C H A P T E R **7**

Troubleshooting High Availability Issues

This chapter describes how to identify and resolve problems related to high availability (HA).

This chapter includes the following sections:

Information About Cisco VSG High Availability

Cisco VSG high availability (HA) is a subset of NX-OS HA. The following HA features minimize or prevent traffic disruption in the event of a failure:

Redundancy

Cisco VSG redundancy is equivalent to HA pairing. The two possible redundancy states are **active** and **standby**. An active VSG is always paired with a standby Cisco VSG. HA pairing is based on the Cisco VSG ID. Two Cisco VSGs assigned the identical ID are automatically paired. All processes running in the Cisco VSG are data path critical. If one process fails in an active Cisco VSG, failover to the standby Cisco VSG occurs instantly and automatically.

Isolation of Processes

The Cisco VSG software contains independent processes, known as services. These services perform a function or set of functions for a subsystem or feature set of Cisco VSG. Each service and service instance runs as an independent, protected process. This operational process ensure highly fault-tolerant

Send document comments to vsg-docfeedback@cisco.com.

software infrastructure and fault isolation between services. Any failure in a service instance does not affect any other services running at that time. Additionally, each instance of a service runs as an independent process. For example, two instances of a routing protocol run as separate processes.

Cisco VSG Failovers

The Cisco VSG HA pair configuration allows uninterrupted traffic forwarding by using stateful failover when a failure occurs.

Problems with High Availability

Following are the some key problems found in Cisco VSG HA. In addition to these, some of the common NX-OS HA symptoms are grouped in Table 7-1. Provided are symptoms related to high availability, their possible causes, and recommended solutions.

- Cisco VSG pair communication problems

- Cisco VSGs do not reach active/standby status

- Sometimes the Cisco VSG reboots continuously when tenants share the management network (for example, collisions of the domain ID):

  - In the multi-tenant environment, if there is a shared management network and if a collision occurs in the domain ID (two or more tenants using the same domain ID), that triggers spontaneous reboots of the Cisco VSGs involved in collision. There is no workaround for this issue. When domain IDs are provisioned, they must be unique across all tenants

- Cisco VSGs in the HA pair get stuck in bash# prompt during reboots/upgrades/switchovers

- Cisco VSGs in the HA pair get stuck in boot# prompt during reboots/upgrades/switchovers

Table 7-1   Problems with High Availability

Symptom	Possible Cause	Solution
The active Cisco VSG does not see the standby Cisco VSG.	Roles are not configured properly: • primary • secondary	Verify roles, update an incorrect role, and save the configuration. 1. Verify the role of each Cisco VSG by entering the **show system redundancy status** command. 2. Update an incorrect role by entering the **system redundancy role** command. 3. Save the configuration by entering the **copy run start** command.
	Network connectivity problems between the Cisco VSG and the upstream and virtual switches. Problem could be in the control or management VLAN.	Restore connectivity. 1. From the vSphere client, shut down the Cisco VSG, which should be in standby mode. 2. From the vSphere client, bring up the standby Cisco VSG after network connectivity is restored.

Table 7-1   Problems with High Availability (continued)

Symptom	Possible Cause	Solution
The active Cisco VSG does not complete synchronization with the standby Cisco VSG.	Version mismatch between Cisco VSGs.	Verify that the Cisco VSGs are using the same software version. If not, then reinstall the image. **1.** Verify software version on both Cisco VSGs by entering the **show version** command. **2.** Reinstall the secondary Cisco VSG with the same version used in the primary.
	Fatal errors during gsync process. • Check the gsyncctrl log by entering the **show system internal log sysmgr gsyncctrl** command and look for fatal errors.	Reload the standby Cisco VSG by entering the **reload module** *standby_module_number* command. See the "Reloading a Module" section on page 7-8.
The standby Cisco VSG reboots periodically.	The Cisco VSG has connectivity only through the management interface. When a Cisco VSG is able to communicate through the management interface, but not through the control interface, the active Cisco VSG resets the standby to prevent the two Cisco VSGs from being in HA mode and out of sync.	Check control VLAN connectivity between the primary and secondary Cisco VSG by entering the **show system internal redundancy info** command. In the output, **degraded_mode flag** = true. If there is no connectivity, restore it through the control interface.
Both Cisco VSGs are in active mode.	Network connectivity problems. • Check for control and management VLAN connectivity between the Cisco VSG at the upstream and virtual switches. • When the Cisco VSG cannot communicate through any of these two interfaces, they will both try to become active.	If network problems exist: **1.** From the vSphere client, shut down the Cisco VSG, which should be in standby mode. **2.** From the vSphere client, bring up the standby Cisco VSG after network connectivity is restored.

Table 7-1   Problems with High Availability (continued)

Symptom	Possible Cause	Solution

Active and standby Cisco VSGs are not synchronized	**Incompatible versions**
	
	The boot variables for active and standby Cisco VSGs are set to different image names, or if image names are the same, the files are not the correct files.
	
	When active and standby Cisco VSGs are running different versions that are not HA compatible, they are unable to synchronize.
		Update the software version or the boot variables.
		
		1. From each Cisco VSG (active and standby), verify the software version by entering the **show version** command.
		
		2. Reload the standby Cisco VSG with the version that is running in the active by doing one of the following:
		
		   – correcting the boot variable names
		
		   – replacing the incorrect software files
		
		See the "Reloading a Module" section on page 7-8.

	Broadcast traffic problem: Broadcast traffic from standby to active Cisco VSG may prevent the Cisco VSGs from synchronizing. The standby Cisco VSG tries to contact the active Cisco VSG periodically, but if broadcast traffic problems persist for over a minute when the standby is booting up, the system cannot synchronize.	Fix the traffic problem and reload the standby Cisco VSG. 1. From the standby Cisco VSG, verify the broadcast traffic problem by entering the **show system internal log sysmgr verctrl** command. If so, the following message will be displayed: `standby_verctrl: no response from the active System Manager` 2. Fix network connectivity. 3. Reload standby Cisco VSG by entering the **reload module** *standby_module_number* command. See the "Reloading a Module" section on page 7-8.

	False standby removal
	
	The active Cisco VSG falsely detects a disconnect with the standby. The standby is removed and reinserted and synchronization does not occur.
		Verify redundancy states and reload the standby Cisco VSG.
		
		1. Verify active Cisco VSG redundancy by entering the **show system internal redundancy status** command. Output is as follows:
		
		`RDN_DRV_ST_AC_NP`
		
		2. Verify standby Cisco VSG redundancy by entering the **show system internal redundancy status** command. Output is:
		
		`RDN_DRV_ST_SB_AC`
		
		3. Reload the standby Cisco VSG by entering the **reload module** *standby_module_number* command.
		
		See the "Reloading a Module" section on page 7-8.

High Availability Troubleshooting Commands

This section lists commands that can be used troubleshoot problems related to high availability.

This section includes the following topics:

Checking the Domain ID of the Cisco VSG

Returns the domain ID information by entering the **show vsg** command.

This example shows the output for the command:

```
vsg# show vsg
Model: VSG
HA ID: 3000
VSG Software Version: 4.2(1)VSG1(1) build [4.2(1)VSG1(1)]
VNMC IP: 10.193.75.145
vsg#
```

Checking Redundancy

This section includes the following topics:

Checking the System Redundancy Status

Check the system redundancy status by entering the **show system redundancy status** command.

This example shows the output for the command:

```
vsg# show system redundancy status
Redundancy role
---------------
    administrative:   primary <-- Configured redundancy role
       operational:   primary <-- Current operational redundancy role

Redundancy mode
---------------
    administrative:   HA
       operational:   HA

This supervisor (sup-1)
-----------------------
   Redundancy state:   Active <-- Redundancy state of this VSG
   Supervisor state:   Active
     Internal state:   Active with HA standby
```

```
Other supervisor (sup-2)
-----------------------
    Redundancy state:   Standby <-- Redundancy state of the other VSG
    Supervisor state:   HA standby
      Internal state:   HA standby <-- The standby VSG is in HA mode and in sync
vsg#
```

Checking the System Internal Redundancy Status

Check the system internal redundancy status by entering the **show system internal redundancy info** command.

This example shows the output for the command:

```
vsg# show system internal redundancy info
My CP:
  slot: 0
  domain: 184 <-- Domain id used by this VSG
  role:   primary <-- Redundancy role of this VSG
  status: RDN_ST_AC <-- Indicates redundancy state (RDN_ST) of the this VSG is Active (AC)
  state:  RDN_DRV_ST_AC_SB
  intr:   enabled
  power_off_reqs: 0
  reset_reqs:     0
Other CP:
  slot: 1
  status: RDN_ST_SB <-- Indicates redundancy state (RDN_ST) of the other VSG is Standby
(SB)
  active: true
  ver_rcvd: true
  degraded_mode: false <-- When true, it indicates that communication through the control
interface is faulty
Redun Device 0: <-- This device maps to the control interface
  name: ha0
  pdev: ad7b6c60
  alarm: false
  mac: 00:50:56:b7:4b:59
  tx_set_ver_req_pkts:   11590
  tx_set_ver_rsp_pkts:   4
  tx_heartbeat_req_pkts: 442571
  tx_heartbeat_rsp_pkts: 6
  rx_set_ver_req_pkts:   4
  rx_set_ver_rsp_pkts:   1
  rx_heartbeat_req_pkts: 6
  rx_heartbeat_rsp_pkts: 442546 <-- Counter should be increasing, as this indicates that
communication between VSG is working properly.
  rx_drops_wrong_domain: 0
  rx_drops_wrong_slot:   0
  rx_drops_short_pkt:    0
  rx_drops_queue_full:   0
  rx_drops_inactive_cp:  0
  rx_drops_bad_src:      0
  rx_drops_not_ready:    0
  rx_unknown_pkts:       0
Redun Device 1: <-- This device maps to the mgmt interface
  name: ha1
  pdev: ad7b6860
  alarm: true
  mac: ff:ff:ff:ff:ff:ff
  tx_set_ver_req_pkts:   11589
  tx_set_ver_rsp_pkts:   0
  tx_heartbeat_req_pkts: 12
  tx_heartbeat_rsp_pkts: 0
```

```
  rx_set_ver_req_pkts:   0
  rx_set_ver_rsp_pkts:   0
  rx_heartbeat_req_pkts: 0
  rx_heartbeat_rsp_pkts: 0 <-- When communication between VSG through the control
interface is interrupted but continues through the mgmt interface, the
rx_heartbeat_rsp_pkts will increase.
  rx_drops_wrong_domain: 0
  rx_drops_wrong_slot:   0
  rx_drops_short_pkt:    0
  rx_drops_queue_full:   0
  rx_drops_inactive_cp:  0
  rx_drops_bad_src:      0
  rx_drops_not_ready:    0
  rx_unknown_pkts:       0
vsg#
```

Checking the System Manager State

Check the system internal sysmgr state by entering **show system internal sysmgr state** command.

This example shows the output for the command:

```
vsg# show system internal sysmgr state

The master System Manager has PID 1988 and UUID 0x1.
Last time System Manager was gracefully shutdown.
The state is SRV_STATE_MASTER_ACTIVE_HOTSTDBY entered at time Tue Apr 28 13:09:13 2009.

The '-b' option (disable heartbeat) is currently disabled.

The '-n' (don't use rlimit) option is currently disabled.

Hap-reset is currently enabled.

Watchdog checking is currently disabled.

Watchdog kgdb setting is currently enabled.


        Debugging info:

The trace mask is 0x00000000, the syslog priority enabled is 3.
The '-d' option is currently disabled.
The statistics generation is currently enabled.


        HA info:

slotid = 1    supid = 0
cardstate = SYSMGR_CARDSTATE_ACTIVE.
cardstate = SYSMGR_CARDSTATE_ACTIVE (hot switchover is configured enabled).
Configured to use the real platform manager.
Configured to use the real redundancy driver.
Redundancy register: this_sup = RDN_ST_AC, other_sup = RDN_ST_SB.
EOBC device name: eth0.
Remote addresses:  MTS - 0x00000201/3      IP - 127.1.1.2
MSYNC done.
Remote MSYNC not done.
Module online notification received.
Local super-state is: SYSMGR_SUPERSTATE_STABLE
Standby super-state is: SYSMGR_SUPERSTATE_STABLE
Swover Reason : SYSMGR_SUP_REMOVED_SWOVER <-- Reason for the last switchover
```

```
Total number of Switchovers: 0 <-- Number of switchovers
                             >> Duration of the switchover would be listed, if any.

        Statistics:

Message count:          0
Total latency:          0               Max latency:            0
Total exec:             0               Max exec:               0
vsg#
```

Reloading a Module

Reload a module by entering the **reload module** command.

✎

Note   Using the **reload** command, without specifying a module, reloads the whole system.

This example shows the output for the command:

```
vsg# reload module 2
This command will reboot the system (y/n)? y
vsg#
```

Attaching to the Standby Cisco VSG Console

The standby VSG console is not accessible externally. Access the standby Cisco VSG console on the active VSG by entering the **attach module** *module-number* command.

This example shows the output for the command:

```
vsg# attach module 2
Attaching to module 2...
To exit type 'exit', to abort type '$.'
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2011, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
vsg#
```

Checking for the Event History Errors

Check for errors in the event history by entering the **show system internal sysmgr event-history errors** command.

This example shows the output for the command:

```
vsg# show system internal sysmgr event-history errors
Event:E_DEBUG, length:122, at 370850 usecs after Thu Feb  3 09:45:28 2011
[101] sysmgr_sdb_set_standby_state: Setting standby super state in sdb for vdc 1 to
SYSMGR_SUPERSTATE_STABLE, returned
```

```
0x0

Event:E_DEBUG, length:73, at 408277 usecs after Thu Feb  3 09:44:52 2011
[101] active_gsyncctrl_info_parse: UUID 0xB6 in vdc 1 service not running

Event:E_DEBUG, length:73, at 593428 usecs after Thu Feb  3 09:44:49 2011
[101] active_gsyncctrl_info_parse: UUID 0xE0 in vdc 1 service not running

Event:E_DEBUG, length:80, at 624613 usecs after Thu Feb  3 09:44:40 2011
[101] process_plugin_load_complete_msg: Start done rcvd for all plugins in vdc 1

Event:E_DEBUG, length:89, at 624611 usecs after Thu Feb  3 09:44:40 2011
[101] process_plugin_load_complete_msg: Received plugin start done for plugin 1 for vdc 1

Event:E_DEBUG, length:99, at 518152 usecs after Thu Feb  3 09:44:39 2011
[101] perform_bootup_plugin_manager_interactions: all bootup plugins have now been loaded
in vdc: 1

Event:E_DEBUG, length:79, at 518150 usecs after Thu Feb  3 09:44:39 2011
[101] perform_bootup_plugin_manager_interactions:incrementing number of plugins

Event:E_DEBUG, length:118, at 518020 usecs after Thu Feb  3 09:44:39 2011
[101] perform_bootup_plugin_manager_interactions: plugin has been loaded in vdc 1 -
sending response to Plugin Manager

Event:E_DEBUG, length:58, at 631599 usecs after Thu Feb  3 09:44:38 2011
[101] process_reparse_request: on vdc 1, plugin start rcvd
vsg#
```

Standby Synchronization

This section includes the following topic:

Synchronization Fails

If the standby Cisco VSG is stuck in the synchronization stage, follow these steps on the active Cisco VSG:

Step 1   Enter the **show system internal sysmgr state** command and check for a line similar to the following:

```
Gsync in progress for uuid: xxxx
```

If this entry is present and shows the same **xxxx** value for a long time, the system has trouble synchronizing the state for one of the processes.

Step 2   Identify the process by entering the **show system internal sysmgr service running | grep xxxx** command.

This message appears in the first few lines of the output:

```
BL-bash# show system internal sysmgr state
The master System Manager has PID 1350 and UUID 0x1.
Last time System Manager was gracefully shutdown.
Gsync in progress for uuid: 0x18
The state is SRV_STATE_MASTER_ACTIVE_HOTSTDBY entered at time Mon Feb 21 17:56:3
9 2011.
```

```
The '-b' option (disable heartbeat) is currently disabled.
…

If synchronization for each process occurs quickly, you might not have the chance to see
the line (you might have to enter the command repeatedly as the standby Cisco VSG). If
gsync for a particular process gets stuck, the line stays in the output for a while.
```

Step 3   If you are able to login to the console of the standby VSG (you might need to press Ctrl-C after giving the password), check the output of these two commands:

- **show system internal sysmgr state**

- **show system internal sysmgr service running | grep xxxx**
  where xxxx is from the line "Gsync in progress for uuid: xxxx" (found by running the **show system internal sysmgr state** command)

Step 4   If access to the system is available only after the standby server has booted up and synchronized with the active server, use the following commands:

- Enter the **show system internal sysmgr bootupstats** command and look for processes that took much longer than the rest, in the order of the time that the system took to boot.

- On the standby console, enter the **show system internal sysmgr gsyncstats** command and look for processes with large Gsync time values.