HP Network Node Manager i-series Software

Causal Analysis White Paper

Software Version 8.11

Communications and data networks have grown significantly in size and complexity, and so have the number of faults that occur. A single failure can trigger many alarms, and distinguishing the real problem from the anecdotal alarms has become a bottleneck for the network operator. Traditional event correlation systems have been able to reduce alarms, but these systems fall short in terms of identifying the root cause in an automated way.

The HP Network Node Manager i-series Software (NNMi) Causal Engine technology applies **root cause analysis (RCA)** to network symptoms, using a causality-based approach. NNMi event correlation actively models the behavioral relationship between managed objects, determining root cause and impact based on a MINCAUSE algorithm. The causal analysis software can deal with ambiguity and partial symptoms.

In addition, NNMi actively solicits symptoms during analysis and reacts dynamically to topology changes. NNMi provides an end-to-end diagnosis of network faults, with the ability to handle a hierarchy of models.

CONTENTS

The Causal Engine and NNMi Incidents

Communications and data networks have grown significantly in size and complexity, and so have the number of faults that occur. A single failure can trigger many alarms, and distinguishing the real problem from the anecdotal alarms has become a bottleneck for the network operator. Traditional event correlation systems have been able to reduce alarms, but these systems tend to fall short in terms of identifying the root cause in an automated way.

The HP Network Node Manager i-series Software Causal Engine technology applies **root cause analysis (RCA)** to network symptoms, using a causality-based approach.

Causality Analysis - High Level Discussion

Causal Engine technology provides the following high-level features:

- Uses the `NmsApa` jboss service to analyze your network
- Model-based approach to RCA
  - Models the behavioral relationship between managed objects
  - Uses an object model in addition to event causality to drive analysis
  - Determines root cause and impact based on the MINCAUSE algorithm
  - Effectively deals with ambiguity and partial symptoms
- Dynamic
  - Actively solicits symptoms during analysis
  - Reacts dynamically to topology changes
- Extensible
  - Employs a hierarchy of modules (import/export)
  - Provides an end-to-end diagnosis of network faults
  - Provides the ability to add rule sets in future offerings

Causal Engine Concepts

Causal Engine technology uses the following sequential approach:

1. Formally define the root-cause problems and symptoms.
2. Perform analysis by relating symptoms to root-cause problems using the model. Symptoms come from two sources:
   - StatePoller, where the symptoms are state changes
   - Events, where the symptoms are traps
3. Generate conclusions that relate to the root cause.

Causal Engine conclusions contain artifacts that are associated with the model. Artifacts include the following details:

- Incident generation
- Incident correlation
- Incident suppression
- Incident cancellation
- Status on relevant objects

Concept of Status

In addition to incident manipulation, the `NmsApa` service sets status on relevant objects. Status indicates the overall health of an object and is determined from the outstanding conclusions. Every conclusion has a severity associated with it; the status reported is the most severe of all outstanding conclusions. In addition, conclusions inform the user of the underlying cause (or reason) for an object's status.

The `NmsApa` service uses the following status categories in decreasing order of severity:

- Unknown
- Disabled
- Critical
- Minor
- Warning
- Normal
- No Status

What is an Episode?

The goal of the `NmsApa` service is to present a single incident that the operator or network engineer can work with. To do this, the `NmsApa` service uses the concept of an episode. An episode exists for a specific duration, during which secondary failures are either correlated or suppressed based on configuration. For example:

- The `AddressNotResponding` incident is suppressed by the `InterfaceDown` incident, according to the following scenario:
    - When an IPv4 address stops responding to ICMP, an episode begins, which exists for the duration of 60 seconds.
    - Within that duration, if the interface associated with that IPv4 address goes down, the `NmsApa` service concludes that the interface down condition caused the IPv4 address to stop responding.
    - Therefore, the `AddressNotResponding` incident is not generated. Only the `InterfaceDown` incident is generated.
    - To ensure that the `InterfaceDown` incident is detected within the duration, the `NmsApa` service issues a named poll for that interface. The incident enables the network engineer to fix the root cause of the problem which, in this case, is the interface.
    - If the interface does not go down during the episode, the `NmsApa` service generates an `AddressNotResponding` incident. If the interface goes down after the episode, the `InterfaceDown` incident is generated. In this case, the network engineer has to treat the two problems separately.
- The `NodeDown` incident correlates the `InterfaceDown` incident from one-hop neighbor interfaces, according to the following scenario:
    - When an interface goes down, a `NodeDown` episode begins for the neighboring node, which exists for the duration of 300 seconds.
    - Within that duration, if the node goes down, the `InterfaceDown` incident is correlated beneath the `NodeDown` incident.

- The `InterfaceDown` incidents from all one-hop neighbors are correlated beneath the `NodeDown` incident. You can review the `InterfaceDown` incidents as supporting evidence for the `NodeDown` incident.

What Does NNMi Analyze?

NNMi analyzes a variety of network elements (Ports, Interfaces, Addresses, and so forth). NNMi monitors these devices using either the SNMP protocol or ping to retrieve information about the network element.

The following list shows the specific network elements that NNMi monitors and analyzes:

- SNMP Agent
- IPv4 Address
- Interface
- Node
- Connections
- Node Groups
- Link Aggregated Ports
- Link Aggregated Connections
- Redundant Router Groups
- Component Health Sensors

An **SNMP agent** is a process running on the managed node, which provides management functions. The SNMP agent is responsible for managing interfaces and ports on the managed node; it can be associated with one or more nodes.

The following list shows the possible NNMi status categories associated with an SNMP agent:

`Critical` - SNMP Agent doesn't respond to SNMP queries.

`Normal` - SNMP Agent responds to SNMP queries.

`No Status` - SNMP Agent is not polled.

An **IPv4 address** is a routable address that responds to ICMP. IPv4 addresses are typically associated with nodes. NNMi reports the status of a node as follows:

`Disabled` - The interface associated with this IPv4 address is administratively down or disabled.

`Critical` - IPv4 address doesn't respond to ICMP queries (ping the device).

`Normal` - IPv4 address responds to ICMP queries.

`No Status` - IPv4 address is not polled.

An **interface** is a physical port that can be used to connect a node to the network. NNMi reports the status of an interface as follows:

`Unknown` - The SNMP Agent associated with the interface doesn't respond to SNMP queries. Unknown indicates that the `NmsApa` service cannot determine the health because `ifAdminStatus` and `ifOperStatus` cannot be measured.

`Disabled` - Interface is administratively down (`ifAdminStatus` = down).

`Critical` - Interface is operationally down (`ifOperStatus` = down).

`Normal` - Interface is operationally up (`ifOperStatus` = up).

`No Status` - Interface is not polled.

A **node** is a device that NNMi finds as a result of the spiral discovery process. A node can contain interfaces, boards, and ports. You can separate nodes into two categories:

- Network nodes, which are active devices such as switches, routers, bridges, and hubs
- End nodes, such as UNIX or Windows servers

NNMi typically manages network nodes, reporting node status and component health status as follows:

`Unknown` – The SNMP Agent associated with the node doesn't respond to SNMP queries and polled IPv4 addresses do not respond to ICMP queries. This indicates that NNMi is unable to manage the node.

`Critical` – Any one of the following:

- The node is down as determined by neighbor analysis.
- The node is marked as important and is unmanageable (NNMi cannot access the node from the NNMi server).
- The node is an island (it has no neighbors) and, therefore, is unmanageable.
- The `NmsApa` service cannot determine if the node is down or if the incoming connection is down.
- Fan failure (component health).

`Minor` – Node status can be minor if a managed object contained in the node has an issue, as is the case with the following:

- The SNMP Agent associated with the node doesn't respond to SNMP queries.
- One or more interfaces in the node are down.
- One or more IPv4 addresses on the node do not respond to ICMP.

`Normal` – The SNMP Agent, polled interfaces, and polled IPv4 addresses of the node are up.

`No Status` – The SNMP Agent, all interfaces, and all IPv4 addresses of the node are not polled.

Connections are Layer 2 physical connections and Layer 3 network connections. NNMi discovers connection information by reading forwarding database (FDB) tables from other network devices and by using devices that support discovery protocols such as CDP and EDP. NNMi reports the status of a connection as follows:

`Unknown` – All endpoints of the connection have unknown status.

`Disabled` – Any one endpoint of the connection is disabled.

`Critical`– All endpoints are operationally down.

`Minor` – Any one endpoint is down.

`Warning` – Endpoints have unknown and non-critical status.

`Normal` – All endpoints are operationally up.

`No Status` – Any one endpoint is not polled.

A **node group** is a logical collection of nodes for separating the polling configuration. An administrator creates node-type groupings. For example, some nodes, such as routers, are critical to your business; you might want to poll these routers more frequently. To do so, define a node group containing the critical routers and configure them for a shorter polling cycle.

An NNMi administrator can configure node group status calculations. The out-of-the-box configuration propagates the most severe status as follows:

`Critical` – At least one node in the group has critical status.

`Major` – There are no nodes with critical status, and at least one node in the group has major status.

`Minor` – There are no nodes with critical or major status, and at least one node in the group has minor status.

`Warning` – There are no nodes with critical, major, or minor status, and at least one node in the group has warning status.

`Normal` – There are no nodes with critical, major, minor, or warning status, and at least one node in the group has normal status.

`Unknown` – There are no nodes with critical, major, minor, warning, or normal status, and at least one node in the group has unknown status.

`No Status` – All nodes in the group have no status.

A **link aggregated port** is a set of ports (interfaces) on a switch that are linked together, usually for the purpose of creating a trunk (high bandwidth) connection to another device. Port aggregations have a designated master port and member ports. An administrator can monitor the overall health of the aggregated port to know when the port is degraded. (For example, the master is down, or any of the members is down.)

NNMi reports the status of a link aggregated port as follows:

`Unknown` – All members of the aggregation are unknown.

`Critical` – The master and/or all members of the aggregation are operationally down.

`Minor` – Some member (but not all members) of the aggregation is operationally down.

`Normal` – All members of the aggregation are operationally up.

`No Status` – All members of aggregation are not polled.

A **link aggregated connection** is a connection with endpoints that are link aggregated ports. These are usually high-bandwidth connections that link switches. Link aggregated connections have a designated master connection and member connections. An administrator can monitor the overall health of the aggregated connection to know when the connection is degraded in any way. (For example, the master is down or any of the members is down.)

NNMi reports the status of a link aggregated connection as follows:

`Unknown` – Any member of the aggregation is unknown.

`Critical` – The master and/or all member connections of the aggregation are operationally down.

`Minor` – Some member connection (but not all connections) of the aggregation is operationally down.

`Normal` – All members of the aggregation are operationally up.

`No Status` – All member connections of aggregation are not polled.

A **redundant router group** is a set of routers that are configured to provide redundancy in the network. Such groups use two types of protocols/technologies:

- Hot standby router protocol (HSRP)
- Virtual router redundancy protocol (VRRP)

Redundant router groups usually have a single device acting as the primary, a single device acting as a secondary, and any number of standby devices. If the primary device fails, the secondary device should take over as primary, and one of the standby devices should become secondary. The router groups employ either the HSRP or VRRP protocol to designate the primary, secondary, and standby routers.

NNMi reports the status of redundant router groups as follows:

`Critical` – The group has no acting primary router.

`Major` – The group primary is not properly configured (for example, there are multiple primary routers).

`Minor` – The group secondary is not properly configured (for example, there is no acting secondary router).

`Warning` – The group is functioning but in some way degraded.

`Normal` – The group is functioning properly.

`No Status` – The group is not yet fully discovered or populated.

Large (or more sophisticated) network devices often require special environments and components in order to function properly. Examples are power supplies, fans, voltage regulators, and internal computers. These device components can be monitored by **component health sensors**.

An administrator can monitor the health of these components to know when any of them has failed or is operating marginally. NNMi reports the status of component health sensors as follows:

`Critical` – The component is not functioning properly.

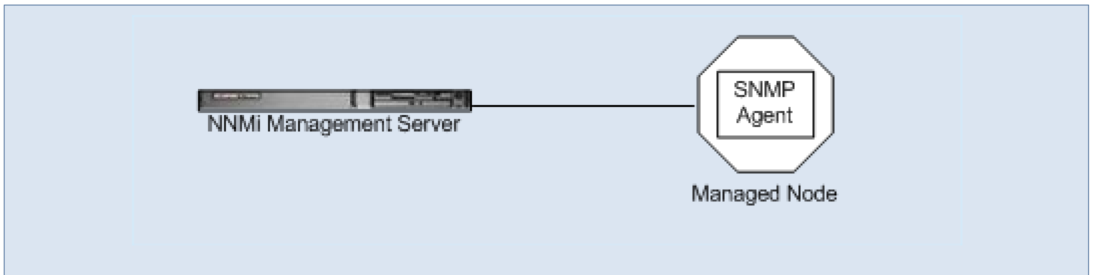
`Normal` – The component is operating properly.

`No Status` – The component is not polled.

What Are the Failure Scenarios?

The following sections describe the fault scenarios that the NNMi Causal Engine analyzes and how the failures are diagnosed. These scenarios describe the symptoms of the failure, as well as the status, conclusions, and incidents that the Causal Engine generates for the failure.

SNMP Agent Not Responding to SNMP Queries



Legend

Managed Node	Network device, such as an Ethernet switch
SNMP Agent	With new community string for managed node
MS communication configuration settings	Not updated with new community strings

Scenario: The SNMP agent is not responding. For instance, the community string for this SNMP agent has been changed, or NNMi's communication configuration settings have not yet been updated, but the node is operational (IPv4 addresses can be pinged).

NOTE: Ping is not enabled out-of-the box. This scenario requires that at least one address is polled.

Root Cause: The SNMP Agent is not responding.

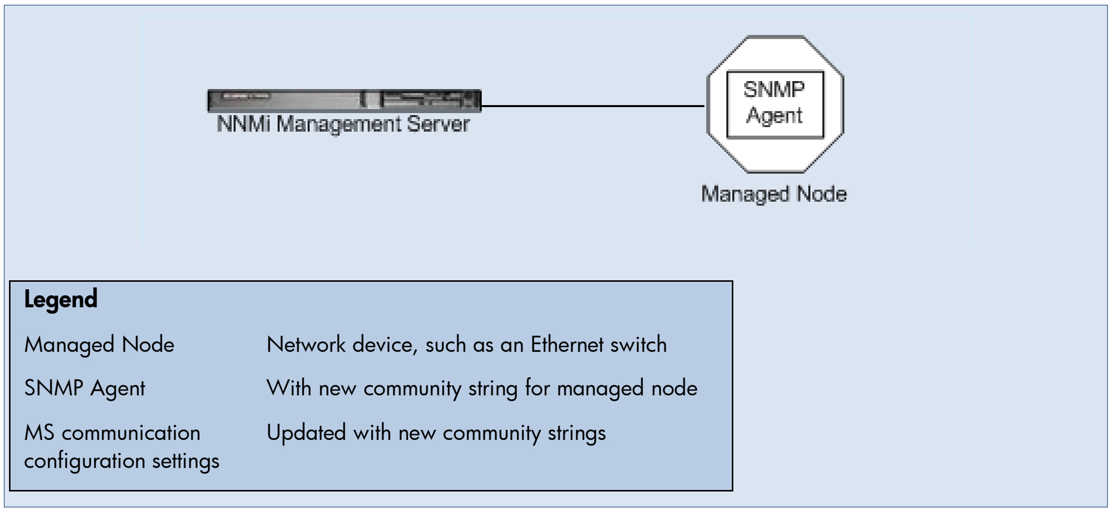
Incident: None generated.

Status: SNMP Agent is in critical status.

Conclusion: `SNMPAgentNotResponding`

Effect: Node status is minor; conclusion on the node is `UnresponsiveAgentInNode`. All polled interfaces have unknown status because they cannot be managed by NNMi. The conclusion on each interface is `InterfaceUnmanageable`.

SNMP Agent Responding to SNMP Queries



Legend

Managed Node	Network device, such as an Ethernet switch
SNMP Agent	With new community string for managed node
MS communication configuration settings	Updated with new community strings

Scenario: This scenario continues the *SNMP Agent Not Responding to SNMP Queries* scenario on page 9. An NNMi administrator has updated the communication configuration settings to include the new community string. The SNMP agent for the managed node starts responding to SNMP queries.

Root Cause: SNMP Agent is responding.

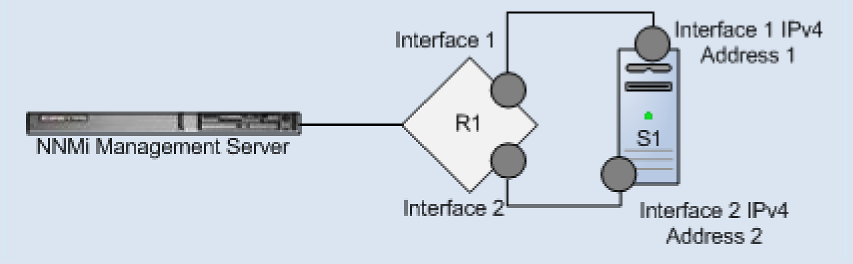
Incident: None generated.

Status: SNMP Agent is in normal status.

Conclusion: `SNMPAgentResponding`

Effect: Node status is normal; conclusion on the node is `ResponsiveAgentInNode`. `InterfaceUnmanagable` is cleared from all polled interfaces and the interfaces revert to their previous status.

IPv4Address Not Responding to ICMP



Legend

R1	Router 1
Route	Changed from interface 1 to interface 2 on the router
S1	Server 1, a multi-homed server
S1 Interface 1	Associated with IPv4 address 1
S1 Interface 2	Associated with IPv4 address 2

Scenario: IPv4 address 1 on Server 1 (S1) is not responding. For instance, the route on Router 1 (R1) has changed from interface 1 to interface 2, so that packets destined for S1 interface 1 are now routed out of R1 interface 2. The associated interface is operational, and the node can be reached because you can ping some IPv4 addresses. The SNMP agent is up.

NOTE: Ping is not enabled out-of-the box. This scenario requires that at least one address is polled.

Root Cause: IPv4 address is not responding.

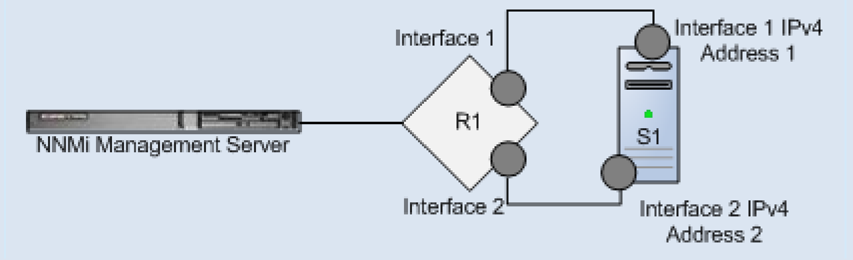
Incident: An `AddressNotResponding` incident is generated.

Status: IPv4 address is in critical status.

Conclusion: `AddressNotResponding`

Effect: Node status is minor; conclusion on the node is `SomeUnresponsiveAddressesInNode`.

IPv4Address Responding to ICMP



Legend

R1	Router 1
Route	Changed from interface 2 to interface 1 on the router
S1	Server 1, a multi-homed server
S1 Interface 1	Associated with IPv4 address 1
S1 Interface 2	Associated with IPv4 address 2

Scenario: This scenario continues the *IPv4Address Not Responding to ICMP* scenario on page 11. The IPv4 address is now responding, the associated interface is operational, and the node can be reached (for example, you can ping some IPv4 addresses, and/or the SNMP agent is up).

Root Cause: IPv4 address is responding.

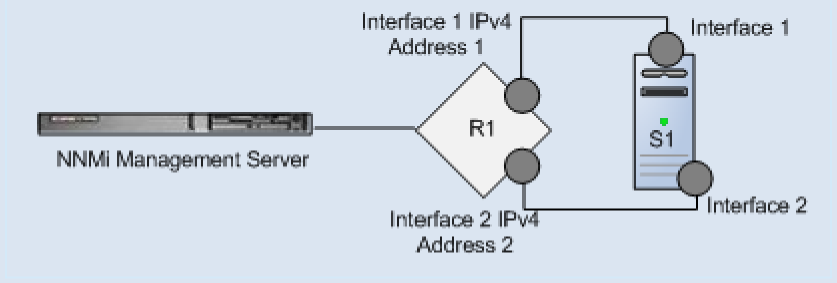
Incident: None generated; the `AddressNotResponding` incident is closed.

Status: IPv4 address is in normal status.

Conclusion: `AddressResponding`

Effect: Node status is normal; conclusion on the node is `ResponsiveAddressesInNode`.

Interface Is Operationally Down



Legend	
R1	Router 1
Interface 1 for R1	Configured to be administratively up but is operationally down Configured for IPv4 address 1
Interface 2 for R1	Configured for IPv4 address 2
S1	Server 1

Scenario: R1 interface 1 is operationally down (`ifOperStatus` = down) and administratively up (`ifAdminStatus` = up). R1 sends a `linkDown` trap. R1 can be reached because you can ping some IPv4 addresses, such as IPv4 address 2. The SNMP agent is up. IPv4 address 1 is associated with interface 1; it has stopped responding to ICMP.

Root Cause: Interface is down.

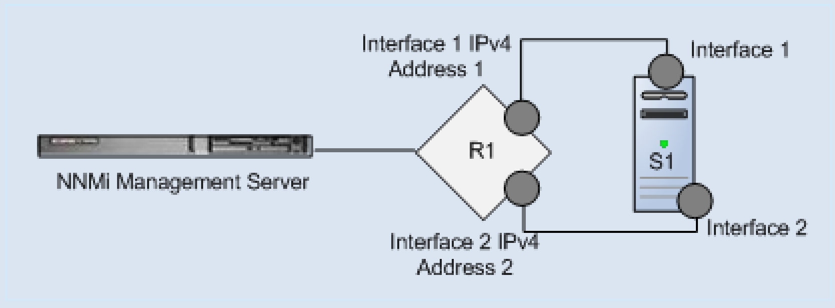
Incident: An `InterfaceDown` incident is generated; the LinkDown incident is correlated beneath the `InterfaceDown` incident.

Status: Interface is in critical status.

Conclusion: `InterfaceDown`

Effect: Node status is minor; conclusions on the node are `InterfacesDownInNode` and `SomeUnresponsiveAddressesInNode`. The address associated with the interface is in critical status; however, no `AddressNotResponding` incident is sent because this incident is suppressed by the `InterfaceDown` incident.

Interface Is Operationally Up



Legend

R1	Router 1
Interface 1 for R1	Configured to be administratively up and operationally up Configured for IPv4 address 1
Interface 2 for R1	Configured for IPv4 address 2
S1	Server 1

Scenario: This scenario continues the *Interface is Operationally Down* scenario on page 13. R1 interface 1 is now operationally up (`ifOperStatus` = up). The node can be reached; you can ping all of its IPv4 addresses. The SNMP agent is up.

Root Cause: Interface is up.

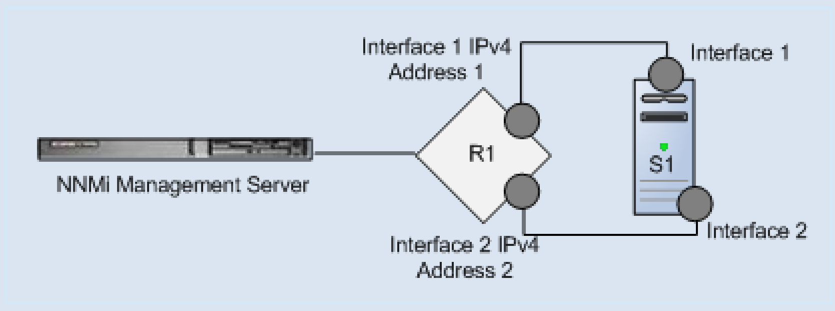
Incident: None generated; the `InterfaceDown` incident is closed.

Status: Interface is in normal status.

Conclusion: `InterfaceUp`

Effect: Node status is normal; conclusion on the node is `InterfacesUpInNode`.

Interface Is Administratively Down



Legend	
R1	Router 1
Interface 1 for R1	Configured to be administratively down Configured for IPv4 address 1

Scenario: R1 interface 1 is administratively down (`ifAdminStatus` = down), but the node can be reached. For instance, you can ping interface 2 and the SNMP agent is up. Disabling R1 interface 1 brings that interface operationally down. The IPv4 address associated with this interface, IPv4 address 1, stops responding to ICMP.

Root Cause: R1 interface 1 is disabled.

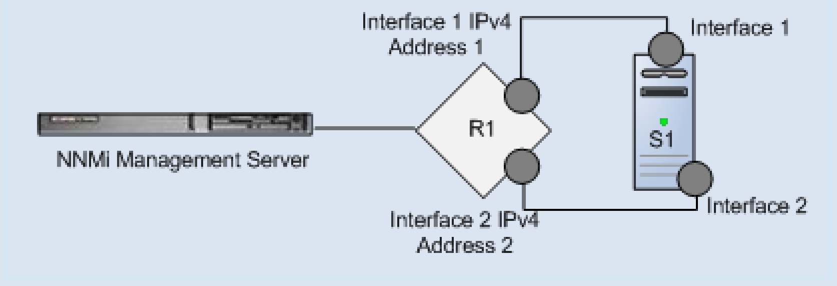
Incident: None generated.

Status: Interface is in disabled status.

Conclusion: `InterfaceDisabled`

Effect: The IPv4 address associated with R1 interface 1 has a status of disabled; the conclusion on the IPv4 address is `AddressDisabled`.

Interface Is Administratively Up



Legend

R1	Router 1
Interface 1 for R1	Configured to be administratively up and operationally up Configured for IPv4 address 1

Scenario: This scenario continues the *Connection is Administratively Down* scenario on page 15. R1 interface 1 is now administratively up (`ifAdminStatus` = up), and you can reach the node by pinging some IPv4 addresses of that interface. The SNMP agent is up. Enabling R1 interface 1 brings it operationally up. The IPv4 address associated with this interface starts responding to ICMP.

Root Cause: Interface is enabled.

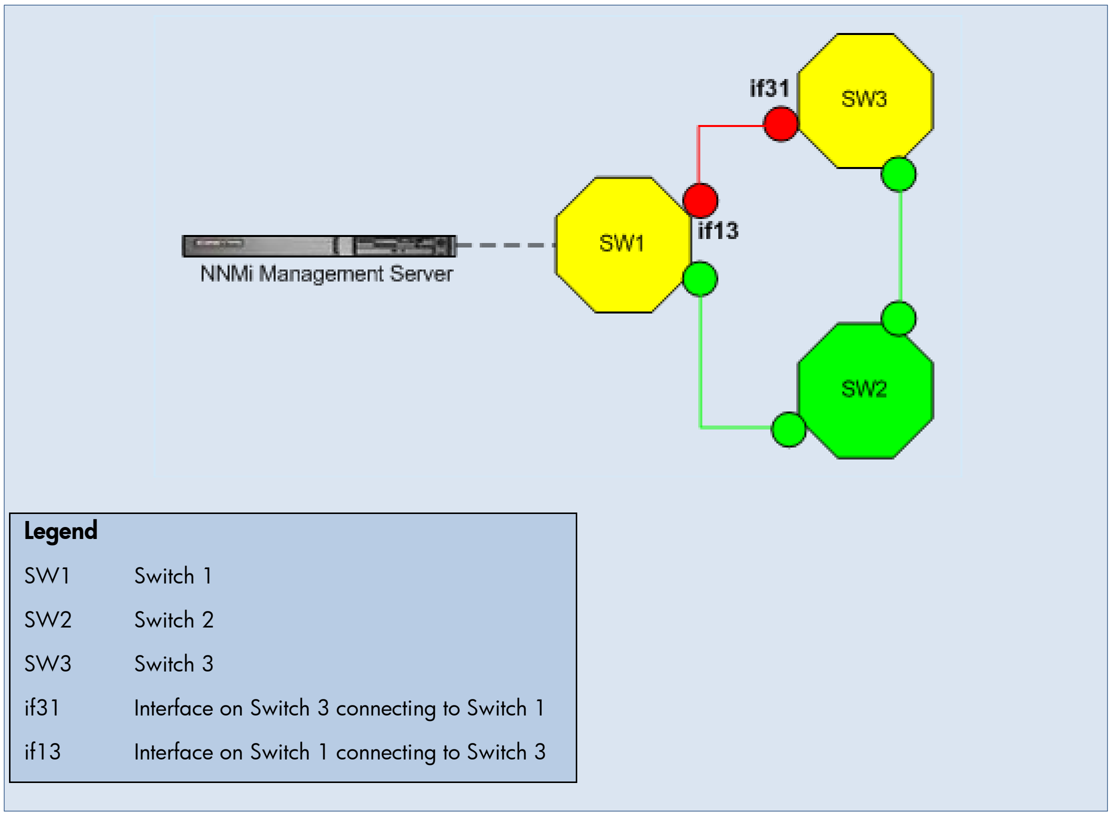
Incident: None generated.

Status: Interface is in normal status.

Conclusion: `InterfaceEnabled`

Effect: The IPv4 address associated with R1 interface 1 has a status of enabled; the conclusion on the IPv4 address is `AddressResponding`.

Connection Is Operationally Down



Legend

SW1	Switch 1
SW2	Switch 2
SW3	Switch 3
if31	Interface on Switch 3 connecting to Switch 1
if13	Interface on Switch 1 connecting to Switch 3

Scenario: The connection between the interface on Switch 3 connecting to Switch 1 (if13) and the interface on Switch 1 connecting to Switch 3 (if31) is down. Traffic flows from the Management Server through Switch 1 (SW1) and Switch 2 (SW2). Both if13 and if31 are marked down.

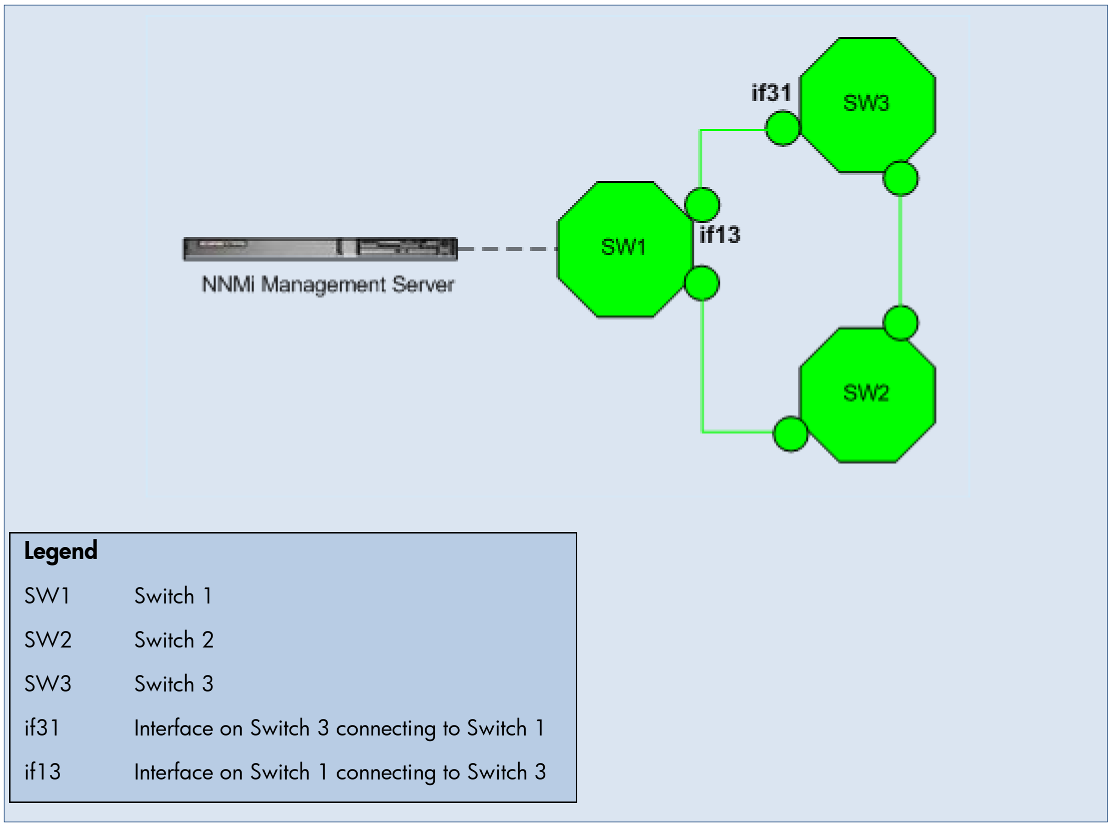
Root Cause: The connection between if13 and if31 is down.

Incident: A `ConnectionDown` incident is generated; the `InterfaceDown` incidents from if13 and if31 are correlated beneath `ConnectionDown`.

Status: Connection is in critical status.

Conclusion: `ConnectionDown`

Connection Is Operationally Up



Legend

SW1	Switch 1
SW2	Switch 2
SW3	Switch 3
if31	Interface on Switch 3 connecting to Switch 1
if13	Interface on Switch 1 connecting to Switch 3

Scenario: This scenario continues the *Connection is Operationally Down* scenario on page 17. The connection between if13 and if31 is now up.

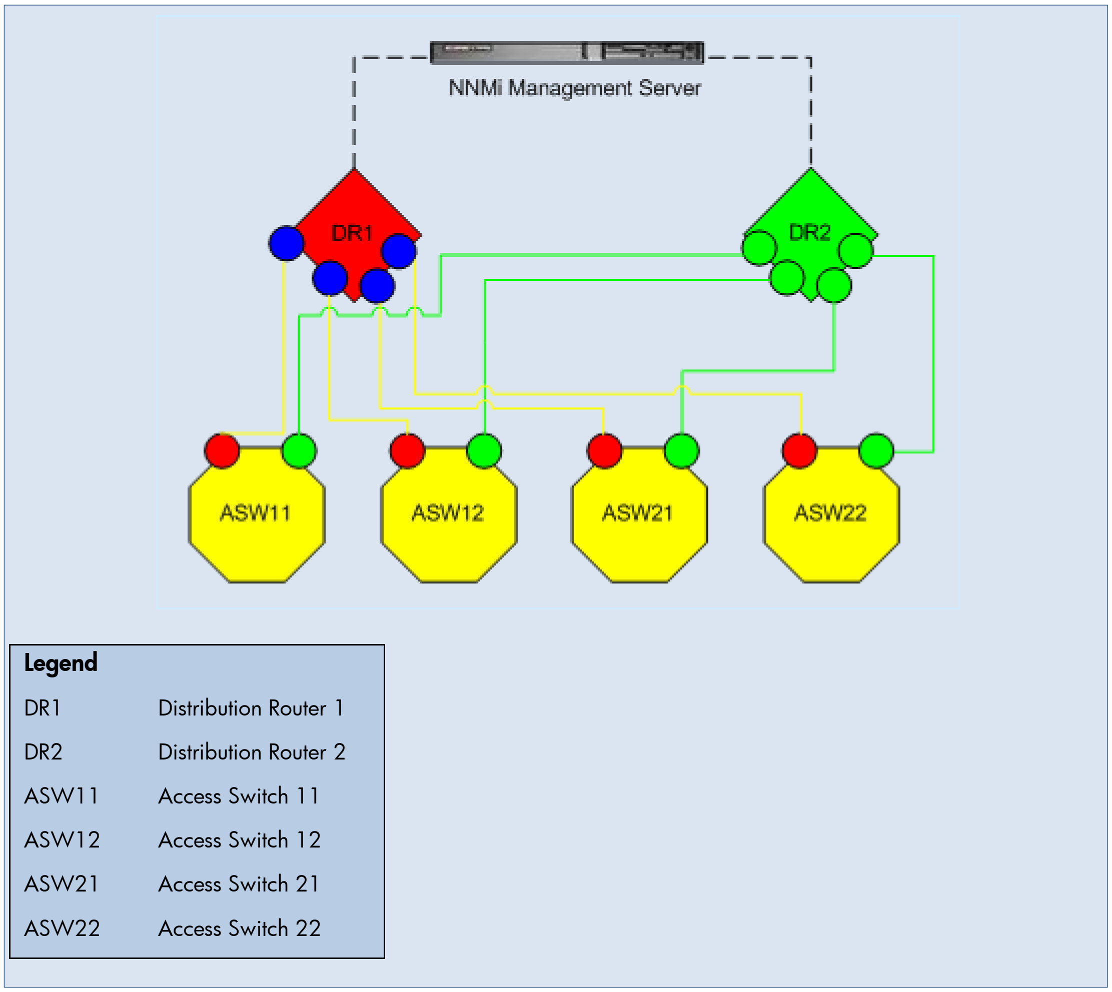
Root Cause: The connection between if13 and if31 is up.

Incident: None generated; the `ConnectionDown` incident is closed.

Status: Connection is in normal status.

Conclusion: `ConnectionUp`

Directly Connected Node Is Down



Legend	
DR1	Distribution Router 1
DR2	Distribution Router 2
ASW11	Access Switch 11
ASW12	Access Switch 12
ASW21	Access Switch 21
ASW22	Access Switch 22

Scenario: Access switches ASW11, ASW12, ASW21, and ASW22 are redundantly connected to the distribution routers, as shown. The distribution routers DR1 and DR2 are directly connected to one another. The distribution router DR1 goes down.

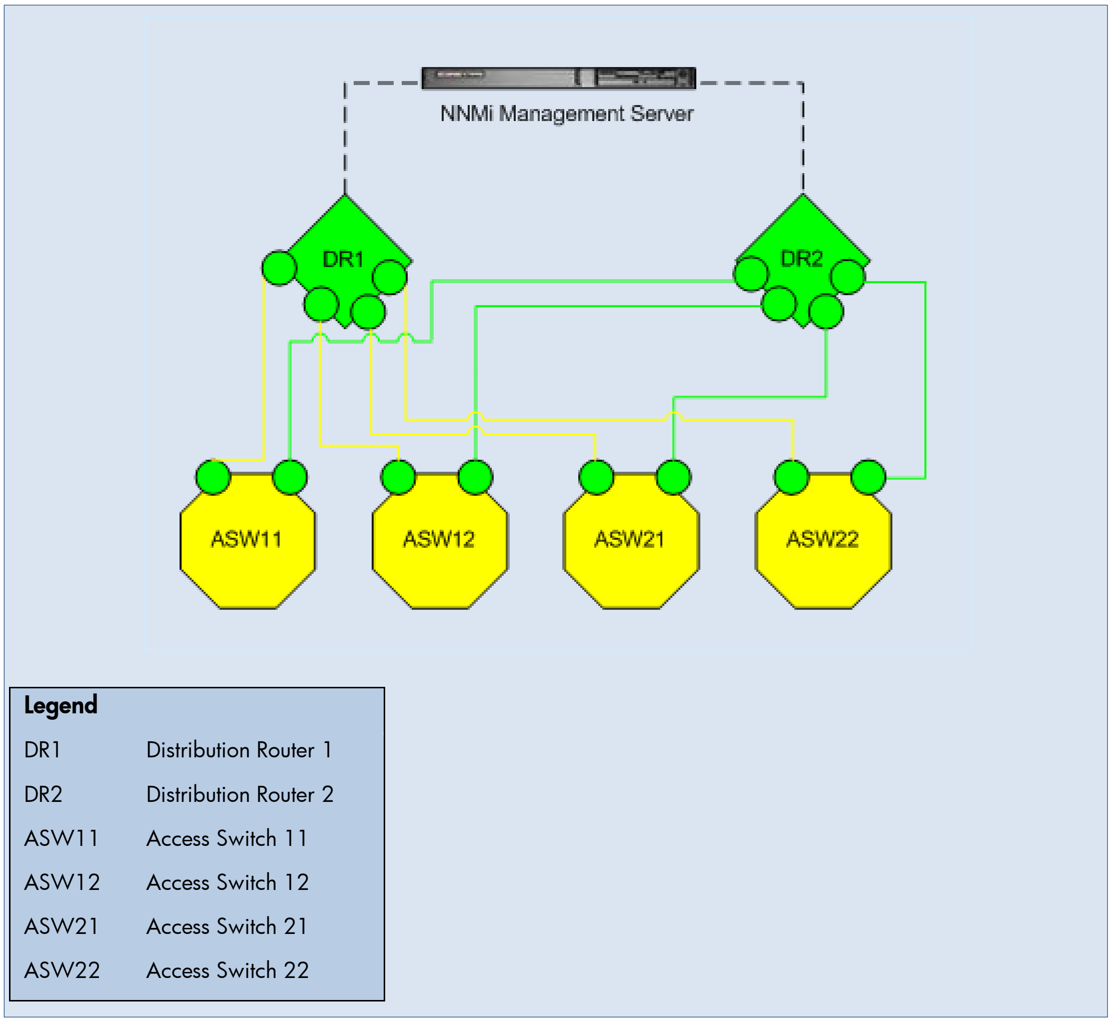
Root Cause: Node DR1 is down according to neighbor analysis.

Incident: A `NodeDown` incident is generated; the `InterfaceDown` incidents from one-hop neighbors are correlated beneath the `NodeDown` incident.

Status: Node is in critical status.

Conclusion: `NodeDown`

Directly Connected Node Is Up



Legend

DR1	Distribution Router 1
DR2	Distribution Router 2
ASW11	Access Switch 11
ASW12	Access Switch 12
ASW21	Access Switch 21
ASW22	Access Switch 22

Scenario: This scenario continues the *Directly Connected Node is Down* scenario on page 19. The distribution router DR1 comes back up.

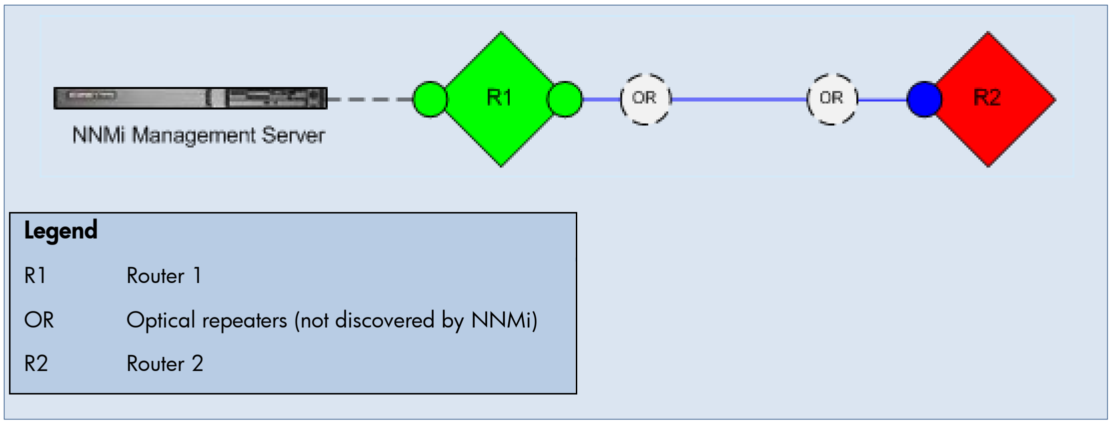
Root Cause: Node DR1 is up.

Incident: A `NodeUp` incident is generated; the `NodeDown` incident is closed by the `NodeUp` incident, and the `NodeUp` incident is also closed.

Status: Node is in normal status.

Conclusion: `NodeUp`

Indirectly Connected Node Is Down



Legend

R1	Router 1
OR	Optical repeaters (not discovered by NNMi)
R2	Router 2

NOTE: The diagram is conceptual. It does not represent an actual NNMi topology map or workspace view.

Scenario: This scenario can occur with any indirect connection where NNMi cannot discover the intermediate devices. In this example, Routers R1 and R2 appear to be directly connected in NNMi topology maps, but in reality these two routers are indirectly connected through optical repeaters. (The optical repeaters do not respond to SNMP or ICMP queries, so they are not discovered by NNMi.)

R2 becomes unreachable, either because its connected interface is down or because the connection between the optical repeaters is down. The interface on R1 that indirectly connects it to R2 is still up because its optical repeater is still up.

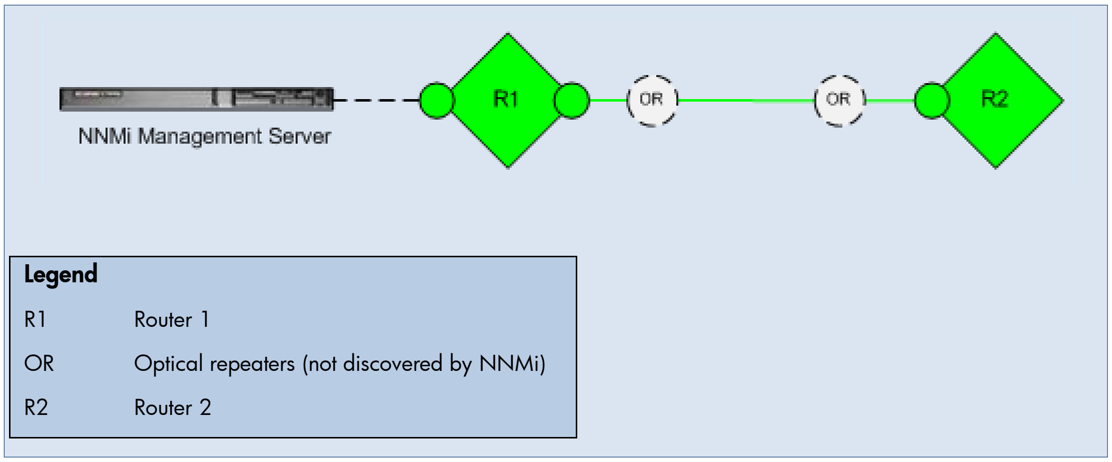
Root Cause: Router R2 is down according to neighbor analysis.

Incident: A `NodeDown` incident is generated.

Status: Node R2 is in critical status.

Conclusion: `NodeDown`

Indirectly Connected Node Is Up



Legend

R1	Router 1
OR	Optical repeaters (not discovered by NNMi)
R2	Router 2

NOTE: The diagram is conceptual. It does not represent an actual NNMi topology map or workspace view.

Scenario: This scenario continues the *Indirectly Connected Node is Down* scenario on page 21. The failed connection comes back up, and R2 becomes reachable.

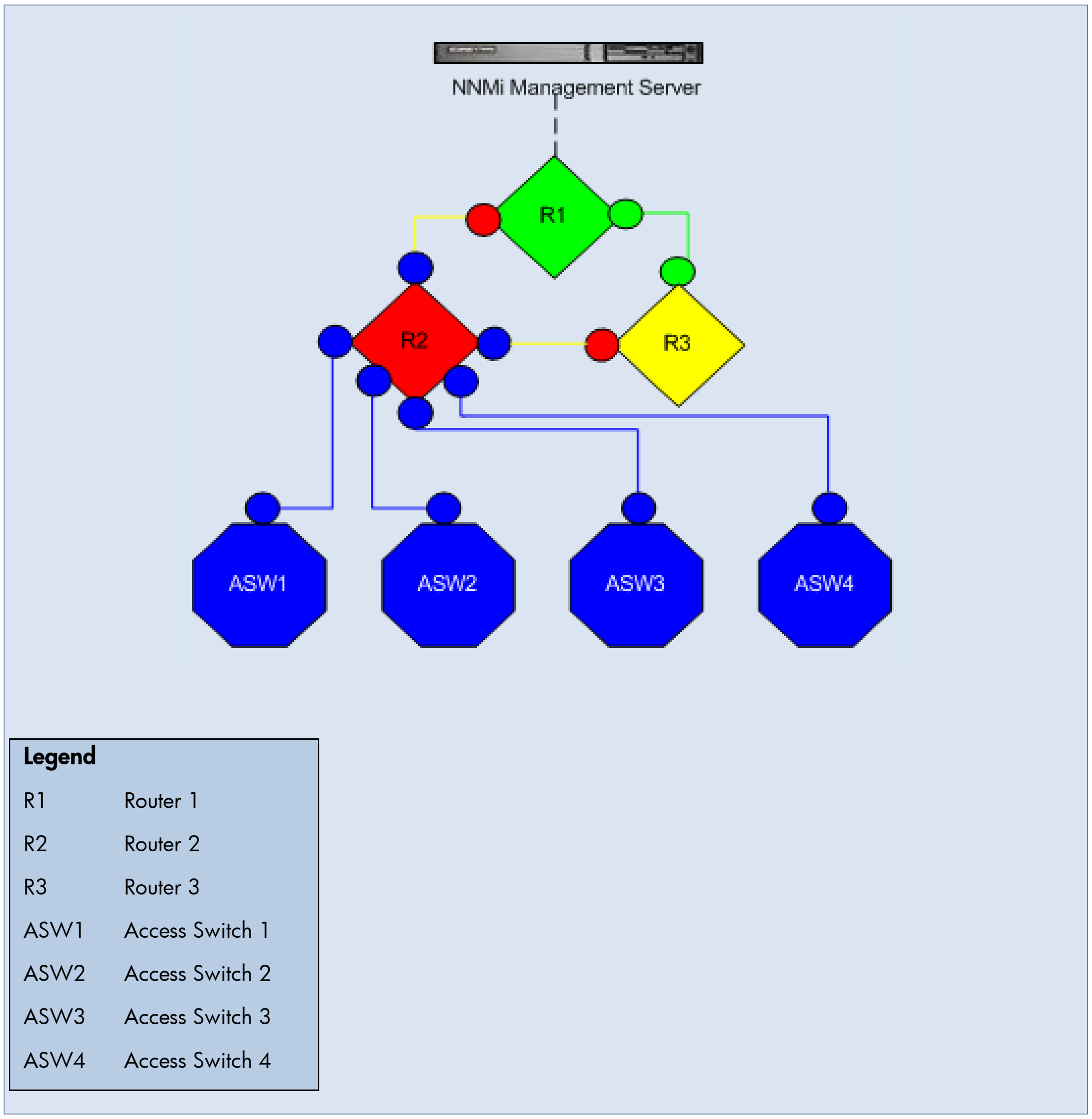
Root Cause: Connection between R1 and R2 is up.

Incident: A `NodeUp` incident is generated; the `NodeDown` incident is closed by the `NodeUp` incident, and the `NodeUp` incident is also closed.

Status: Router R2's status is Normal. Connection status is Normal.

Conclusion: `NodeUp`

Directly Connected Node Is Down and Creates a Shadow



Legend

R1	Router 1
R2	Router 2
R3	Router 3
ASW1	Access Switch 1
ASW2	Access Switch 2
ASW3	Access Switch 3
ASW4	Access Switch 4

Scenario: Router 2 (R2) goes down.

Root Cause: Node R2 is down according to NNMi's neighbor analysis.

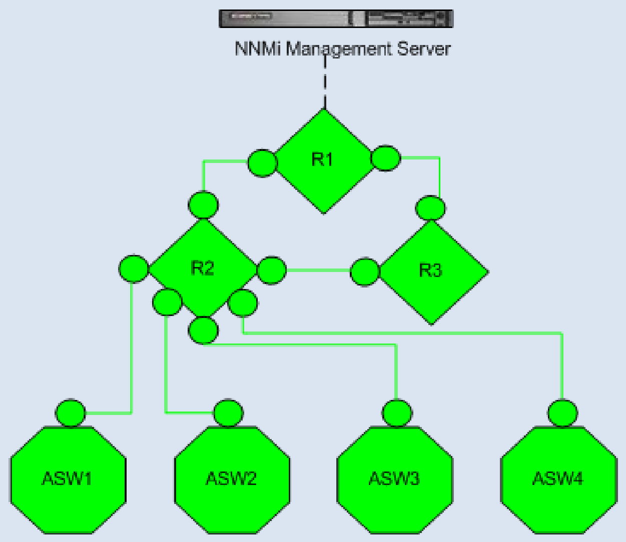
Incident: A `NodeDown` incident is generated; the `InterfaceDown` incidents from one-hop neighbors are correlated beneath the `NodeDown` incident.

Status: Node is in critical status.

Conclusion: `NodeDown`

Effect: All of the access switches are unreachable. The status of all nodes in the shadow is unknown and the conclusion on each of them is `NodeUnmanageable`.

Directly Connected Node Is Up, Clearing the Shadow



Legend	
R1	
R2	Router 2
R3	Router 3
ASW1	Access Switch 1
ASW2	Access Switch 2
ASW3	Access Switch 3
ASW4	Access Switch 4

Scenario: This scenario continues the *Node is Down and Creates a Shadow* scenario on page 23. R2 comes back up.

Root Cause: Node R2 is up.

Incident: A `NodeUp` incident is generated; the `NodeDown` incident is closed by the `NodeUp` incident, and the `NodeUp` incident is also closed.

Status: Node is in normal status.

Conclusion: `NodeUp`

Effect: All of the access switches are now reachable. The status of all nodes in the shadow is normal.

Important Node Is Unreachable

Scenario: A node that is part of the Important Nodes node group cannot be reached.

NOTE: You must add a node to the Important Nodes node group before the `NmsApa` service analyzes the node. If a node becomes unreachable before being added to the Important Nodes node group, the `NmsApa` service does not generate a `NodeDown` incident.

Root Cause: The node is down. The `NmsApa` service does not do neighbor analysis but concludes that the node is down because it was marked as important.

Incident: A `NodeDown` incident is generated; there are no correlated incidents.

Status: The node is in critical status.

Conclusion: `NodeDown`

Important Node Is Reachable

Scenario: This scenario continues the *Important Node is Unreachable* scenario on page 25. The important node comes back up; it can be reached.

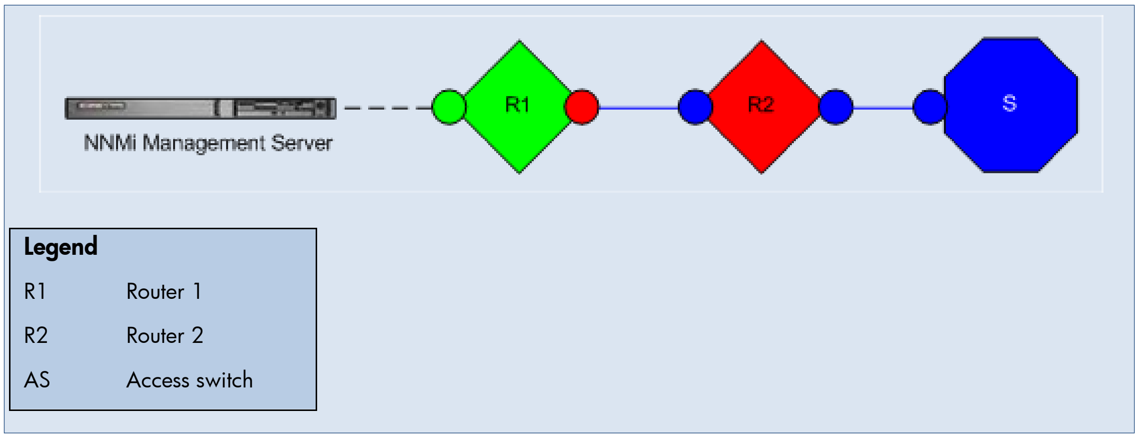
Root Cause: The node is up.

Incident: None generated; the `NodeDown` incident is closed by the `NodeUp` incident.

Status: Node is in normal status.

Conclusion: `NodeUp`

Node or Connection Is Down



Legend

R1	Router 1
R2	Router 2
AS	Access switch

Scenario: There is no redundancy to Router 2 (R2). Either R2 is down or the connection between Router 1 (R1) and R2 is down.

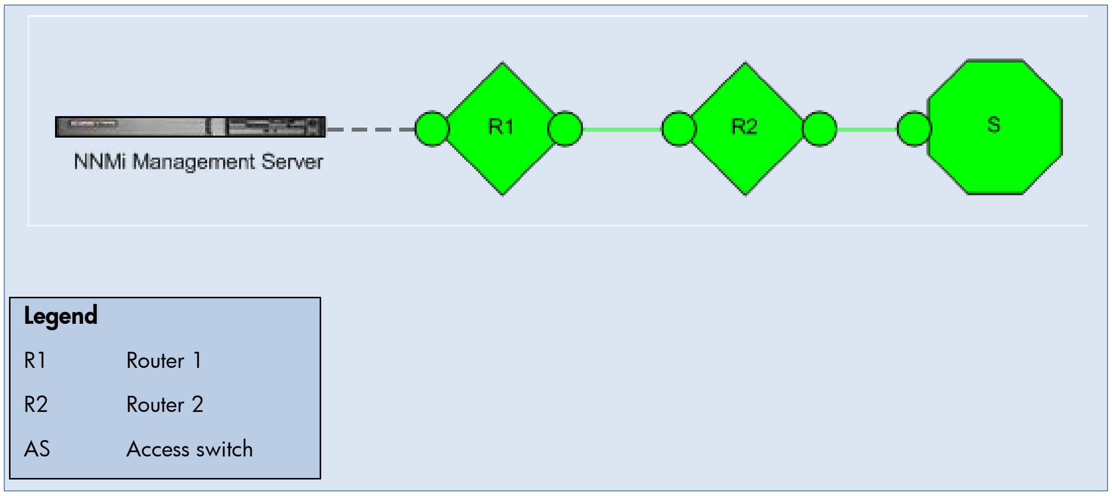
Root Cause: The node or the connection is down.

Incident: A `NodeOrConnectionDown` incident is generated; the source node in this scenario is R2.

Status: Node is in critical status; connection is in warning status.

Conclusion: `NodeOrConnectionDown`

Node or Connection Is Up



Legend

R1	Router 1
R2	Router 2
AS	Access switch

Scenario: This scenario continues the *Node or Connection is Down* scenario on page 26. R2 is now up.

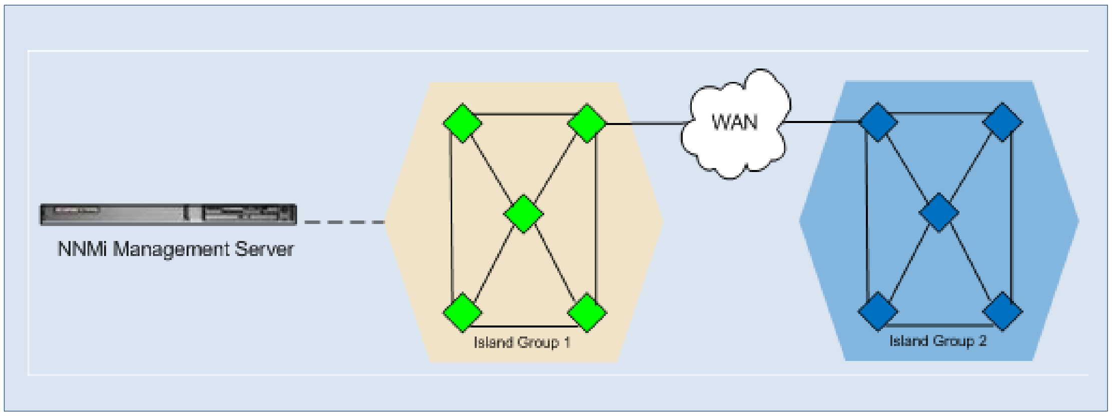
Root Cause: `NodeUp`

Incident: A `NodeUp` incident is generated; the `NodeOrConnectionDown` incident is closed by the `NodeUp` incident, and the `NodeUp` incident is also closed.

Status: The node is in normal status; the connection is in normal status.

Conclusion: `NodeUp`

Island Group Is Down



NOTE: The diagram is conceptual. It does not represent an actual NNMi topology map or workspace view.

Scenario: NNMi has partitioned your network into two Island Groups. The NNMi management server is connected to a node in Island Group 1. Island Group 2 has become unreachable due to problems in your service provider's WAN.

NOTE: Island Groups contain highly-connected sets of nodes that are not connected or are only minimally connected to the rest of the network. For example, NNMi can identify multiple Island Groups for an enterprise network with geographically distributed sites connected by a WAN. Island Groups are created by NNMi and cannot be modified by the user. See For more information about Island Groups, see the NNMi help.

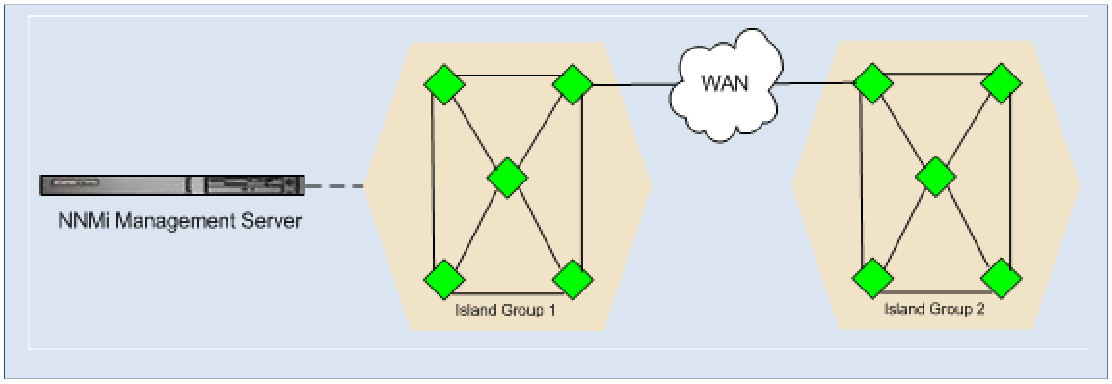
Root Cause: Island Group 2 is down according to neighbor analysis.

Incident: An `IslandGroupDown` incident is generated. NNMi chooses a representative node from Island Group 2 as the source node for the incident.

Status: The status of Island Group 2 is set to `Unknown`. Objects in Island Group 2 have unknown status. The connecting interface from Island Group 1 is UP because the connection from the interface to the WAN is still up.

Conclusion: Not applicable for Island Groups

Island Group Is Up



NOTE: The diagram is conceptual. It does not represent an actual NNMi topology map or workspace view.

Scenario: This scenario continues the *Island Group is Down* scenario on page 27. Your service provider's WAN problems are fixed, and Island Group 2 can be reached.

Root Cause: The WAN connection to Island Group 2 is back up.

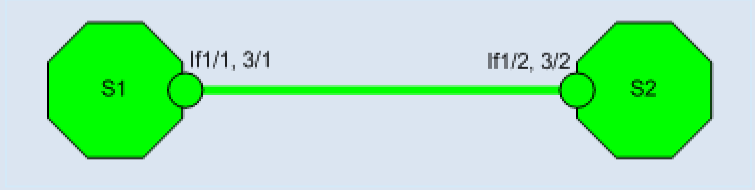
Incident: None generated; the `IslandGroupDown` incident is closed.

Status: The status for Island Group 2 is set to `Normal`. Objects in Island Group 2 return to normal status.

Conclusion: Not applicable for Island Groups

Link Aggregated Ports (NNMi Advanced)

Aggregator Is Up



Legend

S1	Switch 1
S2	Switch 2
If1/1, If1/2, If1/3	Aggregated ports on S1
If2/1, If2/2, If2/3	Aggregated ports on S2
If1/1 and If1/2	Master interfaces

One link aggregated connection exists made up of the following connections:

- If1/1, If2/1
- If2/1, If2/2
- If3/1, If3/3

Scenario: All ports within the port aggregator are operationally and administratively up.

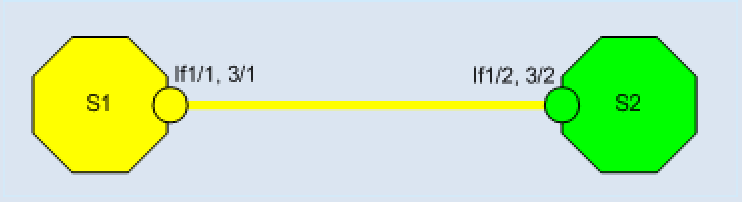
Root Cause: All operational and administrative states are up.

Incident: No incident is generated.

Status: The status of the aggregator is set to normal.

Conclusion: `AggregatorUp`

Aggregator Is Degraded



Legend

S1	Switch 1
S2	Switch 2
If1/1, If1/2, If1/3	Aggregated ports on S1
If2/1, If2/2, If2/3	Aggregated ports on S2
If1/1 and If1/2	Master interfaces

One link aggregated connection exists made up of the following connections:

- If1/1, If2/1
- If2/1, If2/2
- If3/1, If3/3

Scenario: Some (but not all) ports within the port aggregator are operationally down.

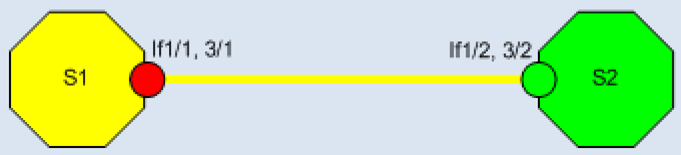
Root Cause: Operational states on some ports are down.

Incident: An `AggregatorDegraded` incident is generated.

Status: The status of the aggregator is set to minor.

Conclusion: `AggregatorDegraded`

Aggregator Is Down



Legend

S1	Switch 1
S2	Switch 2
If1/1, If1/2, If1/3	Aggregated ports on S1
If2/1, If2/2, If2/3	Aggregated ports on S2
If1/1 and If1/2	Master interfaces

One link aggregated connection exists made up of the following connections:

- If1/1, If2/1
- If2/1, If2/2
- If3/1, If3/3

Scenario: All ports within the port aggregator are operationally down.

Root Cause: Operational states on all ports are down.

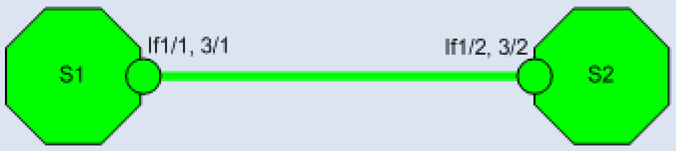
Incident: An `AggregatorDown` incident is generated.

Status: The status of the aggregator is set to critical.

Conclusion: `AggregatorDown`

Link Aggregated Connections (NNMi Advanced)

Link Aggregated Connection Is Up



Legend

S1	Switch 1
S2	Switch 2
If1/1, If1/2, If1/3	Aggregated ports on S1
If2/1, If2/2, If2/3	Aggregated ports on S2
If1/1 and If1/2	Master interfaces

One link aggregated connection exists made up of the following connections:

- If1/1, If2/1
- If2/1, If2/2
- If3/1, If3/3

Scenario: All port aggregator members of the connection are up.

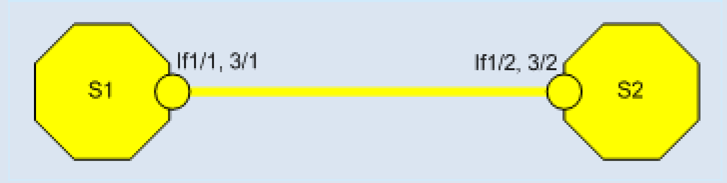
Root Cause: Aggregator up on all members of the connection.

Incident: No incident is generated.

Status: The status of the aggregated connection is set to normal.

Conclusion: `AggregatorLinkUp`

Link Aggregated Connection Is Degraded



Legend

S1	Switch 1
S2	Switch 2
If1/1, If1/2, If1/3	Aggregated ports on S1
If2/1, If2/2, If2/3	Aggregated ports on S2
If1/1 and If1/2	Master interfaces

One link aggregated connection exists made up of the following connections:

- If1/1, If2/1
- If2/1, If2/2
- If3/1, If3/3

Scenario: Some (but not all) port aggregator members of the connection are down.

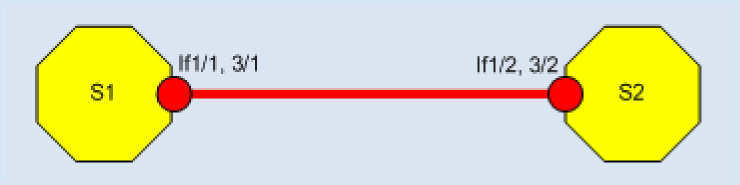
Root Cause: Aggregator down on some members of the connection.

Incident: An `AggregatorLinkDegraded` incident is generated.

Status: The status of the aggregated connection is set to minor.

Conclusion: `AggregatorLinkDegraded`

Link Aggregated Connection Is Down



Legend

S1	Switch 1
S2	Switch 2
If1/1, If1/2, If1/3	Aggregated ports on S1
If2/1, If2/2, If2/3	Aggregated ports on S2
If1/1 and If1/2	Master interfaces

One link aggregated connection exists made up of the following connections:

- If1/1, If2/1
- If2/1, If2/2
- If3/1, If3/3

Scenario: All port aggregator members of the connection are down.

Root Cause: Aggregator down on all members of the connection.

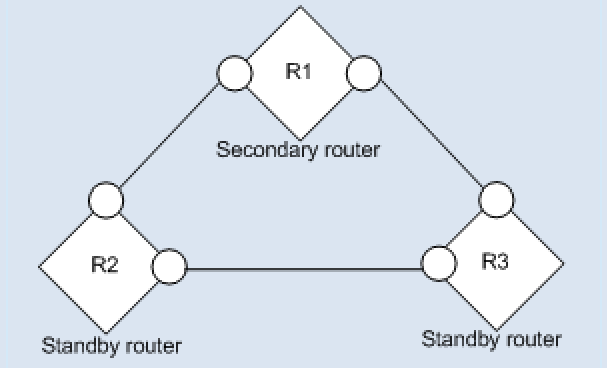
Incident: An `AggregatorLinkDown` incident is generated.

Status: The status of the aggregated connection is set to `CRITICAL`.

Conclusion: `AggregatorLinkDown`

Redundant Router Groups: HSRP and VRRP (NNMi Advanced)

Redundant Router Group Has No Primary



Legend

R1	Router 1, acting as secondary
R2	Router 2, acting as standby
R3	Router 3, acting as standby

Scenario: A redundant router group does not have a primary member. A properly functioning HSRP or VRRP router group should have one operational primary router and one operational secondary router.

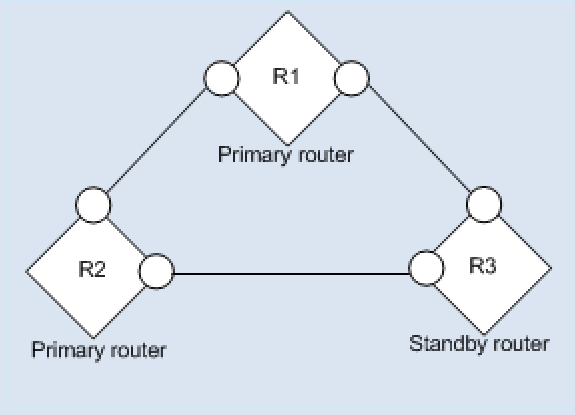
Root Cause: This scenario could be the result of an interface on the primary router failing, when the secondary was not active or mis-configuration of the redundant router group.

Incident: An `RrgNoPrimary` incident is generated. `RrgNoPrimary` is impacted. If there is an identified root cause such as `InterfaceDown`, an impact correlation is generated between `RrgNoPrimary` and `InterfaceDown`.

Status: The status of the redundant router group is set to `CRITICAL`.

Conclusion: `RrgNoPrimary`

Redundant Router Group Has Multiple Primaries



Legend

R1	Router 1, acting primary
R2	Router 2, acting as primary
R3	Router 3, acting as standby

Scenario: A redundant router group has multiple routers reporting as the primary router. A properly functioning HSRP or VRRP router group should have only one operational primary router.

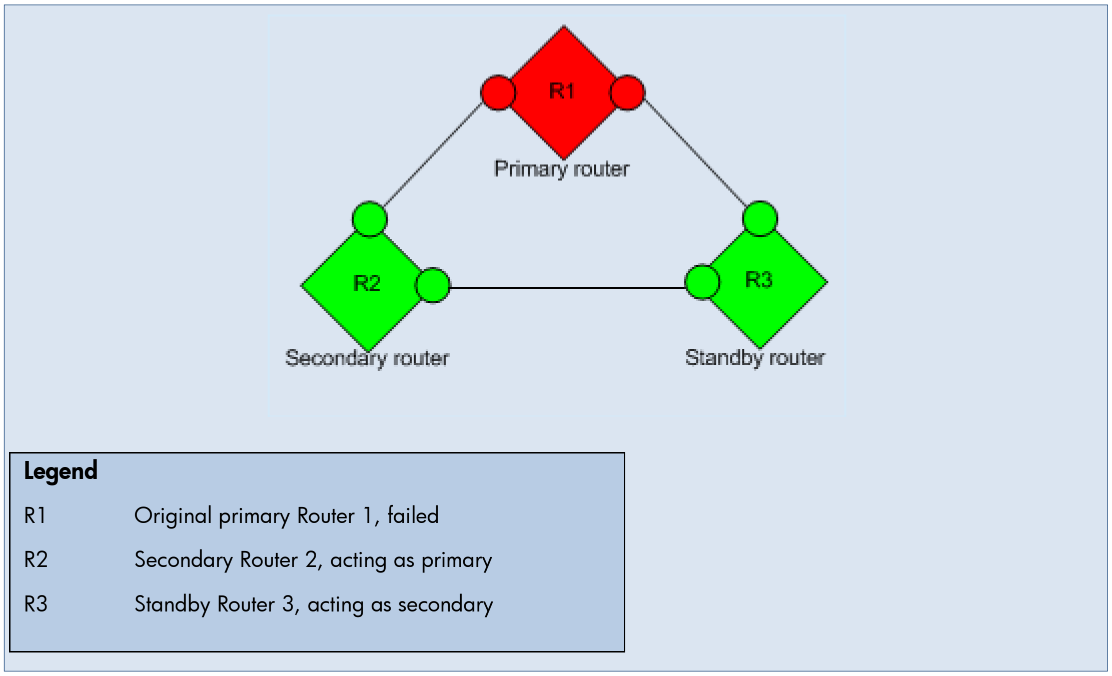
Root Cause: This scenario could be due to a faulty configuration of the redundant router group.

Incident: An `RrgMultiplePrimary` incident is generated; `RrgMultiplePrimary` is impacted.

Status: The status of the redundant router group is set to `CRITICAL`.

Conclusion: `RrgMultiplePrimary`

Redundant Router Group Has Failed Over



Legend

R1	Original primary Router 1, failed
R2	Secondary Router 2, acting as primary
R3	Standby Router 3, acting as secondary

Scenario: A redundant router group has had a failure on the primary router and the secondary router has taken over as primary. Usually the standby becomes the secondary, which is not a problem; the group is functioning as intended. The incident generated for this scenario is for informational purposes to report that the group has had a failover.

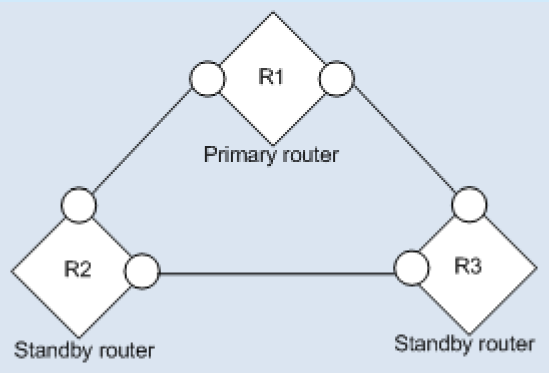
Root Cause: This scenario is most likely due to a failure on the primary router.

Incident: An `RrgFailover` incident is generated. The nature of `RrgFailover` is impacted, and if there is an identified root cause such as `InterfaceDown`, the correlation between the `RrgFailover` and `InterfaceDown` incidents is impacted.

Status: No status is generated for this case.

Conclusion: `RrgFailover`

Redundant Router Group Has No Secondary



Legend

R1	Primary Router 1
R2	Secondary Router 2, failed
R3	Standby Router 3, did not become secondary

Scenario: A redundant router group has had a failure on the secondary router. Either there is no standby, or the standby did not take over as the secondary.

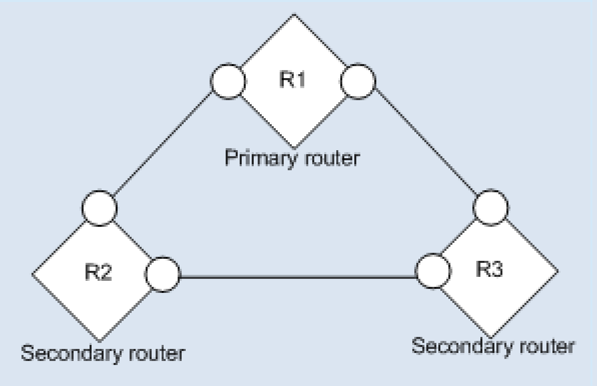
Root Cause: This scenario could be due to an interface failure on the router or some mis-configuration of the router group.

Incident: An `RrgNoSecondary` incident is generated. The nature of `RrgNoSecondary` is impacted, and if there is an identified root cause such as `InterfaceDown`, the correlation between the `RrgNoSecondary` and `InterfaceDown` interfaces is impacted.

Status: The status of the redundant router group is set to minor.

Conclusion: `RrgNoSecondary`

Redundant Router Group Has Multiple Secondaries



Legend	
R1	Primary Router 1
R2	Secondary Router 2
R3	Standby Router 3, acting as secondary

Scenario: A redundant router group has multiple routers reporting as the secondary router. A properly functioning HSRP or VRRP router group should have only one operational secondary router.

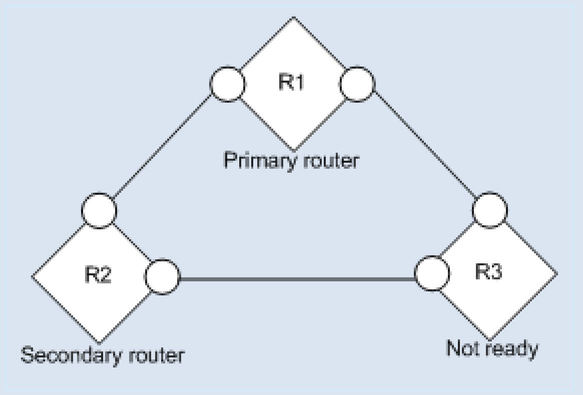
Root Cause: This scenario could be due to mis-configuration of the redundant router group.

Incident: An `RrgMultipleSecondary` incident is generated; the nature of `RrgMultipleSecondary` is impacted.

Status: The status of the redundant router group is set to minor.

Conclusion: `RrgMultipleSecondary`

Redundant Router Group Has Degraded



Legend

R1	Primary Router 1
R2	Secondary Router 2
R3	Standby Router 3

Scenario: The redundant router group has had some change. The group is functioning, and there are one primary router and one secondary router, but there is some non-normal condition that could be an issue. For example, there might be several routers not in a ready state.

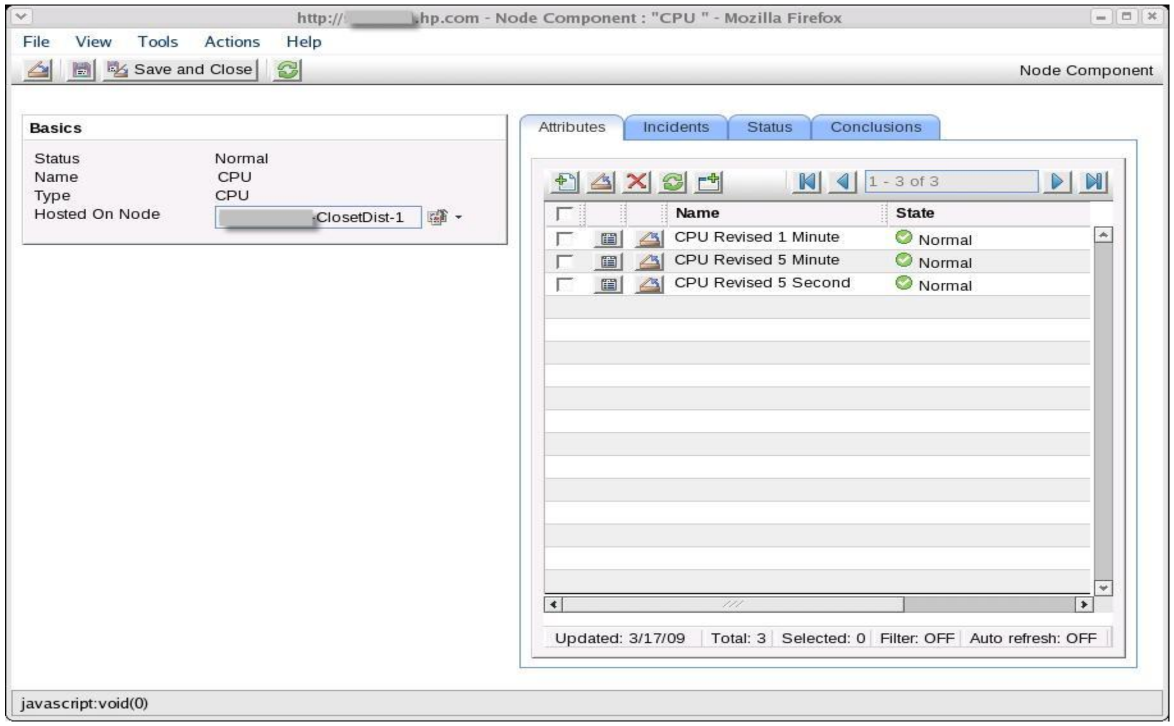
Root Cause: This scenario could be due to some mis-configuration of the router group.

Incident: An `RrgDegraded` incident is generated; the nature of `RrgDegraded` is impacted.

Status: The status of the redundant router group is set to warning.

Conclusion: `RrgDegraded`

Component Health Scenarios



http://[redacted]hp.com - Node Component : "CPU " - Mozilla Firefox

File View Tools Actions Help
Save and Close
Node Component
Attributes Incidents Status Conclusions
1 - 3 of 3

Basics

Status	Normal
Name	CPU
Type	CPU
Hosted On Node	[redacted]ClosetDist-1

Name	State
CPU Revised 1 Minute	Normal
CPU Revised 5 Minute	Normal
CPU Revised 5 Second	Normal

Updated: 3/17/09 | Total: 3 | Selected: 0 | Filter: OFF | Auto refresh: OFF

javascript:void(0)

Fan Failure or Malfunctioning

Scenario: A fan sensor detects a failed fan in a chassis.

Incident: A `FanOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the fan sensor node component is critical. A minor status is propagated to the node.

Conclusion: `FanOutOfRangeOrMalfunctioning`

Power Supply Failure or Malfunctioning

Scenario: A power supply sensor detects a failed power supply in a chassis.

Incident: A `PowerSupplyOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the power supply node component is critical. The status of minor is propagated to the node.

Conclusion: `PowerSupplyOutOfRangeOrMalfunctioning`

Temperature Exceeded or Malfunctioning

Scenario: A temperature sensor detects a high temperature in a chassis.

Incident: A `TemperatureOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the temperature sensor node component is critical.

Conclusion: `TemperatureOutOfRangeOrMalfunctioning`

Voltage Out of Range or Malfunctioning

Scenario: A voltage sensor detects a voltage problem in a chassis.

Incident: A `VoltageOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the voltage sensor node component is critical.

Conclusion: `VoltageOutOfRangeOrMalfunctioning`

Buffer Utilization Exceeded or Malfunctioning (NNM iSPI for Performance)

Scenario: The device operating system detects a problem with buffer utilization.

Incident: A `BufferOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the buffer sensor node component is critical.

Conclusion: `BufferOutOfRangeOrMalfunctioning`

CPU Utilization Exceeded or Malfunctioning (NNM iSPI for Performance)

Scenario: A CPU sensor in a chassis detects a CPU utilization problem.

Incident: A `CpuOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the CPU sensor node component is critical.

Conclusion: `CpuOutOfRangeOrMalfunctioning`

Memory Utilization Exceeded or Malfunctioning (NNM iSPI for Performance)

Scenario: A memory sensor detects a memory problem.

Incident: A `MemoryOutOfRangeOrMalfunctioning` incident is generated.

Status: The status of the memory sensor node component is critical.

Conclusion: `MemoryOutOfRangeOrMalfunctioning`

Network Configuration Changes

During the span of a day, a network operator might complete several configuration changes. The following scenarios illustrate some common network configuration changes and show how NNMi responds to these changes.

- Node updated

  Suppose that a network operator modifies a node: for example, by swapping a failed interface board with a working replacement. When NNMi notices this change, the discovery process sends a notification to the `NmsApa` service. The `NmsApa` service completes the following tasks:

  - Recalculate the status of the node.
  - Close all registered incidents for the deleted IPv4 addresses and interfaces on the node.

- Interface moves to and from connections

  Suppose that a network operator changes the way network devices are connected. When an interface joins a connection or leaves one connection to join another, the NNMi discovery process sends a notification to the `NmsApa` service. The `NmsApa` service recalculates the status of the connection.

- Device-generated traps

  `ColdStart` and **`WarmStart`** traps — The `NmsApa` service subscribes to notifications from the Events system for `ColdStart` and `WarmStart` traps. These notifications trigger the `NmsApa` service to initiate a rediscovery of device information from the node that generated the trap.

  `LinkUp` and **`LinkDown`** traps — The `NmsApa` service subscribes to notifications from the Events system for `LinkUp` and `LinkDown` traps, as well as for some vendor-specific link traps. These notifications trigger the `NmsApa` service to initiate a rediscovery of device information from the node that generated the trap.

  NOTE: For a complete list of the trap incident configurations that NNMi provides, see the NNMi help or select the **SNMP Trap Configuration** tab from the **Incident Configuration** view.

NNMi Management Configuration Changes

During the span of a day, an NNMi administrator might complete several NNMi configuration changes. The following scenarios illustrate some common NNMi management configuration changes and show how NNMi responds to these changes.

- NNMi administrator does not manage an IPv4 address or puts it out-of-service

  The `NmsApa` service receives a notification from StatePoller after the `pingState` is set to Not Polled. The `NmsApa` service sets the status of the IPv4 address to No Status.

- NNMi administrator manages an IPv4 address or puts it back in service

  The `NmsApa` service receives a notification from StatePoller after the `pingState` is set to the measured value. The `NmsApa` service calculates the status of the IPv4 address based upon the measured value.

- NNMi administrator does not manage an interface or puts it out-of-service

  The `NmsApa` service receives a notification from StatePoller after the `operState` is set to Not Polled. The `NmsApa` service sets the status of the interface to No Status.

- NNMi administrator manages an interface or puts it back in service

  The `NmsApa` service receives a notification from StatePoller after the `operState` is set to the measured value. The `NmsApa` service calculates the status of the interface based upon the measured value.

- NNMi administrator does not manage a node or puts it out-of-service

  The `NmsApa` service receives a notification from StatePoller after the `agentState` is set to Not Polled. `operState` is set to Not Polled for all interfaces, and `pingState` is set to Not Polled for all IPv4 addresses. The `NmsApa` service sets the status of the node to No Status.

- NNMi administrator manages a node or puts it back in service

  The `NmsApa` service receives a notification from StatePoller after the `agentState` is set to the measured value. `operState` is set to the measured value for all interfaces, and `pingState` is set to the measured value for all IPv4 addresses. The `NmsApa` service calculates the status of the node.