Troubleshooting

This chapter provides information and instructions for using the system command line interface (CLI) for troubleshooting any issues that may arise during system operation.

Verifying Network Connectivity

There are multiple commands supported by the system to verify and/or troubleshoot network connectivity. Note that network connectivity can only be tested once system interfaces and ports have been configured and bound.

The commands specified in this section should be issued on a context-by-context basis. Contexts act like virtual private networks (VPNs) that operate independently of other contexts. Ports, interfaces, and routes configured in one context cannot be tested from another context without additional configuration.

To switch between contexts enter the following command at the root prompt for the Exec mode:

`[local]host_name# context context_name`

context_name is the name of the context to which you wish to switch. The following prompt appears:

`[context_name]host_name#`

Using the ping or ping6 Command

The **ping** or **ping6** command verifies the system's ability to communicate with a remote node in the network by passing data packets between and measuring the response. This command is useful in verifying network routing and if a remote node is able to respond at the IP layer.

For VPC-DI deployments, use the Exec mode command **system ping** to verify the internal network (DI-network) connectivity between the VPC-DI VMs. Refer to the *Command Line Interface Reference* for more details.

Syntax

The **ping** command has the following syntax:

ping *host_ipv4_address* **[count** *num_packets* **] [flood] [pattern** *packet_pattern* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv4_address* **}] [vrf** *vrf_nam* **]**

ping6 *host_ipv6_address* **[count** *num_packets* **] [flood][pattern** *packet_pattern* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv6_address* **}] [vrf** *vrf_nam* **]**

For complete information on the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

The following displays a sample of a successful **ping** (IPV4) response.

```
PING 192.168.250.1 (192.168.250.1): 56 data bytes
64 bytes from 192.168.250.1: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 192.168.250.1: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=4 ttl=255 time=0.2 ms
--- 192.168.250.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.4 ms
```

Troubleshooting

If no response is received from the target follow these troubleshooting procedures:

- Verify that the correct IP address was entered.
- Attempt to ping a different device on the same network. If the ping was successful then it is likely that your system configuration is correct. Verify that the device you are attempting to ping is powered and functioning properly.
- Verify the port is operational.
- Verify that the configuration of the ports and interfaces within the context are correct.
- If the configuration is correct and you have access to the device that you're attempting to ping, ping the system from that device.
- If there is still no response, it is likely that the packets are getting discarded by a network device. Use the **traceroute** or **traceroute6** and **show ip static-route** commands discussed in this chapter to further troubleshoot the issue.

Using the traceroute or traceroute6 Command

The **traceroute** or **traceroute6** command collects information on the route data will take to a specified host. This is a useful troubleshooting command that can be used to identify the source of significant packet delays or packet loss on the network. This command can also be used to identify bottle necks in the routing of data over the network.

traceroute – IPv4

The **traceroute** command has the following syntax:

traceroute { *host_name* | *host_ipv4_address* } [**count** *packets*] [**df**] [**maxttl** *max_ttl*] [**minttl** *min_ttl*] [**port** *port_number*] [**size** *octet_count*] [**src** { *src_host_name* | *src_host_ipv4_address* }] [**timeout** *seconds*] [**vrf** *vrf_nam*]

For complete information on the above command, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

The following displays a sample output.

```
traceroute to 192.168.250.1 (192.168.250.1), 30 hops max, 40 byte packets
 1  192.168.250.1 (192.168.250.1)  0.446 ms  0.235 ms  0.178 ms
```

traceroute6 – IPv6

The **traceroute6** command has the following syntax:

traceroute6 { *host_name* | *host_ipv6_address* } [**count** *packets*] [**maxttl** *max_ttl*] [**port** *port_number*] [**size** *octet_count*] [**src** { *src_host_name* | *src_host_ipv6_address* }] [**timeout** *seconds*] [**vrf** *vrf_nam*]

For complete information on the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

The following displays a sample output.

```
traceroute6 to 2001:4A2B::1f3F (2001:4A2B::1f3F), 30 hops max, 40 byte packets
 1  2001:4A2B::1f3F (2001:4A2B::1f3F)  0.446 ms  0.235 ms 0.178 ms
```

Viewing IP Routes

The system provides a mechanism for viewing route information to a specific node or for an entire context. This information can be used to verify network connectivity and to ensure the efficiency of the network connection. The command has the following syntax:

show ip route [*route_ip_address*]
show ipv6 route [*route_ipv6_address*]]

For complete information on the above commands, see the *Exec Mode show Commands* chapter of the *Command Line Interface Reference*.

If no keywords are specified, all IP routes within the context's routing table are displayed.

The following displays a sample of this command's output showing a context IPv4 routing table.

```
"*" indicates the Best or Used route.
  Destination     Nexthop     Protocol    Prec    Cost    Interface
*0.0.0.0/0        10.0.4.1    static      0       0       SPIO1
*10.0.4.0/24      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.0/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.3/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.255/32    0.0.0.0     kernel      0       0       SPIO1
```

Viewing the Address Resolution Protocol Table

The system provides a mechanism for viewing Address Resolution Protocol (ARP) table information to a specific node or for an entire context. This information can be used to verify that when the system sends an ARP packet, it receives valid responses from other network nodes.

`[local]host_name# show ip arp [ arp_ip_address ]`

arp_ip_address specifies a specific network node for which to display ARP information. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation. If this keyword is not specified, all entries within the context's ARP table are displayed.

Important Restarting the VPN Manager removes all interfaces from the kernel which in turn removes all ARP entries. However, the NPU still retains all of the ARP entries so that there is no traffic disruption. From a user point of view, **show ip arp** is broken since this command gathers information from the kernel and not the NPU.

The following displays a sample of this command's output showing a context's ARP table.

```
Flags codes:
C - Completed, M - Permanent, P - Published, ! - Not answered
T - has requested trailers
Address         Link Type    Link Address          Flags    Mask Interface
10.0.4.240      ether        00:05:47:02:20:20     C        MIO1
10.0.4.7        ether        00:05:47:02:03:36     C        MIO1
10.0.4.1        ether        00:01:30:F2:7F:00     C        MIO1
```

Using the System Diagnostic Utilities

The system provides protocol monitor and test utilities that are useful when troubleshooting or verifying configurations. The information generated by these utilities can help identify the root cause of a software or network configuration issue.

This section describes how to use these utilities.

Important Only an administrator with Operator or higher privilege can run the diagnostic utilities described in this section.

Using the Monitor Utility

For troubleshooting purposes, the system provides a protocol monitoring utility. This tool displays protocol information for a particular subscriber session or for every session being processed.

Caution The monitor tool may cause session processing delays and/or data loss. Therefore, it should be used only when troubleshooting.

Using the Protocol Monitor

The protocol monitor displays information for every session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Refer also to Packet Capture (PCAP) Trace to enable PCAP functionality for the **monitor protocol** and **monitor subscriber** commands.

Follow the instructions below to invoke and configure the protocol monitoring tool.

Step 1 Invoke the protocol monitor from the Exec mode by entering the **monitor protocol** command.

```
[local]host_name# monitor protocol
```

An output listing all the currently available protocols, each with an assigned number, is displayed.

Step 2 Choose the protocol that you wish to monitor by entering the associated number at the *Select*: prompt. A right arrow (>) appears next to the protocol you selected.

Step 3 Repeat *step 2* as needed to choose multiple protocols.

Step 4 Press **B** to begin the protocol monitor.

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE
Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!
(Under heavy call load, some debugging output may not be displayed)
Proceed? - Select (Y)es or (N)o
```

Step 5 Enter **Y** to proceed with the monitor or **N** to go back to the previous menu.

```
C - Control Events    (ON )
D - Data Events               (ON )
E - EventID Info              (ON )
H - Display ethernet          (ON )
I - Inbound Events            (ON )
O - Outbound Events           (ON )
S - Sender Info               (OFF)
T - Timestamps                (ON )
X - PDU Hexdump               (OFF)
A - PDU Hex/Ascii             (OFF)
+/- Verbosity Level           (    1)
L - Limit Context             (OFF)
M - Match Newcalls            (ON )
R - RADIUS Dict               (no-override)
G - GTPP Dict                 (no-override)
Y - Multi-Call Trace          ((OFF))
      (Q)uit,      <ESC> Prev Menu,      <SPACE> Pause,      <ENTER> Re-Display Options
```

Step 6 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter associated with that option (C, D, E, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Step 7 Press the **Enter** key to refresh the screen and begin monitoring.

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Using the Protocol Monitor for a Specific Subscriber

The protocol monitor can be used to display information for a specific subscriber session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions in this section to invoke and configure the protocol monitoring tool for a specific subscriber session.

Step 1 To invoke the session-specific protocol monitor from the Exec mode enter the **monitor subscriber** command.

`[local]host_name# monitor subscriber { callid | imei | imsi | ipaddr | ipv6addr | msid | msisdn | next-call | pcf | peer-fa | peer-lac | sgsn-address | type | username }`

Step 2 Specify the method the monitor should use by entering the appropriate keyword.

Step 3 Select other options and/or enter the appropriate information for the selected keyword.

If no session matching the specified criteria was being processed when the monitor was invoked, a screen of available monitoring options appears.

Step 4 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter or 2-digit number associated with that option (C, D, E, 11, 12, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Option **Y** for performing multi-call traces is only supported for use with the GGSN.

Step 5 Repeat *step 6* as needed to enable or disable multiple protocols.

Step 6 Press **Enter** to refresh the screen and begin monitoring.

The following displays a portion of a sample of the monitor's output for a subscriber named *user2@aaa*. The default protocols were monitored.

```
------------------------------------------------------------------------------
Incoming Call:
------------------------------------------------------------------------------
    MSID: 0000012345 Callid: 002dc6c2
    Username: user2@aaa SessionType: unknown
    Status: Active Service Name: xxx1
    Src Context: source Dest Context:
------------------------------------------------------------------------------

<<<<OUTBOUND 10:02:35:415 Eventid:25001(0)
PPP Tx PDU (9)
PAP 9: Auth-Ack(1), Msg=

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Req(1), IP-Addr=192.168.250.70

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (27)
CCP 27: Conf-Req(1), MPPC, Stac-LZS, Deflate, MVRCA

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (30)
```

```
IPCP 30: Conf-Req(1), IP-Comp VJ-Comp, IP-Addr=0.0.0.0, Pri-DNS=0.0.0.0,
Sec-DNS=0.0.0.0

<<<<OUTBOUND 10:02:35:517 Eventid:25001(0)
PPP Tx PDU (26)
IPCP 26: Conf-Rej(1), IP-Comp VJ-Comp, Pri-DNS=0.0.0.0, Sec-DNS=0.0.0.0

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Ack(1), IP-Addr=192.168.250.70

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (31)
LCP 31: Prot-Rej(1), Rejected-Protocol=CCP (0x80fd)

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(2), IP-Addr=0.0.0.0

<<<<OUTBOUND 10:02:35:518 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Nak(2), IP-Addr=192.168.250.87

INBOUND>>>>> 10:02:35:519 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(3), IP-Addr=192.168.250.87
```

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Generating an SSD

An SSD is an instance of the output when the Exec mode **show support details** command is run. It displays a comprehensive list of system information that is useful for troubleshooting purposes. In most cases, the output of this command is requested by the Technical Assistance Center (TAC).

An SSD output .tar file can redirected to a local or remote location (URL).

The .tar file includes:

- **support_summary** - An ASCII text file that contains the support detail information.
- **information.minicores.tar** - A .tar file that contains any minicore files found on the system. Minicore files contain memory core dumps that are captured during some events. These core dumps provide specific memory locations and other information about the event. This information is useful to the technical support team in identifying where and when an event occurred along with its probably cause.

The **show support details** command includes information that is not otherwise accessible to users but that is helpful in the swift resolution of issues by TAC.

Important Platforms with large configuration files can take up to 30 minutes to complete an SSD. Executing the **show support details** command consumes system resources and may reduce traffic throughput.

For releases prior to 20.0, an operator could initiate another SSD while an SSD was already running. In a large configuration, the SSD request often timed out while waiting for the first one to complete. The operator

was not aware of the failure until the entire timeout expired. An operator could have more than one SSD running simultaneously.

For release 20.0 and higher, if an SSD is in progress when the operator enters the **show support details** command, StarOS responds with a warning message stating that an SSD is already in progress and the user should try again later. The operator is restricted to running only one SSD instance at a time.

There are optional keywords to the **show support details** command that can target the SSD to only report specific type of information. These keywords can reduce the amount of time required to generate the SSD/

For additional information about the **show support details** command, see the *Exec Mode show Commands (Q-S)* chapter in the *Command Line Interface Reference*.

Configuring and Using the Support Data Collector

The task of collecting the support data is performed by a background CLI task called the record collector. The administrator configures the Support Data Collector (SDC) via the CLI with the commands to be executed on a periodic basis. The record collector always runs in the background and checks if there are records to be collected.

When it is time to collect support data, the scheduler executes the configured sequence of CLI commands and stores the results in a gunzipped (.gz) file on the hard-disk. This file is called an SDR (Support Data Record), and represents a snapshot of the overall state of the system at that time.

Technical Assistance Center (TAC) personnel and local administrators can review the SDRs on-line or by transferring them off the system. They may also wish to investigate the collector state information.

Refer to the *Support Data Collector* chapter for a complete description of SDC functionality.

Hypervisor Forced Reboot

The hypervisor is required to provide a virtual watchdog device. If StarOS stops servicing this watchdog device because of an underlying host OS failure, the hypervisor must forcefully reboot the VM.

In the virtualized chassis that is a VPC-DI instance, there is no way for the CF to remotely force a reboot of another VM resulting from a host failure. Under normal conditions, the CF can remotely reboot another VM by sending a message to StarOS running on that VM. However, under host failure conditions this may not work and the hypervisor watchdog must function as a fall-back mechanism.

Under KVM, a virtual watchdog device can be provided using the "--watchdog i6300esb" command line arguments.

VMware provides a proprietary watchdog mechanism refer to VMware documentation for additional information.

Table 1: Failure Conditions and Reboot Methods

Failure Condition	Reboot Method	Recovery	Notes
card reboot *x*	CF tells remote StarOS to do local reboot.		

Failure Condition	Reboot Method	Recovery	Notes
Critical task failure	Hypervisor watchdog	Hypervisor reboots VM	StarOS HATCPU task stops servicing the watchdog.
Kernel hang/crash	Kernel reboot or Hypervisor watchdog	Hypervisor reboots VM	
Host failure	Hypervisor HA	CF tells remote StarOS to do local reboot	Reference: a VMware HA cluster.

Manually Switching to a Standby CF

You can manually force the active CF "card" to switch to the standby CF in the redundant pair.

The following Exec mode CLI switches the active CF to the standby CF:

`card switch to` *`slot`*

Notes:

- This command only works for redundant CF VMs in a VPC-DI instance.
- You must be logged into the Active CF to initiate the switchover.
- *slot* can be 1 or 2.

The switchover process proceeds as follows:

1. Active CF transitions to Standby mode.
2. Standby CF transitions to Active mode.
3. A reload is initiated on the formerly Active CF (now the Standby CF).