



Changing VNE Polling, Reachability, Discovery, and Persistency and Working with Unmanaged Segments (Cloud VNEs)

These topics provide advanced technical information about VNEs, including the configurable points:

- [Changing VNE Polling Settings, page 12-1](#)
- [Changing VNE and Protocol Settings That Determine Device Reachability, page 12-24](#)
- [Changing Device Discovery Timeouts and Investigation State Reporting, page 12-31](#)
- [Changing How VNE Commands Are Executed \(Collectors and Command Priorities\), page 12-32](#)
- [Changing Settings That Control VNE Data Saved After Restarts, page 12-37](#)
- [Creating Connections Between Unmanaged Network Segments \(Cloud VNEs and Links\), page 12-42](#)
- [Improving TACACS Server Performance by Changing VNE Telnet/SSH Login Rates \(Staggering VNEs\), page 12-51](#)
- [Tracking VNE-Related Events, page 12-53](#)

Changing VNE Polling Settings

Prime Network uses a variety of polling methods to model and monitor the network. Working together these mechanisms maintain the balance between ensuring model fidelity (frequent polling cycles) while protecting system performance (less polling cycles). [Table 12-1](#) lists the polling methods used by Prime Network, their default behavior, and where you can find more information on each method.



Note

If you are going to make changes to a large group of VNEs, do it during a maintenance window so you can test the changes locally and then restart the entire system to apply your changes throughout the system.

If you are experiencing high CPU usage, see [Responding to High CPU Utilization Problems, page 12-2](#).

Table 12-1 *Polling Mechanisms Used by Prime Network*

Polling Mechanism	Description	Default Setting	For information, see:
Reduced polling	An event-driven polling that is triggered by changes in the managed device. It reduces the load on devices and the network by minimizing the use of periodic polling.	Enabled for all VNEs (but not supported on all device types). Can be disabled per VNE or across the system from GUI.	Configuring Reduced (Event-Based) Polling, page 12-3
Regular Polling (VNE polling groups)	Periodic polling that is done according to a group setting, in a repetitive fashion. You can create new polling groups using Prime Network Administration, and apply them to network elements. Changes to the model are updated according to the polling cycles.	Enabled on devices that do not support reduced polling. Can be controlled from GUI.	Configuring Basic Polling Settings for Status, Configuration, System, Layer 1 and Layer 2 Data, page 12-18
Adaptive polling	When CPU usage is high, introduces an interval between executions of device commands. Changes to the model are updated according to the interval. You can create adaptive polling groups using Prime Network Administration, and apply them to network elements.	Enabled. Settings can be modified or disabled per VNE or across the system from GUI.	Configuring Adaptive Polling for High CPU Events, page 12-10
Smooth polling	Takes commands in the same polling cycle and spreads their execution throughout the polling cycle using a random number within the polling interval, rather than using a timer-based approach.	Enabled. Can be enabled by editing the registry.	Using Smooth Polling To Spread Out Commands in a Polling Cycle, page 12-22
Smart polling	For repetitive queries, introduces a polling protection interval that specifies the minimum amount of time that must pass before a query can be sent to a device a second time.	Disabled. Can be enabled using the Registry Controller.	Adjusting the Polling Protection Interval Between Repeated Device Queries (Smart Polling), page 12-23

Responding to High CPU Utilization Problems

If you suspect ongoing CPU utilization problems, start with these troubleshooting steps:

1. Review the device log files to find any recurring polling spikes that extend for prolonged periods. If the CPU spikes are *not* occurring at a constant interval, it is likely a network events rather than a device problem.
2. Verify whether other applications (besides Prime Network) are managing the devices, and check those applications for problems before proceeding with Prime Network changes.
3. If you think the problem resides in Prime Network, analyze the CPU over a 24-hour period as follows:
 - Log onto the device and check the usage for different timelines. (Refer to the operating system documentation that applies to the device type.)
 - Check the audit log for any open sessions that correspond with the usage problems.

4. Read the following topics:
 - [Configuring Adaptive Polling for High CPU Events, page 12-10](#)
 - [Configuring Basic Polling Settings for Status, Configuration, System, Layer 1 and Layer 2 Data, page 12-18](#)
5. Consider disabling MAC-based topology. To disable this topology, use the following registry command, where *devicetype* is the registry location for the device type.

```
# ./runRegTool.sh -gs 127.0.0.1 set 0.0.0.0 "site/device-type/ipcore/software
versions/default version/amsi/topology/ethernet/MacTestEnable" false
```

For example, this command disables MAC-based topology for Cisco 7600 routers:

```
# ./runRegTool.sh -gs 127.0.0.1 set 0.0.0.0 "site/ciscorouter2/76xx/product/software
versions/default version/amsi/topology/ethernet/MacTestEnable" false
```

Configuring Reduced (Event-Based) Polling



Note

For VNEs using reduced polling, add the event-generating IP address to the VNE (in the Events tab) so the VNE will listen to that address for syslogs and traps. See [VNE Properties: Events, page D-17](#).

These topics provide procedures for adjusting the reduced polling mechanism:

- [Finding Out Which Device Types Support Reduced Polling, page 12-5](#)
- [Finding Out Whether a VNE is Using Reduced Polling, page 12-7](#)
- [Changing the Default Reduced Polling Approach for a Single VNE or All VNEs, page 12-7](#)
- [Preventing Repeated Executions of the Same Command \(Reduced Polling Throttling Mechanism\), page 12-9](#)

All VNEs either use reduced polling or regular polling. When a VNE is using reduced polling, Prime Network will poll the device whenever it receives a configuration change event. Changes to the model are updated immediately. Reduced polling is the default polling method for new VNEs. If a device type does not support reduced polling, Prime Network uses regular polling.

Because the syslog facility is sometimes unreliable, the reduced polling mechanism has a *fail-safe* mechanism that polls the device's complete command history (from the archive log) to ensure that no device configuration changes were missed.

If you expect a device to receive multiple syslogs in a short period of time, you can enable a *throttling* mechanism which prevents the same command from being executed repeatedly. See [Preventing Repeated Executions of the Same Command \(Reduced Polling Throttling Mechanism\), page 12-9](#).

If a VNE using reduced polling is moved to the Currently Unsynchronized state, it means it failed to identify one or more changes, or there is a gap in the configuration archive buffer. The device configuration archive buffer contains the configuration commands that were executed on the device. For Cisco IOS devices, it is possible for the buffer to overflow when a large number of commands are executed; thus some commands can be lost, a gap is identified, and the VNE is assumed to be out of synch with the device. VNEs using reduced polling are more sensitive to these changes due to their different polling frequency.

To quickly synchronize the VNE model without having to wait for the next polling cycle, click the **Poll Now** button in the Network Element Properties window. You can open this window from:

- Vision by right-clicking a device and choosing **Properties**

- Administration by right-clicking a VNE and choosing **Inventory**

Figure 12-1 provides an example of the Network Element Properties window with the **Poll Now** button.

Figure 12-1 Poll Now Button in Device Properties Window



The information refresh is similar to the VNE discovery process, the main difference being what triggers the process.

Like any discovery process the VNE refresh has the potential of raising the CPU usage on the device. However, several factors work together to keep CPU usage low: the queuing mechanism that controls command execution, the VNE logic that reuses command results, and adaptive polling's throttle mechanism that introduces a delay between commands.

The amount of time needed for the VNE refresh depends on many factors, such as device and network latency, and gateway server activities. To help you understand when the refresh is in process and when it has completed:

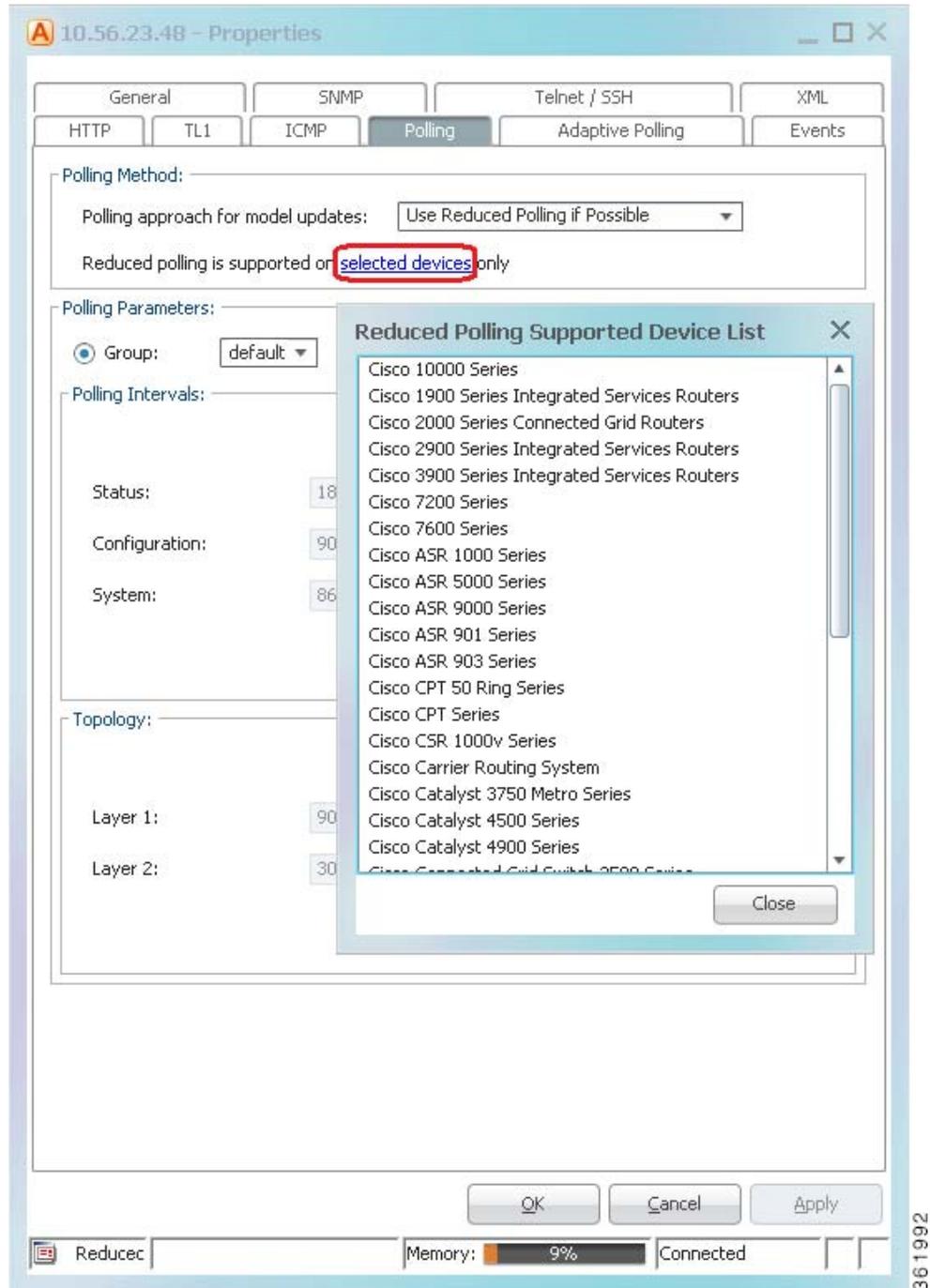
- The VNE moves to Currently Unsynchronized investigation state and its icon changes to an hourglass (see Figure 12-1).
- You can configure Prime Network to generate a System event when a VNE enters or exits the Currently Unsynchronized state (or any other investigation state). See Table 12-6 on page 12-32.

Finding Out Which Device Types Support Reduced Polling

To find out whether or not a VNE supports reduced polling, check the listing in the VNE properties dialog as follows.

-
- Step 1** Open the VNE properties window from the Prime Network Administration by right-clicking the VNE and choosing **Properties**.
 - Step 2** Click the Polling tab and go to the Polling Method area.
 - Step 3** Click **Supported on selected devices only** to list the device types that support reduced polling, as shown in [Figure 12-2](#), and verify it against the VNE device type.

Figure 12-2 Listing the Devices That Support Reduced Polling



Finding Out Whether a VNE is Using Reduced Polling

The VNE Status Details window displays a true/false setting for Reduced Polling that indicates whether the VNE is using reduced polling. If you were not sure that your device support reduced polling and you choose **Use Reduced Polling if Possible** as your polling method, this window is where you can find the result.

- Step 1** Open the device inventory window from the Prime Network Administration by right-clicking the VNE and choosing **Inventory**.



Note Users with Operator privileges can open the Communications Details window from Prime Network Vision.

- Step 2** Click **VNE Status** at the bottom of the window to open the VNE Status Details window, and check the reduced polling setting as shown in [Figure 12-3](#).

Figure 12-3 Reduced Polling Setting in VNE Status Details Window



The value true means that the VNE is using reduced polling to monitor the device.

Changing the Default Reduced Polling Approach for a Single VNE or All VNEs

For reduced polling to work as designed, devices must be properly configured to generate device change events. See [Configuring Devices So They Can Be Properly Modeled and Managed by Prime Network, page A-1](#).

By default, all new VNEs use reduced polling. If the device type does not support reduced polling, the VNE uses regular polling (you are not notified that this is happening). If you want to get a notification that a VNE does not support reduced polling, or you just want to use regular polling, change the default polling method as described in this procedure. The change will take effect for all new VNEs.

- Step 1** Select **Tools > Registry Controller > Advanced VNE Configurations** from the main menu of the Administration GUI client. The Default Polling Mode list shows the current setting for the system.
- Step 2** To make a change, choose one of the following from the Default Polling Mode drop-down list.

Table 12-2 Default Polling Approaches

What You Want To Do:	Choose:	Approach	Description
You want the VNE to use reduced polling, but if the device type does not support it, you want to receive a notification (event).	0	Always use reduced polling	Prime Network will define the settings based on the recommended offset of model fidelity vs. interference. If the device type does not support event-based polling, Prime Network generates a Device Unsupported event. Use this if you want to be notified that the device type does not support reduced polling.
You want the VNE to use reduced polling, but if the device type does not support it, you want the VNE to use regular polling.	1	Used reduced polling if possible	Prime Network will define the settings based on the recommended offset of model fidelity vs. interference. If the device type does not support event-based polling, Prime Network uses regular polling. Note This is the default method for all VNEs.
You do not want the VNE to use reduced polling (even if the device supports it).	2	Use regular polling	Instructs Prime Network to proactively poll configuration data using a configuration interval (usually every 15 minutes). This means that even in extreme circumstances where events are lost, the VNE would be synchronized after a maximum of 15 minutes (not 24 hours).

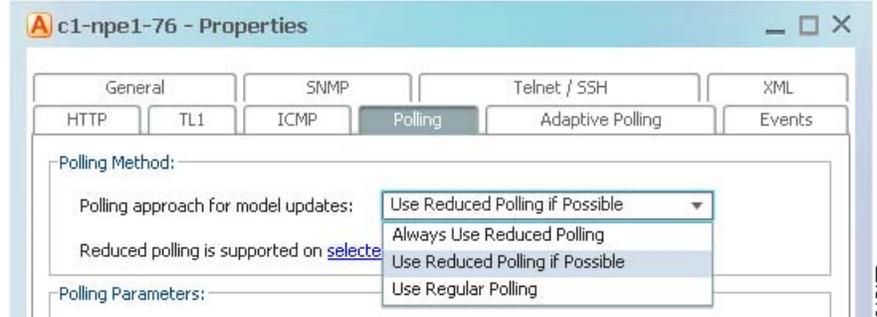
- Step 3** Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.
- Step 4** Click **Apply** and restart the VNEs (by right-clicking each VNE and choosing **Actions > Stop**, then **Actions > Start**).
- Step 5** Restart the gateway. See [Stopping and Restarting Prime Network Components, page 3-16](#).

**Note**

There may be a delay in updates for Cisco ASR 5000 Series devices. This is because although the device sends an SNMP config change trap, it does not send it immediately.

If you only want to change the polling method for a single VNE, use this method.

- Step 1** From the Administration GUI client, open the VNE Properties by double-clicking a VNE.
You can also do this from the Vision GUI client using the device properties window (by clicking the **VNE Details** button at the bottom of the window; this opens the VNE Properties).
- Step 2** In the VNE Properties window, check whether your device supports reduced polling by clicking the **Supported on selected devices only** link. (See [Figure 12-2 on page 12-6](#) for an example.)
- Step 3** Click the Polling tab and choose an approach from the drop-down list.

Figure 12-4 Reduce Polling Setting in VNE Properties Dialog Box

- Step 4** Save your changes, and restart the VNE by right-clicking it and choosing **Actions > Stop**. When the Status changes to Down, right-click the VNE and choose **Actions > Start**.

Preventing Repeated Executions of the Same Command (Reduced Polling Throttling Mechanism)

For cases where a VNE using reduced polling receives multiple configuration change syslogs from the same device in a short time span, a *throttling* mechanism can be used to prevent the same command from being executed repeatedly. The throttle mechanism collects all change notifications that are received within a predefined interval, and when the interval expires, the VNE polls the device for updated information at one time. The throttle feature is turned off by default (the interval is set to 0). If a change is not immediately reflected in Prime Network Vision because the throttle is enabled, you can manually update the GUI using the Poll Now button (see [Figure 12-1](#)).

The interval should allow enough time for the change to be applied, including being applied to other affected devices. In the following example we change the interval to five minutes. This may not be a suitable interval in the following scenarios:

- If multiple large configuration changes are bulked and run over a period of time, a larger interval might reduce CPU usage.
- If multiple small configurations are run throughout the day, a smaller interval would be appropriate because it would reflect the changes more quickly.

To check, enable, or disable the throttling mechanism for an individual VNE, use the following procedure.

- Step 1** Log into the gateway as *pnuser* and change to the Main directory.
- ```
cd $ANAHOME/Main
```
- Step 2** For a VNE where *unit-IP* is the unit IP address, *avmxxx* is the AVM ID, *vne-key* is the VNE name), use the following commands. If you are running this command on AVMs that are on the gateway server, *unit-IP* should be **127.0.0.1**.
- To check whether throttling is enabled (and an interval is set):
- ```
# ./runRegTool.sh -gs 127.0.0.1 get -entry unit-IP "avmxxx/agents/da/vne-key/evne polling interval"
```

- To set the throttling interval to *minutes*:

```
# ./runRegTool.sh -gs 127.0.0.1 set unit-IP "avmxxx/agents/da/vne-key/evne polling interval" minutes
```

- To unset (disable) the throttling interval:

```
# ./runRegTool.sh -gs 127.0.0.1 unset unit-IP "avmxxx/agents/da/vne-key/evne polling interval"
```

For example, this command would set the throttling interval to 5 minutes for a VNE named c7-npe1-76 on AVM 600, and would make the change to the Golden Source registry:

```
# ./runRegTool.sh -gs 127.0.0.1 set 127.0.0.1 "avm600/agents/da/c7-npe1-76/evne polling interval" 5
```

Step 3 Restart the VNE by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**.

Configuring Adaptive Polling for High CPU Events

These topics provide procedures for adjusting the adaptive polling mechanism:

- [Customizing How Prime Network Responds to High CPU Events](#), page 12-13
- [Apply Customized Adaptive Polling Settings to a VNE](#), page 12-14
- [Turning Off Adaptive Polling and Disabling Customized Adaptive Polling Groups](#), page 12-15
- [Changing the CPU Usage Polling Interval for Adaptive Polling](#), page 12-16
- [Adjusting Adaptive Polling for Devices with Large Configurations \(and Telnet Responses\)](#), page 12-17

Adaptive polling is a feature that preserves device integrity in extreme network scenarios or when your system encounters device caveats. When device CPU is exceedingly and consistently high, the adaptive polling mechanism issues an informational Service alarm and moves the VNE to slow polling. A delay is introduced between SNMP packets or Telnet CLI commands sent to the device, which allows the device to recover. Because some devices have exceptionally large configurations which generate very large Telnet responses—literally thousands of output lines—the adaptive polling mechanism breaks the Telnet responses into chunks. It also inserts a delimiter (such as --More--) and waits for the VNE to respond before continuing. This technique is sometimes called *flow control*.

This mechanism ensures that an NE's CPU utilization is not monopolized by polling commands and allows the NE to continue to address other priorities. Although this may result in a longer time to receive all of the information, this is a desirable tradeoff to all CPU utilization being consumed by polling.

You can make the following adjustments to the adaptive polling mechanism:

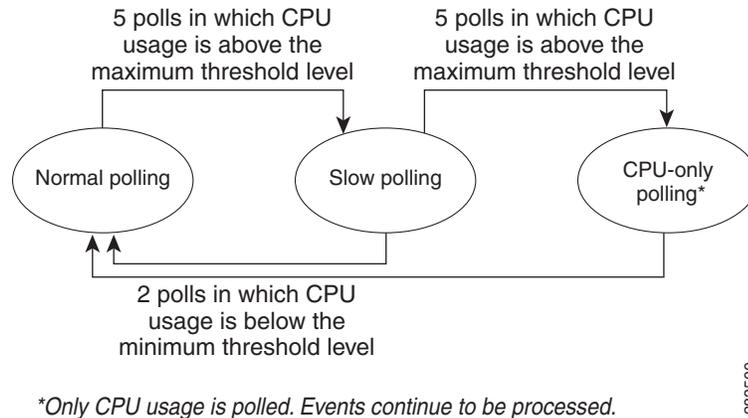
- Create an adaptive polling group with customized settings that can be easily applied to VNEs
- Fine-tune the adaptive polling thresholds for individual VNEs
- Adjust the terminal length and delimiter

The XML protocol also supports adaptive polling due to the fact that XML is a protocol that is handled over Telnet. Although adaptive polling is not formally supported over HTTP, because other (non-HTTP) protocols are involved in data collection, an overall improved result is also seen for HTTP.

If a VNE keeps moving to slow polling or CPU-only polling, you should adjust the adaptive polling thresholds. See [Changing the CPU Usage Polling Interval for Adaptive Polling](#), page 12-16.

Figure 12-5 illustrates the adaptive polling mechanism with its default settings. You can adjust these settings as described in [Apply Customized Adaptive Polling Settings to a VNE](#), page 12-14.

Figure 12-5 How Adaptive Polling Works



Note

In this figure, the term *slow polling* does *not* refer to the preconfigured polling group called *slow*, that is described in [Table 12-5](#) on page 12-19.

The adaptive polling mechanism issues Service alarms as the device CPU usage changes. The following steps provide more detail about the adaptive polling algorithm illustrated in [Figure 12-5](#).

1. When a *normal polling* VNE exceeds the maximum CPU usage threshold value, an informational Service alarm is issued. If the threshold is exceeded for five consecutive polls, it is moved to *slow polling*.

Slow polling introduces a delay (interval) between sending commands to the NE. In SNMP, the delay is between SNMP packets sent to the device (500 ms); in Telnet or SSH, the delay is between CLI commands sent to the device.) In addition, Telnet responses are divided into smaller parts, separated by a delimiter to adjust throughput.

2. A *slow polling* VNE can do either of the following, depending on CPU usage polling results:
 - If CPU usage is below the minimum threshold level for two consecutive polls, the VNE returns to *normal polling*. A Service alarm is issued as the VNE return to normal polling.
 - If CPU usage exceeds the maximum threshold for five additional consecutive polls (a total of ten polls), the VNE moves to *CPU-only polling* and a critical Service alarm is issued.

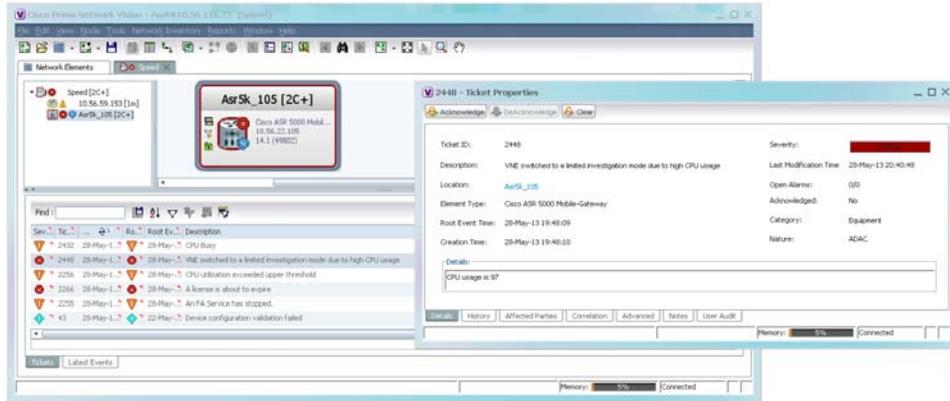
All polling is suspended except for CPU usage; however, syslogs and traps continue to be processed.

3. When a *CPU-only polling* VNE has CPU usage that is below the minimum threshold level for two consecutive polls, it returns to *normal polling*.

The average CPU usage is calculated using a CPU polling *interval*. The *interval* controls how often to poll the VNE for its CPU usage (for example, every 30 seconds). The *interval* is described in [Table 12-4](#) on page 12-16.

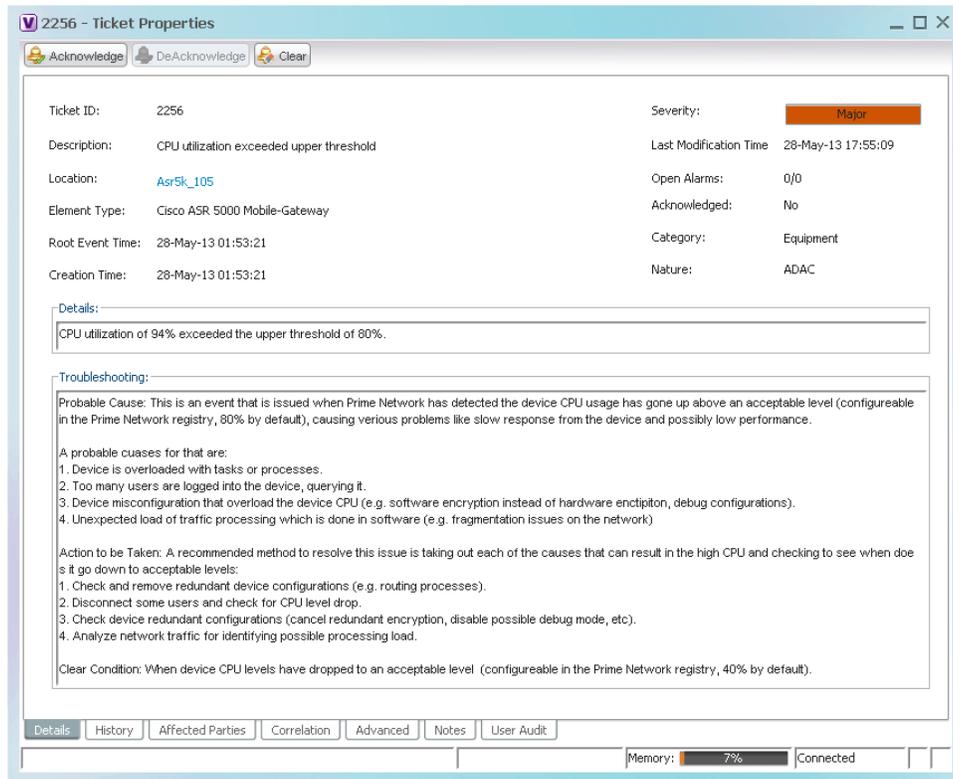
[Figure 12-6](#) shows the an example of what you will see in Prime Network Events and Prime Network Vision when a VNE is experiencing high CPU usage. (The Communication Details window can be launched from Prime Network Vision and by clicking **VNE Status** from the device properties window.)

Figure 12-6 What Prime Network Reports When a VNE Experiences High CPU Usage



The Event Details can also provide troubleshooting information for the adaptive polling problem, as shown in Figure 12-7.

Figure 12-7 Adaptive Polling Troubleshooting Information in Ticket Details

**Note**

If a parent AVM is stopped during this process, the VNE retains its previous polling data. When the AVM is restarted, the VNE continues from the point at which its polling was interrupted. See [Instrumentation Persistency](#), page 12-40.

Customizing How Prime Network Responds to High CPU Events

If you want to apply customized thresholds to a group of devices, create an adaptive polling group. Once you have created it, it becomes available to all VNEs in the VNE properties Adaptive Polling tab. Prime Network provides one predefined adaptive polling group named **PN Settings Group**. It uses whichever settings are recommended by Prime Network.

If you are not sure what settings to apply, use the default (Device Type Settings).

- Step 1** Right-click **Global Settings > Adaptive Polling Groups** and choose **New Adaptive Polling Group**.
- Step 2** Enter a name and description, and check the Enable check box.
- Step 3** Enter the customized settings for the new adaptive polling group.

Table 12-3 Adaptive Polling Local Settings

Thresholds	Description	Default
Upper Threshold	Upper CPU usage threshold. When CPU usage exceeds this value for a specified number of (tolerance) polls, the adaptive polling mechanism is triggered and the VNE moves to <i>slow polling</i> or <i>CPU-only polling</i> .	90%
Lower Threshold	Lower CPU usage threshold. When CPU usage drops below this value for a specified number of polls (2 by default), the VNE reverts from <i>slow polling</i> to <i>normal polling</i> and related alarms are cleared.	60%
Upper Tolerance	Number of high-CPU polls required to move the VNE to <i>slow polling</i> . When the Upper Threshold is crossed this number of consecutive CPU polls, the VNE moves from <i>normal polling</i> to <i>slow polling</i> . (To be more conservative, enter a lower number.) For example, using the default settings, a Cisco IOS-XR VNE would move from <i>normal polling</i> to <i>slow polling</i> after 5 minutes — that is, 5 Upper Tolerance polls with a 60-second interval (see Table 12-4 on page 12-16).	5
Lower Tolerance	Number of low-CPU polls required to revert the VNE to <i>normal polling</i> . When CPU utilization falls below the Lower Threshold for this number of consecutive polls, the VNE reverts from <i>slow polling</i> or <i>CPU-only polling</i> to <i>normal polling</i> . (To be more conservative, enter a higher number.)	2

Table 12-3 Adaptive Polling Local Settings (continued)

Thresholds	Description	Default
Maintenance Tolerance	<p>Total number of high-CPU polls required to move the VNE to <i>CPU-only polling</i>. This number includes the Upper Tolerance polls.</p> <p>For example, an Upper Tolerance of 5 and a Maintenance Tolerance of 10 means:</p> <ul style="list-style-type: none"> • The VNE would move from <i>normal polling</i> to <i>slow polling</i> after 5 high-CPU polls (Upper Tolerance). • The VNE would move from <i>slow polling</i> to <i>CPU-only polling</i> after 5 more high-CPU polls, for a total of 10 (Maintenance Tolerance) high-CPU polls. <p>Using the default settings, this means that Cisco IOS-XR VNEs, which have a 60-second polling interval, would move from <i>normal polling</i> to <i>CPU-only polling</i> in 10 minutes:</p> <ul style="list-style-type: none"> • The VNE would move from <i>normal polling</i> to <i>slow polling</i> after 5 minutes. • The VNE would move from <i>slow polling</i> to <i>CPU-only polling</i> after 5 more minutes. <p>See Table 12-4 on page 12-16 for the default <i>interval</i> settings.</p>	10
SNMP Delay	Delay (in milliseconds) between SNMP packets that are sent from the VNE to the device.	500
Telnet Delay	Delay (in milliseconds) between Telnet commands that are sent from the VNE to the device.	500

- Step 4** Click **OK**. The new group is added to the list of adaptive polling groups and can be applied to new and existing VNEs.

Apply Customized Adaptive Polling Settings to a VNE

Use this procedure to apply adaptive polling settings to a VNE. You can also use this procedure to change a VNE's existing adaptive polling settings.

- Step 1** If you want to apply customized settings to multiple VNEs, create an adaptive polling group.
- Right-click Global Settings > Adaptive Polling Groups and choose **New Adaptive Polling Group**. You can also edit an existing group by double-clicking it; all fields are editable except for the name.
 - Enter a name and description.
 - Check the Enable check box.

- d. Enter the customized settings for the new adaptive polling group. The settings are described in [Table 12-3 on page 12-13](#).
- e. Click **OK** (or, if you are editing an existing group, **Apply** and **OK**). The new adaptive polling group is added to the list of groups under Global Settings.



Note Make sure you have checked the Enable check box if you want to use the new adaptive polling group.

Step 2 Apply adaptive polling settings to a VNE, or change its existing settings.

- a. Select a VNE and right-click **Properties**.
- b. Click the Adaptive Polling tab.
- c. Choose the source for the VNE adaptive polling settings. If you are not sure what to choose, use Device Type Settings (which is the default).

Settings Type	Description
Group	Use a customized adaptive polling group. If any adaptive polling groups have been created and enabled, they are displayed in the drop-down list. (Prime Network comes with one predefined adaptive polling group named PN Settings Group ; it uses whichever settings are recommended by Prime Network.)
Device Type Settings	Use the settings specified for this device type (as delivered with Prime Network). If the device does not support adaptive polling (no device type settings exist), the Prime Network Settings are used.
Local Settings	Specify your own settings, overriding the defaults. The settings are applied to this VNE only. If you select Local Settings, enter the adaptive polling settings as shown in Table 12-3 on page 12-13 .

- d. Apply your settings.
 - If you are editing an existing VNE, click **Apply**. You do not have to restart the VNE.
 - If you are creating a new VNE, click **OK** to create the new VNE, or continue with the VNE configuration.

Turning Off Adaptive Polling and Disabling Customized Adaptive Polling Groups

When you turn off adaptive polling, if a VNE experiences any high CPU problems, Prime Network will not use any of the safeguards provided by the adaptive polling mechanism. Use this procedure to turn off adaptive polling for a specified VNE.

Step 1 Select the VNE and right-click **Properties**.

Step 2 In the Adaptive Polling tab, choose **Local Settings** and uncheck the Enable check box.

Step 3 Save and restart the VNE (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).

When you disable an adaptive polling group, the adaptive polling mechanism is turned off for all VNEs using the group settings.

-
- Step 1** Click **Global Settings > Adaptive Polling Groups** and double-click the adaptive polling group you want to disable.
- Step 2** Uncheck the Enable check box.
- Step 3** Save and restart the VNE (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).
-

You can reenable the group at any time using this same procedure and re-checking the Enable check box.

Changing the CPU Usage Polling Interval for Adaptive Polling

The command for retrieving CPU utilization data is sent to the device according to the *interval* setting in Table 12-4. Therefore, if Prime Network reports a high CPU utilization on a VNE, it means that for last five CPU polls, the average CPU utilization has been crossing the recommended threshold.

For example, the CPU usage information for some devices is gathered using the following command (other devices may use SNMP):

show processes cpu | include CPU utilization

Table 12-4 lists the parameters that control how often the data is polled. Complete directory paths to the registry entries are provided in the procedure that follows the table.

Table 12-4 Registry Settings—CPU Polling

Registry Entry	Description	Default Value				
		IOS XR	IOS	Cat OS	NX-OS	Star OS
interval	How often (milliseconds) to poll the CPU usage when determining the average usage.	60000 (1 min)	30000 (30 secs)	30000 (30 secs)	30000 (30 secs)	30000 (30 secs)
cpu-util-counter-bucket	(Cisco IOS XR only) Parameter for CPU measurement (see examples below)	5	N/A	N/A	N/A	current

Example for Cisco IOS XR Devices

As shown in Table 12-4, Prime Network provides a *cpu-util-counter-bucket* variable to calculate average CPU usage for Cisco IOS XR devices. The following table provides examples of values you might see for the same interval setting, but with different *cpu-util-counter-bucket* settings.

cpu-util-counter-bucket Setting	If <i>interval</i> =1 minute, CPU usage is checked every:	Hypothetical CPU average usage
1	1 x <i>interval</i> = 1 minute	10%
5	5 x <i>interval</i> = 5 minutes	16%
15	15 x <i>interval</i> = 15 minutes	14%

With a `cpu-util-bucket-counter` setting of 5, the adaptive polling mechanism would recognize average CPU usage on the device to be 16%.

Use the following procedure to adjust how often CPU utilization is polled by a specific VNE.

**Note**

Changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative. For information on the format of the `runRegTool.sh` script, see [Changing Global Registry Settings Using the CLI \(runRegTool\)](#), page B-4.

-
- Step 1** Log into the gateway as `pnuser` and change to the Main directory.
- ```
cd $ANAHOME/Main
```
- Step 2** To change the current CPU polling interval for an individual VNE, where `avmxxx` is the AVM ID, `vne-key` is the VNE name, and `unit-IP` is the IP address of the unit where the AVM resides (if you are running this command on AVMs on the gateway server, `unit-IP` should be `127.0.0.1`):
- To change the default polling interval to 60000 milliseconds (60 seconds, the recommended interval for Cisco IOS XR devices):
 

```
./runRegTool.sh -gs 127.0.0.1 set unit-IP "avmxxx/agents/da/vne-key/dcs
/registrations/com.sheer.metrocentral.coretech.common.dc.ManagedElement/cpu
usage/instrumentation services/interval" 60000
```
  - To change the default polling interval to 30000 milliseconds (30 seconds, the recommended interval for Cisco IOS and Cisco Cat OS devices):
 

```
./runRegTool.sh -gs 127.0.0.1 set unit-IP "avmxxx/agents/da/vne-key/dcs
/registrations/com.sheer.metrocentral.coretech.common.dc.ManagedElement/cpu
usage/instrumentation services/interval" 30000
```
- Step 3** (Cisco IOS XR devices only) To change the number of times to poll a device to 15, where `avmxxx` is the AVM ID on the gateway server, `vne-key` is the VNE name:
- ```
# ./runRegTool.sh -gs 127.0.0.1 set 127.0.0.1 "avmxxx/agents/da/vne-key/dcs
/registrations/com.sheer.metrocentral.coretech.common.dc.ManagedElement/cpu
usage/instrumentation services/command/parsing params/cpu-util-counter-bucket" 15
```
- Step 4** Restart the VNE for your changes to take effect (by right-clicking each VNE and choosing **Actions > Stop**, then **Actions > Start**).
-

Adjusting Adaptive Polling for Devices with Large Configurations (and Telnet Responses)

Some device have an exceptionally large configuration and can generate Telnet responses that contain thousands of output lines. If this happens, to protect system performance, Prime Network moves the VNE to slow polling and:

- Inserts a delimiter between commands (300 milliseconds, by default), and waits for a response before continuing. By default, this delay is 300 milliseconds.
- Breaks the response into segments according to a terminal length (512 lines, by default).

If you want to adjust the delimiter or terminal length, use the Registry Controller.

**Note**

Changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

- Step 1** Select **Tools > Registry Controller > Advanced VNE Configurations > Adaptive Polling** from the main menu of the Administration GUI client.
- Step 2** Adjust the following adaptive polling settings as needed.

Flow Control Settings	What the Setting Controls	Default
Telnet Delimiter Delay	Inserts a delimiter (such as --More--) and stops sending information until the VNE responds (sends a space character). For Telnet and SSH, the delay is inserted between CLI commands sent to the device.	300 (ms)
Terminal Length	Breaks the Telnet responses into segment of x lines.	512

- Step 3** Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.
- Step 4** Click **Apply** and restart the VNEs (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).

Configuring Basic Polling Settings for Status, Configuration, System, Layer 1 and Layer 2 Data

Prime Network VNEs poll the network element in a repetitive fashion according to a predefined time interval, called a polling cycle. The Polling Groups window enables you to manage these cycles by specifying the intervals you want, creating a group with those intervals, and then assigning VNEs to use that polling group.

Prime Network comes with two predefined polling groups named **default** and **slow**. You can employ these or, alternatively, define a new polling group, apply configured polling intervals to the group, and assign the polling group to managed elements. The VNE will poll the network element according to the preset values. This ensures polling of devices for different information consistently and in accordance with technical and business requirements.

**Note**

Any changes that are made in the Polling Groups window are automatically saved and immediately registered in Prime Network.

Alternatively, you can create a new polling group to fine-tune the frequency at which information is retrieved from the managed elements, thus controlling the amount of network traffic used by the various VNEs. For example, these are cases where a polling group with a longer polling interval would be useful:

- Define a core-device polling group with a long interval for configuration changes, because core devices seldom undergo configuration changes. Access devices, which are more likely to adjust to service provisioning changes, would have a shorter interval. This enables you to differentiate the same device type based on the device role.
- Define a group for legacy architectures and in-band management, that has an overall long interval (slow polling cycle).

Table 12-5 identifies the settings for the default and slow polling groups.

Table 12-5 Polling Rates for default and slow Polling Groups

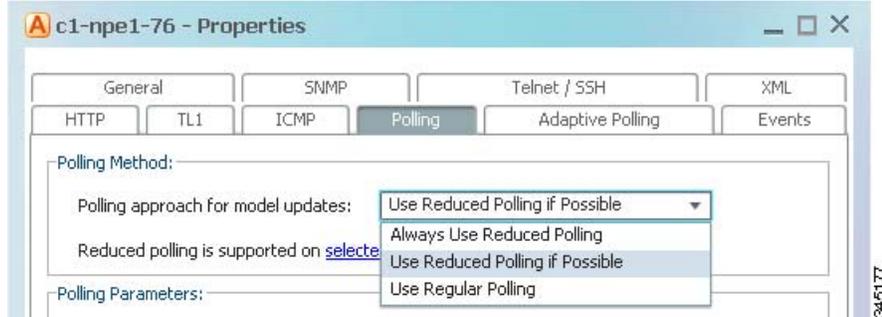
Attribute	Description	Preconfigured Polling Groups	
		default	slow
Status	The polling rate for status-related information, such as device status (up or down), CPU usage, port status, admin status, operational status.	180 seconds (3 minutes)	360 seconds (6 minutes)
Configuration	The polling rate for configuration-related information, such as IP address, device name and type; communication and investigation state; system name, description, location.	900 seconds (15 minutes)	1800 seconds (30 minutes)
System	The polling rate for system-related information, such software version.	86400 seconds (24 hours)	172800 seconds (48 hours)
Layer 1	The polling rate of the topology process as an interval for the Layer 1 counter. This is an ongoing process.	90 seconds	90 seconds
Layer 2	The polling rate of the topology process as an interval for the Layer 2 counter. This process is available on demand.	30 seconds	30 seconds

Configure a VNE To Use Regular Polling

By default, all VNEs using reduced polling. To change a VNE to use regular polling, use this procedure. If you want all new VNEs to use regular polling, you must edit the registry setting as described in [Changing the Default Reduced Polling Approach for a Single VNE or All VNEs, page 12-7](#).

-
- Step 1** Select a device (for example, using Prime Network Vision map view or properties view, or Prime Network PathTracer). Right-click the device and choose **Properties**, then click the **VNE** button.
- Step 2** Double-click the VNE to open the VNE Properties dialog box.
- Step 3** Choose an item from the Polling approach for model updates drop-down list. [Figure 12-8](#) provides an example of the drop-down list.

Figure 12-8 Reduce Polling Setting in VNE Properties Dialog Box



- Step 4** Save your changes, and restart the VNE by right-clicking it and choosing **Actions > Stop**. When the Status changes to Down, right-click the VNE and choose **Actions > Start**.

How to Create a New Polling Group

In the following example, a new polling group is created that polls for all device information every 24 hours. The polling group is then applied to a new VNE.

- Step 1** Choose **Global Settings > Polling Groups**.
- Step 2** Open the New Polling Group dialog box by right-clicking **Polling Groups**, then choose **New Polling Group**.
- Step 3** Complete the New Polling Group dialog. [Figure 12-9](#) provides an example of the new 24-hour polling group.

Figure 12-9 Creating a Polling Group Called 24 Hrs Cycle

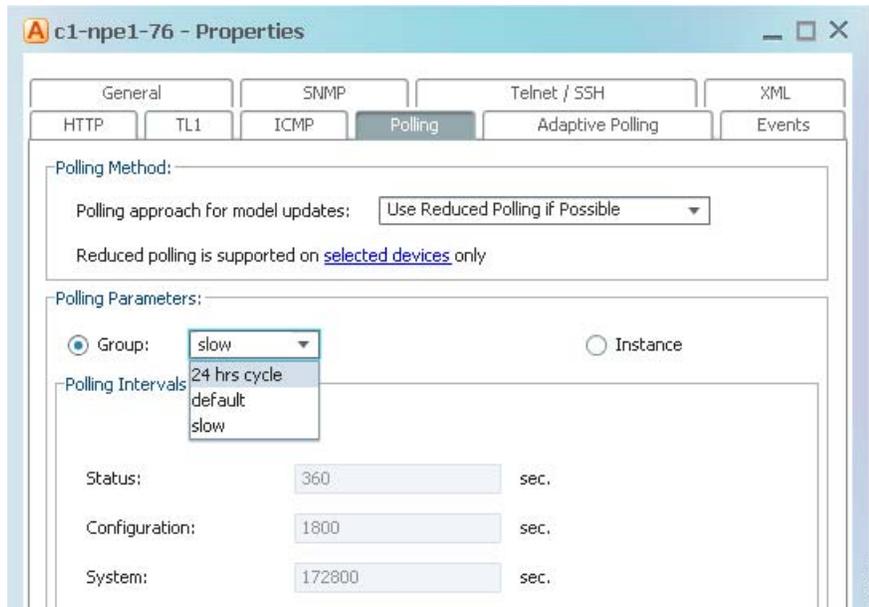
The following table describes the fields in this dialog box.

Field	Description
Name	Name for the polling group.
Description	Description for the polling group.
Polling Intervals	
Status	Number of seconds between collections of status-related information.
Configuration	Number of seconds between collections of configuration-related information.
Topologies	
System	Number of seconds between collections of system-related information.
Layer 1	Number of seconds in the topology Layer 1 counter. This is an ongoing process.
Layer 2	Number of seconds in the topology Layer 2 counter. This process is available on demand.

- Step 4** Save the changes by clicking **OK**. The new polling group is displayed in the content area and will be displayed when users create new VNEs.
- Step 5** To apply the new polling group to a new VNE, select the required gateway or unit and AVM in the navigation tree.

- Step 6** Right-click the AVM, then choose **New VNE**. The New VNE dialog box is displayed, opened to the General tab.
- Step 7** Complete the dialog as described in [Adding a New Device Type to Prime Network, page 4-17](#).
Apply the **24 hrs cycle** polling group to the VNE by clicking the Polling tab and selecting **24 hrs cycle** from the Polling Parameters Group drop-down list, as shown in [Figure 12-10](#).

Figure 12-10 Applying the 24 Hrs Cycle Polling Group to a VNE



- Step 8** Click **OK**. The new VNE is created, and it will poll the device according to the settings in the **24 hrs cycle** polling group.

Using Smooth Polling To Spread Out Commands in a Polling Cycle

Each VNE uses device registrations (commands) to collect different kinds of data from the associated network element. Each registration specifies the commands required to obtain a specific given item of data, and can be configured with a specific polling interval or logically associated with one of the polling intervals on a per device/VNE basis.

The smooth polling mechanism, which is enabled by default, spreads out the execution of commands in a single polling cycle. Rather than using a timer-based approach (where a large number of commands will be potentially scheduled for execution at the same time), the smooth polling method generates a random number (within the polling interval) for the next execution. This ensures that the commands get executed at least once within the required period, while also reducing the probability that two or more commands will run at the same time. This “smooths out” the load of the management protocols on the network and reduces their impact. Obviously, the longer the polling interval, the more effective smooth polling can be.

Note that smooth polling augments regular polling only after the completion of the first poll. Smooth polling is enabled in Prime Network by default.

How to Enable or Disable Smooth Polling

While it is rare that you will need to change the smooth polling setting, you can disable it if a VNE's polling intervals are extremely small.

**Note**

Changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Step 1 Log into the gateway as *pnuser* and change to the Main directory.

```
# cd $ANAHOME/Main
```

Step 2 Issue the appropriate command for a VNE where *unit-IP* is the unit IP address, *avmxxx* is the AVM ID, *vne-key* is the VNE name (if you are running this command on AVMs on the gateway server, *unit-IP* should be **127.0.0.1**):

- To disable smooth polling:

**Note**

Disabling smooth polling will likely result in higher CPU usage.

```
# ./runRegTool.sh -gs 127.0.0.1 set unit-IP
"avmxxx/agents/da/vne-key/smoothpollingenabled" false
```

- To revert to the default setting (enabled):

```
# ./runRegTool.sh -gs 127.0.0.1 unset unit-IP
"avmxxx/agents/da/vne-key/smoothpollingenabled"
```

Step 3 Restart the VNE (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).

Adjusting the Polling Protection Interval Between Repeated Device Queries (Smart Polling)

When Prime Network receives an incoming notification about a model change, the event provides information about the change but not about other components that may be affected by the change. For this reason Prime Network polls for this information that can affect the VNE model.

Sometimes queries are repeatedly submitted to a device. Common cases for this are when a user opens a Prime Network Path Tracer, window, and when an expedited event is received by Prime Network. To prevent overpolling, the smart polling mechanism uses a polling protection interval that specifies the minimum amount of time that must pass before a query can be sent to a device a second time.

For example, if multiple GUI or BQL users are concurrently using Prime Network Path Tracer, if the paths being viewed have common network elements, the details are collected according to the smart polling interval, and the data is shared without performing duplicate polls.

This example shows how Prime Network uses smart polling when receiving multiple instances of an expedited event:

1. An incoming event notification is classified as an expedited event, so Query A is immediately sent.
2. A few milliseconds later, the same incoming event arrives on an adjacent interface, triggering Query A again.

If the interval was 10 seconds, and the second instance of Query A arrived 7 seconds after the first instance of Query A, the second query would be dropped.

For expedited queries, Prime Network will queue the query to run when the interval is complete. Using the previous example, suppose the first instance of the query arrived at 12:00:00. The second instance arrives at 12:00:07. Because the query is expedited, the second query is queued to run at 12:00:10 (10 seconds after the first query).

You can change the polling protection interval using the Registry Controller. The default is 30000 ms (30 seconds).



Note You must restart the gateway to apply your changes.

-
- Step 1** Select **Tools > Registry Controller > Advanced VNE Configurations > Smart Polling** from the main menu of the Administration GUI client.
- Step 2** Adjust the Polling Protection Interval. Make your changes based on the amount of time required for the network to stabilize after a change, and keep the following in mind:
- If the interval is too short, Prime Network might report false alarms.
 - If the interval is too long, Prime Network will not report current data.
- Step 3** Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.
- Step 4** Click **Apply**.
- Step 5** Restart the gateway. See [Stopping and Restarting Prime Network Components, page 3-16](#).
-

Changing VNE and Protocol Settings That Determine Device Reachability

Prime Network VNEs communicate with network devices using a variety of protocols, and traps and syslogs. To determine the reachability of specific protocols, Prime Network runs multiple connectivity tests to check the device reachability.

The status of all of these protocols determine whether a device is reachable. By default, Prime Network marks a device as unreachable only when all enabled protocols are down; that is, the protocols are not responding, and the device is not generating syslogs or traps. However, you can change this behavior to fit your network.

These topics describe how reachability is determined and how you can change this behavior to fit the needs of your network:

- [Changing Reachability Settings for VNEs, page 12-25](#)
- [Changing Reachability Settings for Individual Protocols', page 12-26](#)

Changing Reachability Settings for VNEs

The management communication policy determines when Prime Network changes a VNE communication state to Device Unreachable or Device Partially Reachable. You can choose a policy based on how strictly you want to track and report device connectivity.

By default, Prime Network moves a VNE to Device Unreachable state when all of its enabled protocols are down, even if the device is still generating traps or syslogs. The management communication policy can be changed using the Registry Controller.



Note All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.



Note This procedure requires a gateway restart.

Step 1 Choose **Tools > Registry Controller > Advanced VNE Configurations > VNE Communication Policies** from the main menu of the Administration GUI client.

Step 2 Select the required management communication policy for Prime Network.

Management Policy	Use This Policy When You Want This VNE Reachability Reporting:
ensure-management (Default)	Change the VNE communication state to Device Unreachable when <i>all</i> of its enabled protocols are down, even if traps and syslogs are still being generated. In this scenario, Prime Network will never change the VNE communication state to Device Partially Reachable.
notstrict	Change the VNE communication state to Device Unreachable when <i>all</i> of its enabled protocols are down, and the device has not generated traps or syslogs for 6 minutes. Change the VNE communication state to Device Partially Reachable when <i>all</i> of its enabled protocols are down but the device is still generating traps or syslogs.
strict	Change the VNE communication state to Device Unreachable when <i>at least one</i> of the enabled protocols is down (even if traps and syslogs are still being generated). In this scenario, Prime Network will never change the VNE communication state to Device Partially Reachable.

Step 3 Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.

Step 4 Click **Apply**.

Step 5 Restart the gateway server. See [Stopping and Restarting Prime Network Components, page 3-16](#).

Changing Reachability Settings for Individual Protocols'

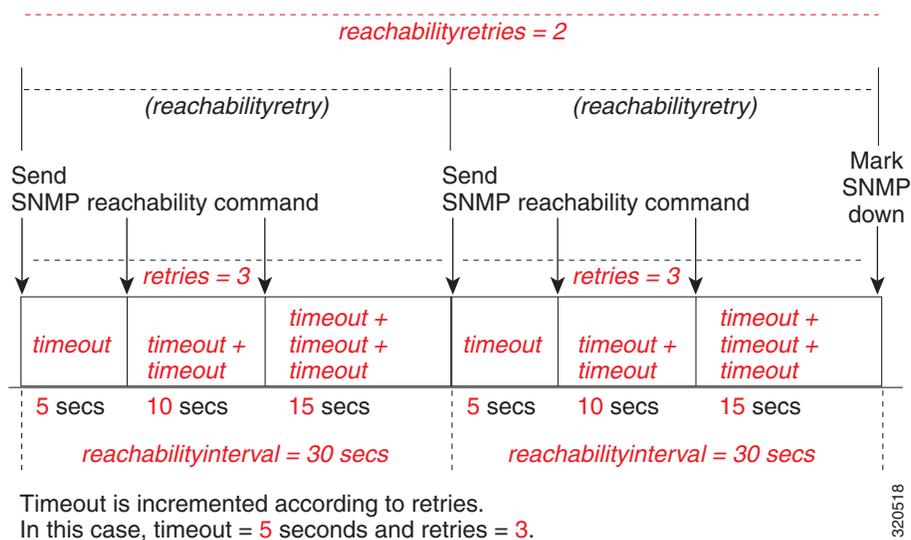
Changing Reachability Settings for SNMP

SNMP reachability is determined by sending a SNMP GET request to the device (by default, a GET request for the SysObjectId of the device) and waiting for a response. The following steps describe how Prime Network checks the health of the SNMP protocol.

Step	Description
Step 1	<p>The VNE begins an SNMP reachability command cycle (the cycle is represented by <i>reachabilityretry</i>). The number of commands that are sent in each command cycle is determined by the value of <i>retries</i>. In this illustration, <i>retries</i>=3 and <i>timeout</i>=5 seconds.</p> <ol style="list-style-type: none"> The VNE sends an SNMP GET request for the device <i>sysObjectId</i> to the device. This is the first retry; <i>retry</i>=1. If the device does not respond within <i>timeout x retry</i> (5 seconds x 1), the SNMP command is repeated. The VNE sends another SNMP reachability command (this is retry 2). If the device does not respond within <i>timeout x retry</i> (5 seconds x 2), the SNMP command is repeated (this is retry 3). <p>This continues until <i>retries</i> SNMP commands have been sent. This completes one reachability command cycle.</p>
Step 2	The value of <i>reachabilityretries</i> is decremented by 1.
Step 3	The mechanism waits the period of time specified by <i>reachabilityinterval</i> .
Step 4	The mechanism repeats the reachability command cycle until <i>reachabilityretries</i> equals 0.
Step 5	The SNMP protocol is marked Down.

How these values work together is illustrated in [Figure 12-11](#).

Figure 12-11 SNMP Reachability Testing



By default, lazyreachability is disabled. This means the default reachability algorithm is proactive—the VNE sends an SNMP request to the device and expects a response. If a response is not received within a certain amount of time, the SNMP protocol is marked as Down. However, if the lazyreachability registry key is enabled, the VNE will not be proactive. Instead, the VNE will wait until a regular query is sent to the device, and if no result is received, the VNE marks the protocol as Down.

You can adjust the settings that determine SNMP reachability using the Registry Controller.

**Note**

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Before You Begin

Because many VNEs may be impacted, we recommend that you change these settings during a maintenance window. Avoid setting values too low (which can trigger false “unreachable” messages) or too high (which may cause real problems to go undetected).

- Step 1** Select **Tools > Registry Controller > Advanced VNE Configurations > Device Protocol Reachability > SNMP** from the main menu of the Administration GUI client.
- Step 2** Adjust the SNMP reachability settings as needed. Refer to [Figure 12-11](#) for an illustration of what some of the settings control.

SNMP Reachability Settings	Default
Enable reachability detection process for SNMP	true
Note A false setting disables the reachability detection process, not the protocol.	
Duration in milliseconds that the VNE should wait for the device to respond to the SNMP GET request. (The first retry waits this duration; the 2nd retry is 2 x the duration; the 3rd retry is 3 x the duration, and so forth.)	5000
Number of retries for each request (retries)	3
Interval for device reachability commands, in milliseconds (reachabilityinterval)	30000
Number of retries until a reachability problem is determined (reachabilityretries)	1
Send reachability request when normal polling occurs rather than sending a dedicated command	false

- Step 3** Verify your changes to ensure you want to overwrite the current registry settings because after you click **Apply**, you cannot retrieve your settings using the **Restore** button.
- Step 4** Click **Apply**.

Changing Reachability Settings for Telnet and XML

Telnet connectivity is determined by sending a space and carriage return to the device and waiting for the device to echo the prompt.

**Note**

Prime Network uses these same tests for XML reachability testing. The only difference is that instead of sending a space and a carriage return, the VNE sends a request to sample the serial number of the device.

When a running command times out and the connection to the device is lost, the VNE will attempt to start a new connection with the device as shown in the following table.

Step	Description
Step 1	The VNE sends a message (a space and carriage return) to the device to initiate a login sequence.
Step 2	Starting from when the login sequence was initiated, if there is no response within <i>logintimeout</i> , the protocol remains marked Down.

If an open Telnet session is idle for an amount of time that exceeds *idletime*, Prime Network closes the connection. If the protocol connection is dropped, it is possible that reachability problems may go undetected by Prime Network until the Telnet connection is needed.

By default, lazyreachability is disabled. This means that the VNE does not wait until a normal polling cycle to perform its testing, but instead sends a dedicated Telnet request to the device (and a space and a newline character) and expects a response. If a response is not received within a certain amount of time, the Telnet protocol is marked as Down. If the lazyreachability registry key is enabled, the VNE will wait until a regular polling query is sent to the device, and if no result is received, the VNE marks the protocol as Down.

You can adjust the settings that determine Telnet and XML reachability using the Registry Controller.



Note

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Before You Begin

Because many VNEs may be impacted, change these settings during a maintenance window. Avoid setting values too low (which can trigger false “unreachable” messages) or too high (which may cause real problems to go undetected).

-
- Step 1** Select **Tools > Registry Controller > Advanced VNE Configurations > Device Protocol Reachability > Telnet** (or **XML**) from the main menu of the Administration GUI client.

- Step 2** Adjust the Telnet or XML reachability settings as needed. Refer to [Figure 12-11](#) for an illustration of what some of the settings control.

Telnet and XML Reachability Settings	Default
Enable reachability detection process	true
Note A false setting disables the reachability detection process, not the protocol.	
Interval for device reachability command, in milliseconds (reachabilityinterval)	30000
Send reachability request when normal polling occurs rather than sending a dedicated command	false
Timeout for login part, in milliseconds (logintimeout)	28000
Timeout for receiving initial device response to a command or for executing a “more” or other interactive user signal (for responses that have multiple pages or bulk), in milliseconds (receivetimeout)	20000
Timeout for not receiving a device response to any commands, in milliseconds (workingtimeout)	1800000
Amount of time, in milliseconds, where no commands are sent to device (after which the session is disconnected)	300000

- Step 3** Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.
- Step 4** Click **Apply** and restart the VNEs (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).

Changing Reachability Settings for ICMP

ICMP connectivity is determined by attempting to establish a TCP connection.

1. The VNE tries to establish a TCP connection on port 7 (Echo), and the device does not respond within *timeout*.
2. The first step is repeated *retries* times.
3. If there is still not response, the ICMP protocol is marked Down, and the VNE starts this process again.

You can adjust the settings that determine ICMP reachability using the Registry Controller.



Note

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Before You Begin

Because many VNEs may be impacted, we recommend that you change these settings during a maintenance window. Avoid setting values too low (which can trigger false “unreachable” messages) or too high (which may cause real problems to go undetected).

Step 1 Select **Tools > Registry Controller > Advanced VNE Configurations > Device Protocol Reachability > ICMP** from the main menu of the Administration GUI client.

Step 2 Adjust the ICMP reachability settings as needed.

ICMP Reachability Settings	Default
Enable reachability detection process for ICMP	true
Note A false setting disables the reachability detection process, not the protocol.	
Number of ICMP retries	1
Timeout for not receiving a device response to the ICMP TCP connection (in milliseconds)	5000

Step 3 Verify your changes to ensure you want to overwrite the current registry settings because after you click **Apply**, you cannot retrieve your settings using the **Restore** button.

Step 4 Click **Apply** and restart the VNEs (by right-clicking each VNE and choosing **Actions > Stop**, then **Actions > Start**).

Changing Reachability Settings for HTTP

HTTP connectivity is determined by trying to log into the device. If the device does not respond within *timeout*, the device is marked as Down. You can adjust these settings using the Registry Controller.

**Note**

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Before You Begin

Because many VNEs may be impacted, we recommend that you change these settings during a maintenance window. Avoid setting values too low (which can trigger false “unreachable” messages) or too high (which may cause real problems to go undetected).

Step 1 Select **Tools > Registry Controller > Advanced VNE Configurations > Device Protocol Reachability > HTTP** from the main menu of the Administration GUI client.

Step 2 Adjust the following HTTP reachability settings as needed.

HTTP Reachability Settings	Default
Enable reachability detection process for HTTP	true
Note A false setting disables the reachability detection process, not the protocol.	
Send reachability request when normal polling occurs rather than sending a dedicated command	false
Timeout for login (in milliseconds)	20000
HTTP keepalive (uses the same connection to send and receive multiple HTTP requests/responses instead of opening a new connection for each request/response pair)	true
Require device username and password when using HTTP	true

Step 3 Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.

Step 4 Click **Apply**.

Changing Device Discovery Timeouts and Investigation State Reporting

Table 12-6 lists registry settings you can change to control the following discovery and state reporting behaviors:

- Whether Prime Network should generate a Service event and long event description when an investigation state changes. This is not done by default because it can affect performance and cause unnecessary concern to operators. (Service events are generated for communication state changes by default.)
- The number of retries for device commands issued during the discovery process, and whether the device command is required.
- Whether Prime Network should use the timeout mechanism or the convergence mechanism to determine when the discovery process is complete. (You can also adjust the length of the discovery timeout.)



Note

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 12-6 Registry Settings for Discovery and Investigation States

Registry Entry	Description	Default Value
Investigation and Communication State Reporting		
site/agentdefaults/da/investigation-progress/investigation-state-update-event	Generate a Service event (in Prime Network Events) when investigation state changes	false
site/agentdefaults/da/investigation-progress/investigation-state-result-summary-event	Include an elaborated report about the investigation state change in the Long Description field of the Service event	false
Device Commands Used for Discovery		
site/interfacebasedscheme/defaultregistration/errorupdate tolerance	Allowable number of device command failures, after which an error is generated	3
site/interfacebasedscheme/defaultregistration/required	Designate the device command as required for evaluating an investigation state (insert this after the device command key name)	false
VNE Discovery Period Controls		
site/agentdefaults/da/investigation-progress/max-delay-before-managed-state-in-milliseconds	Timeout for VNE discovery process (in milliseconds) (ignored if convergence is being used)	1800000 (30 minutes)
site/agentdefaults/da/investigation-progress/convergence	Use the VNE convergence mechanism to control discovery	false

Changing How VNE Commands Are Executed (Collectors and Command Priorities)

The following topics provide a high-level description of how VNE collectors execute the commands required to build a model of a device, and how to adjust the way Prime Network executes these commands:

- [What Are Collectors and Command Priorities?](#), page 12-32
- [Considerations for Using Fast Commands and Fast Collectors](#), page 12-34
- [Expedited Commands and Activation Scripts and Fast Collectors](#), page 12-34
- [Configuring a Command With the “Fast” Command Priority](#), page 12-35
- [Creating a Fast Collector for a VNE](#), page 12-36

What Are Collectors and Command Priorities?

Prime Network discovers and models a network element using commands that are called *registrations*. Registrations are forwarded to a VNE’s *collectors*, which are the VNE components that communicate with the physical network element. By default, each VNE is configured to have two collectors: an SNMP collector and a Telnet collector. These collectors can execute only one command at a time. Because many commands are sent to the network element during modeling, each collector maintains a queue of commands. When a collector is busy, any new incoming commands are placed at the end of the queue (FIFO, or first in, first out). When a collector finishes with one command, it executes the next command in the queue in a serial fashion.

In most cases, executing commands in a serial fashion is adequate. However, it may not be efficient enough for network elements with large configurations, for the following reasons:

- When modeling begins, the collector receives many commands in a short amount of time. This results in a very long command queue.
- Some commands require extra time to execute (for example, when sampling a routing table for a Cisco CRS-1). The result is that commands at the end of the queue experience long delays before execution. This is particularly problematic for expedited commands and activation script commands (these cases are discussed in [Expedited Commands and Activation Scripts and Fast Collectors](#), page 12-34).

**Note**

Slow response could be the result of a high CPU utilization problem. See [Responding to High CPU Utilization Problems](#), page 12-2.

Command Priorities and Command Queues: Normal and Fast

To prevent delays in command execution, Prime Network uses a *command priority* mechanism. Every command is given one of the following priorities:

- **Fast**—High priority
- **Normal**—Normal priority (the default)

To deal with the two priorities, each collector maintains two *queues*: a fast queue for the fast commands and a normal queue for the normal commands. When a collector is available it will execute commands in the fast queue first. It will not execute any commands in the normal queue until the fast queue is empty.

Fast Collectors

Even a fast priority command can suffer a delay if, when it is sent, the collector is already busy executing a very large normal priority command.

For this situation, you can configure an additional collector called a *fast collector*. The fast collector is a special collector that is dedicated to commands in the fast queue. When the fast queue is empty, the fast collector is dormant.

For example, if you configure a fast collector for the Telnet protocol, Prime Network will have:

- One Telnet *fast* collector that only executes commands in the Telnet fast queue. If the Telnet fast queue is empty, the Telnet fast collector is dormant.
- One Telnet (default) collector that executes commands in both the Telnet normal and fast queues. (Remember that the default collector always executes commands in the fast queue first. If the Telnet fast collector is occupied, the Telnet (default) collector will execute the next command in the fast queue.)

Collectors and Thread Sharing

To decrease the overall number of threads used at the VNE layer, each AVM maintains pools of threads that are shared by the VNEs. VNEs acquire and release the threads as needed, in an asynchronous fashion.

One thread pool is dedicated to activation scripts. This thread pool grows dynamically, up to the number of VNEs in the AVM. Each thread is destroyed after 60 seconds of inactivity. Even if you expect a large number of activation scripts to run in parallel, you should see no IO degradation. However, we recommend that you do not run more than 100 concurrent activation scripts on a unit.

Considerations for Using Fast Commands and Fast Collectors

There are obvious benefits of marking commands with a fast priority, and configuring an additional fast collector. But these methods also have some cost and possible risks.

Risks of Using the Fast Command Priority

Only a small number of registration commands should have a fast command priority. If too many commands are marked as fast, the queue for the fast commands can become long, with the following results:

- The purpose of command priorities is defeated because even fast commands have to wait in a queue.
- The normal commands are delayed even further because they are not executed until the (long) fast queue is empty.

Risks of Using Fast Collectors

We recommend that you do not configure an additional fast collector for the following reasons:

- The additional collector can impact system scale performance. In Prime Network, because each collector works in a separate thread, every VNE configured with a fast collector will consume an additional thread. If a large number of VNEs are configured with fast collectors, system performance can be significantly degraded.
- The additional collector could significantly reduce overall management traffic throughput. Every VNE configured with a fast collector opens an additional management connection to a device. Opening multiple connections in parallel can cause a significant increase in NE CPU levels, which can greatly reduce the overall throughput of management traffic.

General Recommendation for Fast Commands and Fast Collectors

For commands that are high priority, mark the command with the fast command priority. Do *not* configure an additional fast collector unless the command takes an unusually long time to execute.

Expedited Commands and Activation Scripts and Fast Collectors

By default, all expedited commands, activation scripts, and CPU monitoring commands have a fast command priority.

CPU monitoring commands have a fast command priority so that Prime Network can quickly identify and respond to high CPU issues that may affect the device and overall system.

Expedited commands have a fast command priority, but only for their *first* execution. Normally, expedited commands execute with little delay. When it has successfully executed, the expedited command returns to a normal command priority. You should only consider using an additional fast collector if expedited commands are consistently delayed by other commands that require a long time to execute. To find out which commands are expedited, refer to the specific syslog, trap, and command descriptions in:

- [Cisco Prime Network Supported Syslogs](#)
- [Cisco Prime Network Supported Traps](#)
- [Cisco Prime Network 4.2 User Guide](#)

Activation scripts (which are converted into commands) have a fast command priority by default. However, activation scripts must adhere to a more strict timeout mechanism than expedited commands.

All commands—expedited commands or commands in activation scripts—have a timeout period which begins when command execution starts. But activation scripts have an additional timeout on the gateway. This gateway timeout begins when the commands are sent to the VNE. If a collector is occupied for an extended period, the gateway timeout may expire and the activation will fail.

If activation commands are timing out, consider the following approaches:

- For devices with marginal timeouts (that is, devices for which there is a very small difference between the script timeout and the time required for the longest command to execute), consider slightly increasing the activation script timeout. However, this is not appropriate for complex device configuration commands.
- For very complex devices with commands that require several minutes to execute, consider configuring an additional fast collector. Increasing the timeout is not appropriate because the increase would have to be sizable. This would result in Prime Network taking a long time to detect activation script failures, hence reducing the system throughput.

General Recommendation for Using Fast Collectors with Expedited Commands and Activation Scripts

The default behavior (described earlier) should be sufficient for both activation scripts and expedited commands. Consider an additional fast collector only if commands are experiencing unacceptable delays.

If you decide to configure additional fast collectors, limit it to the smallest possible number of VNEs—in other words, *only* for VNEs with the most critical need. Also be sure to monitor the system for any effects on device CPU and system scale performance.

Configuring a Command With the “Fast” Command Priority

By default, all commands have a normal command priority and are executed by the collector in a FIFO basis. You can mark a command to have the fast (high) command priority, which means it will be placed in the collector’s fast queue rather than its normal queue. Use the following procedure to edit the command priority in the registry.



Note

We recommend that you do not change any of these settings. Changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Before You Begin

- Read [Risks of Using the Fast Command Priority](#), page 12-34.
- Read [General Recommendation for Fast Commands and Fast Collectors](#), page 12-34.



Note

This procedure requires a gateway restart.

To set a command priority to fast, use the following procedure.

Step 1

Log into the gateway as *pnuser* and change to the Main directory:

```
# cd $ANAHOME/Main
```

Step 2 Issue the following command to configure commands with the fast command priority. The variable *registry-path* is the path to the command to be configured. For example, for the CPU usage command in Cisco IOS devices, use the following:

```
# ./runRegTool.sh -gs 127.0.0.1 set 0.0.0.0 "site/registry-path/cpu usage
snmp/instrumentationservices/command/priority" fast
```

Step 3 Restart the gateway server. See [Stopping and Restarting Prime Network Components, page 3-16](#).

Creating a Fast Collector for a VNE

By default, every protocol has only one collector (that is, no fast collector). You can configure a fast Telnet or SNMP collector for a VNE by editing the registry.



Note Before you configure a fast collector, try using the fast command priority mechanism. See [Configuring a Command With the “Fast” Command Priority, page 12-35](#).



Note We recommend that you do not change any of these settings. Changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Before You Begin

- Read [Risks of Using Fast Collectors, page 12-34](#).
- Read [General Recommendation for Using Fast Collectors with Expedited Commands and Activation Scripts, page 12-35](#).

To create a fast Telnet or SNMP collector *for a specific VNE*, use the following procedure.

Step 1 Log into the gateway as *pnuser* and change to the Main directory:

```
# cd $ANAHOME/Main
```

Step 2 Issue the following command to create a new fast collector for a specific VNE. In the following, *avmxxx* is the AVM ID, *vne-key* is the VNE name, and *vne-ip* is the VNE IP address.

If the VNE is on the gateway server, *unit-IP* should be **127.0.0.1**.

If the VNE is on a unit server, *unit-IP* should be the unit's IP address.

- To create an SNMP fast collector for the VNE with the ID *vne-key*:

```
# ./runRegTool.sh -gs 127.0.0.1 set unit-IP
"avmxxx/agents/da/vne-key/ips/vne-ip/protocols/snmp/maxfastcollectors" 1
```

- To create a Telnet fast collector for the VNE with the ID *vne-key*:

```
# ./runRegTool.sh -gs 127.0.0.1 set unit-IP
"avmxxx/agents/da/vne-key/ips/vne-ip/protocols/telnet/maxfastcollectors" 1
```

Step 3 Restart the VNE (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).



Note

Be sure to monitor the system for any effects on device CPU and system scale performance.

Changing Settings That Control VNE Data Saved After Restarts

Persistency is the ability to store information in the unit for later use. These topics describe the VNE persistency mechanism in Prime Network:

- [Persistency Overview, page 12-37](#)
- [Alarm Persistency, page 12-38](#)
- [Instrumentation Persistency, page 12-40](#)
- [Topology Persistency, page 12-41](#)



Note

These topics describe some of the persistency registry settings. Changes to the registry should be performed only with the support of Cisco. For details, contact your Cisco account representative.

Persistency Overview

Persistency information is stored across unit, AVM, and VNE restarts. This accelerates the startup time after restarts because Prime Network does not have to re-poll the complete NE.

VNE data persists during runtime when a VNE polls data from a device, and the VNE updates the files in the file system for changes in the device's response according to the persistency variables. When a VNE is started or restarted, the persistency information is read from these files once. Every normal polling or refresh that takes place after the first time will read the data from the device itself and not from the files.

VNE data persistency is lost in the following scenarios (but alarm persistency is saved):

- A user manually moves the VNE to another AVM, or moves the parent AVM to another unit.
- A unit server high availability event occurs, causing a unit to switch over to the standby unit.
- The device the VNE models is reconfigured (for example, a new sysOID or software version change).

The upgrade mechanism automatically clears all persistency files on Prime Network gateways and units. This option does not clear the alarm history that is stored in the Fault Database.

Instrumentation Persistency

Instrumentation persistency is used mainly to:

- Shorten the starting time of VNEs for devices. When the information from the local file system is used, the device's response time and network latency are eliminated; thus the VNE finishes modeling its first state very quickly.
- Provide information about the old state of the VNE, to initiate alarms if the status has changed while the VNE was unloaded. For example, a Port Down alarm is initiated only if the port status was up and changed to down. This ensures that an alarm is not issued on ports which should be down. By maintaining information about the old state of the port, the system understands whether or not the current state is valid.
- Help lower the CPU load on the device while starting when many polling commands are generated. Also, when persistence data is loaded from the unit, traffic bandwidth between the unit and device is much lower than when the system is loaded using "ordinary" device discovery and modeling.

For more information, see [Instrumentation Persistency, page 12-40](#).

Topology Persistency

Topology persistency creates topology between devices on startup when the VNE is loaded, instead of performing the entire discovery process. Verification of the links is then performed. For more information, see [Topology Persistency, page 12-41](#).

Alarm Persistency

Alarm persistency saves information about the VNE components that send alarms. When a VNE sends an alarm, the VNE can save this information (that it has sent an alarm of type X). This information can then be used by the VNE components after restarts to verify whether the VNE needs to send clearing alarms where changes have occurred in the device when the VNE was down. For more information, see [Alarm Persistency, page 12-38](#).

Alarm Persistency

Alarm persistency enables the system to clear alarms that relate to events that occurred while the system was down. For example, a Link Down alarm is generated, and then the system goes down. While the system is down, a Link Up event occurs in the network, but because the system is down, it does not monitor the network. When the system goes up, the alarm is cleared because the system remembers that a Link Down alarm exists, and the system needs to clear it by sending a corresponding alarm.

Persisting events are held in the AlarmPersistencyManager. Each VNE contains an AlarmPersistencyManager object. Alarms are added to and removed from the AlarmPersistencyManager object in order to maintain the status of an event, whether it exists in the repository or not; that is, whether an up alarm or a clearing alarm has been generated. Persistency files are associated with a VNE using the VNE's agent ID (not the VNE IP address). Two copies of alarm persistency information are maintained: one in the memory, and the other on disk.

At startup, the AlarmPersistencyManager retrieves the events persisted for the containing VNE.

Event data in the files is updated at the following times:

- At shutdown.
- After a change, when an event is added or removed.
- After a specific interval of time has passed. This prevents data from being rewritten to the persistency file when a stream of events is added or removed during a short period of time, because the data is saved only after the specified period of time has elapsed.

Initialization

Alarm persistency is controlled by settings in the registry. Global alarm persistency information is stored in `agentdefaults.xml`. The major settings are listed in [Table 12-7](#). The settings for these configurable items only apply when trying to retrieve data from the persistency files. Individual event persistency information is described in [Configuring Alarm Persistency for a Specific Event, page 12-40](#).



Note

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 12-7 *Default Settings for Alarm Persistency*

Registry Entry	Description	Default Value
enabled	Enabled the persistency mechanism for this VNE.	true
writing-delay	Interval (in milliseconds) between the arrival of a new event or the removal of an existing event, and the writing activity of the persistency file.	300000 (5 minutes)
max-alarm-age-in-days	How many days an event remains in a persistency file before it becomes obsolete.	7

Retrieving Events

At startup, each VNE calls its `AlarmPersistencyManager` to load the persisting events.

If the file does not exist or is corrupt, no events are loaded. Faulty event objects are not loaded. Events which have been in the file for longer than the configured maximum age are not loaded. No age tests are held during ordinary runtime.

Storing Events

At shutdown, events are saved to the VNE's event persistency file as a precaution in case the events have not already been saved. These files are associated with a VNE using the VNE's agent ID (not IP address).

Removing an Event

An event is searched for and removed using the same information which was used to add it. The event is removed from memory because a clearing event (for example, a Link Up alarm) has been generated, and the persistency information is no longer required. After the removal, the `AlarmPersistencyManager` stores the events after a writing delay, as specified in the registry.

Removing an Event and Clearing an Alarm

The `AlarmPersistencyManager` is able to search for and remove an event, and send a clearing alarm for the event, if it is found that this information is no longer required because the alarm has been cleared.

After an event has been added to or removed from the `AlarmPersistencyManager`, a delayed message is sent to the `AlarmPersistencyManager`. Upon its arrival, the message triggers the events to be stored to the file.

Configuring Alarm Persistency for a Specific Event

Alarm persistency can be configured per event using the setting described in [Table 12-8](#). Event-specific persistency information is stored in `event-persistency-application.xml`.



Note

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 12-8 Registry Setting for Alarm Persistency for a Specific Event

Registry Entry	Description	Default Value
alarm-persistency	Enable persistency for a specific event.	See Cisco Prime Network 4.2 Supported Service Alarms

In the following LDP Neighbor Loss alarm, the LDP Neighbor Down event marks the alarm as present in the system (persisted), and the LDP Neighbor up event is used to clear the alarm from persistency (unpersist):

```
<key name="LDP neighbor loss">
  <entry name="default">event-persistency-application/templates/generic persistency
  event</entry>
  <key name="sub-types">
    <key name="LDP neighbor down">
      <entry name="alarm-persistency">persist</entry>
    </key>
    <key name="LDP neighbor up">
      <entry name="alarm-persistency">unpersist</entry>
    </key>
  </key>
</key>
```

Instrumentation Persistency

The instrumentation layer persists the information that was collected from the device to the file system. When the VNE restarts, it uses this information to emulate the device's response, and thus the VNE can be modeled according to its last persistent state. The next polling instance is performed against the real device.

The registry entries that control instrumentation persistency are provided in [Table 12-9](#).



Note

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 12-9 Registry Settings for Instrumentation Persistency

Registry Entry	Description	Default Value
persistencydir	Specifies the directory in which persistency information is saved on the local file system. This is a relative path. Allowed values are a string that represents the relative directory in the file system.	instrumentor-persistency

Table 12-9 Registry Settings for Instrumentation Persistency (continued)

Registry Entry	Description	Default Value
persistencylevel	<p>Controls the level of persistency to be used. The allowed values are Full (persisted) or Off (not persisted).</p> <p>These values can be used for certain commands to make sure some are persisted and some are not.</p> <p>Note If a compound command contains both Full and Off persistency levels, Prime Network will use the full level for all commands.</p>	Full
persistencystorageenabled	Controls whether the whole storage mechanism is enabled.	true
persistencystorageinterval	<p>Interval (in milliseconds) for which the data to be persisted is accumulated and then written to the persistent storage in bulk. Files are only updated if they have changed.</p> <p>The default value (20 minutes) is a compromise between small intervals (which cause more I/O operations in the local file system) and long intervals (which result in stored information not being up-to-date).</p>	1200000 (20 minutes)
persistencytimeout	<p>Timeout period (in milliseconds) at which initial data is marked as obsolete; all subsequent commands will run directly on the device.</p> <p>If the persistency mechanism is enabled when the instrumentation layer starts, it loads all the data from the files. This data can be used for the commands only the first time they are executed. Some commands can be used for the first time, long after other commands have finished multiple cycles; for example, commands which run only when the status on the device has changed.</p> <p>The default value (1 minute) is a compromise between a small value (which can cause the instrumentation layer to ignore the persistent data) and a large value (which causes the data to be retrieved long after the VNE has finished loading).</p> <p>Note We recommend that this value be at least 600000 (1 minute).</p>	600000 (1 minute)

Topology Persistency

Prime Network supports persistency for Layer 1 topological connections. Layer 1 topology supports one connection per Device Component (DC), so the physical topology reflects a single port connected by a single link.

The following topologies are persisted:

- Layer 1 counter-based topologies.
- Static topologies.

Static topology, which identifies physical links configured by the user, is persisted once a user configures the static link between the two entities. This link is then stored in the registry, in the AVM key that contains the specific VNE registrations.

For other topologies, every time a link is created, the persistency mechanism writes the link to this file. When a link is disconnected, the file representing the link is removed.

**Note**

Topology persistency assumes that the XID (the unique device component ID) is persistable. For example, the port XID should remain the same after the device reboots or after the VNE reboots. This is not dependent on whether the ifIndex is changed from time to time.

Topology persistency is controlled by the setting listed in [Table 12-10](#).

**Note**

All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 12-10 Registry Setting for Topology Persistency

Registry Entry	Description	Default Value
persistency	Enable physical topology persistency. Note We recommend that this entry remain enabled.	true

Creating Connections Between Unmanaged Network Segments (Cloud VNEs and Links)

Cloud VNEs represent *unmanaged* network segments that are connected to two or more *managed* segments. This prevents interruptions to alarm correlations and affected subscribers for the managed segments.

These topics describe how to add and remove links between two ports of two network elements in the network that are connected to some unmanaged network segment through a Cloud VNE. Dynamic links are used to connect these ports to a cloud.

Static links override any existing autodiscovered topology in the system. A static link is identical in all respects to a link that is autodiscovered.

- [Unmanaged Segments and Cloud VNEs, page 12-43](#)
- [Creating and Deleting Static Links, page 12-50](#)

**Note**

If you create a cloud VNE with a static connection to a device, and you upgrade to a later version of Prime Network, the connection between the cloud VNE and the device may be lost. You should delete and recreate the link.

Unmanaged Segments and Cloud VNEs

Three types of technology simulations are supported for Cloud VNEs: Frame Relay, ATM, and Ethernet. If you want to work with Cloud VNEs with Ethernet support, see [Ethernet on Cloud VNEs, page 12-43](#).

Administrators can create Cloud VNEs that represent:

- A single device to which two or more *managed* segments of the network can be connected. In this case, the Cloud VNE builds a model with port type and technology that is identical to its adjacent VNEs and virtual forwarding components. Each physical port in a VNE can connect to only one Cloud VNE.
- Multiple unmanaged segments and multiple technologies, as long as each technology is in a different network segment.
- Multiple Cloud VNEs, each one representing a portion of an unmanaged network.

All VNEs can also be configured to connect dynamically to a Cloud VNE. When loading, the VNE gathers whatever data is relevant to the Cloud VNE, and sends the data to it. Upon receiving this information, the Cloud VNE builds the corresponding model to allow the topology to connect the two VNEs.

To create a Cloud VNE, you must do the following:

1. Create the VNE using Prime Network Administration. You only have to provide a name for the VNE. No additional protocols need to be configured for the Cloud VNE. See [Ethernet on Cloud VNEs, page 12-43](#).
2. Connect the cloud VNE to a device, which will automatically populate the Cloud VNE with technology and topology information. See [Connecting the Cloud VNE to a Device, page 12-45](#).

**Note**

Unmanaged segments must be pure switches; no routing can be involved with the segment.

Ethernet on Cloud VNEs

When using an Ethernet LAN cloud to represent unmanaged network segments, be aware of the following:

- For Ethernet interfaces with duplicate IPs, see [Configuring Duplicate IP Addresses on Ethernet Interfaces, page 12-44](#).
- Devices on both sides of the cloud must communicate so that a Cloud VNE can build the forwarding information properly; otherwise, their MAC addresses do not appear in each other's ARP or bridging tables.
- The logic that builds the bridging table assumes that each port in the network has a unique MAC address. If multiple ports with the same MAC address do exist in the network, the Cloud VNE will not function properly.
- The logic that builds the bridging table assumes there all VLANs in the network have different IDs. If multiple VLANs with the same ID do exist on any of the VNEs connected to the cloud, the VLANs will be connected together on the cloud.
- A router with an interface that is an ingress point of a Martini tunnel (with no IP address configuration) cannot be connected to a cloud. A Layer 2 tunnel represents a point-to-point pseudowire in the network.

- The size of the Ethernet Cloud VNE depends on the number of devices, their configurations and the number of VLANs that are connected to it.
- The Layer 2 devices in the unmanaged cloud segment cannot contain VLAN rewrite configurations that are not supported by the Cloud VNE.
- The Cloud VNE does not support the Q-in-Q technology. If VLAN stacking is configured on an unmanaged segment, or if ports with Q-in-Q configuration are connected to the cloud, the cloud might not be able to simulate the behavior of the unmanaged segment.
- The Cloud VNE does not have Spanning Tree Protocol (STP) awareness, so any link from a device to the unmanaged network is assumed to be in a nonblocking state. This might cause the forwarding information calculated by the Cloud VNE to be inaccurate.
- By default, Prime Network does not display VLANs that are present on the device and that cannot be deleted, such as restricted Fiber Distributed Data Interface (FDDI), Token Ring, and other nonEthernet VLANs.

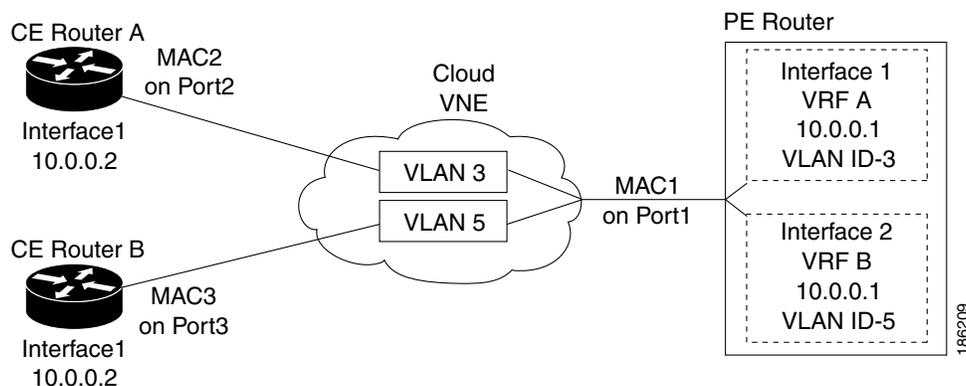
**Note**

Most of the Ethernet functionality—namely, MAC and VLAN support—is only available for dynamic links.

Configuring Duplicate IP Addresses on Ethernet Interfaces

Figure 12-12 provides an example of a configuration of duplicate IP addresses on Ethernet interfaces that are connected to the same Cloud VNE.

Figure 12-12 Duplicate IP Addresses on Ethernet Interfaces



In Figure 12-12, a PE router and two CEs are connected to an unmanaged Ethernet access network, represented by a Cloud VNE.

The PE router is connected to the Cloud VNE through Port1. Two interfaces configured on Port1 are connected to different VRFs (VRF A and VRF B). Both VRF interfaces are configured with the same IP address (10.0.0.1). Each interface is configured with a different VLAN encapsulation (VLAN-ID 3 and VLAN-ID 5), and is connected to a different VLAN in the unmanaged network (VLAN 3 and VLAN 5).

The two CEs are connected to different VLANs in the unmanaged network: CE A is connected to VLAN 3 through Port2, and CE B is connected to VLAN 5 through Port3. Both Port2 and Port3 are access ports (that is, untagged ports with no VLAN encapsulation) and are configured with identical IP addresses (10.0.0.2).

The Cloud VNE creates a similar port for each port connected to it, and two bridges, one per VLAN (that is, a bridge for VLAN 3 and a bridge for VLAN 5). Each bridge contains a forwarding table with the MAC addresses of the ports connected to that VLAN. In this example, the bridge representing VLAN 3 contains MAC1 and MAC2, and the bridge representing VLAN 5 contains MAC1 and MAC3.

Connecting the Cloud VNE to a Device

Each Cloud VNE has a unique agent ID (that is used as the Cloud VNE's identifier) that cannot be used to access any network element. To connect a regular VNE to a Cloud VNE, the VNE must be configured with the physical port that should be connected, and the agent ID of the Cloud VNE.

When configuring a Cloud VNE for dynamic operation, the cloud model and the topology (that is, the link between the Cloud VNE and the adjacent VNE) are discovered and managed automatically by Prime Network.

To configure the Cloud VNE to operate dynamically, after creating a new Cloud VNE, you must:

1. Identify the OID of the physical port layer of the port that will connect to the Cloud VNE.
2. Connect the ports on the adjacent VNEs to the Cloud VNE.
3. For Cloud VNEs with Ethernet support, configure the Cloud VNE's permissible subnets.

Before You Begin

If you are creating an Cloud VNE with Ethernet support, read [Ethernet on Cloud VNEs, page 12-43](#).

Step 1 Identify the physical port layer OID of the ports that will connect to the Cloud VNE.

- a. Perform a **GET** on the PhysicalRoot to retrieve all the physical models of the VNE up to the physical layer. The **GET** command can be optimized to retrieve only necessary information using a specific retrieval specification.

The following is an example of an optimized **GET** command for VNE PE_South:

```
<command name="Get">
  <param name="oid">
    <value>{ [ManagedElement (Key=PE_South) ] [PhysicalRoot]}</value>
  </param>
  <param name="rs">
    <value>
      <key name="imo-view-controller">
        <entry name="depth">10</entry>
        <entry name="register">true</entry>
        <entry name="cachedResultAcceptable">>false</entry>
      <key name="requiredProperties">
        <key name="com.sheer.imo.IPhysicalRoot">
          <entry name="EquipmentHolders"/>
        </key>
        <key name="com.sheer.imo.IEquipmentHolder">
          <entry name="ContainedEquipmentHolder"/>
          <entry name="ContainedEquipment"/>
        </key>
        <key name="com.sheer.imo.IEquipment">
          <entry name="SupportedPTPs"/>
        </key>
        <key name="com.sheer.imo.IPhysicalTerminationPoint">
          <entry name="ContainedCurrentCTPs"/>
        </key>
      </key>
      <key name="requiredAspects">
    </key>
  </param>
</command>
```

```

        </key>
      </value>
    </param>
  </command>

```

- b. Identify the physical layer (port) OID according to port name or location. You will need For example, from the result of the previous step's GET command, this would be the physical layer OID of port FastEthernet1/0 in PE_South.

```

<?xml version="1.0" encoding="UTF-8" ?>
<IPhysicalRoot>
  <ID type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] }</ID>
  <EquipmentHolders type="IMObjects_Array">
    <IChassis>
      <ID type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] }</ID>
      <ContainedEquipmentHolder type="IMObjects_Array">
        ....
        <IEquipmentHolder>
          <ID
type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] [Slot (SlotNum=1) ] }</I
D>
          <ContainedEquipment type="IModule">
            <ID
type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] [Slot (SlotNum=1) ] [Mod
ule] }</ID>
            <SupportedPTPs type="IMObjects_Array">
              <IPortConnector>
                <ID
type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] [Slot (SlotNum=1) ] [Mod
ule] [Port (PortNumber=FastEthernet1/1) ] }</ID>
                <ContainedCurrentCTPs type="IMObjects_Array">
                  <IPhysicalLayer>
                    <ID
type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] [Slot (SlotNum=1) ] [Mod
ule] [Port (PortNumber=FastEthernet1/1) ] [PhysicalLayer] }</ID>
                    </IPhysicalLayer>
                  </ContainedCurrentCTPs>
                </IPortConnector>
              <IPortConnector>
                <ID
type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] [Slot (SlotNum=1) ] [Mod
ule] [Port (PortNumber=FastEthernet1/0) ] }</ID>
                <ContainedCurrentCTPs type="IMObjects_Array">
                  <IPhysicalLayer>
                    <ID
type="Oid">{ [ManagedElement (Key=PE_South) ] [PhysicalRoot] [Chassis] [Slot (SlotNum=1) ] [Mod
ule] [Port (PortNumber=FastEthernet1/0) ] [PhysicalLayer] }</ID>
                    </IPhysicalLayer>
                  </ContainedCurrentCTPs>
                </IPortConnector>
              </SupportedPTPs>
            </ContainedEquipment>
          </IEquipmentHolder>
        ....
      </ContainedEquipmentHolder>
    </IChassis>
  </EquipmentHolders>
</IPhysicalRoot>

```

The OID is

```
{[ManagedElement(Key=PE_South)][PhysicalRoot][Chassis][Slot(SlotNum=1)][Module][Port(Po
rtNumber=FastEthernet1/0)][PhysicalLayer]}
```

- c. Replace / (the slash) in the port name with **!\slash!** when specifying the OID in the CLI command.

For example, the OID from the preceding step should be changed to:

```
{[ManagedElement(Key=PE_South)][PhysicalRoot][Chassis][Slot(SlotNum=1)][Module][Port(Po
rtNumber=FastEthernet1!\slash!0)][PhysicalLayer]}
```

- Step 2** Connect the ports to the Cloud VNE. For each VNE that represents a device that is connected to the unmanaged network represented by the Cloud VNE, do the following:

- a. Log into the gateway as *pnuser* and change to the Main directory:

```
# cd $ANAHOME/Main
```

- b. Obtain the cloud agentId by running the following command, where *cloudAvmId* is the ID of the AVM in which the cloud was defined:

```
cat registry/avmcloudAvmId.xml
```

In the following example, a cloud was defined on AVM 358:

```
# cat registry/avm358.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<key name="avm358">
  <entry name="default">mcvm</entry>
  <entry name="avmkey">AVM 358</entry>
  <key name="agents">
    <key name="da">
      <key name="Cloud">
        <entry name="default">sheer/cloud/product/software versions/default
version</entry>
        <entry name="element type">SHEER_NETWORKS_CLOUD</entry>
        <entry name="deletePersistency">>true</entry>
        <entry name="adaptivePollingType">1</entry>
        <key name="creationTime">
          <entry name="time">1311516933201</entry>
        </key>
        <key name="pollingrates">
          <entry name="default">pollinggroups/default</entry>
        </key>
        <key name="amsi">
          <key name="topology">
            <key name="dynamic">
              <key name="permissible-subnet">
                <entry name="subnet">0.0.0.0/0</entry>
              </key>
            </key>
            <key name="static"></key>
          </key>
        </key>
        <key name="maintenance">
          <entry name="activated">>false</entry>
        </key>
        <key name="ips">
          <entry name="agentId">784</entry>
        </key>
      </key>
    </key>
  </key>
</key>
```

...

- c. From the gateway, run the following CLI commands:

```
# ./runRegTool.sh -gs 127.0.0.1 add unit-IP
"avmxxx/agents/da/vne-key/dcs/instance/physical-layer-oid/cloud topology"
# ./runRegTool.sh -gs 127.0.0.1 set unit-IP
"avmxxx/agents/da/vne-key/dcs/instance/physical-layer-oid/cloud topology/id"
cloud-agent-ID
```

The following lists the parameters you must define:

Parameter	Meaning
<i>unit-IP</i>	The IP address of the machine on which the parent AVM resides (for the VNE that will connect to the Cloud VNE). If the AVM is on the gateway server, <i>unit-IP</i> should be 127.0.0.1 .
<i>avmxxx</i>	The ID of the parent AVM (for the VNE that will connect to the Cloud VNE).
<i>vne-key</i>	The name of the VNE which will connect to the Cloud VNE.
<i>physical-layer-oid</i>	The OID of the VNE port which will connect to the Cloud VNE. This is the OID you identified in Step 1 of this procedure.
<i>cloud-agent-ID</i>	The agent ID of the Cloud VNE. (This is the Cloud VNE you created in Adding a New Device Type to Prime Network, page 4-17.)

Example:

```
# ./runRegTool.sh -gs 127.0.0.1 add 192.168.100.1
"avm900/agents/da/PE_South/dcs/instance/{[ManagedElement(Key=PE_South)][PhysicalRoot][Chassis][Slot(SlotNum=1)][Module][Port(PortNumber=FastEthernet1\!slash\!0)][PhysicalLayer]}/cloud topology"
# ./runRegTool.sh -gs 127.0.0.1 set 192.168.100.1
"avm900/agents/da/PE_South/dcs/instance/{[ManagedElement(Key=PE_South)][PhysicalRoot][Chassis][Slot(SlotNum=1)][Module][Port(PortNumber=FastEthernet1\!slash\!0)][PhysicalLayer]}/cloud topology/id" 784
```

The previous example connects a VNE named PE_South (which resides in avm900 on unit 192.168.100.1) with a Cloud VNE that has the agent ID 784. The connection with the Cloud VNE is made using the physical layer of PE_South that has the OID:

```
{[ManagedElement(Key=PE_South)][PhysicalRoot][Chassis][Slot(SlotNum=1)][Module][Port(PortNumber=FastEthernet1/0)][PhysicalLayer]}
```

is connected to the Cloud VNE with the agent ID 784.

- d. Restart the VNE (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).

Step 3 If the cloud represents an Ethernet access network, configure the permissible subnets on the Cloud VNE. This will permit IP interfaces to connect to other entities only if the interfaces are on the specified subnets. This minimizes the number of connections the Cloud VNE handles.



Note

This configuration applies to the Cloud VNE, not to the adjacent VNEs. The most common use case is to configure permissible subnets to allow the connection through all subnets that are connected to the cloud (by configuring 0.0.0.0/0, or 0::0/0 for IPv6).

For each Cloud VNE, do the following:

- a. Log into the gateway as *pnuser* and change to the Main directory:

```
# cd $ANAHOME/Main
```

- b. From the gateway, run the following CLI commands:

```
# ./runRegTool.sh -gs 127.0.0.1 add unit-IP
"avmxxx/agents/da/cloud-vne-key/amsi/topology/dynamic/permisible-subnet"
# ./runRegTool.sh -gs 127.0.0.1 set unit-IP
"avmxxx/agents/da/cloud-vne-key/amsi/topology/dynamic/permisible-subnet/subnet"
permisible-subnet
```

The following lists the parameters you must define:

Parameter	Meaning
<i>unit-IP</i>	The IP address of the machine on which the parent AVM resides (for the VNE that will connect to the Cloud VNE). If the AVM is on the gateway server, <i>unit-IP</i> should be 127.0.0.1 .
<i>avmxxx</i>	The ID of the parent AVM (for the VNE that will connect to the Cloud VNE).
<i>cloud-vne-key</i>	The name of the Cloud VNE (not the adjacent VNE).
<i>permisible-subnet</i>	The permisible subnet in the address/mask (such as 192.168.1.0/24).



Note You can add multiple subnets by running the second CLI command multiple times. Each entry has a different name (e.g., *subnet-2*, *subnet-3*, and so on).

Example:

```
# ./runRegTool.sh -gs 127.0.0.1 add 192.168.100.1
"avm900/agents/da/EthernetCloud/amsi/topology/dynamic/permisible-subnet"
# ./runRegTool.sh -gs 127.0.0.1 set 192.168.100.1
"avm900/agents/da/EthernetCloud/amsi/topology/dynamic/permisible-subnet/subnet"
0.0.0.0/0
```

The previous example configures the permisible subnet 0.0.0.0/0 (meaning all IPv4 subnet connections are allowed), on a Cloud VNE named EthernetCloud (which resides in avm900 on unit 192.168.100.1). To allow all IPv6 subnet connections, use subnet 0::0/0.

- c. Restart the Cloud VNE (by right-clicking it and choosing **Actions > Stop**, then **Actions > Start**).

Creating and Deleting Static Links

**Note**

If you create a cloud VNE with a static connection to a device, and you upgrade to a later version of Prime Network, the connection between the cloud VNE and the device may be lost. You should delete and recreate the link.

You can create a static link between devices by selecting the two end ports from the device physical inventory in Prime Network Administration. To create a static topological link, you need to supply the exact location of the two end ports (at both ends of the link). The physical hierarchy in which the port is located defines the location of a port, as follows:

Device > [shelf] > module > [submodule] > port

Links are bidirectional, and need to be added only once.

**Note**

By default, a user can view a link in Prime Network Vision only if *both* link endpoints are in the user's device scope. If you want to make links viewable if only *one* endpoint is in a user's scope, you must edit the registry as described in [Displaying Links Based On Whether Endpoints Are In User's Scope](#), page 6-4.

The new link is validated after the two ports are selected, but before the link is added. Validation checks:

- The similarity of the connector port types (for example, RJ45 on both sides).
- Layer 2 technology type (for example, ATM OC-3 on both sides).
- The physical layer.
- The operation status of both ports.
- One of the ports is part of another link.

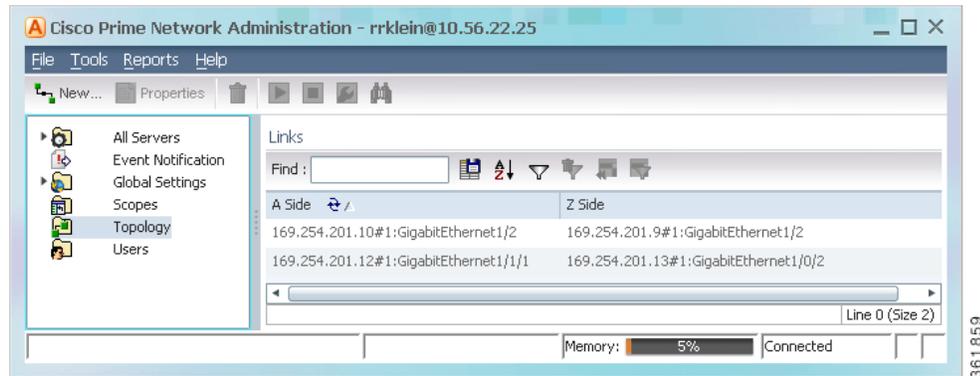
For links between LAGs (IEEE 802.3ad), Prime Network also validates the following:

- The underlying dynamically discovered physical connections do not contradict the new static link.
- Different number of ports configured under the two LAGs.

If validation reveals that one of the ends is part of a static link, you are asked to delete the previous link manually. If validation reveals that one of the ends is part of a dynamic link, the previous link is overridden.

Figure 12-13 provides an example of the Topology window.

Figure 12-13 Topology Window



The Topology window displays all static links defined in the system, including the A side and Z side of the link.

To create a new static link:

-
- Step 1** Right-click **Topology** and choose **New Static Link**.



Note Any changes made in the Topology window are saved automatically and are registered immediately in Prime Network.

The A Side and Z Side lists enable you to choose the devices and the ports for the static link. When you select a device from the list, the physical inventory of the device is displayed in the dialog box.

- Step 2** From the A Side and Z Side lists choose a device. The physical inventory of each device is displayed in the related area of the dialog box.
-

To delete a static link, right-click the link in the Topology window and choose **Delete**.

Improving TACACS Server Performance by Changing VNE Telnet/SSH Login Rates (Staggering VNEs)

The VNE staggering mechanism controls the rate at which VNEs initiate Telnet/SSH connections across the network. This prevents degraded performance on TACACS servers, which can result when there are many concurrent connections.

A gateway service controls whether VNEs on the unit are permitted to initiate Telnet login sequences. It does this by limiting the number of concurrent connections, and distributing those connections based on how AVMs and VNEs are allocated. The service runs on AVM 99 on the gateway server and units. This service does not monitor the TACACS server; it only controls the number and distribution of connections.

When the gateway receives a Telnet authorization request, it queues the requests in a FIFO (first in, first out) manner. If the gateway denies the request, the VNE communication state is changed to Device Partially Managed and a System event is generated. You can verify that the gateway denied the service by checking the VNE Status Details window. The VNE will continue to request the login, and once a connection is permitted, the VNE communication state changes accordingly and a clearing System event is generated.

You can enable the VNE staggering mechanism using the Registry Controller.

**Note**

Changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

- Step 1** Select **Tools > Registry Controller > Advanced VNE Configurations > VNE Staggering Mechanism** from the main menu of the Administration GUI client.

VNE Staggering Settings	What the Setting Controls	Default
Enable VNE staggering mechanism	Enables the VNE staggering service on all VNEs managed by the gateway and units.	false
Authorize before login	Instructs VNE protocol to contact the gateway or unit for permission before allowing a login to proceed (permission is controlled by unit's management service)	false
Number of permitted concurrent logins	Number of concurrent connections allowed by the gateway service. The connections are distributed based on how AVMs and VNEs are allocated. The gateway service runs on AVM 99 on the gateway and all units, in a distributed fashion.	10
VNE login timeout	Specify the amount of time allotted for the VNE to successfully log in. If exceeded, the login is disallowed. (This allows the next VNE in the queue to proceed with its login.)	300000 (ms)
VNE login supervisor		(none)

- Step 2** Verify your changes to ensure you want to overwrite the current registry settings because after you click Apply, you cannot retrieve your settings using the Restore button.

- Step 3** Click **Apply**.

- Step 4** Start the gateway service.

```
# ./runRegTool.sh -gs 127.0.0.1 set 0.0.0.0 avm99/services/initlevel15/vneLoginSupervisor
com.sheer.system.os.services.vne.login.VneLoginSupervisorServiceImpl
```

- Step 5** Restart AVM 99 on all units.

```
# runall.csh networkctl -avm 99 restart
```

Tracking VNE-Related Events

The following table provides ways you can get historical information on VNE-related events.

For historical events related to:	See:
Editing VNE polling settings VNE communication state issues (Device Unreachable or Partially Reachable) VNE investigation state issues (Device Unsupported) Events related to reduced polling and adaptive polling	AVM and other appropriate log files (see Log Files Reference , page C-3) The following reports, which you can launch from the main menu by choosing Reports > Run Report > Events Reports > : <ul style="list-style-type: none"> • Detailed Service Events • Detailed System Events • Detailed Security Events

