


```

Local site KEY (IP)           : 10.3.0.106
Neighbour IP                 : 10.3.0.106
Site type                    : SD-WAN
Site ID                      : 6a:00
Site Name                    : SDWAN-Branch1
Branch ID                   : 106
Tenant ID                   : 5
Neighbour mgmt VRF id       : 16
Neighbour global tint id    : 3 (Master Tenant)
Neighbour master global tint id : 3
Neighbour flags              : [ AX]
Neighbour num transport IPs  : 3

WAN lcl vrf-id              : 9
WAN lcl link ifindex        : 1148
WAN lcl link name           : vni-0/0.0
WAN lcl circuit info        : (name: WAN1, provider: , media: Unknown, type: Unknown)
WAN lcl link id             : 1
WAN lcl link behind NAT     : 1
WAN lcl link shaping rate   : 0 (min 0)
WAN lcl link addr(public)   : 192.168.11.101
WAN lcl link addr(priv)     : 192.168.11.101
WAN lcl link flags          : [ BN CT PT SLA-P]
WAN lcl transport domain    : (1) [ 2 ]
WAN SLA interval            : [ ]

WAN lcl vrf-id              : 30
WAN lcl link ifindex        : 1150
WAN lcl link name           : vni-0/1.0
WAN lcl circuit info        : (name: WAN2, provider: , media: Unknown, type: Unknown)
WAN lcl link id             : 2
WAN lcl link behind NAT     : 1
WAN lcl link shaping rate   : 0 (min 0)
WAN lcl link addr(public)   : 192.168.12.101
WAN lcl link addr(priv)     : 192.168.12.101
WAN lcl link flags          : [ BN CT PT SLA-P]
WAN lcl transport domain    : (1) [ 2 ]
WAN SLA interval            : [ ]

WAN lcl vrf-id              : 31
WAN lcl link ifindex        : 1152
WAN lcl link name           : vni-0/2.0
WAN lcl circuit info        : (name: WAN3, provider: , media: Unknown, type: Unknown)
WAN lcl link id             : 3
WAN lcl link behind NAT     : 0
WAN lcl link shaping rate   : 0 (min 0)
WAN lcl link addr(public)   : 192.168.13.101
WAN lcl link addr(priv)     : 192.168.13.101
WAN lcl link flags          : [ CT PT SLA-P]
WAN lcl transport domain    : (1) [ 3 ]
WAN SLA interval            : [ ]

```

4. To check whether the remote site objects were learned from BGP or from configuration, issue the **show vsm p2mp tunnel-remote-endpoint tenant** CLI command on the vsm control plane. For example:

https://docs.versa-networks.com/Secure_SD-WAN/03_Troubleshooting/Troubleshoot_SD-WAN_Data_Path

Updated: Wed, 23 Oct 2024 08:07:02 GMT

Copyright © 2024, Versa Networks, Inc.

```
vsm-vcsn0> show vsm p2mp tunnel-remote-endpoint tenant 3
```

Legend:

AX: Access ckt id's translated

SD: Stale state pending delete

CAP-I: Capability config inherited from master

AP: Access ckt id update pending (child tenant)

Neighbor update max time elapsed: 502 usecs

Neighbour Endpoint: 0 (gen: 2)

```
Neighbour KEY (IP)      : 10.10.64.1
Neighbour IP            : 10.10.64.1
Neighbour sibling IP     : 10.10.0.1
Site type               : SD-WAN
SDWAN Site type         : Controller
Site ID                 : 01:00
Site Name                : SDWAN-Controller1
Branch ID               : 1
Tenant ID               : 3
Neighbour mgmt VRF id   : 12
Neighbour global tnt id : 10 (Master Tenant)
Neighbour master global tnt id : 10
Neighbour OBJID         : 5
Neighbour flags         : [ AX]
Neighbour num transport IPs : 3
Neighbour SA v1 str     :
Neighbour SA v1 len     : 0
Neighbour SA v2 str     :
Neighbour SA v2 len     : 0
Neighbour SA v1         : 0x00000000
Neighbour SA v2         : 0x00000000
Neighbour Ptv Intf      : ptvi20

WAN lcl circuit info    : (name: WAN1, media: Unknown, type: Unknown)
WAN rmt link id        : 1
WAN rmt behind NAT     : 0
WAN rmt link shaping rate : 0 (min 0)
WAN rmt link address (priv): 192.168.211.1
WAN rmt link address (pub): 192.168.211.1
WAN rmt link nat port   : 4790
WAN rmt link flags     : []
WAN rmt transport domain : (1) [ 2 ]
WAN rmt link nat binding : 0
WAN lcl circuit info    : (name: WAN2, media: Unknown, type: Unknown)
WAN rmt link id        : 2
WAN rmt behind NAT     : 0
WAN rmt link shaping rate : 0 (min 0)
WAN rmt link address (priv): 192.168.212.1
WAN rmt link address (pub): 192.168.212.1
WAN rmt link nat port   : 4790
WAN rmt link flags     : []
WAN rmt transport domain : (1) [ 2 ]
WAN rmt link nat binding : 0
WAN lcl circuit info    : (name: WAN3, media: Unknown, type: Unknown)
```

```

WAN rmt link id      : 3
WAN rmt behind NAT   : 0
WAN rmt link shaping rate : 0 (min 0)
WAN rmt link address (priv): 192.168.213.1
WAN rmt link address (pub) : 192.168.213.1
WAN rmt link nat port   : 4790
WAN rmt link flags     : []
WAN rmt transport domain : (1) [ 3 ]
WAN rmt link nat binding : 0

```

Check the vsm Data Plane State

Routes to all remote branches must be present to ensure connectivity among the branches. Note that if no IKE and IPsec tunnel has been established, the ptvi-esp interface toward the Controller is in the Down state and a route to the Controller is not present

To check that routes to all remote branches are present:

1. To display the routes in the core FIB and customer FIB for a given tenant, issue the **show vunet route summary** CLI command. For example:

```
vsm-vcsn0> show vunet 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 vunet 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 vunet 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

- To check the incoming label table in the data path to ensure the correct distribution of labels, issue the **show vsm mpls-label-table** CLI command. For example:

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

MPLS Label Table:

Number of label entries: 42

Label	NH type	FIB	Core FIB	Lcl TNT	Proto	Hit Count
24705	VRF-table-label	13	12	3	IPv4	0
16474	VRF-label-proto	23	22	8	NSH CMN	0
8284	VRF-table-label	27	26	10	IPv4	0
68	VRF-table-label	18	18	6	IPv4	29097
24713	VRF-table-label	29	28	11	IPv4	0
16477	VRF-label-proto	29	28	11	NSH CMN	0
24710	VRF-table-label	23	22	8	IPv4	0
67	VRF-table-label	16	16	5	IPv4	29085
84	VRF-table-label	32	32	12	IPv4	46818
16469	VRF-label-proto	11	10	2	NSH CMN	0
8283	VRF-table-label	25	24	9	IPv4	0
24711	VRF-table-label	25	24	9	IPv4	0
17	Next-proto	0	0	0	Ether	268082
74	VRF-table-label	12	12	3	IPv4	29091
16476	VRF-label-proto	27	26	10	NSH CMN	0
8282	VRF-table-label	23	22	8	IPv4	0
66	VRF-table-label	14	14	4	IPv4	29099

- To check whether the branch table is programmed correctly and to verify that all the configured local site and learned remote branch information is present, issue the **show vsf tunnel branch-table local** CLI command. For example:

```
vsm-vcsn0> show vsf tunnel branch-table local
```

Control thread

<Br ID,Glbl Tnt>	Branch Name	CT PTVI (Overlay IP)	ET PTVI (Overlay IP)	Tnt ID
C-FIB IKE Status(Uptime)(LST)(LLUT)(LCL Site)(Site-type)				
< 106, 1> SDWAN-Branch1		1027 (10.1.0.106)	1031 (10.1.64.106)	2 10
N/A (0s)(0s)(0s)(B) (B)				
< 106, 10> SDWAN-Branch1		1039 (10.10.0.106)	1043 (10.10.64.106)	3
12 N/A (0s)(0s)(0s)(B) (B)				
< 106, 2> SDWAN-Branch1		1049 (10.2.0.106)	1053 (10.2.64.106)	4 14
N/A (0s)(0s)(0s)(B) (B)				
< 106, 3> SDWAN-Branch1		1061 (10.3.0.106)	1065 (10.3.64.106)	5 16
N/A (0s)(0s)(0s)(B) (B)				
< 106, 4> SDWAN-Branch1		1071 (10.4.0.106)	1075 (10.4.64.106)	6 18
N/A (0s)(0s)(0s)(B) (B)				
< 106, 5> SDWAN-Branch1		1081 (10.5.0.106)	1085 (10.5.64.106)	7 20
N/A (0s)(0s)(0s)(B) (B)				
< 106, 6> SDWAN-Branch1		1091 (10.6.0.106)	1095 (10.6.64.106)	8 22
N/A (0s)(0s)(0s)(B) (B)				

```

< 106, 7> | SDWAN-Branch1 | 1101 ( 10.7.0.106) | 1105 ( 10.7.64.106) | 9 | 24
| N/A ( 0s)( 0s)( 0s)( B) ( B) |
< 106, 8> | SDWAN-Branch1 | 1111 ( 10.8.0.106) | 1115 ( 10.8.64.106) | 10 |
26 | N/A ( 0s)( 0s)( 0s)( B) ( B) |
< 106, 9> | SDWAN-Branch1 | 1121 ( 10.9.0.106) | 1125 ( 10.9.64.106) | 11 |
28 | N/A ( 0s)( 0s)( 0s)( B) ( B) |
< 106, 20> | SDWAN-Branch1 | 1139 ( 10.20.0.106) | 1143 ( 10.20.64.106) | 12 |
32 | N/A ( 0s)( 0s)( 0s)( B) ( B) |

```

vsm-vcsn0> **show vsf tunnel branch-table**

Legend:

CT -> Clear Text

ET -> Encrypted Text

C/H -> Local site is Controller/Hub

B -> Local site is Branch

LST -> Last SA-INIT time

LLUT -> Last link update notif time

GT - Global Tenant ID

C-FIB - Core-facing FIB

Control thread

```

=====
<Br ID,Glbl Tnt>| Branch Name | CT PTVI (Overlay IP) | ET PTVI (Overlay IP) | Tnt ID |
C-FIB | IKE Status(Uptime)(LST)(LLUT)(LCL Site)(Site-type) |
=====
< 104, 3> | SDWAN-Branch2 | 1252 ( 10.3.0.104) | 1253 ( 10.3.64.104) | 5 | 16
| N/A ( 0s)( 0s)( 0s)( B)( B) |
< 2, 8> | SDWAN-Controller2 | 1113 ( 10.8.0.2) | 1117 ( 10.8.64.2) | 10 | 26 | N/
A ( 0s)( 0s)( 0s)( B)( C) |
< 108, 7> | SDWAN-Branch4 | 1274 ( 10.7.0.108) | 1275 ( 10.7.64.108) | 9 | 24
| N/A ( 0s)( 0s)( 0s)( B)( B) |
< 108, 6> | SDWAN-Branch4 | 1272 ( 10.6.0.108) | 1273 ( 10.6.64.108) | 8 | 22
| N/A ( 0s)( 0s)( 0s)( B)( B) |
< 2, 9> | SDWAN-Controller2 | 1123 ( 10.9.0.2) | 1127 ( 10.9.64.2) | 11 | 28 |
IKE_UP ( 4389s)( 0s)( 0s)( B)( C) |
< 101, 1> | SDWAN-Branch5 | 1234 ( 10.1.0.101) | 1235 ( 10.1.64.101) | 2 | 10
| N/A (

```

- To check the forwarding plane state of a site and to check the network paths between the local and remote site, issue the **show vsf tunnel access-circuits ptvi brief** CLI command. In this command, use the clear text and cipher text ptvi ifindex from the output of the **show vsf tunnel branch-table** command (shown in Step 3). For example:

vsm-vcsn0> **show vsf tunnel access-circuits ptvi 1253 5 brief**

Legend:

ED: Endpoint Dependent NAT enabled

I: SLA in INIT state

U: SLA in UP state

D: SLA in DOWN state

S: Skip Route IF check

Access Circuits to Neighbor: [Branch-id: 104, core-fib:16, tnt:5, IP:10.3.64.104]

Encap chain info (in order of imposition):

```

Number of Encaps    : 4
Encap 0            : VMLH
Encap 1            : MPLS-over-GRE
Encap 2            : IPSec-ESP
Encap 3            : VXLAN
Max total encap overhead    : 129
Tunnel check for branch/ack/route required : TRUE
Vxlan transport compatibility version    : 2
Crypto operation          : SYNC

SPI Ctxt: 0x0x7fbf8a114e00
Out SPI   : 0x51710003
In SPI    : 0x000e0068

Control Thread:
Default valid transport-path id: 34
Default mgmt transport-path id : N/A
Tunnel MTU          : 1336
SLA Mask over all ackts: 0x0000000000000000, 0x0000000000000000, 0x0000000000000000,
0x0008000600060000
Remote Branch behind NAT    : FALSE
Remote Ackt id's translated : TRUE
Remote Intf has mgmt access  : FALSE
Remote Ackt map atomic refcount : 0
Remote Ackt map packet refcount : 0

ID  Transp Source IP    Destination IP  NAT-P NAT VRF  Cap  Ifldx Pipe  RTGen Flags MgmtP
PMTU EMTU
-----
17(1,1) V4UDP 192.168.11.101 192.168.21.101 4790 N 9  (P,E) 1148 65535 0  U  00 1500
1500
18(1,2) V4UDP 192.168.11.101 192.168.22.101 4790 N 9  (P,E) 1148 65535 0  U  00 1500
1500
33(2,1) V4UDP 192.168.12.101 192.168.21.101 4790 N 30 (P,E) 1150 65535 0  U  00 1500
1500
34(2,2) V4UDP 192.168.12.101 192.168.22.101 4790 N 30 (P,E) 1150 65535 0  U  00 1500
1500
51(3,3) V4UDP 192.168.13.101 192.168.23.101 4790 N 31 (P,E) 1152 65535 0  U  00 1500
1500

```

- To check the session state for transit packets from the client (behind Branch1) to the server (behind Branch2), issue the **show vsf session all detail** CLI command. The output displays information about dropped packets if the session infrastructure dropped any packets. For 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 Tmestamp: 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

```

View vsm Data Plane Statistics

Note that Releases 20.2 and later add support for displaying IP multicast statistics.

To check for packet drops in the data path:

1. Run the **show vsm statistics port** CLI command. For example:

```

vsm-vcsn0> show vsm statistics port
Interface : vni-0/0 (port: 0)
Successfully received packets      : 237560
Successfully transmitted packets   : 230317
Successfully received bytes        : 43859029
Successfully transmitted bytes     : 49224192
Erroneous received packets         : 0
Failed transmitted packets         : 0
RX mbuf allocation failures        : 0
Pause mode                        : 0

Interface : vni-0/1 (port: 1)
Successfully received packets      : 152823

```



```

Successfully transmitted packets      : 162765
Successfully received bytes           : 29468694
Successfully transmitted bytes        : 31093865
Erroneous received packets            : 0
Failed transmitted packets            : 0
RX mbuf allocation failures           : 0
Pause mode                           : 0

```

```

Interface : vni-0/2 (port: 2)
Successfully received packets         : 77542
Successfully transmitted packets      : 78215
Successfully received bytes           : 14980017
Successfully transmitted bytes        : 15045268
Erroneous received packets            : 0
Failed transmitted packets            : 0
RX mbuf allocation failures           : 0
Pause mode                           : 0

```

```

Interface : vni-0/3 (port: 3)
Successfully received packets         : 11649
Successfully transmitted packets      : 6421
Successfully received bytes           : 713704
Successfully transmitted bytes        : 473102
Erroneous received packets            : 0
Failed transmitted packets            : 0
RX mbuf allocation failures           : 0
Pause mode                           : 0

```

```

Interface : vni-0/4 (port: 4)
Successfully received packets         : 0
Successfully transmitted packets      : 0
Successfully received bytes           : 0
Successfully transmitted bytes        : 0
Erroneous received packets            : 0
Failed transmitted packets            : 0
RX mbuf allocation failures           : 0
Pause mode                           : 0

```

2. To check statistics about packets between the infmgr and vsm control threads, issue the **show vsm statistics infmgr** CLI command. For 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 socket error : 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/ack/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 0
18(1,2) 192.168.101.3 192.168.101.104 4790 10 (P,E) 1058 65535 0 I 0 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 0
34(2,2) 192.168.101.103 192.168.101.104 4790 10 (P,E) 1060 65535 0 I 0 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 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 0
18(1,2) 192.168.101.3 192.168.101.104 4790 10 (P,E) 1058 65535 0 I 0 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 0
34(2,2) 192.168.101.103 192.168.101.104 4790 10 (P,E) 1060 65535 0 I 0 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 0
18(1,2) 192.168.101.3 192.168.101.104 4790 10 (P,E) 1058 65535 0 I 0 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 0
34(2,2) 192.168.101.103 192.168.101.104 4790 10 (P,E) 1060 65535 0 I 0 0 0 0 0

```

- To check whether the remote endpoint is behind a NAT and to check the translated IP address and port number, issue the **show vsf tunnel nat-info ptvi detail** CLI command. For example:

```
vsm-vcsn0> show vsf tunnel nat-info ptvi 1117 10 detail
```

```
Access Circuit's NAT info for Neighbor: [Branch-id: 2, core-fib:26, tnt:10, IP:10.8.64.2]
```

```
-----
Control Thread, Branch-id: 2, core-fib:26, tnt:10, IP:10.8.64.2
```

```
-----
AC |L-VBP|R-VBP| Idx | Priv-Dest IP (dport) | Public-Dest IP | Dport | ED-IP |ED-Port| DP | SLA-mask |
-----
```

```

17| 1 | 0 |PUB-1| 192.168.221.1 ( 4790) | => 192.168.221.1 | 4790 | 192.168.221.1 | 4790 | 0 | 0x0000 |
-----
18| 1 | 0 |PUB-1| 192.168.222.1 ( 4790) | => 192.168.222.1 | 4790 | 192.168.222.1 | 4790 | 0 | 0x0000 |
-----
33| 1 | 0 |PUB-1| 192.168.221.1 ( 4790) | => 192.168.221.1 | 4790 | 192.168.221.1 | 4790 | 0 | 0x0000 |
-----
34| 1 | 0 |PUB-1| 192.168.222.1 ( 4790) | => 192.168.222.1 | 4790 | 192.168.222.1 | 4790 | 0 | 0x0000 |
-----
51| 0 | 0 |PUB-1| 192.168.223.1 ( 4790) | => 192.168.223.1 | 4790 | 192.168.223.1 | 4790 | 0 | 0x0000 |
-----

```

Worker Thread: 0, Branch-id: 2, core-fib:26, tnt:10, IP:10.8.64.2

```

AC |L-VBP|R-VBP| Idx | Priv-Dest IP (dport) | Public-Dest IP | Dport | ED-IP |ED-Port| DP | SLA-mask |
-----
17| 1 | 0 |PUB-1| 192.168.221.1 ( 4790) | => 192.168.221.1 | 4790 | 192.168.221.1 | 4790 | 0 | 0x0000 |
-----
18| 1 | 0 |PUB-1| 192.168.222.1 ( 4790) | => 192.168.222.1 | 4790 | 192.168.222.1 | 4790 | 0 | 0x0000 |
-----
33| 1 | 0 |PUB-1| 192.168.221.1 ( 4790) | => 192.168.221.1 | 4790 | 192.168.221.1 | 4790 | 0 | 0x0000 |
-----
34| 1 | 0 |PUB-1| 192.168.222.1 ( 4790) | => 192.168.222.1 | 4790 | 192.168.222.1 | 4790 | 0 | 0x0000 |
-----
51| 0 | 0 |PUB-1| 192.168.223.1 ( 4790) | => 192.168.223.1 | 4790 | 192.168.223.1 | 4790 | 0 | 0x0000 |
-----

```

Worker Thread: 1, Branch-id: 2, core-fib:26, tnt:10, IP:10.8.64.2

```

AC |L-VBP|R-VBP| Idx | Priv-Dest IP (dport) | Public-Dest IP | Dport | ED-IP |ED-Port| DP | SLA-mask |
-----
17| 1 | 0 |PUB-1| 192.168.221.1 ( 4790) | => 192.168.221.1 | 4790 | 192.168.221.1 | 4790 | 0 | 0x0000 |
-----
18| 1 | 0 |PUB-1| 192.168.222.1 ( 4790) | => 192.168.222.1 | 4790 | 192.168.222.1 | 4790 | 0 | 0x0000 |
-----
33| 1 | 0 |PUB-1| 192.168.221.1 ( 4790) | => 192.168.221.1 | 4790 | 192.168.221.1 | 4790 | 0 | 0x0000 |
-----
34| 1 | 0 |PUB-1| 192.168.222.1 ( 4790) | => 192.168.222.1 | 4790 | 192.168.222.1 | 4790 | 0 | 0x0000 |
-----
51| 0 | 0 |PUB-1| 192.168.223.1 ( 4790) | => 192.168.223.1 | 4790 | 192.168.223.1 | 4790 | 0 | 0x0000 |
-----

```

- To check the available turn relays in case the branch is behind an ED NAT box, issue the **show vsf tunnel stun-info tenant** CLI command. For example:

```
vsm-vcsn0> show vsf tunnel stun-info tenant 2
```

STUN Info for Tenant: 2

Control thread [0] :

```
| Pri [ 0] |
```

```

|          Stun-Group   |          Stun-hndl   |          State   |
|-----|-----|-----|

```

Default-Controller	273	[1,0x11]	Connected (Active)	
Default-Controller	274	[1,0x12]	Connected	
Default-Controller	529	[2,0x11]	Connected	
Default-Controller	530	[2,0x12]	Connected	
Default-Controller	289	[1,0x21]	Connected	
Default-Controller	290	[1,0x22]	Connected	
Default-Controller	545	[2,0x21]	Connected	
Default-Controller	546	[2,0x22]	Connected	
Default-Controller	307	[1,0x33]	Connected	
Default-Controller	563	[2,0x33]	Connected	
	27665	[108,0x11]	Connected	
	27666	[108,0x12]	Connected	
	27681	[108,0x21]	Connected	
	27682	[108,0x22]	Connected	
	27699	[108,0x33]	Connected	
	25873	[101,0x11]	Connected	
	25874	[101,0x12]	Connected	
	25889	[101,0x21]	Connected	
	25890	[101,0x22]	Connected	
	25907	[101,0x33]	Connected	
	26641	[104,0x11]	Connected	
	26642	[104,0x12]	Connected	
	26657	[104,0x21]	Connected	
	26658	[104,0x22]	Connected	
	26675	[104,0x33]	Connected	

Current Active STUN Server : 1

Current local acct active mask : 0xe000

Current stun_hdl : 273,289,307,0,0,0,0,0,0,0,0,0,0,0,0,

Worker thread [0] :

| Pri [0] |

Stun-Group	Stun-hndl	State	
Default-Controller	273 [1,0x11]	Connected	(Active)
Default-Controller	274 [1,0x12]	Connected	
Default-Controller	529 [2,0x11]	Connected	
Default-Controller	530 [2,0x12]	Connected	
Default-Controller	289 [1,0x21]	Connected	
Default-Controller	290 [1,0x22]	Connected	
Default-Controller	545 [2,0x21]	Connected	
Default-Controller	546 [2,0x22]	Connected	
Default-Controller	307 [1,0x33]	Connected	
Default-Controller	563 [2,0x33]	Connected	
	27665 [108,0x11]	Connected	
	27666 [108,0x12]	Connected	
	27681 [108,0x21]	Connected	
	27682 [108,0x22]	Connected	
	27699 [108,0x33]	Connected	
	25873 [101,0x11]	Connected	
	25874 [101,0x12]	Connected	
	25889 [101,0x21]	Connected	
	25890 [101,0x22]	Connected	
	25907 [101,0x33]	Connected	
	26641 [104,0x11]	Connected	
	26642 [104,0x12]	Connected	
	26657 [104,0x21]	Connected	
	26658 [104,0x22]	Connected	

```
|          | 26675 [ 104,0x33] | Connected |
|-----|
Current Active STUN Server   : 1
Current local ackt active mask : 0xe000
Current stun_hdl             : 273,289,307,0,0,0,0,0,0,0,0,0,0,0,0,
```

Supported Software Information

Releases 20.2 and later support all content described in this article.