - Endpoint Independent
```

```
Access Circuits NAT Info to Neighbor IP: 20.20.220.3
```

```
Control Thread:
```

```
-----
-----
AC | Dest IP | DPort | ED-NAT-IP | ED-NAT-Port | SLA-mask | VBP | DP |
-----
-----
17| 192.168.101.4 | 4790 | 192.168.101.4 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
18| 192.168.101.104 | 4790 | 192.168.101.104 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
33| 192.168.101.4 | 4790 | 192.168.101.4 | 4790 | 0x0000 | 0 | 0 |
0 | 0 |
-----
-----
34| 192.168.101.104 | 4790 | 192.168.101.104 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
```

```
Worker Thread: 0
```

```
-----
-----
AC | Dest IP | DPort | ED-NAT-IP | ED-NAT-Port | SLA-mask | VBP | DP |
-----
-----
17| 192.168.101.4 | 4790 | 192.168.101.4 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
18| 192.168.101.104 | 4790 | 192.168.101.104 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
33| 192.168.101.4 | 4790 | 192.168.101.4 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
34| 192.168.101.104 | 4790 | 192.168.101.104 | 4790 | 0x0000 | 0 | 0 |
0 |
-----
-----
```

```
Worker Thread: 1
```

```
-----
---
```

AC	Dest IP	DPort	ED-NAT-IP	ED-NAT-Port	SLA-mask	VBP	DP
17	192.168.101.4	4790	192.168.101.4	4790	0x0000	0	0
18	192.168.101.104	4790	192.168.101.104	4790	0x0000	0	0
33	192.168.101.4	4790	192.168.101.4	4790	0x0000	0	0
34	192.168.101.104	4790	192.168.101.104	4790	0x0000	0	0

- Run the `show vsf tunnel stun-info 1` CLI command to check the available turn relays in case the branch is behind ED-NAT box.

Example

```
vsm-vcsn0> show vsf tunnel stun-info 1
```

```
Control thread [0] :
```

```
-----
```

	Pri	[	0]	
Stun-Group		Stun-hndl		State
Default-Controller		273 [1,0x11]		Disconnected
Default-Controller		274 [1,0x12]		Disconnected
Default-Controller		289 [1,0x21]		Disconnected
Default-Controller		290 [1,0x22]		Disconnected

```
Current Active STUN Server:0
```

```
Worker thread [0] :
```

```
-----
```

	Pri	[	0]	
Stun-Group		Stun-hndl		State
Default-Controller		273 [1,0x11]		Disconnected
Default-Controller		274 [1,0x12]		Disconnected
Default-Controller		289 [1,0x21]		Disconnected
Default-Controller		290 [1,0x22]		Disconnected

```

|-----
-----
Current Active STUNServer:0
Worker thread [1] :
-----
| Pri [ 0] |

-----
-----
|          Stun-Group |          Stun-hndl          |          State          |
|-----
| Default-Controller |          273 [ 1,0x11] |Disconnected          |
|-----
| Default-Controller |          274 [ 1,0x12] |Disconnected          |
|-----
| Default-Controller |          289 [ 1,0x21] |Disconnected          |
|-----
| Default-Controller |          290 [ 1,0x22] |Disconnected          |
|-----
-----
Current Active STUN Server:0

```

6 Troubleshooting SD-WAN Branches

This section covers:

6.1 Branch Staging

This section explains the factory default configuration and staging configuration of branch devices,

6.1.1 Controller Configuration

These are the minimum configuration elements, on the controller, required for branch staging. Without these configurations, controller cannot send requisite notifications to SteelConnect EX Director for branch staging:

- Specify the tenant org belonging to the Service Provider in the **system sd-wan site provider-org <org-name>**. This provider org specifies the routing-instance that are used for branch management purposes under **orgs org <org-name> sd-wan site management-routing-instance <rtt-name>**.
- The management-routing-instance specified needs to be marked for SD-WAN-management usage with configuration: **routing-instances <rtt-name> usage-type SD-WAN-management**.
- The **provider-org system sd-wan site provider-org <org-name>** specifies the value of global-tenant-id in **orgs org <org-name> sd-wan site global-tenant-id <id>**. This need to match the global-tenant-id in factory-default configuration on the branch device.

MP-BGP should be configured in the provider org for SD-WAN deployments to deliver notifications for all relevant branch events to SteelConnect EX Director.

6.1.2 Controller Show Commands

A notification is sent by Controller to SteelConnect EX Director after a successful establishment of IPsec connection by branch device to Controller.

Run the showCLI command to verify details of the sent notification.

Example

```
admin@SD-WAN-Controller1-cli> show alarms | match branchd | match br101
branchd          SD-WAN-branch-connect      2016-03-30T15:01:53   chassis-id LR201510017703, branch-id 64,
branch br101
branchd          SD-WAN-branch-connect      2016-03-30T15:06:57   chassis-id LR201510017703, branch-id 65,
branch br101
branchd          SD-WAN-branch-connect      2016-03-30T15:09:48   chassis-id LR201510017703, branch-id 101,
branch br101, wan-ip 11.11.12.102, wan-ip 11.11.11.103
branchd          SD-WAN-branch-disconnect 2016-03-30T15:55:06   chassis-id LR201510017703, branch-id 101,
branch br101, wan-ip 11.11.11.103
```

6.1.3 Branch Life-cycle

These are the typical branch life-cycle:

1. Branch with site-name br101 is connected using factory-default configuration.

Example: Notification for a factory-default branch

```
branchd SD-WAN-branch-connect 2016-03-30T15:01:53 chassis-id LR201510017703, branch-id 64,
branch br101
```

If above connect notification is not seen, then either data path or IPsec connectivity from branch to controller needs to be debugged. Refer to [Stage-3 Debugging on Branch](#)(see page 68).
In response to the branch connect notification, SteelConnect EX Director pushes staging configuration to the branch device and requests reboot of branch device.

2. After rebooting in staging configuration, branch is connected to controller.

Example: Notification for a staged branch

```
branchd SD-WAN-branch-connect 2016-03-30T15:06:57 chassis-id LR201510017703, branch-id 65, branch
br101
```

If above connect notification is not seen, then either Datapath or IPsec connectivity from branch to controller needs to be debugged. Refer to [Stage-3 Debugging on Branch](#)(see page 68).

3. After rebooting in final configuration, branch is connected to controller.

Example: Notification after branch staging

```
branchd SD-WAN-branch-connect 2016-03-30T15:09:48 chassis-id LR201510017703, branch-id 101, branch
br101, wan-ip 11.11.12.102, wan-ip 11.11.11.103
```

If above notification is not seen, then branch is not able to connect to controller after staging. Refer to [Stage-3 Debugging on Branch](#)(see page 68).

A branch operating with configuration after completion of staging is referred as Stage-3 branch. The stage numbering corresponds to steps involved in staging.

4. Branch is disconnected from controller. This is applicable only after the completion of staging.

Example: Notification when a Stage-3 branch that was connected to controller loses its IPsec connectivity to controller

```
branchd SD-WAN-branch-disconnect 2016-03-30T15:55:06 chassis-id LR201510017703, branch-id 101,
branch br101, wan-ip 11.11.11.103
```

Refer to [Stage-3 Debugging on Branch](#)(see page 68) for details on debugging connectivity for Stage-3 branches.

6.2 Stage Three Debugging on SD-WAN Appliance

SteelConnect EX SD-WAN appliance is designed for multi-tenancy, and is applicable even for branch site appliances. All SD-WAN show commands are at tenant level. Provider tenant is the starting point for most of the debugging as branch life cycle is managed in context of the provider tenant.

The debug commands and workflow covered in this section are applicable for all tenants. Several elements of configuration and runtime state are common for SD-WAN controller, branch and hub sites.

These are the CLI commands that provide visibility into the common state:

- Run the `show orgs org ServiceProvider sd-wan site wan-intfs` CLI command to check details of WAN interfaces used for SD-WAN uplink connectivity.

Example

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site wan-intfs
```

											MIN
	CIRCUIT	LINK		NAT		PUBLIC	LINK	SHAPING	SHAPING		
INTF NAME	NAME	ID	ENDPT IP	STATUS	PUBLIC IP	PORT	ENCRYPTION	RATE	RATE		
vni-0/0.0	Braodband1	1	192.168.211.2	unknown	0.0.0.0	4790	optional	0	0		
vni-0/1.0	Broadband2	2	192.168.212.2	unknown	0.0.0.0	4790	optional	0	0		
vni-0/2.0	MPLS	3	192.168.213.2	unknown	192.168.213.2	4790	optional	0	0		

- Run the `show orgs org ServiceProvider sd-wan site statistics vni` CLI command to check statistics for WAN interfaces used for SD-WAN uplink connectivity.

Example

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site statistics vni
```

	RX		TX		TX	RX
VNI NAME	PKTS	RX BYTES	PKTS	TX BYTES	BPS	BPS
vni-0/0.0	96780	50753136	62047	12240536	0	0
vni-0/1.0	13423	2469832	34393	7409432	0	0

- Run the `request clear statistics sd-wan vni all` CLI command to clear WAN interface statistics.

6.2.1 Low-level vty Commands

Low-level commands for in-depth debugging are available from infmgr shell. From Linux shell prompt, execute `vsh connect infmgr` to get infmgr shell.

These are the commands to fetch low level commands:

- Run the `show p2mp nbrs detail ServiceProvider all` CLI command to get low-level commands.

Example

```

infmgr> show p2mp nbrs detail ServiceProvider all
network-id 1, site-id 0x0a00, rtt-index 14, branch-id 10, site-name East-Coast-Controller-1, flags:
SELF
  site-type = SD-WAN, chassis-id East-Coast-Controller-1
  tunnel: (local: 10.10.110.2, remote: 10.10.110.2) [[ encap-outer 5, encap-inner 3, nbrtun_cfgidx
0 ]]
  tunnel: (local: 10.10.10.2, remote: 10.10.10.2) [[ encap-outer 5, encap-inner 1, nbrtun_cfgidx 0 ]]
  transport-ips:
    (local-ip 192.168.211.2, routing-instance TransportVRF, link-id 1, port 4790, seq 0, ckt-name
    Braodband1, tunnels encrypted/plaintext, nat_status:unknown transport-domain-ids [ 2 ]
    (local-ip 192.168.212.2, routing-instance TransportVRF, link-id 2, port 4790, seq 0, ckt-name
    Broadband2, tunnels encrypted/plaintext, nat_status:unknown transport-domain-ids [ 2 ]
  dynamic-endpt-info:
    (link-id 1, public-ip 0.0.0.0, public-port 4790, seq 0, shaping_rate 0, shaping_rate_min 0
    (link-id 2, public-ip 0.0.0.0, public-port 4790, seq 0, shaping_rate 0, shaping_rate_min 0
  mgmt_ip: 10.10.110.2
  cookie: 0x91037d0c
  local_conf: tenant_id 4
  ifname vni-0/0.0: (ifindex 1050, ip 192.168.211.2, link-id 1, circuit-name Braodband1, shaping_rate
0, shaping_rate_min 0, tunnels:encrypt,plaintext, IKE-link)
  SLA-cfg (fc, sla-interval, sla-log-interval, no-encrypt): (0, 0, 0, 0) (1, 0, 0, 0), (2, 0, 0, 0),
(3, 0, 0, 0), (4, 3, 300, 0), (5, 0, 0, 0), (6, 0, 0, 0), (7, 0, 0, 0),
(8, 0, 0, 0), (9, 0, 0, 0), (10, 0, 0, 0), (11, 0, 0, 0), (12, 0, 0, 0), (13, 0, 0, 0), (14, 0, 0,
0), (15, 0, 0, 0),
  ifname vni-0/1.0: (ifindex 1052, ip 192.168.212.2, link-id 2, circuit-name Broadband2, shaping_rate
0, shaping_rate_min 0, tunnels:encrypt,plaintext, IKE-link)
  SLA-cfg (fc, sla-interval, sla-log-interval, no-encrypt): (0, 0, 0, 0) (1, 0, 0, 0), (2, 0, 0, 0),
(3, 0, 0, 0), (4, 0, 0, 0), (5, 0, 0, 0), (6, 0, 0, 0), (7, 0, 0, 0),
(8, 3, 300, 0), (9, 0, 0, 0), (10, 0, 0, 0), (11, 0, 0, 0), (12, 0, 0, 0), (13, 0, 0, 0), (14, 0, 0,
0), (15, 0, 0, 0),
  ctrlr_info: branch-vnf-mgr 192.168.75.2/24 ]
  [[ source: config; state: , ipc: ]]
network-id 1, site-id 0x411f, rtt-index 14, branch-id 8001, site-name Branch1-SanFrancisco, flags:
  site-type = SD-WAN, chassis-id 001branch1
  tunnel: (local: 10.10.10.2, remote: 10.10.11.2) [[ encap-outer 5, encap-inner 1, nbrtun_cfgidx
42 ]]
  tunnel: (local: 10.10.110.2, remote: 10.10.111.2) [[ encap-outer 5, encap-inner 3, nbrtun_cfgidx
52 ]]
  transport-ips:
    (local-ip 101.101.101.1, routing-instance TransportVRF, link-id 1, port 12983, seq 1, ckt-name ,
  tunnels /, nat_status:unknown transport-domain-ids [ 1 ]
  dynamic-endpt-info:
    (link-id 1, public-ip 101.101.101.1, public-port 12983, seq 1, shaping_rate 0, shaping_rate_min 0
  mgmt_ip: 10.10.111.2
  cookie: 0xc289b9a9
  [[ source: vsmd; state: ike_complete, , ipc: add_tun, remote_obj ]]

```

- Run the `show p2mp nbrs detail ServiceProvider <Branch>` CLI command to filter the output to include only a specific node.

Example

```
infmtgr> show p2mp nbrrs detail ServiceProvider Branch1-SanFrancisco
```

6.3 Stage-3 Debugging on Controller

SD-WAN controller manages multiple branch/hub sites for one or more customer tenants. Typical debug workflow involves getting a high level view of all the sites for a particular tenant, and then drilling down into a specific site.

6.3.1 High Level Summary Commands

Run these CLI commands to get a high level view of all the sites:

- `show orgs org ServiceProvider sd-wan site summary`
- `show orgs org ServiceProvider sd-wan site brief`
- `show orgs org ServiceProvider sd-wan site history current-status`

6.3.1.1 Examples

- **Getting summary of number of sites in connected, disconnected or error state:**

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site summary
Branches in connected state      : 6
Branches in disconnected state   : 0
Branches in erroneous state     : 1
```

- **Getting brief information about all SD-WAN sites managed by the controller:**

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site brief
SITE                                MANAGEMENT      CONNECTIVITY IS
ID SITE NAME                       IP              TYPE UP TIME STATUS   CTRLR
-----
10 East-Coast-Controller-1 10.10.110.2 local  1h:3m:49s -         yes
8001 Branch1-SanFrancisco 10.10.111.2 remote 1h:3m:30s Connected no
8002 Branch2-Phoneix      10.10.112.2 remote 1h:3m:33s Connected no
8003 Hub-SaltLakeCity      10.10.113.2 remote 46m:25s Connected no
8004 Branch4-Detroit       10.10.114.2 remote 1h:3m:19s Connected no
8005 Branch5-Tampa         10.10.115.2 remote 1h:3m:20s Connected no
8006 Hub-Denver            10.10.116.2 remote 1h:3m:17s Connected no
```

- **Getting history of connectivity events for all SD-WAN sites in error state:**

The show command allows for listing history of connect events for branches in UP or DOWN state too.

```

admin@Controller1-cli> show orgs org ServiceProvider sd-wan site history current-status ?
Possible completions:
  DOWN ERRONEOUS  UP
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site history current-status ERRONEOUS
Branch History:
  Branch-id      : 65
  Branch-name    : Hub-SaltLakeCity
  current-status : ERRONEOUS
  Logs          :
  Record :
    Event-message : Poststaging done
    Time-stamp    : 2016-03-30 17:23:14
  Record :
    Event-message : Poststaging done
    Time-stamp    : 2016-03-30 17:37:50

```

6.3.2 Branch Specific Commands

Run these CLI commands to get details of a specific SD-WAN site:

- `show orgs org ServiceProvider sd-wan site detail <branch ID>`
- `show orgs org ServiceProvider sd-wan site history branch-name <branch-name>`
- `show orgs org ServiceProvider sd-wan site statistics aggregate <site ID>`
- `admin@Controller1-cli> request clear statistics sd-wan acct all`

6.3.2.1 Examples

- Checking details of a specific SD-WAN site:

```

admin@Controller1-cli> show orgs org ServiceProvider sd-wan site detail 8001
Branch Id      : 8001
State         : Connected
Uptime        : 1h:3m:51s
Branch Name    : Branch1-SanFrancisco
Chassis Id     : 001branch1
Global Tenant Id : 1
Management IP  : 10.10.111.2
SA Available   : no
ESP Tunnel Info
Local Endpoint : 10.10.110.2
Remote Endpoint : 10.10.111.2
VXLAN Tunnel Info
Local Endpoint : 10.10.10.2
Remote Endpoint : 10.10.11.2

```

LINK ID	ACCESS CIRCUIT	LOCAL IP	NAT STATUS	PUBLIC IP	PUBLIC PORT	LINK ENCRYPTION	SHAPPING RATE	MIN SHAPPING RATE
1	101.101.101.1		Unknown	101.101.101.1	12983	plain-text	0	

- Checking details connectivity events for a specific SD-WAN site:

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site history branch-name Branch1-
SanFrancisco
Branch History:
  Branch-id      : 8001
  Branch-name    : Branch1-SanFrancisco
  current-status : UP
  Logs           :
  Record :
    Event-message : IKE attempted
    Time-stamp    : 2016-03-30 17:23:08
  Record :
    Event-message : IKE completed
    Time-stamp    : 2016-03-30 17:23:12
```

This list of events is in chronologically ascending order. So the last event can give an idea in case connectivity failures. For example, if the last event was “IKE attempted” and the time-stamp is more than a several seconds in the past, there is a high chance that IKE attempt from branch is failing. So this information provides a cue to for next step in debugging connectivity issue - either check IPsec or Datapath connectivity.

- Checking traffic statistics per-network path to a specific SD-WAN site:

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site statistics aggregate 8001
SITE PTVI  ENCAP  RX RX TX TX
ID INDEX TYPE   PKTS BYTES PKTS BYTES
-----
8001 1066  plaintext 0 0  0 0
8001 1076  encrypted 0 0  0 0
```

- Clearing traffic statistics for network paths to a specific SD-WAN site:

```
admin@Controller1-cli> request clear statistics sd-wan ackt all
```

6.4 Debugging NAT Connectivity Issues

SteelConnect EX SD-WAN controller acts as a STUN server for branch devices to resolve their NAT bindings.

- Run the `show orgs org ServiceProvider sd-wan site statistics vbp branch` CLI command to get the details of number of NAT binding resolution requests received from each SD-WAN appliance, per remote WAN interface.

```
admin@Controller1-cli> show orgs org ServiceProvider sd-wan site statistics vbp branch
```

BRANCH	LINK	PUBLIC		TX		TX		RX	
ID	SITE NAME	ID	PUBLIC IP	PORT	PKTS	BYTES	PKTS	BYTES	

10	East-Coast-Controller-1								
8001	Branch1-SanFrancisco	1	101.101.101.1		12983	798	12768	798	9576
		2	192.168.12.2		4790	798	12768	798	9576
8002	Branch2-Phoneix	1	102.102.101.1	4134	800	12800	800	9600	
		2	192.168.22.2		4790	800	12800	800	9600
8003	Hub-SaltLakeCity	1	192.168.31.2	4790	592	9472	592	7104	
		2	192.168.32.2		4790	592	9472	592	7104
8004	Branch4-Detroit	1	104.104.101.1		14851	794	12704	794	9528
		2	192.168.42.2		4790	794	12704	794	9528
8005	Branch5-Tampa	1	105.105.101.1		52559	795	12720	795	9540
		2	192.168.52.2		4790	795	12720	795	9540
8006	Hub-Denver	1	192.168.61.2	4790	793	12688	793	9516	
		2	192.168.62.2		4790	793	12688	793	9516

- Run the request `clear statistics sd-wan vbp all` CLI command to clear the statistics.
- Run the `show orgs org ServiceProvider sd-wan site statistics vbp branch` CLI command to check and determine if NAT binding resolution is attempted by a branch device. If NAT resolution request fails to reach controller, then this indicates either a basic connectivity issue between branch and controller, or a potential datapath issue causing VBP packets to drop.

6.5 Stage Three Debugging on Branch

This section covers additional points to be noted while debugging SD-WAN connectivity from Branch node.

Follow these points while debugging SD-WAN connectivity from a branch node:

- Configure branch device to establish secure connectivity to controller(s), and possibly hub(s). The appliance continuously tries to establish connectivity.
- Pre-requisites for successful establishment of connectivity to controller, apart from correct configuration, are as follows:
 - Ensure that WAN interface(s) are up with an IP address.
 - Run the `show interfaces brief` CLI command to check interface status, and presence of IP address.

Example: ptvi11 interface to controller is in up state indicating that IKE based IPsec connectivity to the corresponding controller is established

```
admin@SD-WAN-br2-cli> show interfaces brief
```

NAME	IP	MAC	OPER	ADMIN	TNT	VRF
eth-0/0	[10.40.60.134/16]	00:50:56:8a:25:78	up	0		global
ptvi1	[41.41.40.2/32]	n/a	down		down	1 RT_Provider
ptvi11	[1.1.4.2/32]	n/a	up	1		RT_Provider
ptvi2	[41.2.2.2/32]	n/a	down		down	1 RT_Provider
tvi-0/1		n/a			up	
tvi-0/1.0	[10.10.12.2/24]	n/a	up	1		RT_Provider
tvi-0/2		n/a			up	
tvi-0/2.0	[20.20.22.3/24]	n/a	up	1		RT_Provider
vni-0/0		00:50:56:8a:e5:cc			up	
vni-0/0.0	[80.80.80.102/24]	00:50:56:8a:e5:cc	up	0		global
vni-0/1		00:50:56:8a:00:e9		up		up
vni-0/1.0	[192.168.101.4/24]	00:50:56:8a:00:e9	up		up	0 grt-vrf
vni-0/2		00:50:56:8a:96:34	down		down	

- After the completion of transport connectivity from a branch to controller, at least one of the WAN interfaces will be available, and the branch establishes IKE-based IPsec connectivity to the controller. If establishment of IKE-based IPsec connectivity is successful, ptvi interface to the particular controller will be in up state.
- If establishing IKE-based IPsec connectivity fails, corresponding ptvi interface will be in down state. For debugging these issue, refer to [Datapath](#) and [Ipsec](#) debugging sections.
- After IPsec connectivity, to at-least one controller, is established, branch determines whether each of its WAN interfaces is behind a NAT device. Run the show orgs org Provider sd-wan site statistics vbp self CLI command to check the statistics for this activity.

Example

```
admin@SD-WAN-br2-cli> show orgs org Provider sd-wan site statistics vbp self
```

INTF	NAME	ID	LINK	CIRCUIT	TX PKTS	TX BYTES	RX PKTS	RX BYTES
vni-0/0.0		1	wan-isp-a		2030	178640	2030	32480
vni-0/1.0		2	wan-isp-b		2030	178640	24	384

- Run the show orgs org Provider sd-wan site statistics vbp self CLI command along with the request clear statistics sd-wan vbp all CLI command to determine if NAT binding resolution requests are being originated from the branch, and if response is arriving from the controller.
- Run the show orgs org ServiceProvider sd-wan site detail <site-id> CLI command to verify any resultant NAT binding discovered by the branch device. NAT binding is listed as public IP and public port for each WAN interface.

6.6 Troubleshooting SteelConnect EX FlexVNF Deployment Failure

Deployment of SteelConnect EX FlexVNF on a branch fails after getting the management IP address, in spite of running the `staging.py` script. To resolve this issue, ensure you have:

- Appropriately configured pre-staging and post staging templates.
- Connectivity between the SteelConnect EX Director to a controller, and from a controller

to a branch. Follow these steps to resolve this issue and deploy the SteelConnect EX FlexVNF on a branch:

1. Check the route in branch. The branch must have the southbound IP Address (192.10.1.1) of SteelConnect EX Director.
2. Run the `show route routing-instance grt` CLI command to check the route between the branch to the SteelConnect EX Director.

Example

```
admin@BRANCH1001-cli> show route routing-instance grt
Routes for Routing instance : grt AFI: ipv4
Codes: E1 - OSPF external type 1, E2 - OSPF external type 2
IA - inter area, iA - intra area,
L1 - IS-IS level-1, L2 - IS-IS level-2
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
RTI - Learnt from another routing-instance
+ - Active Route
Prot   Type  Dest Address/Mask   Next-hop   Age      Interface name
-----
conn   N/A   +192.20.11.0/30     0.0.0.0    2w1d04h  vni-0/0.0
local  N/A   +192.20.11.2/32     0.0.0.0    2w1d04h  directly connected
```

3. Open the `/var/versa/vnms/data/conf/vnms.properties` file and check for the southbound IP address (192.10.1.1 of the SteelConnect EX Director).
4. Edit the `/var/versa/vnms/data/conf/vnms.properties` file to add the southbound IP Address of SteelConnect EX Director, if this IP is missing in the file.
5. Run this `sudo` command to open the log file:

```
admin@LAB2-DIRECTOR:~$ sudo vi /var/versa/vnms/data/conf/vnms.properties
```

```
##added property
#Wed May 10 08:15:12 UTC 2017
VNMS_API_ENDPOINT_PORT=9182
NETWORK_MAPPING=vni0/0\ :WAN1,vni0/1\ :WAN2,vni0/2\ :LAN1
SERVICES_API_SECURE_ENDPOINT_HOST=https://10.192.88.31
AVAILABLE_ROUTING_INSTANCES=mgmt
VNMS_API_ENDPOINT_HOST=https://10.192.88.31
ALARM_PROTOCOL=tcp
SERVICES_API_ENDPOINT_PORT=9182
STARTUP_MODE=STANDALONE
VNF_MANAGEMENT_IP=192.10.1.1
SERVICES_API_ENDPOINT_HOST=https://10.192.88.31
VNMS_NCS_COMMIT_TIMEOUT_IN_SECS=30
MAX_TASKS=5000
MANAGEMENT_IP=localhost
REDIS_DATASTORE_PORT=6379
DASHBOARD_THREAD_POOL_SIZE=50
VNMS_REBOOT_APPLIANCE_AFTER_IN_SEC=10
REDIS_DATASTORE_HOST=127.0.0.1
SB_ADDRESS_LIST=192.10.1.1
ALARM_DATA_FORMAT=syslog
DESIGNATED_MASTER=TRUE
DASHBOARD_REFRESH_INTERVAL_IN_SECONDS=300
SERVICES_API_SECURE_ENDPOINT_PORT=9183
```

7 Troubleshooting CG-NAT

This section covers:

7.1 CGNAT Summary

Run the `show orgs org-services <tenant-name> cgnat summary` CLI command to view CGNAT summary.

Example

```
admin@vcsn> show orgs org-services Customer1 cgnat summary
```

VSN	ADDRESS	TOTAL INDEPENDENT MAPPINGS	TOTAL ENDPOINT INDEPENDENT FILTERS	DSLITE	TOTAL ENDPOINT TOTAL 6RD SOFTWARES	TOTAL MAPT SOFTWARES	TOTAL SUBSCRIBERS
2	0	0	0	0	0	0	0
0							

7.2 CGNAT Per Pool Statistics

Run the `show orgs org-services <tenant-name> cgnat pools <pool-name> statistics` CLI command to view CGNAT per pool statistics.

Example

```
admin@vcsn> show orgs org-services Customer1 cgnat pools Customer1_NAPT_Pool statistics
statistics
id                2
bindings-allocated 33
bindings-freed    33
alloc-failures    0
free-failures     7
out-of-address-errors 0
out-of-ports-errors 0
```

7.3 Inspecting CGNAT State in vsmd

Run these CLI commands to inspect CGNAT state in vsmd:

- `show cgnat tenants`
- `show cgnat acl info <tenant-id>`
- `show cgnat acl counters <tenant-id>`
- `show cgnat acl history <tenant-id>`
- `show cgnat config pool brief`

- `show cgnat config pool detail <pool-id> <tenant-id>`
- `show cgnat config rule brief`
- `show cgnat config rule detail <rule-id> <tenant-id>`
- `show cgnat counters active`
- `show cgnat counters internal`
- `show cgnat resources state 100`
- `show cgnat resources counters Customer1_NAPT_Pool 100`
- `show cgnat static-nat <tenant-id>`
- `show rfm table src-port allocated`
- `show rfm table src-ip allocated`
- `show filter info all table <tenant-id>`
- `show filter stats all <tenant-id>`

7.3.1 Configured Tenants and Summary

Run the `show cgnat tenants` CLI command to view configured tenants and summary.

Example

```
vsm-vcsn> show cgnat tenants
TENANT (id: 1)
  Num of rules configured      : 0
  Num of pools configured     : 0
  Num of DSLite SC configured : 0
  Num of 6RD SC configured    : 0
  Num of Static NAT entries   : 2
  Num of subscribers         : 0
  Num of EIM entries         : 0
  Num of EIF entries         : 0
  Num of DSLite softwires     : 0
  Num of 6RD softwires       : 0
TENANT (id: 200)
  Num of rules configured      : 3
  Num of pools configured     : 3
  Num of DSLite SC configured : 0
  Num of 6RD SC configured    : 0
  Num of Static NAT entries   : 0
  Num of subscribers         : 0
  Num of EIM entries         : 0
  Num of EIF entries         : 0
  Num of DSLite softwires     : 0
  Num of 6RD softwires       : 0
TENANT (id: 100)
  Num of rules configured      : 3
  Num of pools configured     : 3
  Num of DSLite SC configured : 0
  Num of 6RD SC configured    : 0
  Num of Static NAT entries   : 0
  Num of subscribers         : 0
  Num of EIM entries         : 0
  Num of EIF entries         : 0
  Num of DSLite softwires     : 0
  Num of 6RD softwires       : 0
```

7.3.2 CGNAT ACLs for Traffic Matching

Run the `show cgnat acl info <tenant-id>` CLI command to view CGNAT ACLs for traffic matching.

Example

```
vsm-vcsn> show cgnat acl info 100
```

Legends

C - Category (A: Allow, D: Deny)

P - Rule precedence

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
|  ACL Hd1  | RuleId | C | P |      VRF      |      SOURCE-IP      |      DEST-IP  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 0x00010003 |      3 | A | 1 | 14-14 | 192.168.130.0/24 | 192.168.130.0/24 |
99.88.77.0/28 |
| 0x00020001 |      1 | A | 1 | 14-14 | 192.168.130.0/24 | 192.168.130.0/24 |
88.77.66.0/28 |
| 0x00030002 |      2 | A | 1 | 14-14 | 192.168.130.0/24 | 192.168.130.0/24 |
77.66.55.0/28 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
Tenant id      : 100
Total filters : 3
```

7.3.3 CGNAT ACL Counters

Run the `show cgnat acl counters <tenant-id>` CLI command to view CGNAT ACL counters (incremented when traffic matches). Use Rule IDs to co-relate with above table.

Example

```
vsm-vcsn> show cgnat acl counters 100
```

```
+-----+-----+-----+-----+
|  ACL Hd1  | Rule ID | COUNTER |
+-----+-----+-----+-----+
| 0x00010003 |      3 |      14 |
| 0x00020001 |      1 |       0 |
| 0x00030002 |      2 |      26 |
+-----+-----+-----+-----+
Tenant id      : 100
Total filters : 3
```

7.3.4 CGNAT ACL History

Run the `show cgnat acl history <tenant-id>` CLI command to view CGNAT ACL history (traffic pattern which hits/matches).

Example

```
vsm-vcsn> show cgnat acl history 100
```

SOURCE-IP	DEST-IP	SPORT	DPORT	P	RES
192.168.130.50	77.66.55.1	46859	53	17	Hit
192.168.130.50	77.66.55.1	58285	53	17	Hit
192.168.130.50	77.66.55.1	54768	21	6	Hit
192.168.130.50	77.66.55.1	54772	21	6	Hit
192.168.130.50	77.66.55.1	54776	21	6	Hit
192.168.130.50	77.66.55.1	54779	21	6	Hit <==

7.3.5 CGNAT Pool Config

Run the `show cgnat config pool brief` CLI command to see CGNAT pool config in brief across all tenants.

Example

```
vsm-vcsn> show cgnat config pool brief
```

TNT-ID	POOL-NAME	ID
200	Customer2_SNAT_Pool	3
200	Customer2_DYN_Pool	1
200	Customer2_NAPT_Pool	2
TNT-ID	POOL-NAME	ID
100	Customer1_SNAT_Pool	3
100	Customer1_DYN_Pool	1
100	Customer1_NAPT_Pool	2

7.3.6 CGNAT Pool Config for Tenant

Run the `show cgnat config pool detail <pool-id> <tenant-id>` CLI command to see CGNAT pool config in detail for a particular tenant. Fetch the Pool-id from above table.

Example

```
vsm-vcsn> show cgnat config pool detail 3 200
TENANT                : [200]
NAT Pool name         : [Customer2_SNAT_Pool]
Object ID             : [3]
Pool Flags            : [0x00001941]
IPv4 Ranges
    102.84.68.0/24
Routing instance      : [ri_Global - 18]
Provider Tenant id    : [1]
TCP idle timeout      : [7440]
TCP idle timeout      : [7440]
UDP idle timeout      : [300]
ICMP idle timeout     : [60]
Addr allocation       : [round-robin]
```

7.3.7 CGNAT Rule Config

Run the `show cgnat config rule brief` CLI command to see CGNAT rule config in brief across all tenants.

Example

```
vsm-vcsn> show cgnat config rule brief
```

TNT-ID	RULE-NAME	ID

200	Customer2_SNAT_Rule	3
200	Customer2_DYN_Rule	1
200	Customer2_NAPT_Rule	2
TNT-ID	RULE-NAME	ID

100	Customer1_SNAT_Rule	3
100	Customer1_DYN_Rule	1
100	Customer1_NAPT_Rule	2

7.3.8 CGNAT Rule Config for Tenant

Run the `show cgnat config rule detail <rule-id> <tenant-id>` CLI command to see CGNAT rule config in detail for a particular tenant. Fetch the Rule-id from above table.

Example

```
vsm-vcsn> show cgnat config rule detail 1 200
TENANT                : [200]
NAT Rule name         : [Customer2_DYN_Rule]
Object ID             : [1]
Precedence            : [1]
TERM
  Routing instance    : [ri_Customer2 - 16]
  Source IPv4 Prefix  :
    192.168.131.0/24
  Dest IPv4 Prefix    :
    55.44.33.0/28
ACTION
  NAT mode            : [dynamic-nat-44]
  Addr Pool Paired    : [False]
  EIM enabled         : [False]
  EIF enabled         : [False]
  Source Pool         : [1]
```

7.3.9 Active Counters

Run the `show cgnat counters active` VTY command to display active counters. You can check for these when the data is flowing to verify if CGNAT is indeed NATing the packets or not.

Example

```
vsm-vcsn> show cgnat counters active
CGNAT_CNTR_ACTIVE_BINDINGS      0
CGNAT_CNTR_ACTIVE_SUBS         0
CGNAT_CNTR_ACTIVE_MAPT_SUBS    0
CGNAT_CNTR_ACTIVE_EI_MAPPINGS  0
CGNAT_CNTR_ACTIVE_EI_FILTERS   0
```

7.3.10 Internal Counters

Run the `show cgnat counters internal` CLI command to display internal counters.

Example

```
vsm-vcsn> show cgnat counters internal
CGNAT_CNTR_BINDINGS_ALLOCD          77
CGNAT_CNTR_BINDINGS_FREED          103
CGNAT_CNTR_ALG_BINDINGS_ALLOCD      26
CGNAT_CNTR_FIRST_PACKET             104
CGNAT_CNTR_SESS_ACCEPT              103
CGNAT_CNTR_SESS_IGNORE               1
CGNAT_CNTR_SESS_CLOSE               103
CGNAT_CNTR_TENANT_LKUP_FAIL           1
CGNAT_CNTR_ST_NAT_FILTER_SUCCESS     28
CGNAT_CNTR_TENANT_SWITCHED           28
CGNAT_CNTR_VRF_SWITCHED              28
CGNAT_CNTR_LOG_EXPORTED              98
```

7.3.11 CGNAT Resources State

Run the `show cgnat resources state 100` CLI command to display CGNAT resources state. Each CGNAT pool is a resource managed by SteelConnect EX Resource and Forwarding Daemon. Ensure these are in a **READY** state before CGNAT can handle traffic for some pool. If it is not ready the traffic is dropped. Check the above CLI command for **MISC_POOL_ERROR** counters.

Example

```
vsm-vcsn> show cgnat resources state 100
Legends:
    A - Address present
    P - Ports present
    S - Static pool
    R - Preserve range
    Y - Preserve parity

+-----+-----+-----+-----+
| ID | Pool-name | State | Flags |
+-----+-----+-----+-----+
| 3 | Customer1_SNAT_Pool | READY | A.S.. |
| 1 | Customer1_DYN_Pool | READY | A. .... |
| 2 | Customer1_NAPT_Pool | READY | AP. .... |
+-----+-----+-----+-----+
```

7.3.12 CGNAT Pool Counters

Run the `show cgnat resources counters Customer1_NAPT_Pool 100` CLI command to display CGNAT pool counters. This displays the number of bindings that are allocated and freed per pool. This command also shows resources assigned by SteelConnect EX Resource and Forwarding daemon to each thread.

Example

```
vsm-vcsn> show cgnat resources counters Customer1_NAPT_Pool 100
Pool      : Customer1_NAPT_Pool
State     : READY
VRF       : ri_Global (18)
Thread 0 : IP [104.74.68.1 - 104.74.68.254]   Ports [4096 - 40959]   <<<<<<
COUNTER          TCP          UDP          OTHER
-----
Bindings allocated      25          4          4
Bindings freed          18          4          11
Alloc failures           0          0          0
  Out of addrs           0          0          0
  Out of ports           0          0          0
Free failures           0          0          7
```

7.3.13 Static NAT Mappings

Run the `show cgnat static-nat <tenant-id>` CLI command to view static NAT mappings used for debugging basic-nat-44 and dnat-44.

Example

```
vsm-vcsn> show cgnat static-nat 100
Legends:
  ID = Rule ID
  T = Type (S - Static NAT)
        (D - Dest NAT)
        (2 - Twice NAT)

+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | T | Rule SPrefix | Rule DPrefix | Pool SPrefix | Pool DPrefix | RPort | Pport |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | S | 192.168.131.0 | | - | 102.84.68.0 | | - | 0 |
0 |
| 3 | S | 192.168.130.0 | | - | 102.74.68.0 | | - | 0 |
0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Use these clear commands for debugging:

- `vsm-vcsn> clear cgnat counters internal`
- `vsm-vcsn> clear cgnat acl history <tnt-id>`
- `vsm-vcsn> clear cgnat acl counters <tnt-id>`

7.3.14 CGNAT Pool with Source-Port

Run the `show rfm table src-port allocated` CLI command to display per thread resource allocation. Here CGNAT pool with source-port is the resource which is divided and then distributed to all worker threads by SteelConnect EX RFD/RFM.

Example

```
vsm-vcsn> show rfm table src-port allocated
src-port: Table Alloc entries
Tenant 0:
StartPort EndPort PoolId Client RmtID LclID Thread Application
=====
4096      4096      0      2      0      1      1      App-RFM
4097      4097      0      2      0      2      1      App-RFM
4114      4114      0      2      0      4      1      App-RFM
4098      4113      0      2      0      3      1      App-LEF
Tenant 1:
StartPort EndPort PoolId Client RmtID LclID Thread Application
=====
Tenant 100:
StartPort EndPort PoolId Client RmtID LclID Thread Application
=====
4096      40959     2      2      0      1      1      App-CGNAT
Tenant 200:
StartPort EndPort PoolId Client RmtID LclID Thread Application
=====
4096      40959     2      2      0      1      1      App-CGNAT
```

7.3.15 CGNAT Pool with Source-IP

- Run the `show rfm table src-ip allocated` CLI command to display per thread resource allocation. Similar to above command but here CGNAT pool with source-ip only is the resource. Displays which thread received what quota from RFD/RFM.

Example

```
vsm-vcsn> show rfm table src-ip allocated
src-ip: Table Alloc entries
Tenant 0:
Start IP          End IP          Pool Client RmtId LclId Thread  Application
=====
Tenant 1:
Start IP          End IP          Pool Client RmtId LclId Thread  Application
=====
Tenant 100:
Start IP          End IP          Pool Client RmtId LclId Thread  Application
=====
103.74.0.0        103.74.119.255 1          2          0          1          1          App-CGNAT
Tenant 200:
Start IP          End IP          Pool Client RmtId LclId Thread  Application
=====
103.84.0.0        103.84.119.255 1          2          0          1          1          App-CGNAT
```

- Run the `show filter info all table <tenant-id>` and the `show filter stats all <tenant-id>` CLI commands to display filters/ACLs per tenant. These are datapath filters used to steer reverse traffic to the right VCSN/VSN in case of multi-VSN deployment. This datapath filter table is also used by CGNAT for installing basic-nat-44 and dnat-44 filters.

Example

```
vsm-vcsn> show filter info all table 1
```

```
Legends
```

```
C/Catg - Category
```

```
L/Prio - Level (Predefined priority)
```

```
P/Prot - Protocol
```

-----+									
Tenant Id -									
1									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
+-----+									
Rule Hdl C L P VRF SOURCE-IP DEST-IP									
SPORT DPORT									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
+-----+									
0x01830403 A 1 17 0-256 0.0.0.0/0 0.0.0.0/0									
67-67 68-68									
0x01840403 A 1 17 0-256 0.0.0.0/0 0.0.0.0/0									
0-65535 3784-3784									
0x01850404 A 1 17 0-256 0.0.0.0/0 0.0.0.0/0									
0-65535 4784-4784									
0x01860403 A 19 6 0-256 0.0.0.0/0 0.0.0.0/0									
0-65535 179-179									
0x01870403 A 19 6 0-256 0.0.0.0/0 0.0.0.0/0									
179-179 0-65535									
0x01885003 A 16 0 18-18 0.0.0.0/0 102.84.68.0/24									
0-65535 0-65535									
0x01895003 A 16 0 18-18 0.0.0.0/0 102.74.68.0/24									
0-65535 0-65535									
0x018a4003 A 19 0 18-18 0.0.0.0/0 103.74.0.0/18									
0-65535 0-65535									
0x018b4003 A 19 0 18-18 0.0.0.0/0 103.74.64.0/19									
0-65535 0-65535									
0x018c4003 A 19 0 18-18 0.0.0.0/0 103.74.96.0/20									
0-65535 0-65535									
0x018d4003 A 19 0 18-18 0.0.0.0/0 103.74.112.0/21									
0-65535 0-65535									
0x018e4003 A 19 0 18-18 0.0.0.0/0 104.74.68.0/24									
0-65535 4096-40959									
0x018f4003 A 19 0 18-18 0.0.0.0/0 103.84.0.0/18									
0-65535 0-65535									
0x01904003 A 19 0 18-18 0.0.0.0/0 103.84.64.0/19									
0-65535 0-65535									
0x01914003 A 19 0 18-18 0.0.0.0/0 103.84.96.0/20									
0-65535 0-65535									
0x01924003 A 19 0 18-18 0.0.0.0/0 103.84.112.0/21									
0-65535 0-65535									
0x01934003 A 19 0 18-18 0.0.0.0/0 104.84.68.0/24									
0-65535 4096-40959									
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+									
+-----+									

```
| Total IPv4 Filters:
```

```
17
```

```
|
```

```
+-----+
```

```
-----+
```

8 Troubleshooting LTE Configuration

This section covers various CLI commands used to troubleshoot the LTE configuration:

- When the interface vni-0/100.0 does not show an IP address, run these debug commands:

Run the `~] # mmcli -m 0` command to verify that the modem is detected.

Example

```
[admin@versa-flexvnf: ~] # mmcli -m 0
```

- Run the `/org/freedesktop/ModemManager1/Modem/0` command to display the various specifications of the device. The value of `m` can vary from 0 to 3 depending on the number of the modem.

Example

```

/org/freedesktop/ModemManager1/Modem/0 (device id 'alae475e8d9be21d63bfa19db19837d2595742f3')
-----
Hardware | manufacturer: 'Sierra Wireless, Incorporated'
          | model: 'MC7455'
          | revision: 'SWI9X30C_02.24.05.06 r7040 CARMD-EV-FRMWR2 2017/05/19 06:23:09'
          | supported: 'gsm-umts
          |         lte
          |         gsm-umts, lte'
          | current: 'gsm-umts, lte'
          | equipment id: '359072060692577'
-----
System   | device: '/sys/devices/pci0000:00/0000:00:16.0/usb1/1-1/1-1.3'
          | drivers: 'option1, qmi_wwan'
          | plugin: 'Sierra'
          | primary port: 'cdc-wdm0'
          | ports: 'ttyUSB0 (qcdm), ttyUSB2 (unknown), cdc-wdm0 (qmi), cdc-wdm1 (qmi), wwan0 (net), wwan1
(net)'
-----
Numbers | own : '4086184766'
-----
Status   | lock: 'sim-pin2'
          | unlock retries: 'sim-pin (3), sim-pin2 (3), sim-puk (10), sim-puk2 (10)'
          | state: 'connected'
          | power state: 'on'
          | access tech: 'lte'
          | signal quality: '55' (recent)
-----
Modes    | supported: 'allowed: 2g, 3g, 4g; preferred: none'
          | current: 'allowed: 2g, 3g, 4g; preferred: none'
-----
Bands    | supported: 'u2100, u1800, u1900, u17iv, u850, u900, eutran-ii, eutran-iv, eutran-v, eutran-xiii'
          | current: 'u2100, u1800, u1900, u17iv, u850, u900, eutran-ii, eutran-iv, eutran-v, eutran-xiii'
-----
IP        | supported: 'ipv4, ipv6, ipv4v6'
-----
3GPP     | imei: '359072060692577'
          | enabled locks: 'none'
          | operator id: '311480'
          | operator name: 'VZW'
          | subscription: 'unknown'
          | registration: 'home'
-----
SIM       | path: '/org/freedesktop/ModemManager1/SIM/0'
-----
Bearers  | paths: '/org/freedesktop/ModemManager1/Bearer/3'[admin@versa-flexvnf: ~] #

```

- Check with service provider whether the SIM card is activated.
- Check the status information, which shows whether the SIM card is connected/enabled.
- 3GPP information shows the operator ID.
- In some cases, there could be a missing operator ID and there might be a need to configure an application name.

- Run the `set interfaces vni-0/100 wwan apn vzwinternet` CLI command to configure the appliance name.

Example

```
admin@versa-flexvnf-cli(config)% set interfaces vni-0/100 wwan apn vzwinternet
[ok] [2018-03-12 11:10:31]
[edit]
admin@versa-flexvnf-cli(config)% commit
Commit complete.
[ok] [2018-03-12 11:10:34]
[edit]admin@versa-flexvnf-cli(config)%
```

- Run the `run show interfaces wwanstatus brief` CLI command to see the wwan interface status on SteelConnect EX FlexVNF.

Example

```
admin@versa-flexvnf-cli(config)% run show interfaces wwanstatus brief
INTERFACE NAME INTERFACE STATUS CONNECTION VENDOR PRODUCT NAME STATUS HW STATUS ID ID HW VENDOR NAME HW
MODEL NAME IMEI
-----
vni-0/100 Configured ModemDetected Connected 1199 9071 Sierra Wireless,
Inc. Sierra_Wireless_MC7455_Qualco 359072060692577
vni-0/101 Not configured
vni-0/102 Not configured
vni-0/103 Not configured
[ok] [2018-03-12 11:11:35]
[edit]
admin@versa-flexvnf-cli(config)%
```

- Run the `run show interfaces wwan status carrier-brief` CLI command to check the carrier information.

Example

```
admin@versa-flexvnf-cli(config)% run show interfaces wwan status carrier-brief
INTERFACE CURRENT OPERATOR RX TX NAME STATE NUMBER
TECH SIGNAL RSSI RSRQ RSRP SNR ID OPERATOR NAME ADDRESS PREFIX GATEWAY DNS UPTIME BYTES BYTES
-----
vni-0/100 disabled 4086184766 unknown 0 311480 Verizon Wireless
vni-0/101
vni-0/102
vni-0/103
[ok] [2018-03-12 11:11:54]
```

- Run the `run show interfaces wwan status cell-location` CLI command to check the location ID.

Example

```
admin@versa-flexvnf-cli(config)% run show interfaces wwan status cell-location

TRACKING
INTERFACE AREA GLOBAL
NAME      MCC MNC NETWORK NAME CODE CELL ID
-----
vni-0/100 311 480 Verizon Wireless 7942 7975180
vni-0/101
vni-0/102
vni-0/103
[ok] [2018-03-12 11:12:55]
```

9 Troubleshooting Link Bandwidth

[Click here to view the topics on this page](#)

SteelConnect EX speed test allows you to measure WAN link bandwidth. SteelConnect EX FlexVNF can be configured as a client or server. Both, the client and server can be a regular CPE or a part of SDWAN topology. After the server is configured, the client can initiate the test using client request commands. The client initiates a TCP-based connection toward the server, which listens on port 5201. The client generates a request ID for each request received from users. Based on the received request ID, users can fetch the latest test results from the device.

Currently, SteelConnect EX speed test measures bandwidths from 2 Mbps to 850 Mbps. This depends on factors like in-path shapers, latency, etc. The test consumes around 2 MB of data in 1-2 seconds to measure the bandwidth. The client device maintains the history of the last 20 test results irrespective of success or failure.

In comparison with other Linux tools like iperf, SteelConnect EX speed test is more efficient and suits well for LTE connections where bandwidth consumption is critical factor.

9.1 Configuring Speed Test Server

You can configure Speed Test server using either:

- GUI
- REST API
- CLI

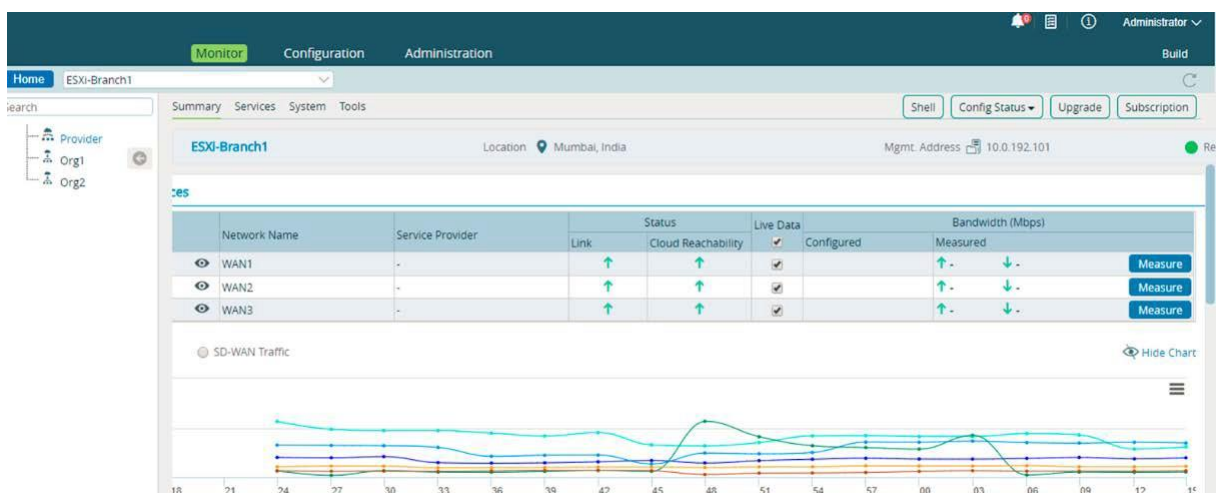
9.1.1 Using GUI

Follow these steps to view the Server mode settings:

1. Login to the SteelConnect EX Director.
2. Select **Administration > Appliances** and select a server from the list displayed in the dashboard. This takes you to the selected servers context page.

Name	Mgmt. Address	Type	Service Start Time	Software Version	Site ID	Organizations	Snapshots	Config Synchronisation	Reachability	Status	Service	Locked
ESXi-Branch1	10.0.192.101	Branch	Wed, Apr 18 20...	16.1-R2-S2.1	101	Org1,Org2,Provider	1	Up	Up	Up	Up	Up
ESXi-Branch2	10.0.192.104	Branch	Wed, Apr 18 20...	16.1-R2-S2.1	104	Org1,Org2,Provider	1	Up	Up	Up	Up	Up
ESXi-Device-001	10.0.192.105	Branch	Wed, Apr 18 20...	16.1-R2-S2.1	105	Org1,Org2,Provider	1	Unavailable	Unknown	Unknown	Unknown	Unknown
SDWAN-Controller-1	10.192.43.143	Controller	Wed, Apr 18 20...	16.1-R2-S2.1	1	Org1,Org2,Provider	1	Up	Up	Up	Up	Up
SPEED-TEST-SERVER	10.128.0.255	Service-vmf	Wed, Apr 18 20...	16.1-R2-S2.1	1	ST-Provider	1	Unavailable	Unknown	Unknown	Unknown	Unknown
ST-Branch1	10.0.192.102	Branch	Wed, Apr 18 20...	16.1-R2-S2.1	102	Org1,Org2,Provider	1	Unknown	Unknown	Unknown	Unknown	Unknown

3. In the Server context page, select **Configuration > Others > System > Configuration** to view the various server related configurations.



9.1.2 Using REST API

Use this API to enable bandwidth measurement on a remote speed test server.

SteelConnect EX REST API	API Type
<code>https://<IP-address>:9182/api/config/devices/device/BW-Test-Server/config/system/bw-measurement</code>	PUT
Payload <pre> { "bw-measurement": { "server": { "routing-instance": "SPEED-SERVER-VR" } } } </pre>	

9.1.3 Using CLI

You can deploy and install the SteelConnect EX speed test on a regular SteelConnect EX FlexVNF device after completing the installation of the appliance and bringing it up with the default configurations.

Ensure you connect the appliances ports (single or multiple) to the network with single routing instance and tenant.

Follow these steps to configure the server to allow the client to start the speed test:

1. Run the `show configuration devices device <device-name> config system bw-measurement` CLI command to get the server mode configuration.

CLI Output

```

show configuration devices device SPEED-TEST-SERVER config system bw-measurement
server {
    routing-instance SPEED-SERVER-VR;
}

```

2. Run the `configure` CLI command to enter the devices configuration mode.

All the steps below will now run inside the devices configuration mode.

3. Run the `set devices device <device-name> config bw-measurement server routing- instance <routing-instance-name>` CLI command to configure the server.

CLI Output

```
set devices device SPEED-TEST-SERVER config system bw-measurement server routing-instance SPEED-
SERVER-VR
```

4. Check the *isolcpus* status and enable it if its disabled.

```
request system isolate-cpu status
request system isolate-cpu enable
```

5. Set the *tx-descriptor* count.

```
set system service-options tx-descriptors-count 1024
```

9.2 Viewing Client Speed Test Capability

Check the clients (device) parameters to ensure its capability to run the remote speed test. Use this API to view the server parameters:

SteelConnect EX REST API	API Type
<a href="https://<IP-address>:9182/vnms/dashboard/appliance/Desk-Silicom-VZ-Wifi-Spoke/live?command=system/bw-measurement/capability">https://<IP-address>:9182/vnms/dashboard/appliance/Desk-Silicom-VZ-Wifi-Spoke/live?command=system/bw-measurement/capability	GET

Payload

```
{
  "system:capability": {
    "status": true
  }
}
```

9.3 Initiating Speed Test From Client

You can install and deploy the Client as a regular SteelConnect EX FlexVNF or as a part of a SD-WAN based network topology. You can start the speed test after completing the basic configuration of the SteelConnect EX FlexVNF device.

Ensure you connect the appliances ports (single or multiple) to the network with single routing instance and tenant. This provides connectivity with the server transport IP address and appropriate routing instance to reach the Server.

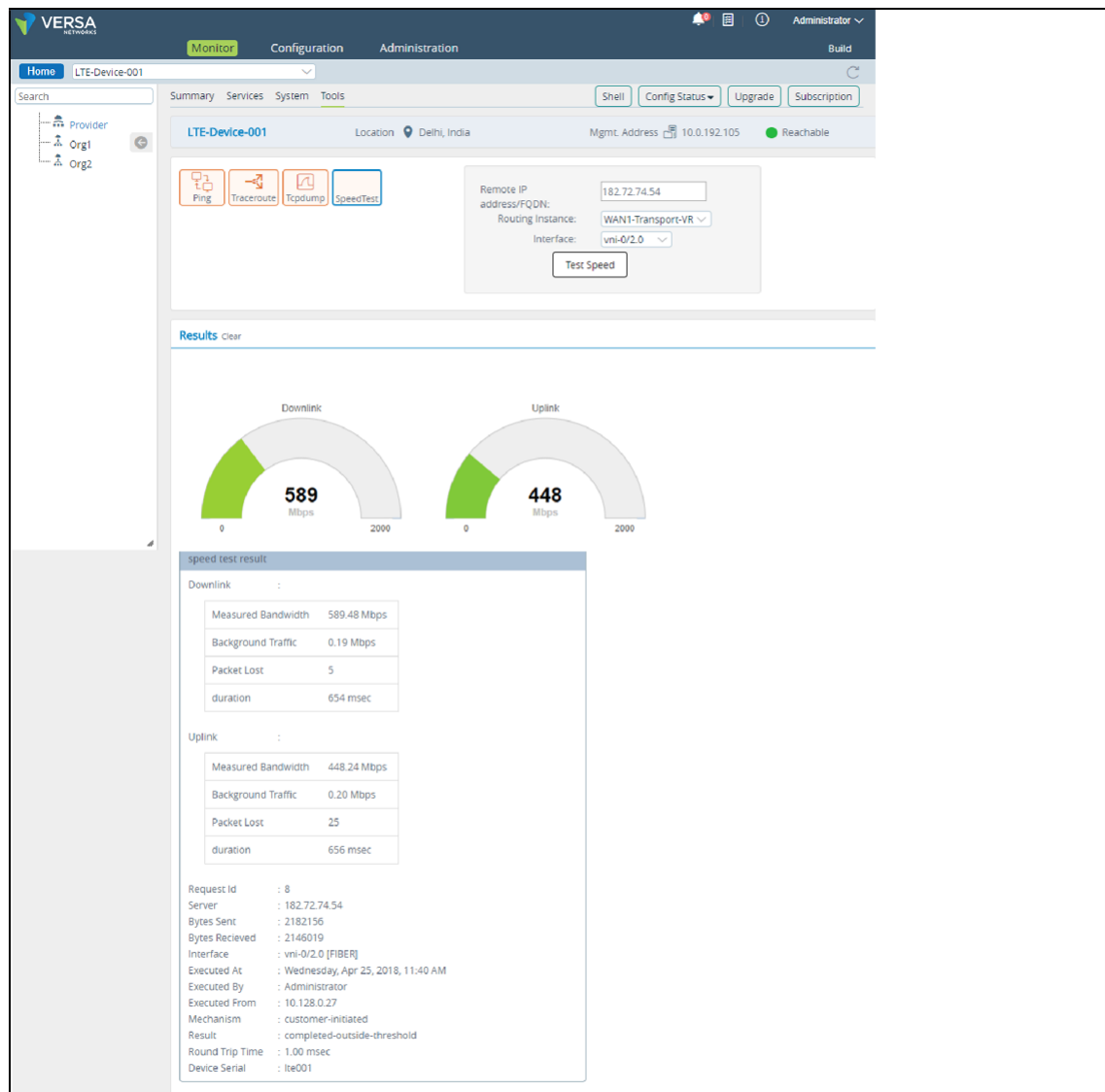
You can initiate a speed test from client using one of these:

- GUI
- REST API
- CLI

9.3.1 Using GUI

Follow these steps to initiate a speed test from the client GUI:

1. Login to the SteelConnect EX Director.
2. Select **Appliances > Administration** and select a Server from the list displayed in the dashboard. This takes you to the selected Servers context page.
3. Select **Monitor > Tools** and click **Speed Test** to initiate speed test from the device (client).



The image above (Step 3) shows the output of an Fibre based interface. For a LTE interface, use a vni in the range of vni-0/100.0 to vni-0/103/.0.

9.3.2 Using REST API

1. Use this API to request speed test on a client:

SteelConnect EX REST API	API Type
<code>https://<IP-address>:9182/versa/ncs-services/bw-measurement/_operations/client?appName=Controller1</code>	POST

Payload
<pre>{ "client": { "interface": "vni-0/0.0", "remote-destination": "10.10.10.01", "routing-instance": "WAN_1-Transport-VR", "executed-by": "Administrator", "executed-from": "10.10.10.01", "executed-at": "2018-04-06T09:44:58.016Z", "mechanism": "customer-initiated" } }</pre>

You may encounter these three scenarios when you request for a speed test on a client:

- a. Speed test is initiated successfully.

Response
<pre>{ "output": { "request-status": "test-initiated", "request-id": 14 } }</pre>

- b. Speed test fails to get a request from the device (the speed test is already running in the background).

Response
<pre>{ "output": { "request-status": "test-already-running", "request-id": 0 } }</pre>

- c. Speed test fails to get a request from the device.

Response

```
{
  "output": {
    "request-status": "test-failed ",
    "request-id": 0
  }
}
```

Use this API to poll the client for a status request:

SteelConnect EX REST API	API Type
<a href="https://<IP-address>:9182/versa/ncs-services/api/operational/devices/device/Controller1/live-status/system/bw-measurement/status/14">https://<IP-address>:9182/versa/ncs-services/api/operational/devices/device/Controller1/live-status/system/bw-measurement/status/14	GET

Response

```
{
  "status": {
    "request-id": 38,
    "server": "10.10.10.01",
    "uplink": {
      "measured-bandwidth": "890.00 Mbps",
      "background-traffic": "90.00 Mbps",
      "duration": "6126 msec",
      "packets-lost": 100
    },
    "downlink": {
      "measured-bandwidth": "880.00 Mbps",
      "background-traffic": "80.00 Mbps",
      "duration": "4129 msec",
      "packets-lost": 20
    },
    "round-trip-time": "3.46 msec",
    "bytes-sent": 2245454,
    "bytes-received": 2127895,
    "interface": "vni-0/100.0",
    "signal-strength": "'-75.00' dBm",
    "device-serial": "ST-Branch1",
    "executed-at": "2018-03-20T12:25:12.477+00:00",
    "executed-by": "Administrator",
    "executed-from": "10.10.10.02",
    "mechanism": "activation",
    "result": "completed-within-threshold"
  }
}
```

Refer to this table for the result codes supported in the speed test status:

#	Result Code
1	"result": "completed-within-threshold"
2	"result": "completed-outside-threshold"
3	"result": "uplink-in-progress"
4	"result": "downlink-in-progress"
5	"result": "failed-internal-error"
6	"result": "failed-timeout"
7	"result": "failed-background-traffic-exceeded"
8	"result": "failed-server-busy"

9.3.3 Using CLI

Follow these CLI commands to initiate the speed test from the client device:

1. Set the *tx-descriptor* count.

```
set system service-options tx-descriptors-count 4096
```

2. Set the cap limit for a day to the maximum value if the user wants to run multiple tests within a single day. The default limit is 5 tests per day.

```
set system bw-measurement test-limit 200
```

3. Run the `configureCLI` command to enter the client's configuration mode.

All the steps below will now run inside the device's configuration mode.

4. Run the `set devices device <device-name> config bw-measurement client-options background-traffic-threshold <1-100>` CLI command to configure the client.

Use this field...	to...
background-traffic-threshold <1-100>	Specify the threshold limit of the background traffic (in percentage) when the speed test under operation. You can configure the limit between 1-100 (in percentage). The default is set to 15%.

CLI Output

```
set devices device ST-Branch1 config bw-measurement client-options background-traffic-threshold 15
```

- Run the request `devices device <device-name> config system bw-measurement client interface <interface-name> routing-instance <routing-instance-name> remote-destination <remote-destination IP> executed-by <Administrator> executed-from <IP-address> executed-at <date and time> mechanism <activation|customer-initiated| installation|performance-monitoring>` CLI command to request SteelConnect EX speed test from the client device.

The `executed-by`, `executed-at`, `executed-from` and `mechanism` parameters are filled only when you are using the REST API. You do not have to enter values for these parameters when using the CLI command.

Use this field...	to...
routing-instance (Mandatory)	Specify the routing instance from which you want to initiate the speed test.
remote-destination (Mandatory)	Specify the Server IP address. The speed test will not start without this value.
interface (optional)	Enter the vni interface that is part of the routing instance. This forces the test to run from a particular vni interface that is part of the routing interface. This is useful when the interface is provided while issuing the CLI command, else the interface is found based on the best routes.
executed-by (optional)	SteelConnect EX Director auto-fills the users login ID when the request is issued. <div>You must not configure this manually.</div>
executed-at (optional)	SteelConnect EX Director auto-fills the date and time when the request is issued. <div>You must not configure this manually.</div>
executed-from (optional)	SteelConnect EX Director auto-fills the users IP address when the request is issued. <div>You must not configure this manually.</div>

Use this field...	to...
mechanism (optional)	<p>Specify the mechanism used to run the speed test. These are the options:</p> <ul style="list-style-type: none"> → activation → customer-initiated → installation → performance-monitoring

CLI Output

```
request devices device ST-Branch1 config system bw-measurement client interface vni-0/100.0
routing-instance ST-B2B-Transport-VR remote-destination 10.10.10.01 executed-by Administrator
executed-from 10.144.1.2 mechanism customer-initiated
```

This CLI command issues a request on the client and returns a request status and request ID in the CLI output. The following are the various request status that you can receive from the client:

- test-initiated
- test-failed
- test-already-running
- device-busy
- uplink-busy
- downlink-busy

Refer to this sample CLI output to check for the test-initiated status.

Sample CLI Output

```
Administrator@versa-director> request devices device ST-Branch1 config system bw-measurement
client routing-instance ST-B2B-Transport-VR remote-destination 10.10.10.01 interface vni-0/100.0
executed-by Administrator executed-from 10.10.10.02 mechanism activation

request-status test-initiated
request-id 38

[ok] [2018-04-17 16:13:17] Administrator@versa-director>
```

6. Run the `show devices device <device-name> live-status system bw-measurement status <request-id>` CLI command to poll the client for status request and fetch the results. Use the request ID as inputs in this CLI command.

CLI Output

```

show devices device ST-Branch1 live-status system bw-measurement status 38
live-status system bw-measurement status 38
server 10.10.10.01
uplink measured-bandwidth "890.00 Mbps"
uplink background-traffic "90.00 Mbps"
uplink duration "6126 msec"
uplink packets-lost 100
downlink measured-bandwidth "880.00 Mbps"
downlink background-traffic "80.00 Mbps"
downlink duration "4129 msec"
downlink packets-lost 20
round-trip-time "3.46 msec"
bytes-sent 2245454
bytes-received 2127895
interface vni-0/100.0
signal-strength "'-75.00' dBm"
device-serial ST-Branch1
executed-at 2018 04-24T18:15:50+00:00
executed-by Administrator
executed-from 10.10.10.02
mechanism activation
result completed-within-threshold

```

Use the request-id 38 to fetch the results from the device for this test run.

9.4 Initiating Speed Test From SD-WAN Branch

You can initiate speed test from a SD_WAN branch using one of these:

- GUI
- REST API
- CLI

9.4.1 Using GUI

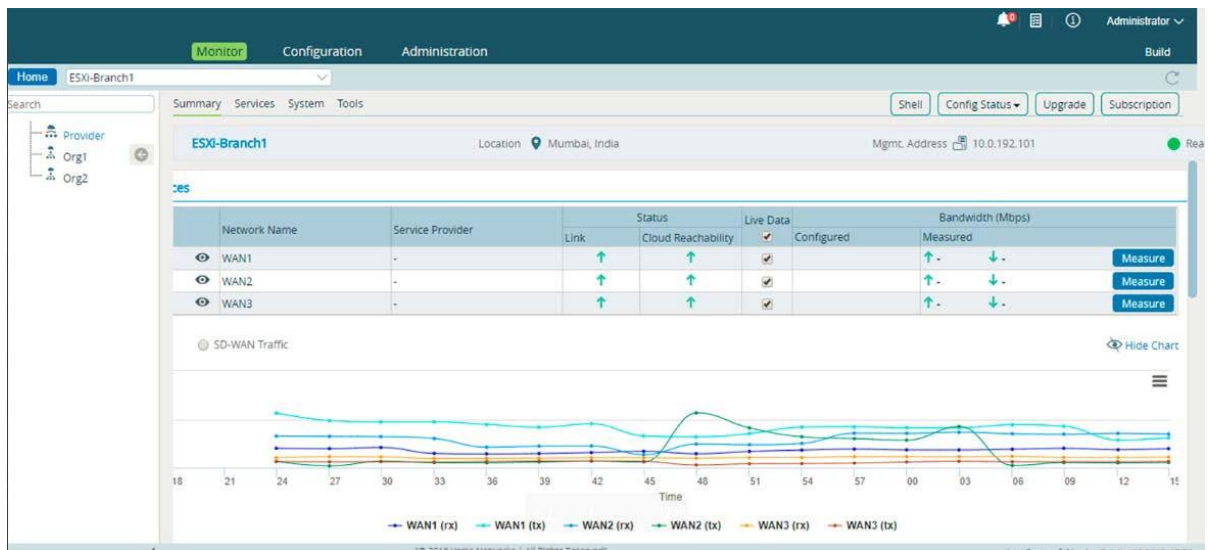
Follow these steps to initiate a speed test from the client GUI:

1. Login to the SteelConnect EX Director.
2. **Select** Administration > Appliances and select a server from the list displayed in the dashboard. This takes you to the selected servers context page.

The screenshot shows the SteelConnect EX Director GUI. The top navigation bar includes 'Monitor', 'Configuration', 'Workflows', 'Administration' (highlighted), and 'Analytics'. The left sidebar shows a tree view with 'Organizations' and 'Appliances' (selected). The main content area displays a table of appliances with the following columns: Name, Mgmt. Address, Type, Service Start T..., Software Version, Site ID, Organizations, Snap..., Config Synchro..., Status, Reachability, Service, and Locked. The table lists six appliances, including ESXi-Branch1, ESXi-Branch2, LTE-Device-901, SDWAN-Controller-1, SPEED-TEST-SERVER, and ST-Branch1.

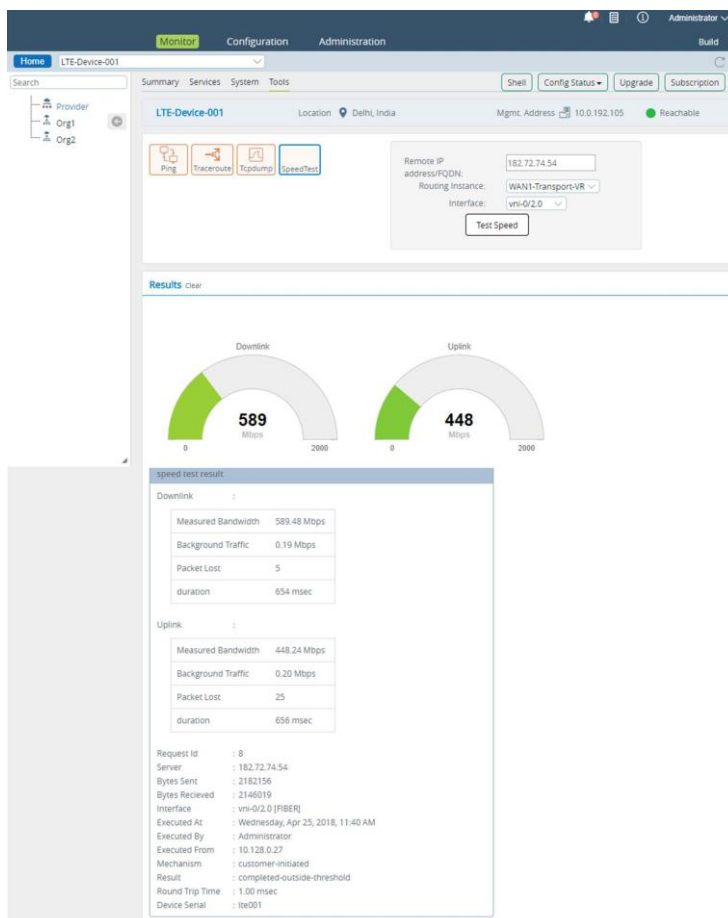
Name	Mgmt. Address	Type	Service Start T...	Software Version	Site ID	Organizations	Snap...	Config Synchro...	Status	Reachability	Service	Locked
ESXi-Branch1	10.0.192.101	Branch	Wed, Apr 18 20...	16.1-R2-S2.1	101	Org1,Org2,Provider			Up	Up	Up	
ESXi-Branch2	10.0.192.104	Branch	Wed, Apr 18 20...	16.1-R2-S2.1	104	Org1,Org2,Provider			Up	Up	Up	
LTE-Device-901	10.0.192.105	Branch			105	Org1,Org2,Provider		Unavailable	Unknown	Unknown	Unknown	
SDWAN-Controller-1	10.192.43.143	Controller	Wed, Apr 18 20...	16.1-R2-S2.1	1	Org1,Org2,Provider			Up	Up	Up	
SPEED-TEST-SERVER	10.128.0.255	Service-vmf				ST-Provider		Unavailable	Unknown	Unknown	Unknown	
ST-Branch1	10.0.192.102	Branch		16.1-R2-S2.1	102	Org1,Org2,Provider		Unknown	Unknown	Unknown	Unknown	

3. In the SD-WAN context, select **Monitor** > **Summary** to view the circuits available with the SD-WAN.



This dashboard displays Measure button only for a WAN. This button is not displayed in the dashboard for a LAN network name.

4. Click **Measure** to initiate the bandwidth measurement speed test.



9.4.2 Using REST API

1. Use this API to request speed test on a client:

SteelConnect EX REST API	API Type
<code>https://<IP-address>:9182/api/config/devices/device/ST-SD-WAN-Branch1/config/orgs/org/Provider/bw-measurement</code>	POST
Payload <pre>{ "bw-measurement": { "local-wan-interface": "vni-0/0.0", "remote-destination": "SPEED-TEST-SERVER", "remote-circuit-name": "WAN3" } }</pre>	

Use this API to poll the client for a status request:

SteelConnect EX REST API	API Type
<code>https://<IP-address>:9182/vnms/dashboard/appliance/ST-SD-WAN-Branch1/live?uuid=3f90c42b-368d-4798-9320-80318aa41fed&command=orgs/org/Provider/sd-wan/bw-measurement/status?deep</code>	GET

Response

```

{
  "collection": {
    "sdwan:status": [
      {
        "site-name": "SPEED-TEST-SERVER",
        "path-bw-status": [
          {
            "path-handle": 6820096,
            "local-site-name": "ST-SD-WAN-Branch1",
            "remote-site-name": "SPEED-TEST-SERVER",
            "local-wan-intf": "vni-0/0.0",
            "local-wan-ckt": "WAN1",
            "remote-wan-ckt": "WAN1",
            "uplink-bw": "524.96 Mbps",
            "downlink-bw": "563.38 Mbps",
            "latency-msec": "1.42",
            "jitter": "1.57",
            "test-status": "failed-background-traffic-exceeded"
          },
          {
            "path-handle": 6820608,
            "local-site-name": "ST-SD-WAN-Branch1",
            "remote-site-name": "SPEED-TEST-SERVER",
            "local-wan-intf": "vni-0/0.0",
            "local-wan-ckt": "WAN1",
            "remote-wan-ckt": "WAN3",
            "uplink-bw": "0.00 Mbps",
            "downlink-bw": "0.00 Mbps",
            "latency-msec": "0.0",
            "jitter": "0.0",
            "test-status": "uplink-in-progress"
          },
          {
            "path-handle": 6824448,
            "local-site-name": "ST-SD-WAN-Branch1",
            "remote-site-name": "SPEED-TEST-SERVER",
            "local-wan-intf": "vni-0/1.0",
            "local-wan-ckt": "WAN2",
            "remote-wan-ckt": "WAN2",
            "uplink-bw": "0.00 Mbps",
            "downlink-bw": "0.00 Mbps",
            "latency-msec": "0.0",
            "jitter": "0.0",
            "test-status": "not-started"
          },
          {
            "path-handle": 6828288,
            "local-site-name": "ST-SD-WAN-Branch1",
            "remote-site-name": "SPEED-TEST-SERVER",
            "local-wan-intf": "vni-0/3.0",
            "local-wan-ckt": "WAN3",
            "remote-wan-ckt": "WAN1",

```

```

        "uplink-bw": "0.00 Mbps",
        "downlink-bw": "0.00 Mbps",
        "latency-msec": "0.0",
        "jitter": "0.0",
        "test-status": "not-started"
    },
    {
        "path-handle": 6828800,
        "local-site-name": "ST-SD-WAN-Branch1",
        "remote-site-name": "SPEED-TEST-SERVER",
        "local-wan-intf": "vni-0/3.0",
        "local-wan-ckt": "WAN3",
        "remote-wan-ckt": "WAN3",
        "uplink-bw": "0.00 Mbps",
        "downlink-bw": "0.00 Mbps",
        "latency-msec": "0.0",
        "jitter": "0.0",
        "test-status": "not-started"
    },
    {
        "path-handle": 6837504,
        "local-site-name": "ST-SD-WAN-Branch1",
        "remote-site-name": "SPEED-TEST-SERVER",
        "local-wan-intf": "vni-0/0.0",
        "local-wan-ckt": "WAN1-v6",
        "remote-wan-ckt": "WAN1-v6",
        "uplink-bw": "0.00 Mbps",
        "downlink-bw": "0.00 Mbps",
        "latency-msec": "0.0",
        "jitter": "0.0",
        "test-status": "not-started"
    },
    {
        "path-handle": 6841856,
        "local-site-name": "ST-SD-WAN-Branch1",
        "remote-site-name": "SPEED-TEST-SERVER",
        "local-wan-intf": "vni-0/1.0",
        "local-wan-ckt": "WAN2-v6",
        "remote-wan-ckt": "WAN2-v6",
        "uplink-bw": "0.00 Mbps",
        "downlink-bw": "0.00 Mbps",
        "latency-msec": "0.0",
        "jitter": "0.0",
        "test-status": "not-started"
    }
]
},
}
}

```

9.4.3 Using CLI

You can also initiate the speed test from a SD-WAN Branch/Device/Hub towards the speed test server. Follow these steps to initiate the speed test from the SDWAN Branch/Device/Hub:

1. Run the request devices device <device-name> config orgs org <org-name> bw- measurement remote-destination <remote-destination site-name> local-wan-interface <interface-name> remote-circuit-name <circuit-name> remote-circuit-media <DSL|LTE|T1|T3|cable|ethernet> remote-circuit-type <Broadband|IP|MPLS> CLI command to request SteelConnect EX speed test from the SD-WAN branch/device/hub.

The executed-by, executed-at, executed-from, and mechanism parameters are filled only when you are using the REST API.

You do not have to enter values for these parameters when using the CLI command.

Use this field...	to...
remote-destination (Mandatory)	Specify the Server IP address. The speed test will not start without this value.
local-wan-interface (optional)	Enter the vni interface that is part of the WAN interface. This forces the test to run from a particular vni interface that is part of the local WAN interface. This is useful when the interface is provided while issuing the CLI command, else the interface is found based on the best routes.
remote-circuit-name	Specify the remote circuit name for the remote destination specified earlier (refer the above parameter).
remote-circuit-media	Select the remote circuit name for the remote destination specified earlier (refer the above parameter). These are the options: → DSL → LTE → T1 → T3 → Cable → Ethernet
remote-circuit-type	Select the remote circuit name for the remote destination specified earlier (refer the above parameter). These are the options: → IP → MPLS

CLI Output

```
request devices device ST-SD-WAN-Branch1 config orgs org tenant1 bw-measurement
remote-destination SPEED-TEST-SERVER local-wan-interface vni-0/0.0 remote-circuit-name WAN1
remote-circuit-media LTE remote-circuit-type Broadband
```

This CLI command issues a request on the SD-WAN branch/Device/Hub.

2. Run the `show devices device <device-name> live-status orgs org <org-name> sd-wan bw-measurement status [remote-site-name]` CLI command to poll the SD-WAN branch and display the result.

CLI Output

```
show devices device ST-SD-WAN-Branch1 live-status orgs org tenant1 sd-wan bw-measurement status
SPEED-TEST-SERVER
sd-wan bw-measurement status SPEED-TEST-SERVER
  path-bw-status 6623488
  local-site-name ST-SD-WAN-Branch1
  remote-site-name SPEED-TEST-SERVER
  local-wan-intf vni-0/0.0
  local-wan-ckt WAN1
  remote-wan-ckt WAN1
  uplink-bw "300.02 Mbps"
  downlink-bw "400.00 Mbps"
  latency-msec 7.2
  jitter 0.5
  test-status completed-within-threshold
```