Technical white paper

HP Network Node Manager i Software 10.10

Causal Analysis

Table of contents

Click here to verify the latest version of this document

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. Distinguishing the real problem from the anecdotal alarms has become a bottleneck for the network operator. Traditional event correlation systems are 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 to incident generation.

Causal Engine Technology

Causal Engine technology provides the following features:

- Uses the NmsApa (NMS Active Problem Analyzer) jboss service to analyze your network
- Uses a 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 handles ambiguity and partial symptoms
- Is Dynamic
    - Actively solicits symptoms during analysis
    - Reacts dynamically to topology changes
- Is 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

Approach to Incident Generation

Causal Engine technology uses the following sequential approach to incident generation:

1. Formally define the root-cause problems and symptoms.
2. Perform analysis by relating symptoms to root-cause problems using the behavioral and object models. 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 track details related to its analysis, including the following:

- Generated incidents
- Correlated incidents
- Suppressed incidents
- Cancelled incidents
- Status on relevant objects

The goal of the NmsApa service is to present a single incident that the operator or network engineer can investigate. 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 incident configuration.

Incident Suppression Example

The Address Not Responding incident is suppressed by the Interface Down incident, according to the following scenario:

- When an IP 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 IP address goes down, the NmsApa service concludes that the interface down condition caused the IP address to stop responding.
- Only the Interface Down incident is generated. (The Address Not Responding incident is suppressed.)
- To ensure that the Interface Down incident is detected within the duration, the NmsApa service issues a request to check the Status for that interface.
- If the interface does not go down during the episode, the NmsApa service generates an Address Not Responding incident. If the interface goes down after the episode, NNMi generates an Interface Down incident. In this case, the network engineer treats the two problems separately.

Incident Correlation Example

The Node Down incident correlates the Interface Down incident from one-hop neighbor interfaces, according to the following scenario:

- When an interface goes down, a Node Down episode begins for the neighboring node, which exists for the duration of 300 seconds.
- Within that duration, if the node goes down, NNMi correlates the Interface Down incident beneath the Node Down incident.
- The Interface Down incidents from all one-hop neighbors are correlated beneath the Node Down incident. You can review the Interface Down incidents as supporting evidence for the Node Down incident.

Object 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
- Major
- Minor
- Warning
- Normal
- No Status

What Does NNMi Analyze?

NNMi analyzes a variety of network objects, including Nodes, Interfaces, and Addresses. NNMi monitors these devices using either the SNMP protocol or ping to retrieve information about the network object.

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

- Card
- Card Redundancy Groups
- Connections
- Field-Replaceable-Unit (FRU) Card
- Interface
- IP Address

- Node
- Node Components
- Node Groups
- Aggregator Layer 2 Connections
- Aggregator Interfaces
- Redundant Router Groups
- SNMP Agent

Cards

A card is a physical component on a device which generally has physical ports that contain one or more interfaces used to connect to other devices. A card can also contain sub-cards. The card containing another card is known in NNMi as the Parent Card. The sub-card is known as a Daughter Card. NNMi supports Daughter cards one level deep.

NNMi reports the Status of a card as follows:

- Unknown – Indicates either of the following:
  - The SNMP Agent associated with the card does not respond to SNMP queries.
  - The NmsApa service cannot determine the health because the cardOperStatus and cardAdminStatus values cannot be measured.
- Disabled – Indicates the card is administratively down (cardAdminStatus = down).
- Critical – Indicates the card is operationally down (cardOperStatus = down).
- Normal – Indicates the card is operationally up (cardOperStatus = up).
- Minor – Indicates the card is neither up nor down (cardOperStatus = unknown or other)
- No Status – Indicates the card is not polled.

Card Redundancy Groups

A Card Redundancy Group is a set of card modules that are configured to provide card redundancy on the device. These cards are management modules on Cisco and HP Procurve platforms. The number of cards supported in a group on both platforms is two. The Card Redundancy Group has one card acting as the primary member, the other acting as the secondary. If the primary card fails, the secondary card takes over as the primary card.

NNMi reports the Status of Card Redundancy Groups as follows:

- Unknown – Indicates all member cards in the group are in Unknown Status.
- Critical – Indicates the group has no acting primary card or has both cards acting as primary.
- Warning – Indicates the group has no acting secondary card.
- Normal – Indicates the group is functioning correctly.
- No Status – Indicates the group has not yet been fully discovered or is not being polled.

Connections

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 Cisco Discovery Protocol (CDP) and Extreme Discovery Protocol (EDP). NNMi reports the Status of a connection as follows:

- Unknown – Indicates all endpoints of the connection have Unknown Status.
- Disabled – Indicates one endpoint of the connection is disabled.
- Critical– Indicates all endpoints are operationally down.
- Minor – Indicates one endpoint is down.
- Warning – Indicates endpoints have unknown and non-critical Status.
- Normal – Indicates all endpoints are operationally up.
- No Status – Indicates all endpoint are not polled.

Field Replaceable Units (FRU Card)

A Field-Replaceable-Unit (FRU) card is a card that can be replaced on a device that is operationally active (not powered down). When an FRU card is removed from or added to the device, NNMi reports the occurrence with an incident. If an FRU card is not recognized by the device, NNMi reports the unrecognized card with an incident.

NNMi reports the Status of an FRU card as follows:

- Unknown – Indicates either of the following:
    - The SNMP Agent associated with the card does not respond to SNMP queries.
    - The NmsApa service cannot determine the health because cardOperStatus and cardAdminStatus values cannot be measured.
- Disabled – Indicates the card is administratively down (cardAdminStatus = down).
- Critical – Indicates the card is operationally down (cardOperStatus = down).
- Normal – Indicates the card is operationally up (cardOperStatus = up).
- Minor – Indicates the card is neither up or down (cardOperStatus = unknown or other)
- No Status – Indicates the card is not polled.

Interfaces

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 – Indicates either of the following:
  - The SNMP Agent associated with the interface does not respond to SNMP queries.
  - The NmsApa service cannot determine the health because ifAdminStatus and ifOperStatus values 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.

IP Addresses

An IP address is a routable address that responds to ICMP. IP addresses are typically associated with nodes. NNMi reports the Status of an IP address as follows:

- Disabled - Indicates the interface associated with this IP address is administratively down or disabled.
- Critical - Indicates the IP address does not respond to ICMP queries (ping the device).
- Normal – Indicates the IP address responds to ICMP queries.
- No Status – Indicates the IP address is not polled.

Nodes

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 node component Status as follows:

- Unknown – Indicates that NNMi is unable to manage the node because of the following:
  - The SNMP Agent associated with the node does not respond to SNMP queries
  - Polled IP addresses do not respond to ICMP queries.
- Critical – Indicates 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.
  - At least one Custom Polled Instance associated with the physical node has a Status of Critical and Custom Polled Instances are configured to affect Node Status.
- Major – Indicates any of the following:

  - A fan (Node Component) failure is detected.
  - A power supply (Node Component) failiure is detected.
  - A backplane (Node Component) failiure is detected.
  - A memory (Node Component) failiure is detected.
  - At least one Custom Polled Instance associated with the physical node has a Status of Major and Custom Polled Instances are configured to affect Node Status.
- Minor – Node Status can be Minor if a managed object contained in the node has a problem, including any of the following:
  - The SNMP Agent associated with the node does not respond to SNMP queries.
  - The management address on the node is not responding to ICMP.
  - One or more interfaces in the node are down.
  - One or more IP addresses on the node do not respond to ICMP.
  - One or more cards on the node is reporting an "Unknown" state (cardOperStatus=Unknown) or an "Other" state (cardOperStatus=Other). This implies that the card is not healthy.
  - At least one interface on the node has a threshold outside the range specified for the device.
  - At least one Custom Polled Instance associated with the physical node has a Status of Minor and Custom Polled Instances are configured to affect Node Status.
  - One or more cards in the node are down.
- Normal – Indicates the SNMP Agent, polled interfaces, and polled IP addresses of the node are up.
- No Status – Indicates the SNMP Agent, all interfaces, and all IP addresses of the node are not polled.

Node Components

Large (or more sophisticated) network devices often require special environments and components to function properly. Examples are power supplies, fans, voltage regulators, and internal computers. These Node 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 Node Components as follows:

- `Critical` – Indicates the component is not functioning properly.
- `Normal` – Indicates the component is operating properly.
- `No Status` – Indicates the component is not polled.

Node Groups

A Node Group is a logical collection of nodes created by an NNMi administrator.

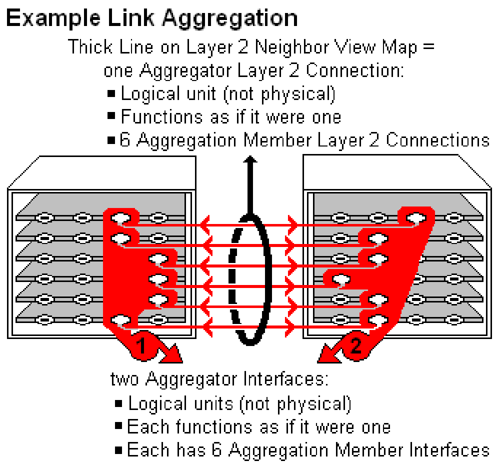
An NNMi administrator can also configure Node Group Status calculations. The out-of-the-box configuration propagates the most severe Status as follows:

- `Critical` – Indicates at least one node in the group has Critical Status.
- `Major` – Indicates no nodes in the group have Critical Status, and at least one node in the group has Major Status.

- `Minor` – Indicates no nodes in the group have Critical or Major Status, and at least one node in the group has Minor Status.
- `Warning` – Indicates no nodes in the group have Critical, Major, or Minor Status, and at least one node in the group has Warning Status.
- `Normal` – Indicates no nodes in the group have Critical, Major, Minor, or Warning Status, and at least one node in the group has Normal Status.
- `Unknown` – Indicates no nodes in the group have Critical, Major, Minor, Warning, or Normal Status, and at least one node in the group has Unknown Status.
- `No Status` – Indicates all nodes in the group have No Status.

Aggregator Layer 2 Connection

An Aggregator Layer 2 Connection is a connection with endpoints that are Aggregator Interfaces. These are usually high-bandwidth connections that link switches. As shown in the following illustration Aggregator Layer 2 Connections have Aggregator Interfaces and Aggregation Members.



Example Link Aggregation

Thick Line on Layer 2 Neighbor View Map =
one Aggregator Layer 2 Connection:
- Logical unit (not physical)
- Functions as if it were one
- 6 Aggregation Member Layer 2 Connections

two Aggregator Interfaces:
- Logical units (not physical)
- Each functions as if it were one
- Each has 6 Aggregation Member Interfaces

An administrator can monitor the overall health of the Aggregator Layer 2 Connection to know when the connection is degraded in any way.

NNMi reports the Status of an Aggregator Layer 2 Connection as follows:

- `Unknown` – Indicates any Aggregation Member Layer 2 Connection of the Aggregator Layer 2 Connection is unknown.
- `Critical` – Indicates the Aggregator Interfaces, the Aggregation Member Layer 2 Connections, or both are operationally down.
- `Minor` – Indicates some Aggregation Member Layer 2 Connections (but not all) of the Aggregator Layer 2 Connection are operationally down.
- `Normal` – Indicates all Aggregation Member Layer 2 Connections of the Aggregator Layer 2 Connection are operationally up.
- `No Status` – Indicates all Aggregation Member Layer 2 Connections of the Aggregator Layer 2 Connection are not polled.

Aggregator Interfaces

An Aggregator Interfaces is a set of interfaces on a switch that are linked together, usually for the purpose of creating a trunk (high bandwidth) connection to another device. Aggregator Interfaces have designated Aggregation Member Interfaces. An administrator can monitor the overall health of the Aggregator Interface to know when the Aggregator Interface is degraded.

NNMi reports the Status of an Aggregator Interface as follows:

- `Unknown` – Indicates all Aggregation Members of the Aggregator Interface are unknown.
- `Critical` – Indicates the Aggregator Interface, the Aggregation Members, or both are operationally down.
- `Minor` – Indicates some Aggregation Members (but not all Aggregation Members) of the Aggregator Interface are operationally down.
- `Normal` — Indicates all Aggregation Members of the Aggregator Interface are operationally up.
- `No Status` – Indicates all Aggregation Members of the Aggregator Interface are not polled.

Router Redundancy Groups

A Router Redundancy Group is a set of routers that are configured to provide redundancy in the network. Such groups use the following two types of protocols:

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

Router Redundancy 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 Router Redundancy Groups as follows:

- `Critical` – Indicates the group has no acting primary router.
- `Major` – Indicates the group primary router is not properly configured (for example, multiple primary routers exist).
- `Minor` – Indicates the group secondary router is not properly configured (for example, no acting secondary router exists).
- `Warning` – Indicates the group is functioning, but is in some way degraded.
- `Normal` – Indicates the group is functioning properly.
- `No Status` – Indicates the group is not yet fully discovered or populated.

SNMP Agents

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. An SNMP Agent can be associated with one or more nodes.

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

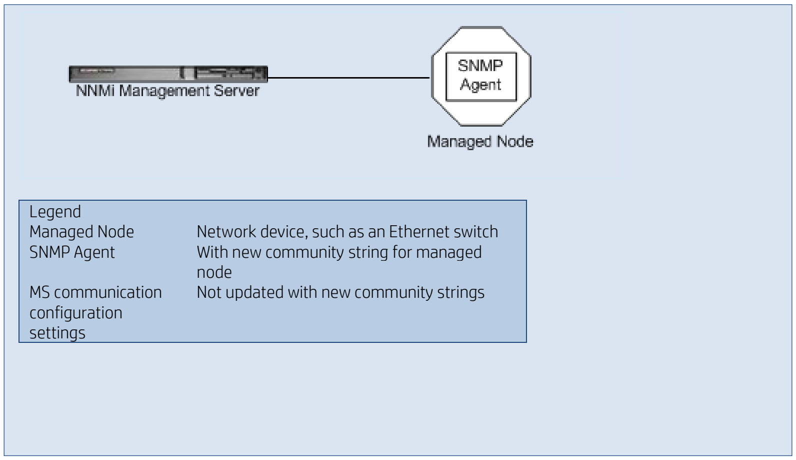
- `Critical` – Indicates the SNMP Agent does not respond to SNMP queries.
- `Minor – Indicates` the address associated with this SNMP Agent is not responding to ping.
- `Normal` – Indicates the SNMP Agent responds to SNMP queries.

- `No Status` – Indicates the SNMP Agent is not polled.
- `Warning – Indicates either a` high or abnormal Internet Control Message Protocol (ICMP) response time from the management station to the selected node.

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 example, 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 (IP addresses can be pinged).

NOTE: This scenario requires that at least one address is polled.

Root Cause: The SNMP Agent is not responding.

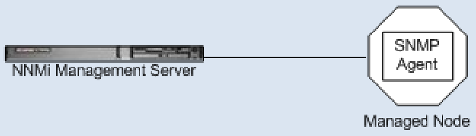
Incident: An `SNMPAgentNotResponding` incident is generated.

Status: The SNMP Agent is in Critical Status.

Conclusion: `SNMPAgentNotResponding`

Effect: The node Status is Minor. The Conclusion on the node is `UnresponsiveAgentInNode`. All polled interfaces and cards, including Daughter cards have Unknown Status because they cannot be managed by NNMi. The Conclusion on each interface is `InterfaceUnmanageable`. The Conclusion on each card is `CardUnmanagable.`

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 previous "SNMP Agent Not Responding to SNMP Queries" scenario. 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.

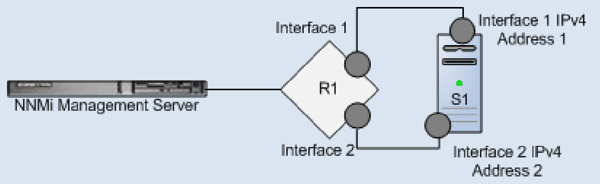
Status: SNMP Agent is in Normal Status.

Conclusion: SNMPAgentResponding

Effect: NNMi closes the `SnmpAgentNotResponding` incident. The Node Status is Normal.

The Conclusion on the node is `ResponsiveAgentInNode`. `InterfaceUnmanagable` is cleared from all polled interfaces and the interfaces return to their previous Status. `CardUnmanagable` is cleared from all polled cards including Daughter cards with their Status restored to the previous Status.

IP Address 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 IP address 1
S1 Interface 2	Associated with IP address 2

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

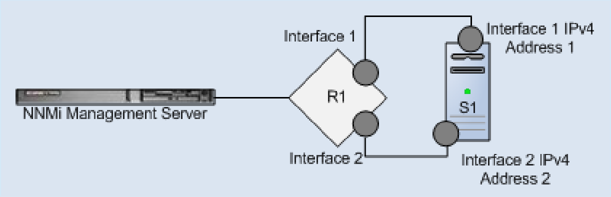
Incident: An `AddressNotResponding` incident is generated.

Status: IP address is in Critical Status.

Conclusion: AddressNotResponding

Effect: The node Status is Minor. The Conclusion on the node is `SomeUnresponsiveAddressesInNode`.

IP Address 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 previous "IP Address Not Responding to ICMP" scenario. The IP address is now responding, the associated interface is operational, and the node can be reached. (For example, you can ping some IP addresses and the SNMP agent is up, or both.)

Root Cause: IP address is responding.

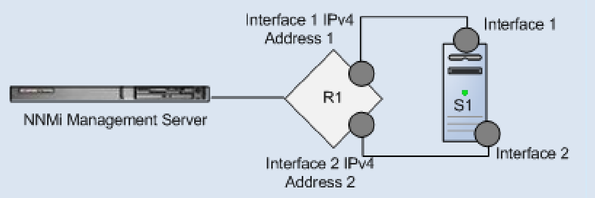
Incident: None generated. The `AddressNotResponding` incident is closed.

Status: The IP address is in Normal Status.

Conclusion: AddressResponding

Effect: The node Status is Normal. The 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: Interface 1 is operationally down (`ifOperStatus` = down) and administratively up (`ifAdminStatus` = up). Router 1 sends a `linkDown` trap. Router 1 can be reached because some IP addresses, such as IP Address 2, respond to ping. The SNMP agent is up. IP Address 1 is associated with Interface 1. IP Address 1 has stopped responding to ICMP.

Root Cause: The interface is down.

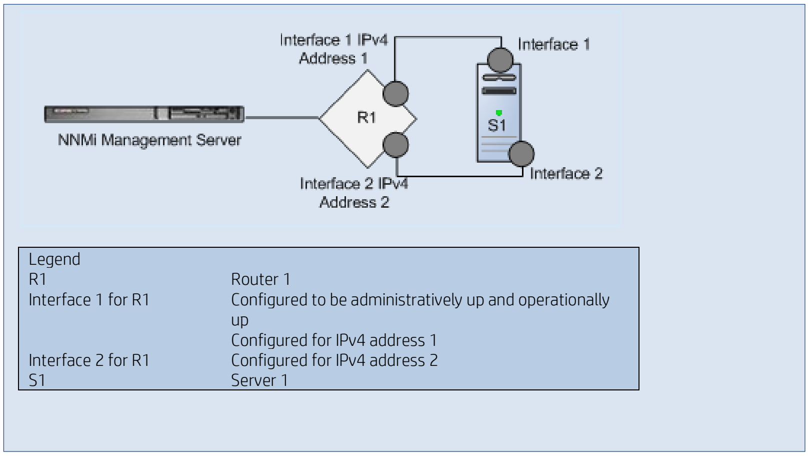
Incident: An `InterfaceDown` incident is generated. The `LinkDown` incident is correlated beneath the `InterfaceDown` incident.

Status: The interface is in Critical Status.

Conclusion: InterfaceDown

Effect: The node status is Minor. The 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 previous “Interface is Operationally Down” scenario. Interface 1 on Router 1 is now operationally up (`ifOperStatus` = up). The node can be reached. All of its IP addresses respond to ping. The SNMP agent is up.

Root Cause: The interface is up.

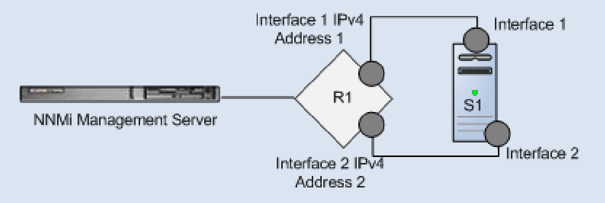
Incident: None generated. The `InterfaceDown` incident is closed.

Status: The interface is in Normal Status.

Conclusion: InterfaceUp

Effect: The Node Status is Normal. The Conclusion on the node is `InterfacesUpInNode`.

Interface Is Administratively Down



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

Scenario: Interface 1 on Router 1 is administratively down (`ifAdminStatus` = down), but the node (Router 1) can be reached. For example, Interface 2 responds to ping and the SNMP agent is up. Disabling Interface 1 brings that interface operationally down. The IP address associated with this interface, IP Address 1, stops responding to ICMP.

Root Cause: Interface 1 is disabled.

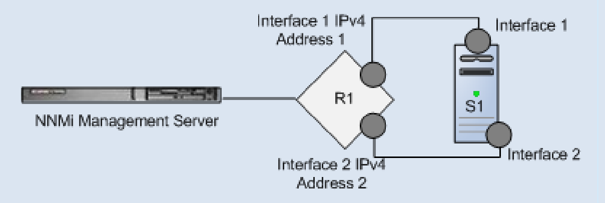
Incident: None generated by default. The NNMi administrator can enable the InterfaceDisabled Incident configuration. See the "Generate Interface Disabled Incidents" help topic in NNMi *Help for Administrators* for more information.

Status: The interface is in Disabled Status.

Conclusion: InterfaceDisabled

Effect: The IP address associated with Interface 1 on Router 1 has a Status of Disabled. The Conclusion on the IP 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 previous “Interface is Administratively Down” scenario. Interface 1 on Router 1 is now administratively up (`ifAdminStatus` = up). Some of the IP addresses of that interface respond to ping. The SNMP agent is up. Enabling Interface 1 on Router 1 brings it operationally up. The IP address associated with this interface starts responding to ICMP.

Root Cause: The interface is enabled.

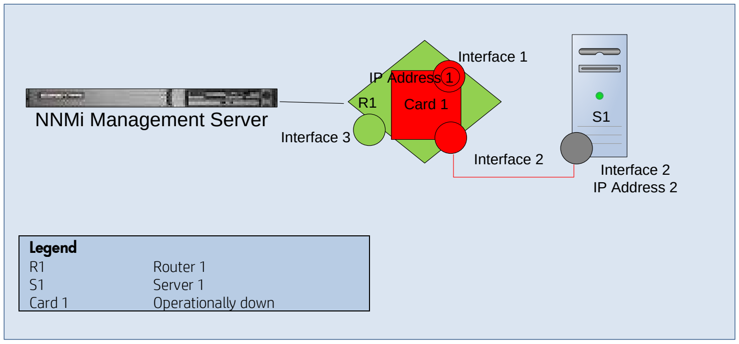
Incident: None generated.

Status: The interface is in Normal Status.

Conclusion: InterfaceEnabled

Effect: The IP address associated with Interface 1 on Router 1 has a Status of Enabled. The Conclusion on the IP address is `AddressResponding`.

Card is Operationally Down



Scenario: Card 1 is operationally down (`cardOperStatus` = down) and administratively up (`ifAdminStatus` = up). Router 1 (R1) sends a `moduleDown` trap. If Card 1 is an FRU card, the trap sent is a `CiscoModuleStatusChange` trap. Router 1 can be reached because Interface 3
re
terface 1 and Interface 2 are associated with ports on
C
d with Interface 1, has stopped responding to ICMP.
In
to Interface 2 on Server 1 (S1) is also down.

Legend	
R1	Router 1
Card 1	Operationally down

Root Cause: Card 1 is down.

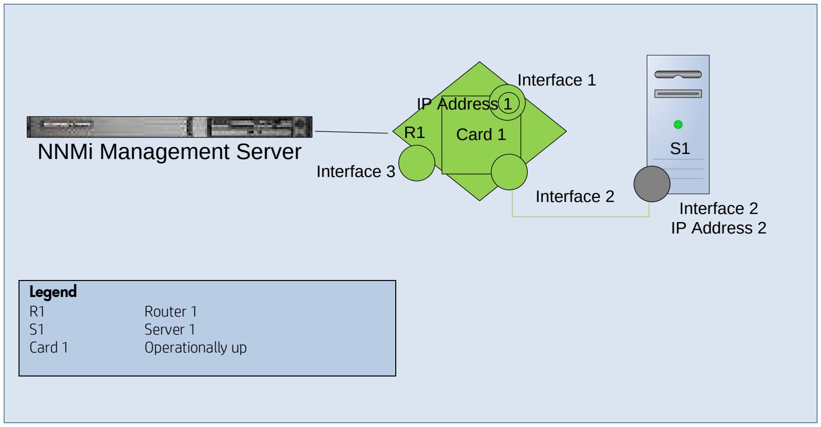
Incident: A `CardDown` incident is generated. The `moduleDown` or `CiscoModuleStatusChange` incident is correlated under the `CardDown` incident. The `InterfaceDown` incident for Interface 1 and the `ConnectionDown` incident between Interface 2 on Router 1 (R1) and Interface 2 on Server 1 (S1) are correlated under the `CardDown` incident. The `InterfaceDown` incident for Interface 2 is correlated under `ConnectionDown`.

Status: The card is in Critical Status.

Conclusion: `CardDown`

Effect: The node Status is Minor. The interfaces associated with the card and the addresses on those interfaces are in Critical Status. Conclusions on the node are `CardsDownInNode`, `InterfacesDownInNode`, and `SomeUnresponsiveAddressesInNode`. No `AddressNotResponding` incident is sent because this incident is suppressed by the `InterfaceDown` incident.

Card Is Operationally Up



Scenario: This scenario continues the previous "Card is Operationally Down" scenario. Card 1 is operationally up (`cardOperStatus` = up). The node (Router 1) can be reached and all of its IP addresses respond to ping. The card is up and all the interfaces and connection on Card 1 are back to normal. The SNMP agent is up.

Root Cause: The card is up.

Incident: No incidents are generated. The `CardDown` incident is closed. The `InterfaceDown` and `ConnectionDown` incidents correlated under the `CardDown` incident are also closed.

Status: The card is in Normal Status.

Conclusion: `CardUp`

Effect: The node Status is Normal. The Conclusion on the node is `CardsUpInNode.`

Card Is Neither Operationally Up nor Operationally Down

Scenario: A card on the router is reporting an "Unknown" state (`cardOperStatus=Unknown`) or an "Other" state (`cardOperStatus=Other`). This implies that the card is not healthy.

Root Cause: The card might be in some faulty state.

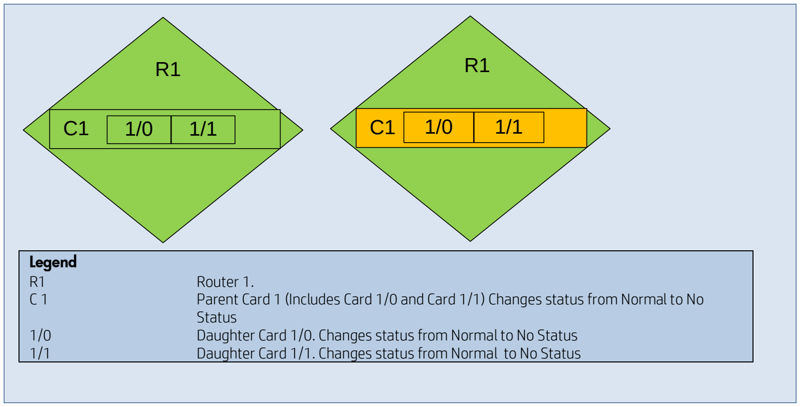
Incident: CardUnderterminedState

Status: The card has Minor Status.

Conclusion: `CardUndeterminedState`

Effect: The node Status is Minor. The Conclusion on the node is `CardsUnderdeterminedStateInNode`.

Parent Card Management Mode is Unmanaged or Out-Of-Service



Legend

R1	Router 1.
C 1	Parent Card 1 (Includes Card 1/0 and Card 1/1) Changes status from Normal to No Status
1/0	Daughter Card 1/0. Changes status from Normal to No Status
1/1	Daughter Card 1/1. Changes status from Normal to No Status

Scenario: Router 1 is managed and is polled. All cards on Router 1 are up. The Parent Card, Card 1, management mode is set to Unmanaged or Out-Of-Service.

Root Cause: The Parent Card is not polled.

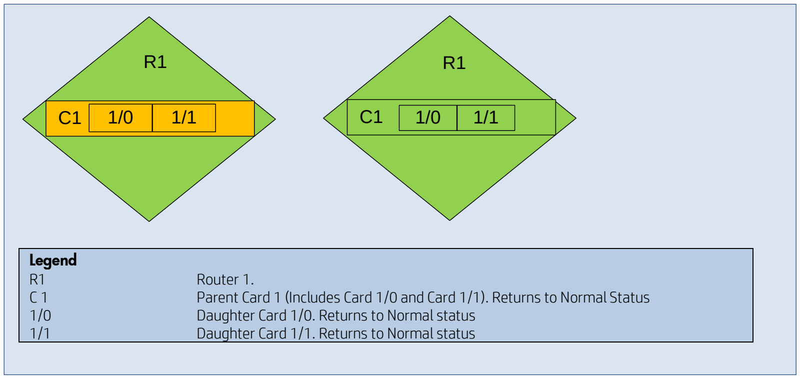
Incident: None generated.

Status: The Parent Card Status changes to No Status.

Conclusion: None

Effect: The Daughter cards, Card 1/0, Card 1/1 change to a Status of No Status. The management mode of the Daughter cards is inherited from the Parent Card so the Daughter cards become unpolled.

Parent Card Management Mode is Inherited


R1
C1
1/0
1/1
R1
C1
1/0
1/1
Legend
R1 Router 1.
C 1 Parent Card 1 (Includes Card 1/0 and Card 1/1). Returns to Normal Status
1/0 Daughter Card 1/0. Returns to Normal status
1/1 Daughter Card 1/1. Returns to Normal status

Scenario: This scenario continues the previous "Parent Card Management Mode is Unmanaged or Out-of-Service" scenario. The Parent Card management mode is now set to Inherited. This means the Parent Card's management mode is inherited from its container node (Router1), which is Managed.

Root Cause: The Parent Card is polled.

Incident: None generated.

Status: The Parent Card Status is restored to its previous Status.

Conclusion: None

Effect: The Status of the Daughter cards (Card 0/1 and Card 1/1) are restored to their previous Status.

Field Replaceable Unit (FRU) Card is Added

Scenario: An FRU card is added to the device. The device sends an `FRUInserted` trap with information on the added card module.

Root Cause: The card is successfully added.

Incident: A `CardInserted` incident is generated. This is an Informational incident. The `FRUInserted` trap is correlated under the `CardInserted` incident.

Status: The new card Status changes from `No Status` (unpolled at the beginning) to a polled Status.

Conclusion: None

Effect: If the new card is not in a Normal Status, the card Status will propagate to the node. In the common case, the node Status should remain unchanged from its previous Status.

Field Replaceable Unit (FRU) Card is Removed

Scenario: An existing FRU card is removed from the device. The device sends an `FRURemoved` trap with information on the removed card module.

Root Cause: The device stops reporting the existence of the card module on rediscovery.

Incident: A `CardRemoved` incident is generated. This is an Informational incident. The `FRURemoved` trap is correlated under the `CardRemoved` incident.

Status: None

Conclusion: None

Effect: If the node Status was previously affected by a faulty FRU card, the node Status is restored to a Normal Status when that card is removed.

Field Replaceable Unit (FRU) Card is not Recognized

Scenario: An inserted FRU card is not recognized by the device. The device sends a `CiscoUnrecognizedFRU` trap with information on the unrecognized card module.

Root Cause: The device does not recognize the card module. One possible reason is that the card might have an incompatible module version.

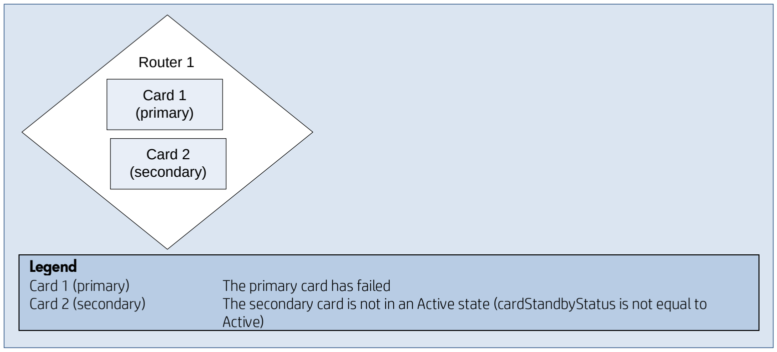
Incident: The `CiscoUnrecognizedFRU` incident is an SNMP trap. It is an Informational incident with a severity of Warning. No additional management event is provided.

Status: None

Conclusion: None

Effect: None

Card Redundancy Group has no Primary Member



Legend

Card 1 (primary)	The primary card has failed
Card 2 (secondary)	The secondary card is not in an Active state (cardStandbyStatus is not equal to Active)

Scenario: A Card Redundancy Group does not have a primary member. The `cardStandbyStatus` of the primary card (Card 1) is not Active. Router 1 sends a `CiscoRFProgressionNotif` or a `CiscoRFSwactNotif` trap on the Card Redundancy Group state changes.

A properly functioning Card Redundancy Group should have one operational primary card one operational secondary card.

Root Cause: The primary card fails and the secondary card fails to switch to primary role.

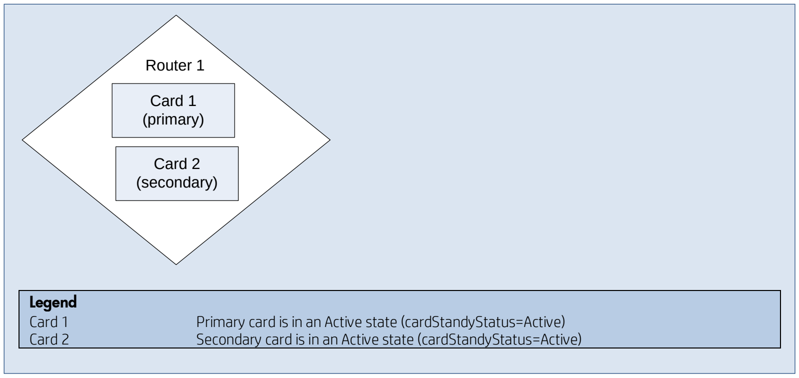
Incident: A `CrgNoPrimary` incident is generated as a Service Impact incident. If there is an identified root cause such as `CardDown`, the `CardDown` incident is correlated under the `CrgNoPrimary` incident as an Impact Correlation.

Status: The Status of the Card Redundancy Group is Critical.

Conclusion: CrgNoPrimary

Effect: The node Status is Warning. The Conclusion on the node is `CrgMalfunctionInNode`.

Card Redundancy Group has Multiple Primary Members



Legend	
Card 1	Primary card is in an Active state (cardStandyStatus=Active)
Card 2	Secondary card is in an Active state (cardStandyStatus=Active)

Scenario: A Card Redundancy Group has both members reporting as the primary card. The `cardStandbyStatus` of both cards is Active. A properly functioning Card Redundancy Group should have only one operational primary card.

Root Cause: This scenario could be due to a misconfiguration of the Card Redundancy Group.

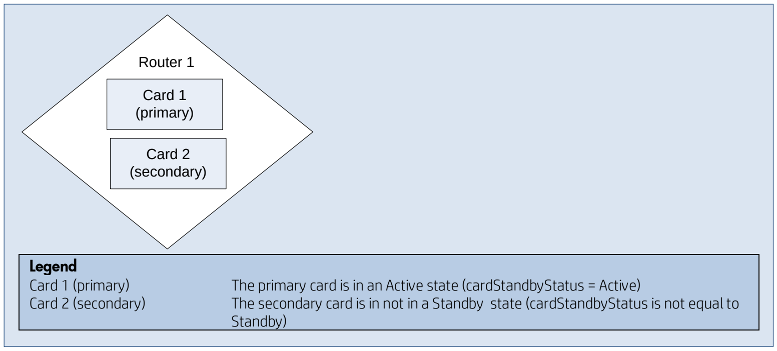
Incident: A `CrgMultiplePrimary` incident is generated.

Status: The Status of the card redundancy group is `Critical`.

Conclusion: CrgMultiplePrimary

Effect: The node Status is Warning. The Conclusion on the node is `CrgMalfunctionInNode.`

Card Redundancy Group has no Secondary Member



Legend	
Card 1 (primary)	The primary card is in an Active state (cardStandbyStatus = Active)
Card 2 (secondary)	The secondary card is in not in a Standby state (cardStandbyStatus is not equal to Standby)

Scenario: A Card Redundancy Group does not have a secondary member. Neither card has a `cardStandbyStatus` equal to Standby. A properly functioning Card Redundancy Group should have one operational primary card and one operational secondary card.

Root Cause: This scenario could result from the secondary card failing or from a misconfiguration of the Card Redundancy Group.

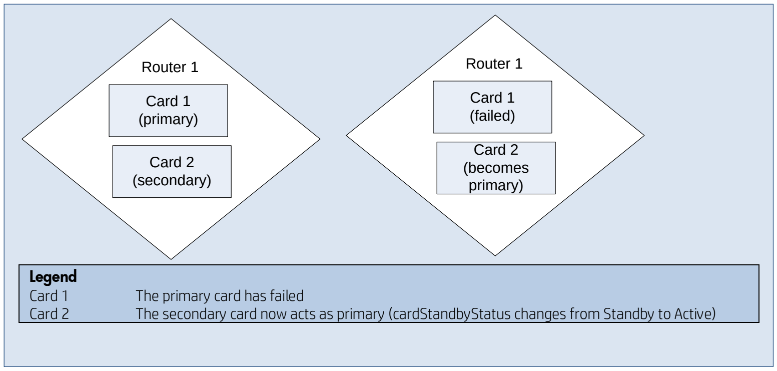
Incident: A `CrgNoSecondary` incident is generated. If there is an identified root cause such as `CardDown`, the `CardDown` incident is correlated under the `CrgNoSecondary` incident as an Impact Correlation.

Status: The Status of the Card Redundancy Group is Warning.

Conclusion: CrgNoSecondary

Effect: The node Status is Warning. The Conclusion on the node is `CrgMalfunctionInNode.`

Card Redundancy Group Fail Over


Router 1
Card 1 (primary)
Card 2 (secondary)
Router 1
Card 1 (failed)
Card 2 (becomes primary)
Legend
Card 1 The primary card has failed
Card 2 The secondary card now acts as primary (cardStandbyStatus changes from Standby to Active)

Scenario: A Card Redundancy Group has a failure on the primary card. The `cardStandbyStatus` on the secondary card changes from the value of Standby to Active. This Active card takes over as the primary member. In this case, the Card Redundancy Group is functioning as intended.

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

Incident: A `CrgFailover` incident is generated. If there is an identified root cause such as `CardDown`, the `CardDown` incident is correlated under the `CrgFailover` incident as an Impact Correlation.

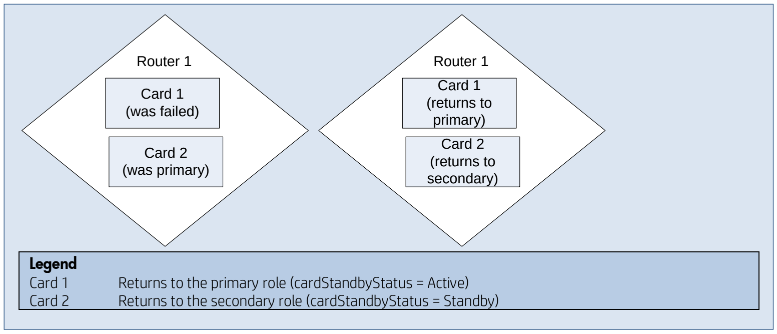
A `CrgNoSecondary` incident is also generated as neither card member is in a Standby state.

Status: The failover Status generates a Status of Normal. The Status of the Card Redundancy Group is Warning because there is no secondary.

Conclusion: CrgFailover and CrgNoSecondary

Effect: The node Status is Warning. The Conclusion on the node is `CrgMalfunctionInNode.`

Card Redundancy Group Failback


Router 1
Card 1 (was failed)
Card 2 (was primary)
Router 1
Card 1 (returns to primary)
Card 2 (returns to secondary)
Legend
Card 1 Returns to the primary role (cardStandbyStatus = Active)
Card 2 Returns to the secondary role (cardStandbyStatus = Standby)

Scenario: This scenario is a continuation of the previous scenario “Card Redundancy Group Fail Over ”. Card 1 is now functional and acts as the primary card. Card 2 resumes the secondary role.

Root Cause: The faulty primary card is now working correctly or the Card Redundancy Group misconfiguration is corrected.

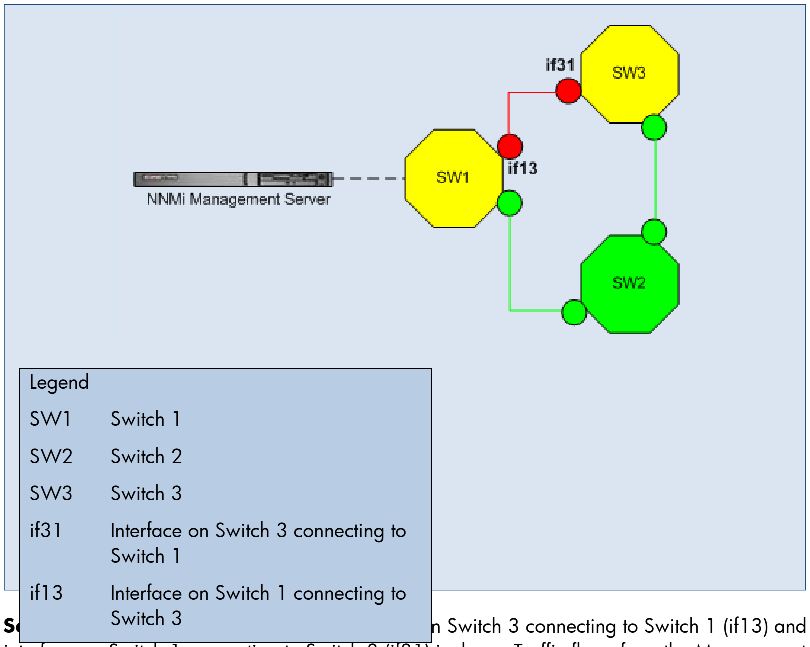
Incident: The incident `CrgFailover` is closed. The `CrgNoSecondary` incident is also closed.

Status: Normal

Conclusion: `CrgNormal`

Effects: The Conclusion on the node is `CrgNormalInNode`.

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

Sc n 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.

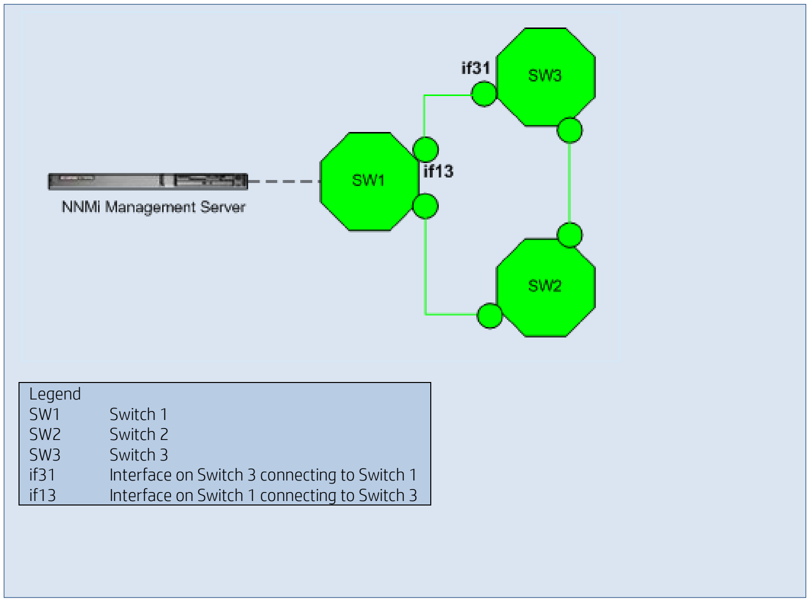
Root Cause: The connection between if13 and if31 is down.

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

Status: The 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 previous “Connection is Operationally Down” scenario. The connection between if13 and if31 is now up.

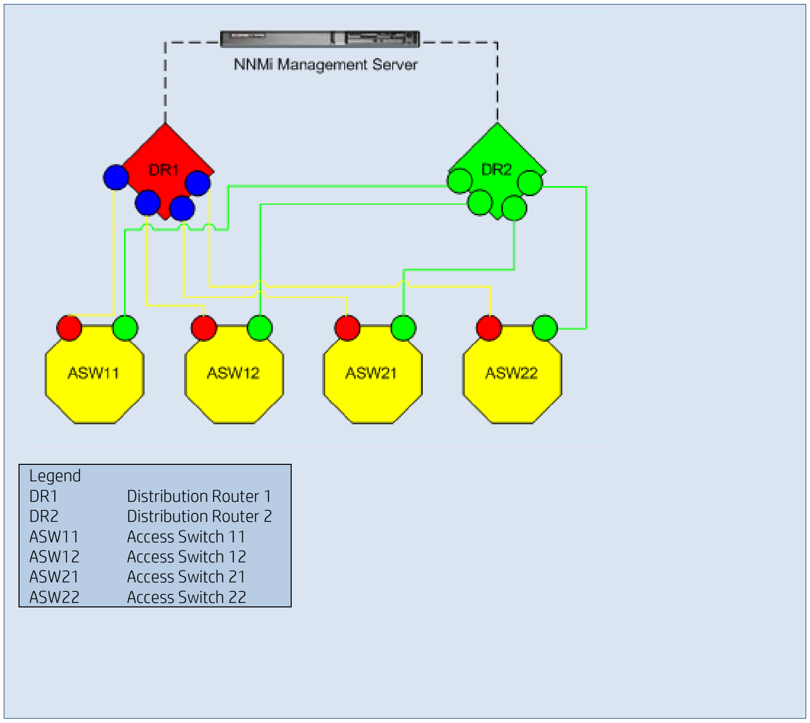
Root Cause: The connection between if13 and if31 is up.

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.

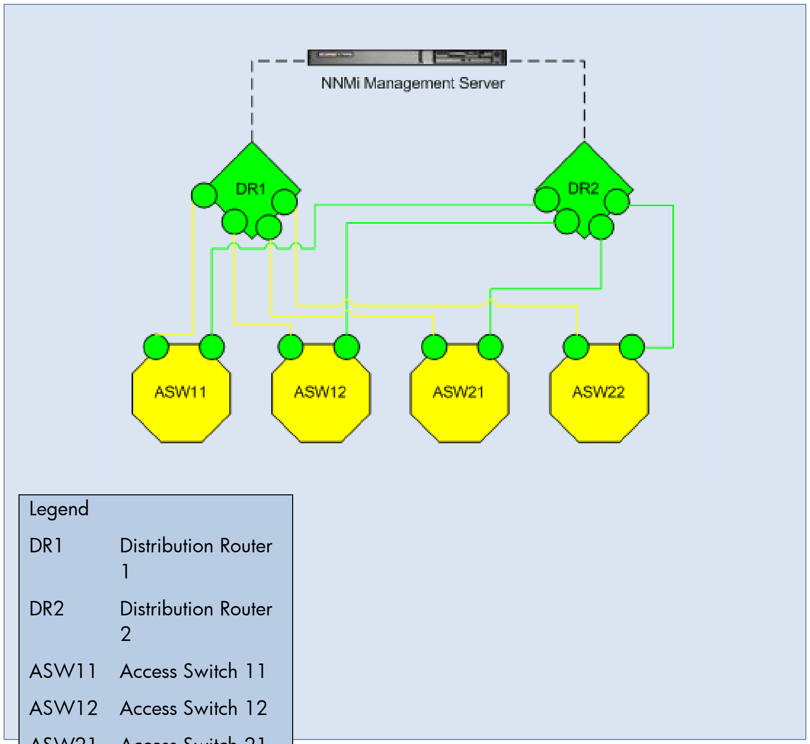
Root Cause: Node DR1 is down according to neighbor analysis.

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

Status: The 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

Sc the previous “Directly Connected Node is Down” scenario. The distribution router DR1 comes back up.

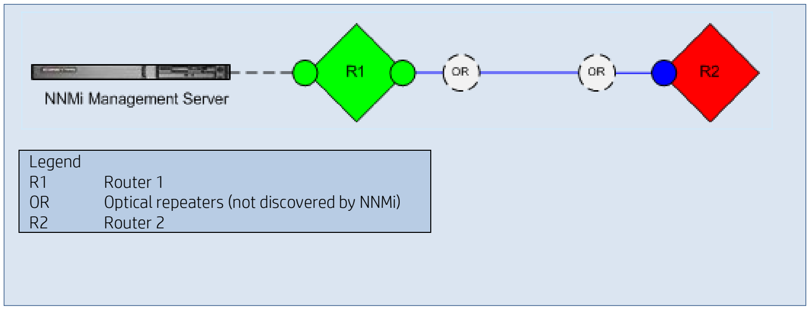
Root Cause: Node DR1 is up.

Incident: None generated. The `NodeDown` incident is closed.

Status: The node is in Normal Status.

Conclusion: `NodeUp`

Indirectly Connected Node Is Down



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.)

Router 2 becomes unreachable, either because its connected interface is down or because the connection between the optical repeaters is down. The interface on Router 1 that indirectly connects it to Router 2 is still up because its optical repeater is still up.

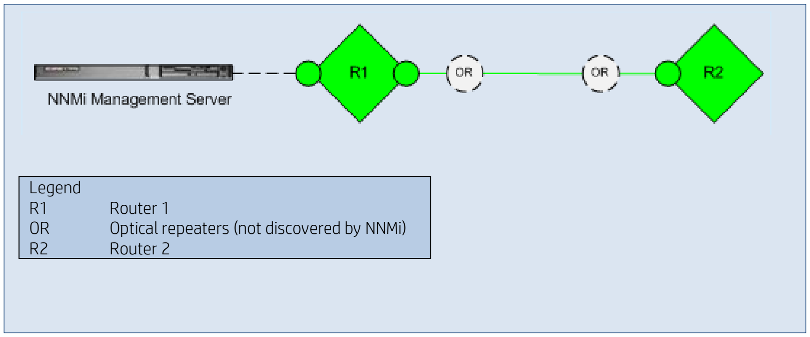
Root Cause: Router 2 is down according to neighbor analysis.

Incident: A `NodeDown` incident is generated.

Status: Node Router 2 is in Critical Status.

Conclusion: `NodeDown`

Indirectly Connected Node Is Up



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

Scenario: This scenario continues the previous "Indirectly Connected Node is Down" scenario. The failed connection comes back up. Router 2 becomes reachable.

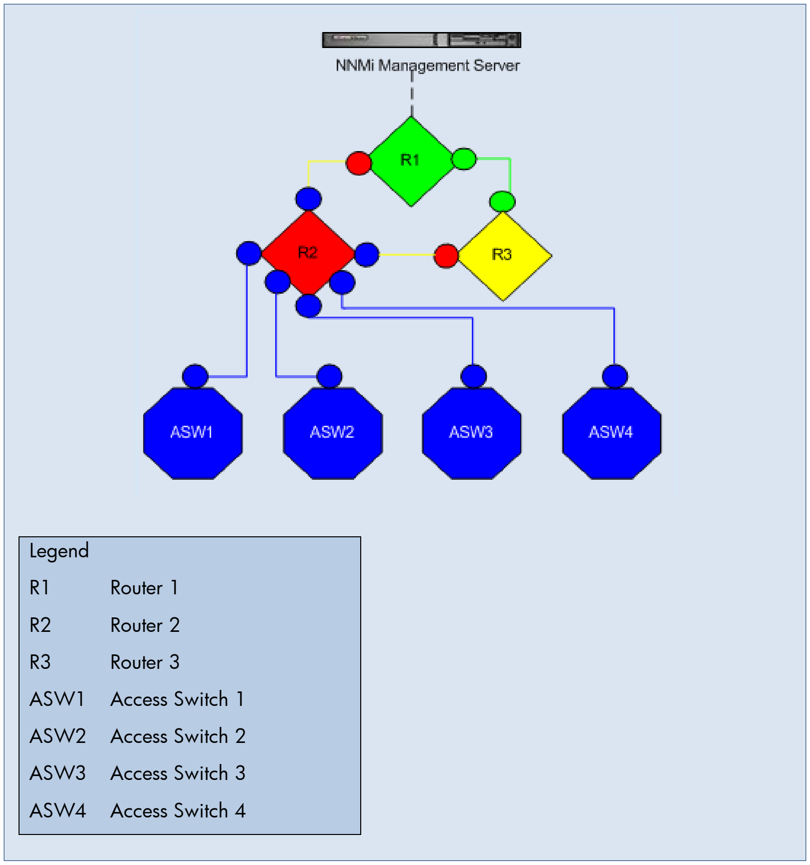
Root Cause: The connection between Router 1 and Router 2 is up.

Incident: None generated. The `NodeDown` incident is closed.

Status: Router 2's status is Normal. The connection Status is Normal.

Conclusion: `NodeUp`

Directly Connected Node Is Down and Creates a Shadow



Scenario: Router 2 (R2) goes down.

Root Cause: Node (Router 2) is down according to NNMi's neighbor analysis.

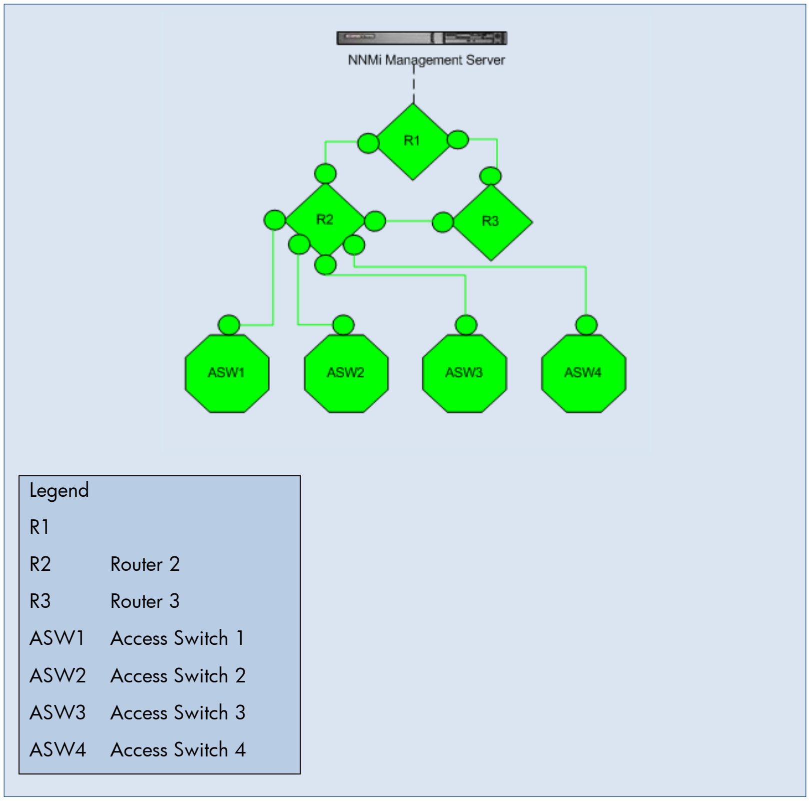
Incident: A `NodeDown` incident is generated. The `InterfaceDown` incidents from one-hop neighbors are correlated beneath the `NodeDown` incident.

Status: The 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



Scenario: This scenario continues the previous "Node is Down and Creates a Shadow" scenario. Router 2 comes back up.

Root Cause: Node Router 2 is up.

Incident: None generated. The `NodeDown` incident is closed.

Status: The 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.

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 previous "Important Node is Unreachable" scenario. The important node comes back up and can be reached.

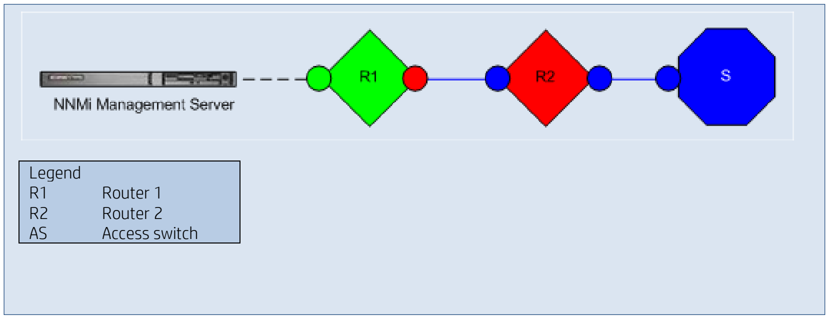
Root Cause: The node is up.

Incident: None generated. The `NodeDown` incident is closed.

Status: The node is in Normal Status.

Conclusion: NodeUp

Node or Connection Is Down



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

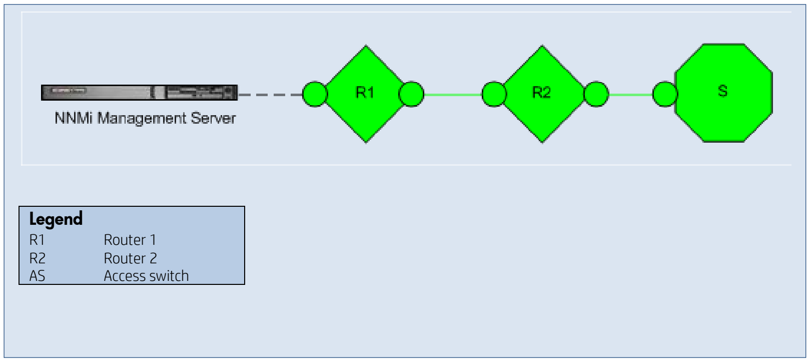
Root Cause: The node or the connection is down.

Incident: A `NodeOrConnectionDown` incident is generated. The Source Node in this scenario is Router 2.

Status: The Node is in Critical Status. The connection is in Warning Status.

Conclusion: NodeOrConnectionDown

Node or Connection Is Up



Scenario: This scenario continues the previous "Node or Connection is Down" scenario. Router 2 is now up.

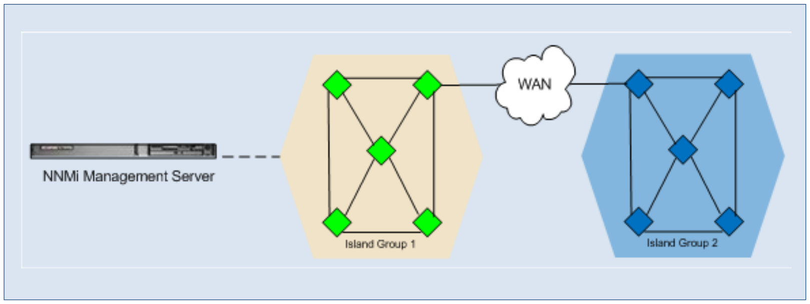
Root Cause: `NodeUp`

Incident: None generated. The `NodeOrConnectionDown` incident is 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. For more information about Island Groups, see the NNMi help.

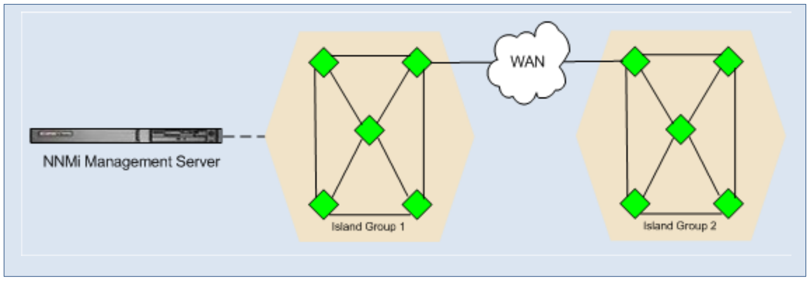
Root Cause: Island Group 2 is down according to neighbor analysis.

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 previous “Island Group is Down” scenario. The 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.

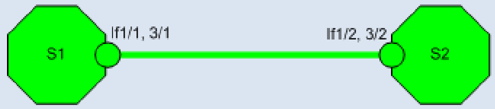
Incident: None generated. The `IslandGroupDown` incident is closed.

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.

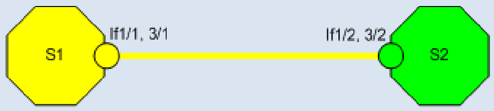
Root Cause: All operational and administrative states are up.

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.

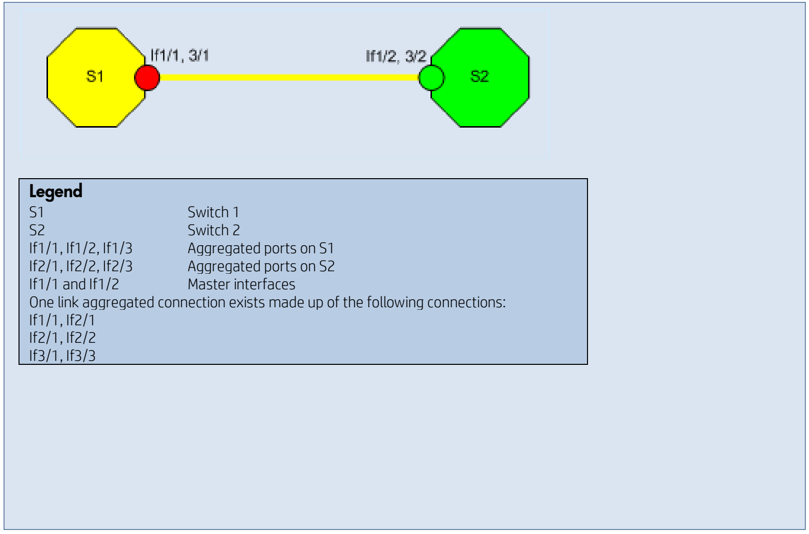
Root Cause: Operational states on some ports are down.

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.

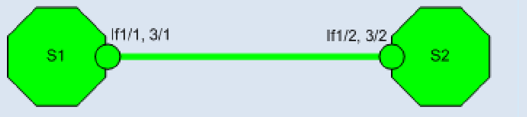
Incident: An `AggregatorDown` incident is generated.

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.

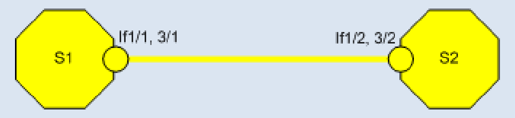
Root Cause: The aggregator is up on all members of the connection.

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.

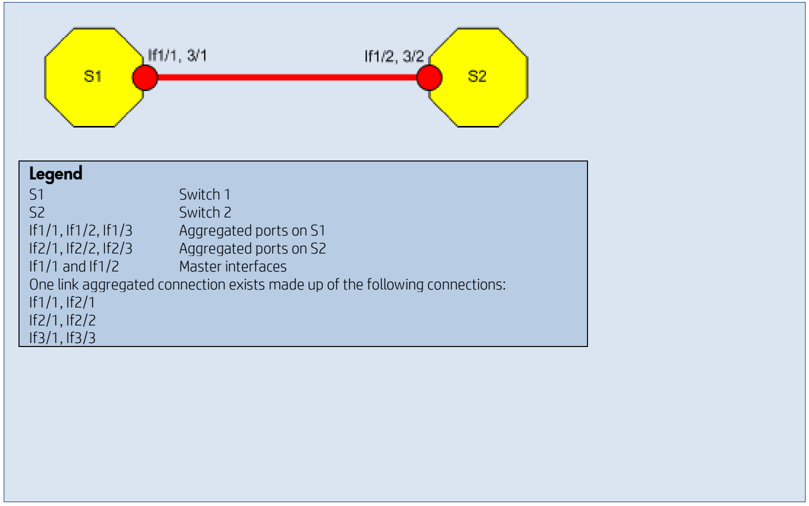
Root Cause: The aggregator is down on some members of the connection.

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: The aggregator is down on all members of the connection.

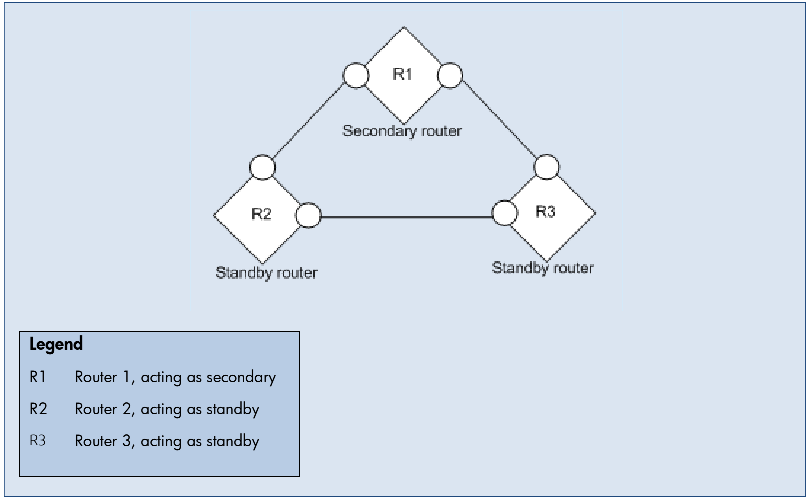
Incident: An `AggregatorLinkDown` incident is generated.

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

Conclusion: `AggregatorLinkDown`

Router Redundancy Groups: HSRP and VRRP (NNMi Advanced)

Router Redundancy Group Has No Primary


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

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

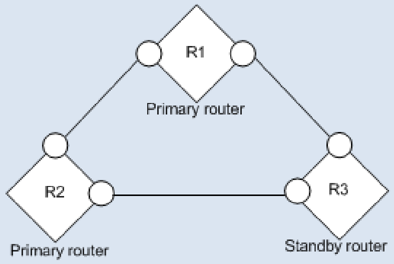
Root Cause: This scenario could be the result of an interface on the primary router failing, when the secondary was not active or a misconfiguration of the Router Redundancy Group.

Incident: An `RrgNoPrimary` incident is generated. If there is an identified root cause such as `InterfaceDown`, the `InterfaceDown` incident is correlated under the `RrgNoPrimary` incident as an Impact Correlation.

Status: The Status of the Router Redundancy Group is set to `Critical`.

Conclusion: `RrgNoPrimary`

Router Redundancy Group Has Multiple Primaries



Legend

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

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

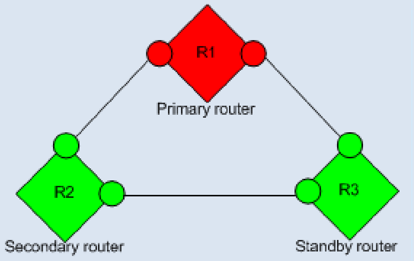
Root Cause: This scenario could be due to a faulty configuration of the Router Redundancy Group.

Incident: An `RrgMultiplePrimary` incident is generated.

Status: The Status of the Router Redundancy Group is set to `Critical`.

Conclusion: `RrgMultiplePrimary`

Router Redundancy 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

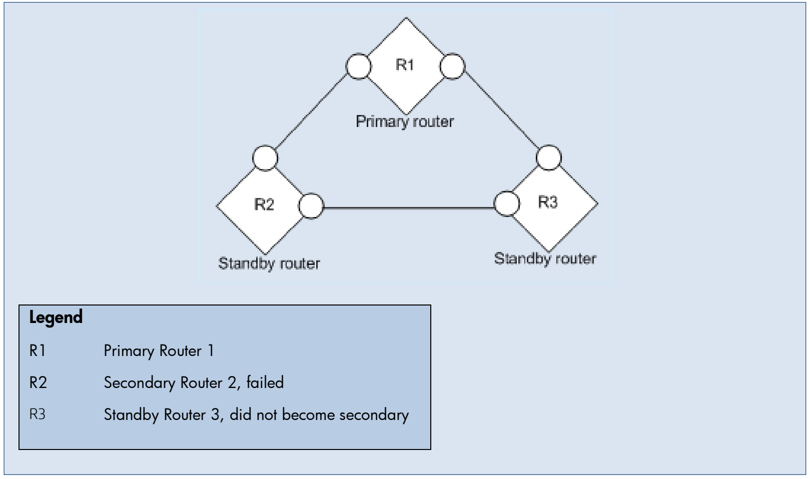
Scenario: A Router Redundancy Group has had a failure on the primary router and the secondary router has taken over as primary. Usually the standby becomes the secondary, which is not a problem. The group is functioning as intended. The incident generated for this scenario is for informational purposes 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 Correlation Nature of `RrgFailover` is Service Impact. If an identified root cause such as `InterfaceDown exists`, the `InterfaceDown` incident is correlated under the `RrgFailover` incident as an Impact Correlation.
Status: None.

Conclusion: `RrgFailover`

Router Redundancy Group Has No Secondary


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

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

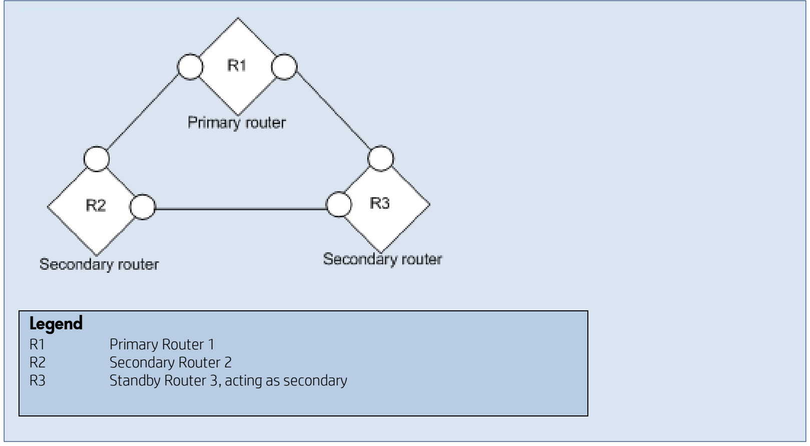
Root Cause: This scenario could be due to an interface failure on the router or some misconfiguration of the Router Redundancy Group.

Incident: An `RrgNoSecondary` incident is generated. The Correlation Nature of `RrgNoSecondary` is Service Impact. If an identified root cause such as `InterfaceDown exists`, the Correlation Nature between the `RrgNoSecondary` and `InterfaceDown` interfaces is Service Impact.

Status: The Status of the Router Redundancy Group is set to Minor.

Conclusion: `RrgNoSecondary`

Router Redundancy Group Has Multiple Secondaries



Legend

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

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

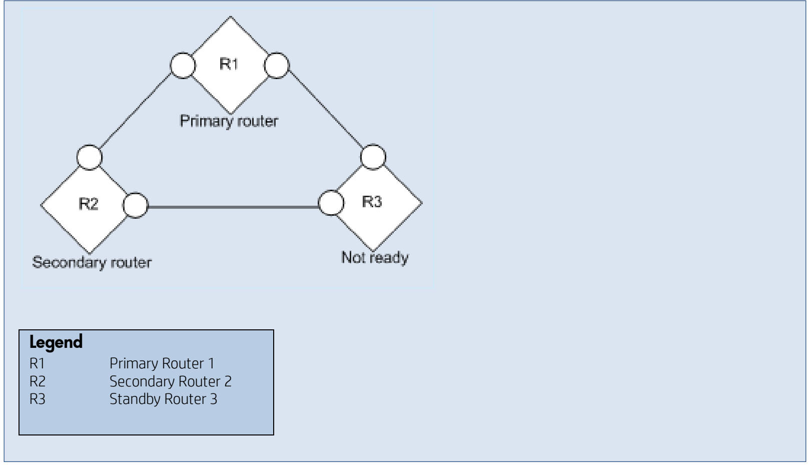
Root Cause: This scenario could be due to misconfiguration of the Router Redundancy Group.

Incident: An `RrgMultipleSecondary` incident is generated. The Correlation Nature of `RrgMultipleSecondary` is Service Impact.

Status: The Status of the Router Redundancy Group is set to Minor.

Conclusion: `RrgMultipleSecondary`

Router Redundancy Group Has Degraded



Scenario: The Router Redundancy 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 Standby state.

Root Cause: This scenario could be due to some misconfiguration of the Router Redundancy Group.

Incident: An `RrgDegraded` incident is generated. The Correlation Nature of `RrgDegraded` is Service Impact.

Status: The Status of the Router Redundancy Group is set to Warning.

Conclusion: `RrgDegraded`

Node Component Scenarios

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 Major Status is propagated to the node.

Conclusion: `NodeWithBadFan`

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 Major is propagated to the node.

Conclusion: `NodeWithBadPowerSupply`

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 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 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. The Status of Major is propagated to the node.

Conclusion: `NodeWithBadMemory`

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 IP 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 IP 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 IP address to No Status.

- NNMi administrator manages an IP 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 IP 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 IP 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 IP addresses. The `NmsApa` service calculates the Status of the node.

Network Functions Virtualization (NFV) Scenarios

Map Symbol

🟢

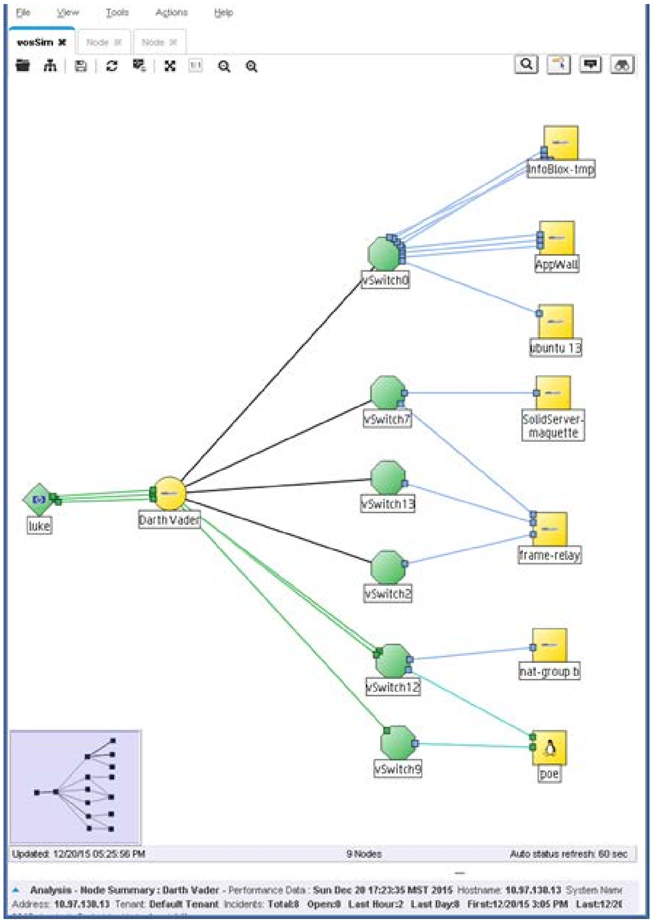
Hypervisor (*NNMi Advanced*)

The virtual machine manager in charge of delegating various aspects from a pool of resources to become virtual devices.

When you double-click the hypervisor map icon, the displayed form includes a Hosted Nodes tab listing the virtual devices being provided.

The virtual devices provided by the hypervisor have a Hosted On attribute that identifies which hypervisor provided the resources.

Web Agent Not Responding to Queries



Updated: 12/20/15 05:25:56 PM 9 Nodes Auto status refresh: 60 sec

Analysis - Node Summary : Darth Vader - Performance Data : Sun Dec 20 17:23:35 MST 2015 Hostname: 10.97.130.13 System Name
Address: 10.97.130.13 Tenant: Default Tenant Incidents: Total:8 Open:0 Last Hour:2 Last Day:8 First:12/20/15 3:05 PM Last:12/2

Scenario: (*NNMi Advanced*) Web Agent on hypervisor is not responding. For example, there is an issue with the hypervisor hosting the Web Agent, and the agent is no longer responding to requests for state information from NNMi but the hypervisor node is operational (IP addresses can be pinged).

Root Cause: Web Agent on hypervisor is not responding.

Status:
🟥 Critical for Web Agent

Incident: None generated by default. The NNMi administrator can enable the `WebAgentNotResponding` Incident configuration. See the "Generate Interface Disabled Incidents" help topic in *NNMi Help for Administrators* for more information

Conclusion: `WebAgentNotResponding`

Effect:
🟨 Minor status for Hypervisor
🟨 Minor status for virtual machines.
The conclusion on these nodes will be `UnresponsiveWebAgentInNode`.

🟦 Unknown status for Virtual ports on the hypervisor
🟦 Unknown status for interfaces on the virtual machines that are non-SNMP nodes.
The conclusion on these nodes will be `InterfaceUnmanagable`.

Web Agent Responding to Queries

Scenario: (*NNMi Advanced*) This scenario continues the previous “Web Agent Not Responding to Queries” scenario. An NNMi administrator has fixed the issue on the hypervisor. The Web Agent on the hypervisor is now responding to queries.

Root Cause: Web Agent on hypervisor is responding.

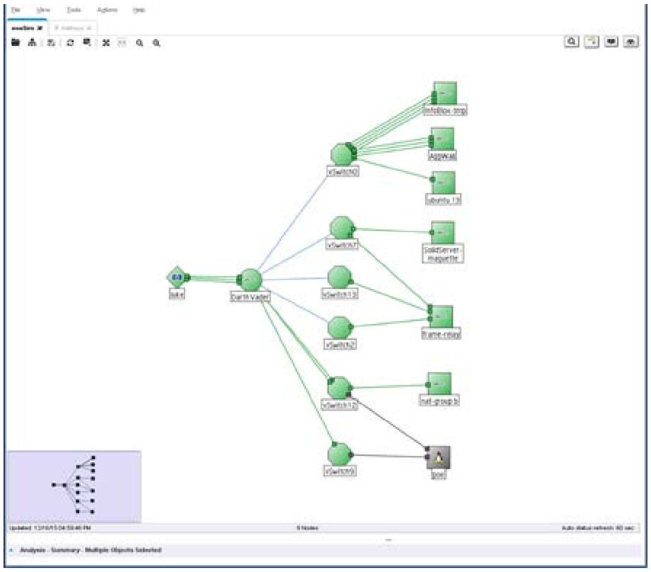
Incident: None generated.

Status:
🟩 Normal for the Web Agent

Conclusion: `WebAgentResponding`

Effect:
🟩 Normal status for the hypervisor
🟩 Normal status for virtual machines hosted on the hypervisor.
NNMi changes the `WebAgentNotResponding` conclusion to `WebAgentResponsiveInNode`.
The `InterfaceUnmanagable` conclusion is cleared from the virtual ports on the hypervisor and interfaces on the virtual machines that are non-SNMP nodes.
.

Virtual Machine is Powered Down



Scenario: (*NNMi Advanced*) A virtual machine is turned off via the operating system or virtual environment manager.

Root Cause: Virtual machine powered down.

Incident: None generated by default. The NNMi administrator can enable the `NodePoweredDown` Incident configuration. See the “Generate Interface Disabled Incidents” help topic in *NNMi Help for Administrators* for more information

Status:
⬜ Disabled for the Virtual Machine

Conclusion: `NodePoweredDown`

Effect:
⬜ Disabled status for Interfaces on Virtual Machine
⬜ Disabled status for virtual ports on the hypervisor
The Conclusion on these interfaces is `InterfaceDisabled`.

Virtual Machine is Powered Up

Scenario: This scenario continues the previous “Virtual Machine is Powered Down” scenario. A virtual machine is turned on via a virtual environment manager.

Root Cause: Virtual machine powered up.

Incident: None generated.

Status:
🟩 Normal for the Virtual Machine

Effect: NNMi closes the `NodePoweredDown` incident. `InterfaceDisabled` is cleared from the interfaces on the virtual machine and cleared from virtual ports on the hypervisor connected to the virtual machine.

Virtual Machine is Paused

Scenario: (*NNMi Advanced*) A virtual machine is paused via a virtual environment manager.

Root Cause: Virtual machine paused.

Incident: None generated by default. The NNMi administrator can enable the `NodePaused` Incident configuration. See the “Generate Interface Disabled Incidents” help topic in *NNMi Help for Administrators* for more information

Status:
⬜ Disabled for the Virtual Machine

Conclusion: `NodePaused`

Effect:
⬜ Disabled status for Interfaces on Virtual Machine
⬜ Disabled status for virtual ports on the hypervisor
The Conclusion on these interfaces is `InterfaceDisabled`.

Virtual Machine is Unpaused

Scenario: (*NNMi Advanced*) This scenario continues the previous “Virtual Machine is Paused” scenario. A virtual machine is unpaused via a virtual environment manager.

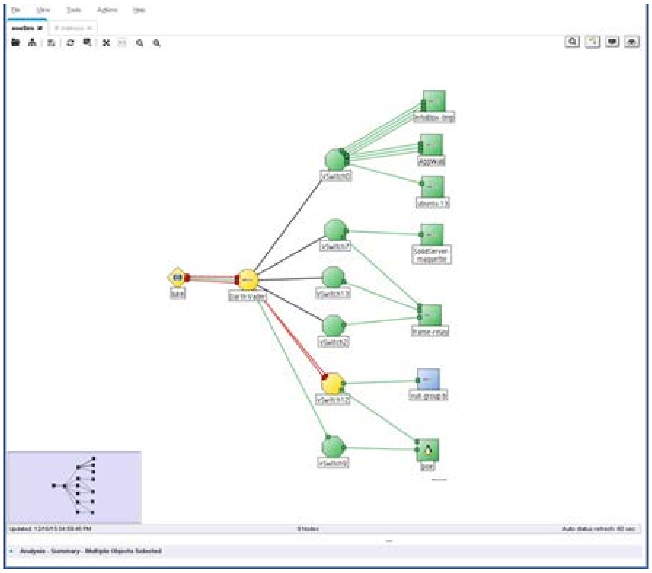
Root Cause: Virtual machine unpaused.

Incident: None generated.

Status:
🟩 Normal for the Virtual Machine

Effect: NNMi closes the `NodePaused` incident. `InterfaceDisabled` is cleared from the interfaces on the virtual machine and cleared from virtual ports on the hypervisor connected to the virtual machine.

Multiple Up Link Connections on a Virtual Switch are Operationally Down



Scenario: (*NNMi Advanced*) Multiple connections between the hypervisor and access switch are down.

Root Cause: Connections down between hypervisor and access switch.

Incident: A `ConnectionDown` incident is generated for each connection between the hypervisor and access switch.
The `InterfaceDown` incidents for the interfaces participating in the connection will be correlated under its `ConnectionDown` incident.

Status:
🟨 Minor for the Virtual Switch
🟥 Critical for Connections
🟥 Critical for Interfaces

Conclusion: `vSwitchDegraded` on Virtual Switch Interface.
`ConnectionDown` on connections.
`InterfaceDown` on interfaces.

Effect:
🟦 Unknown status for virtual machines connected to the virtual switch that has all uplinks down. The conclusion on these nodes is `NodeUnmanagable`.

Multiple Up Link Connections on a Virtual Switch are Operationally Up

Scenario: This scenario continues the previous “Multiple Up Link Connections on a Virtual Switch are Operationally Down” scenario. Multiple connections between the hypervisor and access switch are up.

Root Cause: Connections up between hypervisor and access switch.

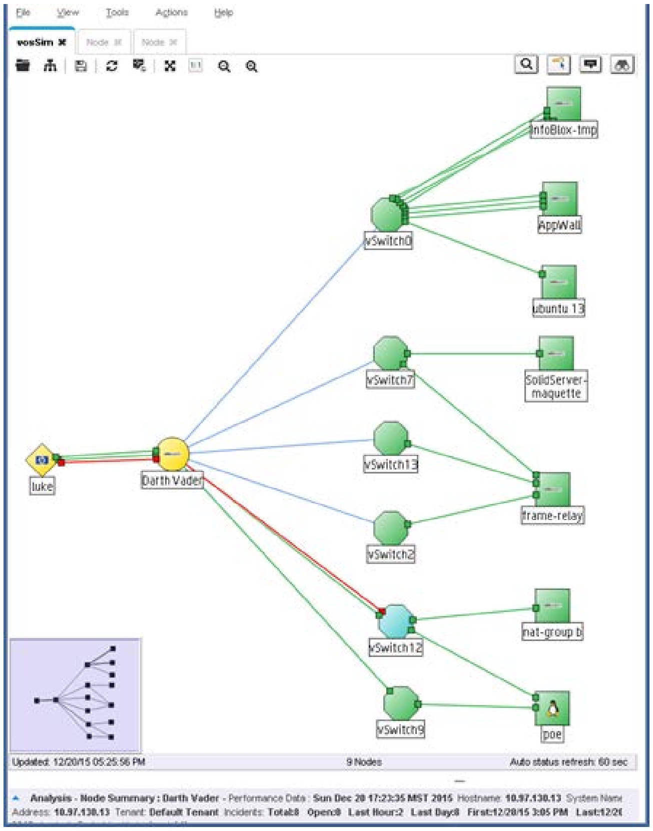
Incident: None generated. `ConnectionDown` incidents are closed.

Status:
🟩 Normal for the Virtual Switch
🟩 Normal for Links \Connections
🟩 Normal for Interfaces

Conclusion:
`vSwitchUp` on Virtual Switch.
`ConnectionUp` on connections.
`InterfaceUp` on interfaces.

Effect: `NodeUnmanagable` conclusion is cleared from Virtual machines connected to the virtual switch that is now responding.

Up Link Connection on a Virtual Switch is Operationally Down



Updated: 12/20/15 05:25:56 PM   9 Nodes   Auto status refresh: 60 sec

Analysis - Node Summary : Darth Vader - Performance Data : Sun Dec 20 17:23:35 MST 2015 Hostname: 10.97.130.13 System Name
Address: 10.97.130.13 Tenant: Default Tenant Incidents: Total:8 Open:0 Last Hour:2 Last Day:8 First:12/20/15 3:05 PM Last:12/2(

Scenario: (*NNMi Advanced*) A connection between the hypervisor and access switch is down.

Root Cause: Connection down between hypervisor and access switch.

Incident: A `ConnectionDown` incident is generated for the connection between the hypervisor and access switch. The `InterfaceDown` incidents for the interfaces participating in the connection will be correlated under the `ConnectionDown` incident.

Status:
🟦 Warning for the Virtual Switch
🟥 Critical for Links \Connections
🟥 Critical for Interfaces

Conclusion:
`vSwitchWarning` on the virtual switch.
`ConnectionDown` on the connection.
`InterfaceDown` on the interfaces.

Up Link Connection on a Virtual Switch is Operationally Up

Scenario: (*NNMi Advanced*) This scenario continues the previous "Up Link Connection on a Virtual Switch is Operationally Down" scenario. A connection between the hypervisor and access switch is up.

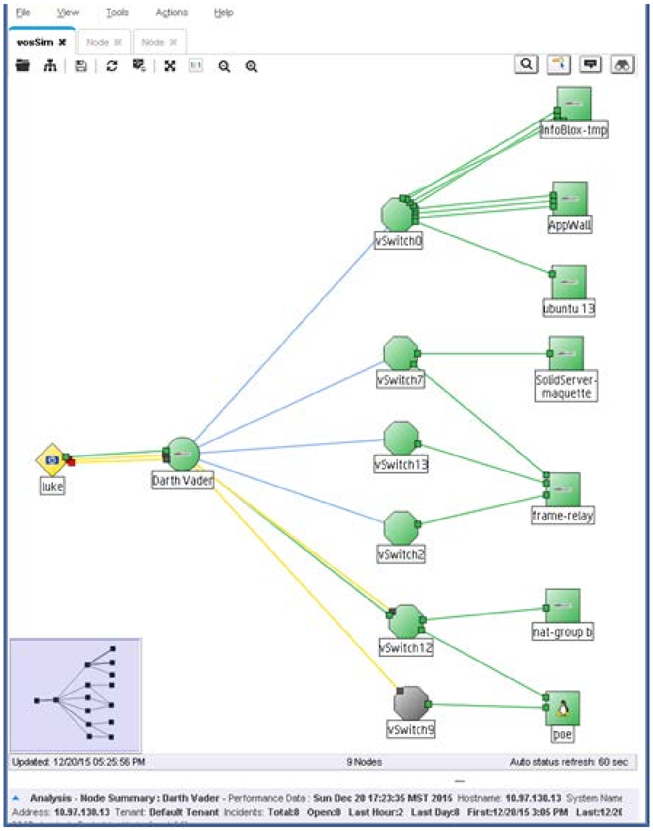
Root Cause: Connection up between hypervisor and access switch.

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

Status:
🟩 Normal for the Virtual Switch
🟩 Normal for Links \Connections
🟩 Normal for Interfaces

Conclusion:
`vSwitchUp` on the virtual switch.
`ConnectionUp` on the connection.
`InterfaceUp` on the interfaces.

All Up Link Interfaces on a Virtual Switch are Administratively Down



Updated: 12/20/15 05:25:56 PM 9 Nodes Auto status refresh: 60 sec

Analysis - Node Summary : Darth Vader - Performance Data : Sun Dec 20 17:23:36 MST 2015 Hostname: 10.97.130.13 System Name
Address: 10.97.130.13 Tenant: Default Tenant Incidents: Total:8 Open:8 Last Hour:2 Last Day:8 First:12/20/15 3:05 PM Last:12/2(

Scenario: (*NNMi Advanced*) All Up Link interfaces on a virtual switch have been administratively downed by a network operator.

Root Cause: Virtual switch Disabled

Incident: None generated by default. The NNMi administrator can enable the `InterfaceDisabled` Incident configuration to see incidents on the up link interfaces. See the "Generate Interface Disabled Incidents" help topic in *NNMi Help for Administrators* for more information.

Status:
▨ Disabled for the Virtual Switch
▨ Disabled for the Up Links

Conclusion: `vSwitchDisabled` on the virtual switch. `InterfaceDisabled` on up links.

All Up Link Interfaces on a Virtual Switch are Administratively Up

Scenario: (*NNMi Advanced*) This scenario continues the previous "All Up Link Interfaces on a Virtual Switch are Administratively Down" scenario. All Up Link interfaces on a virtual switch have been made administratively up by a network operator.

Incident: None generated. `InterfaceDisabled` incidents for the up links are closed.

Status:
🟩 Normal for the Virtual Switch
🟩 Normal for Links

Conclusion: `InterfaceUp` on up links.

We appreciate your feedback!

If an email client is configured on this system, by default an email window opens when you click here.

If no email client is available, copy the information below to a new message in a web mail client, and then send this message to network-management-doc-feedback@hpe.com.

Product name and version: NNMi 10.10

Document title: Causal Engine White Paper

Feedback:

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Oracle Technology — Notice of Restricted Rights
Programs delivered subject to the DOD FAR Supplement are 'commercial computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, programs delivered subject to the Federal Acquisition Regulations are 'restricted computer software' and use, duplication, and disclosure of the programs, including documentation, shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

For the full Oracle license text, see the license-agreements directory on the NNMi product DVD.

Copyright Notices

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Apple is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.

AMD is a trademark of Advanced Micro Devices, Inc.

Google™ is a registered trademark of Google Inc.

Intel®, Intel® Itanium®, Intel® Xeon®, and Itanium® are trademarks of Intel Corporation in the U.S. and other countries.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Internet Explorer, Lync, Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Red Hat® Enterprise Linux Certified is a registered trademark of Red Hat, Inc. in the United States and other countries.

sFlow is a registered trademark of InMon Corp.

UNIX® is a registered trademark of The Open Group.

Acknowledgements

This product includes software developed by the Apache Software Foundation (http://www.apache.org).

This product includes software developed by the Visigoth Software Society (http://www.visigoths.org/).

Sign up for updates
www.hp.com/go/hpsoftwaresupport

Support

Visit the HP Software Support web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

https://hpp12.passport.hp.com/hppcf/createuser.do

To find more information about access levels, go to:

https://softwaresupport.hp.com/web/softwaresupport/access-levels