Network Policies

Network Control Policy

This policy configures the network control settings for the Cisco UCS domain, including the following:

- Whether the Cisco Discovery Protocol (CDP) is enabled or disabled
- How the virtual interface (VIF) behaves if no uplink port is available in end-host mode
- The action that Cisco UCS Central takes on the remote Ethernet interface, vEthernet interface , or vFibre Channel interface when the associated border port fails
- Whether the server can use different MAC addresses when sending packets to the fabric interconnect
- Whether MAC registration occurs on a per-VNIC basis or for all VLANs

Action on Uplink Fail

By default, the **Action on Uplink Fail** property in the network control policy is configured with a value of link-down. For adapters such as the Cisco UCS M81KR Virtual Interface Card, this default behavior directs Cisco UCS Central to bring the vEthernet or vFibre Channel interface down if the associated border port fails. For Cisco UCS systems using a non-VM-FEX capable converged network adapter that supports both Ethernet and FCoE traffic, such as Cisco UCS CNA M72KR-Q and the Cisco UCS CNA M72KR-E, this default behavior directs Cisco UCS Central to bring the remote Ethernet interface down if the associated border port

fails. In this scenario, any vFibre Channel interfaces that are bound to the remote Ethernet interface are brought down as well.

Note if your implementation includes those types of non-VM-FEX capable converged network adapters mentioned in this section and the adapter is expected to handle both Ethernet and FCoE traffic, we recommend that you configure the **Action on Uplink Fail** property with a value of warning. Note that this configuration might result in an Ethernet teaming driver not being able to detect a link failure when the border port goes down.

MAC Registration Mode

MAC addresses are installed only on the native VLAN by default, which maximizes the VLAN port count in most implementations.

Note If a trunking driver is being run on the host and the interface is in promiscuous mode, we recommend that you set the MAC Registration Mode to All VLANs.

Configuring a Network Control Policy

MAC address-based port security for Emulex converged Network Adapters (N20-AE0102) is not supported. When MAC address-based port security is enabled, the fabric interconnect restricts traffic to packets that contain the MAC address that it first learns. This is either the source MAC address used in the FCoE Initialization Protocol packet, or the MAC address in an ethernet packet, whichever is sent first by the adaptor. This configuration can result in either FCoE or Ethernet packets being dropped.

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create nw-ctrl-policy** *policy-name*	Creates the specified network control policy, and enters organization network control policy mode.
Step 4	UCSC(policy-mgr) /org/nw-ctrl-policy # {**disable** | **enable**} **cdp**	Disables or enables Cisco Discovery Protocol (CDP).
Step 5	UCSC(policy-mgr) /org/nw-ctrl-policy # **set uplink-fail-action** {**link-down** | **warning**}	Specifies the action to be taken when no uplink port is available in end-host mode. Use the **link-down** keyword to change the operational state of a vNIC to down when uplink connectivity is lost on the fabric interconnect, and facilitate fabric failover for vNICs. Use the **warning** keyword to maintain

	Command or Action	Purpose
		server-to-server connectivity even when no uplink port is available, and disable fabric failover when uplink connectivity is lost on the fabric interconnect. The default uplink failure action is link-down.
Step 6	UCSC(policy-mgr) /org/nw-ctrl-policy # **set mac-registration-mode**{**all-host-vlans** | **only-native-vlan**	Whether adapter-registered MAC addresses are added only to the native VLAN associated with the interface or added to all VLANs associated with the interface. This can be one of the following: • **Only Native Vlan—**MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count. • **All Host Vlan—**MAC addresses are added to all VLANs with which they are associated. Select this option if your VLANs are configured to use trunking but are *not* running in Promiscuous mode.
Step 7	UCSC(policy-mgr) /org/nw-ctrl-policy # **create mac-security**	Enters organization network control policy MAC security mode
Step 8	UCSC(policy-mgr) /org/nw-ctrl-policy/mac-security # **set forged-transmit** {**allow** | **deny**}	Allows or denies the forging of MAC addresses when sending traffic. MAC security is disabled when forged MAC addresses are allowed, and MAC security is enabled when forged MAC addresses are denied. By default, forged MAC addresses are allowed (MAC security is disabled).
Step 9	UCSC(policy-mgr) /org/nw-ctrl-policy/mac-security # **commit-buffer**	Commits the transaction to the system configuration.

The following example:

- Creates a network control policy named ncp5
- Enables CDP
- Sets the uplink fail action to link-down
- Denies forged MAC addresses (enables MAC security)

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create nw-ctrl-policy ncp5
UCSC(policy-mgr) /org/nw-ctrl-policy* # enable cdp
UCSC(policy-mgr) /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCSC(policy-mgr) /org/nw-ctrl-policy* # create mac-security
UCSC(policy-mgr) /org/nw-ctrl-policy/mac-security* # set forged-transmit deny
UCSC(policy-mgr) /org/nw-ctrl-policy/mac-security* # commit-buffer
UCSC(policy-mgr) /org/nw-ctrl-policy/mac-security #
```

Deleting a Network Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete nw-ctrl-policy** *policy-name*	Deletes the specified network control policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the network control policy named ncp5:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete nw-ctrl-policy ncp5
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in an cluster configuration with two fabric interconnects

Note For Fibre Channel adapter policies, the values displayed by may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and :

- Max LUNs Per Target—SANsurfer has a maximum of 256 LUNs and does not display more than that number. supports a higher maximum number of LUNs.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In , you configure this value in milliseconds. Therefore, a value of 5500 ms in displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. allows you to set values of any size. Therefore, a value of 900 in displays as 512 in SANsurfer.

Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

Important We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for a Windows OS (instead of using the default Windows adapter policy), you must use the following formulas to calculate values that work with Windows:

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

Configuring an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.

	Command or Action	Purpose
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 3	UCSC(policy-mgr) /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 4	UCSC(policy-mgr) /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}	(Optional) Select whether to enable Accelerated Receive Flow Steering (ARFS).
Step 5	UCSC(policy-mgr) /org/eth-policy # **set comp-queue count** *count*	(Optional) Configures the Ethernet completion queue.
Step 6	UCSC(policy-mgr) /org/eth-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 7	UCSC(policy-mgr) /org/eth-policy # **set failback timeout** *timeout-sec*	(Optional) Configures the Ethernet failover.
Step 8	UCSC(policy-mgr) /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	(Optional) Configures the Ethernet interrupt.
Step 9	UCSC(policy-mgr) /org/eth-policy # **set nvgre adminstate** {**enabled** | **disabled**}	(Optional) Select whether to enable NVGRE.
Step 10	UCSC(policy-mgr) /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	(Optional) Configures the Ethernet offload.
Step 11	UCSC(policy-mgr) /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet receive queue.
Step 12	UCSC(policy-mgr) /org/eth-policy # **set roce adminstate** {**enabled** | **disabled**}	(Optional) Select whether to enable RoCE.
Step 13	UCSC(policy-mgr) /org/eth-policy # **set roce memoryregions** *memory_regions_number*	(Optional) Specify the RoCE memory regions. This can be between 1 and 524288.

	Command or Action	Purpose
Step 14	UCSC(policy-mgr) /org/eth-policy # **set roce queuepairs** *queue_pairs_number*	(Optional) Specify the RoCE queue pairs. This can be between 1 and 8192.
Step 15	UCSC(policy-mgr) /org/eth-policy # **set roce resourcegroups** *resource_groups_number*	(Optional) Specify the RoCE queue pairs. This can be between 1 and 128.
Step 16	UCSC(policy-mgr) /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	(Optional) Configures the RSS.
Step 17	UCSC(policy-mgr) /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet transmit queue.
Step 18	UCSC(policy-mgr) /org/eth-policy # **set vxlan adminstate** {**enabled** | **disabled**}	(Optional) Select whether to enable VXLAN.
Step 19	UCSC(policy-mgr) /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures an Ethernet adapter policy:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org* # create eth-policy EthPolicy19
UCSC(policy-mgr) /org/eth-policy* # set comp-queue count 16
UCSC(policy-mgr) /org/eth-policy* # set descr "This is an Ethernet adapter policy example."
UCSC(policy-mgr) /org/eth-policy* # set failover timeout 300
UCSC(policy-mgr) /org/eth-policy* # set interrupt count 64
UCSC(policy-mgr) /org/eth-policy* # set offload large-receive disabled
UCSC(policy-mgr) /org/eth-policy* # set recv-queue count 32
UCSC(policy-mgr) /org/eth-policy* # set rss receivesidescaling enabled
UCSC(policy-mgr) /org/eth-policy* # set trans-queue
UCSC(policy-mgr) /org/eth-policy* # commit-buffer
UCSC(policy-mgr) /org/eth-policy #
```

Deleting an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete eth-policy** *policy-name*	Deletes the specified Ethernet adapter policy.

	Command or Action	Purpose
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an Ethernet adapter policy, and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete eth-policy EthPolicy19
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Dynamic vNIC Connection Policy

The dynamic vNIC connection policy determines how the connectivity between VMs and dynamic vNICs is configured. This policy is required for Cisco UCS domains that include servers with VIC adapters on which you have installed VMs and configured dynamic vNICs.

Ethernet Adapter Policy

Each dynamic vNIC connection policy includes an Ethernet adapter policy and designates the number of vNICs that can be configured for any server associated with a service profile that includes the policy.

Server Migration

Note If you migrate a server that is configured with dynamic vNICs, the dynamic interface used by the vNICs fails and notifies you of that failure.

When the server comes back up, assigns new dynamic vNICs to the server. If you are monitoring traffic on the dynamic vNIC, you must reconfigure the monitoring source.

Creating a Dynamic vNIC Connections Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create dynamic-vnic-conn-policy** *policy-name*	Creates a dynamic vNIC connectivity policy.

	Command or Action	Purpose
Step 4	UCSC(policy-mgr) /org /dynamic-vnic-conn-policy* # **set adapter-policy** *profile-name*	Associates the adapter profile to the policy.
Step 5	UCSC(policy-mgr) /org /dynamic-vnic-conn-policy* # **set dynamic-eth** *value*	(Optional) Displays 54, the default number. You can enter an integer between 0 to 256 for the number of dynamic vNICs this policy affects.
Step 6	UCSC(policy-mgr) /org /dynamic-vnic-conn-policy* # **set protection protected-pref-a**	(Optional) Protects dynamic vNIC connectivity policy. You can choose one of the following: • **Protected Pref A—**Cisco UCS attempts to use fabric A but fails over to fabric B if necessary • **Protected Pref A—** Cisco UCS attempts to use fabric B but fails over to fabric A if necessary • **Protected—**Cisco UCS uses whichever fabric is available
Step 7	UCSC(policy-mgr) /org /dynamic-vnic-conn-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a dynamic vNIC connectivity policy called g-DyVCONPol-1 and sets adapter profile g-ethPol-1 to associate with the policy.

```
UCSC# connect policy-mgr
UCSC(policy-mgr) # scope org /
UCSC(policy-mgr) /org # create dynamic-vnic-conn-policy g-DyVCONPol-1
UCSC(policy-mgr) /org /dynamic-vnic-conn-policy* # set adapter-policy g-ethPol-1
UCSC(policy-mgr) /org /dynamic-vnic-conn-policy* # commit-buffer
UCSC(policy-mgr) /org /dynamic-vnic-conn-policy #
```

Deleting a Dynamic vNIC Connection Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .

	Command or Action	Purpose
Step 3	UCSC(policy-mgr) /org # **delete dynamic-vnic-conn-policy** *policy-name*	Deletes the specified dynamic vNIC connection policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the dynamic vNIC connection policy named sample-1 and commits the transaction:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete dynamic-vnic-conn-policy sample-1
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

Configuring a usNIC Connection Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create usnic-conn-policy** *policy-name*	Creates a usNIC connection policy and enters organization usNIC mode.
Step 4	UCSC(policy-mgr) /org/usnic-conn-policy # **set adapter-profile** *profile-name*	Specifies the adapter profile that you want to use for the usNIC. We recommend that you choose the usNIC adapter profile, which is created by default.
Step 5	UCSC(policy-mgr) /org/usnic-conn-policy # **set usnic-count** *number-of-usNICs*	Specifies the number of Cisco usNICs that you want to create.
Step 6	UCSC(policy-mgr) /org/usnic-conn-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a usNIC connection policy:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org # create usnic-conn-policy usnic-pol1
UCSC(policy-mgr) /org/usnic-conn-policy* # set descr "This is a usNIC connection policy
example."
UCSC(policy-mgr) /org/usnic-conn-policy* # set adapter-profile usnic
```

```
UCSC(policy-mgr) /org/usnic-conn-policy* # set usnic-count 58
UCSC(policy-mgr) /org/usnic-conn-policy* # commit-buffer
UCSC(policy-mgr) /org/usnic-conn-policy #
```

About the LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

Note We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies
- ls-storage—Can create SAN connectivity policies

Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

Creating a LAN Connectivity Policy

You can create a LAN connectivity policy for LAN networks.

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.

	Command or Action	Purpose
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create lan-connectivity-policy** *policy-name*	Creates the specified LAN connectivity policy, and enters organization network control policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period) and you cannot change this name after the object has been saved.
Step 4	UCSC(policy-mgr) /org/lan-connectivity-policy # **set descr** *policy-name*	(Optional) Adds a description to the policy. We recommend that you include information about where and how the policy should be used. Enter up to 256 characters. you can use any characters or spaces except ' (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 5	UCSC(policy-mgr) /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows you how to create a LAN connectivity policy named Local_LAN:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org# create lan-connectivity-policy Local_LAN
UCSC(policy-mgr) /org/lan-connectivity-policy # set descr Local on site LAN policy
UCSC(policy-mgr) /org/lan-connectivity-policy # commit buffer
```

Creating a vNIC for a LAN Connectivity Policy

You can create a vNIC for a LAN connectivity policy.

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.

	Command or Action	Purpose
Step 3	UCSC(policy-mgr) /org # **scope lan-connectivity-policy** *policy-name*	Enters organization network control policy mode.
Step 4	UCSC(policy-mgr) /org/lan-connectivity-policy # **enter vnic** *vnic-name*	Creates a vNIC and enters configuration mode for the specified vNIC.
Step 5	UCSC(policy-mgr) /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows you how to add a vNIC called vNIC1 to an existing LAN connectivity policy:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org# scope lan-connectivity-policy Local_LAN
UCSC(policy-mgr) /org/lan-connectivity-policy # enter vnic vNIC1
UCSC(policy-mgr) /org/lan-connectivity-policy # commit buffer
```

Creating an iSCSI vNIC for a LAN Connectivity Policy

You can create an iscsi vNIC for a LAN connectivity policy.

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **scope lan-connectivity-policy** *policy-name*	Enters organization network control policy mode.
Step 4	UCSC(policy-mgr) /org/lan-connectivity-policy # **enter vnic-iscsi** *iscsi-vnic-name*	Creates an iSCSI vNIC and enters configuration mode for the specified iSCSI vNIC.
Step 5	UCSC(policy-mgr) /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows you how to add an iSCSI vNIC called iSCSI_vNIC1 to an existing LAN connectivity policy:

```
UCSC# connect policy-mgr
```

```
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org# scope lan-connectivity-policy Local_LAN
UCSC(policy-mgr) /org/lan-connectivity-policy # enter vnic-iscsi iSCSI_vNIC1
UCSC(policy-mgr) /org/lan-connectivity-policy # commit buffer
```

UniDirectional Link Detection (UDLD)

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it marks the link as unidirectional. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

UDLD works with the Layer 1 mechanisms to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected interfaces. When you enable both autonegotiation and UDLD, the Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic sent by a local device is received by its neighbor but traffic from the neighbor is not received by the local device.

Modes of Operation

UDLD supports two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected interfaces on fiber-optic connections. In aggressive mode, UDLD can also detect unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and to misconnected interfaces on fiber-optic links.

In normal mode, UDLD detects a unidirectional link when fiber strands in a fiber-optic interface are misconnected and the Layer 1 mechanisms do not detect this misconnection. If the interfaces are connected correctly but the traffic is one way, UDLD does not detect the unidirectional link because the Layer 1 mechanism, which is supposed to detect this condition, does not do so. In case, the logical link is considered undetermined, and UDLD does not disable the interface. When UDLD is in normal mode, if one of the fiber strands in a pair is disconnected and autonegotiation is active, the link does not stay up because the Layer 1 mechanisms did not detect a physical problem with the link. In this case, UDLD does not take any action, and the logical link is considered undetermined.

UDLD aggressive mode is disabled by default. Configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. With UDLD aggressive mode enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor and administratively shuts down the affected port. UDLD in aggressive mode can also detect a unidirectional link on a point-to-point link on which no failure between the two devices is allowed. It can also detect a unidirectional link when one of the following problems exists:

- On fiber-optic or twisted-pair links, one of the interfaces cannot send or receive traffic.
- On fiber-optic or twisted-pair links, one of the interfaces is down while the other is up.
- One of the fiber strands in the cable is disconnected.

Methods to Detect Unidirectional Links

UDLD operates by using two mechanisms:

- Neighbor database maintenance

  UDLD learns about other UDLD-capable neighbors by periodically sending a hello packet (also called an advertisement or probe) on every active interface to keep each device informed about its neighbors. When the switch receives a hello message, it caches the information until the age time (hold time or time-to-live) expires. If the switch receives a new hello message before an older cache entry ages, the switch replaces the older entry with the new one.

  UDLD clears all existing cache entries for the interfaces affected by the configuration change whenever an interface is disabled and UDLD is running, whenever UDLD is disabled on an interface, or whenever the switch is reset. UDLD sends at least one message to inform the neighbors to flush the part of their caches affected by the status change. The message is intended to keep the caches synchronized.

- Event-driven detection and echoing

  UDLD relies on echoing as its detection mechanism. Whenever a UDLD device learns about a new neighbor or receives a resynchronization request from an out-of-sync neighbor, it restarts the detection window on its side of the connection and sends echo messages in reply. Because this behavior is the same on all UDLD neighbors, the sender of the echoes expects to receive an echo in reply.

  If the detection window ends and no valid reply message is received, the link might shut down, depending on the UDLD mode. When UDLD is in normal mode, the link might be considered undetermined and might not be shut down. When UDLD is in aggressive mode, the link is considered unidirectional, and the interface is shut down.

If UDLD in normal mode is in the advertisement or in the detection phase and all the neighbor cache entries are aged out, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbors.

If you enable aggressive mode when all the neighbors of a port have aged out either in the advertisement or in the detection phase, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbor. UDLD shuts down the port if, after the fast train of messages, the link state is still undetermined.

UDLD Configuration Guidelines

The following guidelines and recommendations apply when you configure UDLD:

- A UDLD-capable interface also cannot detect a unidirectional link if it is connected to a UDLD-incapable port of another switch.
- When configuring the mode (normal or aggressive), make sure that the same mode is configured on both sides of the link.
- UDLD should be enabled only on interfaces that are connected to UDLD capable devices. The following interface types are supported:
  - Ethernet uplink
  - FCoE uplink
  - Ethernet uplink port channel member
  - FCoE uplink port channel member

Configuring a UDLD Link Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope domain-group** *domain-group*	Enters domain group root mode and (optionally) enters a sub-domain group under the domain group root. To enter the domain group root mode, type **/** as the *domain-group*.
Step 3	UCSC(policy-mgr) /domain-group # **create udld-link-policy** *policy-name*	Creates a UDLD link policy and enters domain group UDLD link policy mode.
Step 4	UCSC(policy-mgr) /domain-group/udld-link-policy # **set mode** {**aggressive** | **normal**}	Specifies the mode for the UDLD link policy.
Step 5	UCSC(policy-mgr) /domain-group/udld-link-policy # **set admin-state** {**disabled** | **enabled**}	Enables or disables UDLD on the interface.
Step 6	UCSC(policy-mgr) /domain-group/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to:

- Create a UDLD link policy called UDLDPol1
- Set the mode to aggressive
- Enable UDLD on the interface

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope domain-group /
UCSC(policy-mgr) /domain-group # create udld-link-policy UDLDPol1
UCSC(policy-mgr) /domain-group/udld-link-policy* # set mode aggressive
UCSC(policy-mgr) /domain-group/udld-link-policy* # set admin-state enabled
UCSC(policy-mgr) /domain-group/udld-link-policy* # commit-buffer
UCSC(policy-mgr) /domain-group/udld-link-policy #
```

Configuring a Link Profile

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope domain-group** *domain-group*	Enters domain group root mode and (optionally) enters a sub-domain group under the domain group root. To enter the domain group root mode, type / as the *domain-group*.
Step 3	UCSC(policy-mgr) /domain-group # **create eth-link-profile** *profile-name*	Creates a link profile and enters domain group link profile mode.
Step 4	UCSC(policy-mgr) /domain-group/eth-link-profile # **set udld-link-policy** *udld-link-policy-name*	Assigns the specified UDLD link policy to the link profile.
Step 5	UCSC(policy-mgr) /domain-group/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a link profile called LinkProfile1, assign the default UDLD link policy, and commit the transaction.

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope domain-group /
UCSC(policy-mgr) /domain-group # create eth-link-profile LinkProfile1
UCSC(policy-mgr) /domain-group/eth-link-profile* # set udld-link-policy default
UCSC(policy-mgr) /domain-group/eth-link-profile* # commit-buffer
UCSC(policy-mgr) /domain-group/eth-link-profile #
```

Flow Control Policy

Flow control policies determine whether the uplink Ethernet ports in a Cisco UCS domain send and receive IEEE 802.3x pause frames when the receive buffer for a port fills. These pause frames request that the transmitting port stop sending data for a few milliseconds until the buffer clears.

For flow control to work between a LAN port and an uplink Ethernet port, you must enable the corresponding receive and send flow control parameters for both ports. For Cisco UCS, the flow control policies configure these parameters.

When you enable the send function, the uplink Ethernet port sends a pause request to the network port if the incoming packet rate becomes too high. The pause remains in effect for a few milliseconds before traffic is reset to normal levels. If you enable the receive function, the uplink Ethernet port honors all pause requests from the network port. All traffic is halted on that uplink port until the network port cancels the pause request.

Because you assign the flow control policy to the port, changes to the policy have an immediate effect on how the port reacts to a pause frame or a full receive buffer.

Configuring a Flow Control Policy

Note If you have selected global port configuration for **Policy Resolution Control** in Cisco UCS Manager, then any local flow control policies will be overwritten by global flow control policies in the same domain group that have the same name.

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope domain-group** *domain-group*	Enters domain group root mode and (optionally) enters a sub-domain group under the domain group root. To enter the domain group root mode, type / as the *domain-group*.
Step 3	UCSC(policy-mgr) /domain-group # **scope flow-control**	Enters domain group flow control mode.
Step 4	UCSC(policy-mgr) /domain-group/flow-control # **create policy** *flow-control-policy-name*	Creates the specified flow control policy.
Step 5	UCSC(policy-mgr) /domain-group/flow-control/policy # **set prio** {**auto** | **on**}	Specifies whether PPP is enabled or negotiated between Cisco UCS and the network.
Step 6	UCSC(policy-mgr) /domain-group/flow-control/policy # **set receive** {**off** | **on**}	Specifies whether pause requests from the network are honored or ignored.
Step 7	UCSC(policy-mgr) /domain-group/flow-control/policy # **set send** {**off** | **on**}	Specifies whether traffic flows normally regardless of the packet load, or if Cisco UCS sends pause requests if the incoming packet rate becomes too high.
Step 8	UCSC(policy-mgr) /domain-group/flow-control/policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure a flow control policy:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope domain-group /
UCSC(policy-mgr) /domain-group # scope flow-control
UCSC(policy-mgr) /domain-group/flow-control # create policy FlowCon1
UCSC(policy-mgr) /domain-group/flow-control/policy* # set prio auto
UCSC(policy-mgr) /domain-group/flow-control/policy* # set receive on
UCSC(policy-mgr) /domain-group/flow-control/policy* # set send on
UCSC(policy-mgr) /domain-group/flow-control/policy* # commit-buffer
UCSC(policy-mgr) /domain-group/flow-control #
```

Quality of Service Policy

A quality of service (QoS) policy assigns a system class to the outgoing traffic for a vNIC or vHBA. This system class determines the quality of service for that traffic. For certain adapters, you can also specify additional controls on the outgoing traffic, such as burst and rate.

You must include a QoS policy in a vNIC policy or vHBA policy and then include that policy in a service profile to configure the vNIC or vHBA.

Configuring a QoS Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create qos-policy** *policy-name*	Creates the specified QoS policy, and enters org QoS policy mode.
Step 4	UCSC(policy-mgr) /org/qos-policy # **create egress-policy**	Creates the egress policy (for both vNICs and vHBAs) to be used by the QoS policy, and enters org QoS policy egress policy mode.
Step 5	UCSC(policy-mgr) /org/qos-policy/egress-policy # **set host-cos-control** {**full** | **none**}	(Optional) Specifies whether the host or Cisco UCS Central controls the class of service (CoS) for a vNIC. This setting has no effect on a vHBA. Use the **full** keyword to have the host control the CoS. If the packet has a valid CoS value, the host uses that value. Otherwise, it uses the CoS value associated with the specified class priority. Use the **none** keyword to have Cisco UCS Central use the CoS value associated with the specified priority.
Step 6	UCSC(policy-mgr) /org/qos-policy/egress-policy # **set prio** *sys-class-name*	Specifies the system class to be used for the egress policy. The *sys-class-name* argument can be one of the following class keywords: • **FC—**(Fibre Channel) Use this priority for QoS policies that control vHBA traffic only. • **Platinum—**Use this priority for QoS policies that control vNIC traffic only. • **Gold—**Use this priority for QoS policies that control vNIC traffic only. • **Silver—**Use this priority for QoS policies that control vNIC traffic only.

	Command or Action	Purpose
		• **Bronze—**Use this priority for QoS policies that control vNIC traffic only. • **Best Effort—**Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Central does not default to this system class. It defaults to the priority with CoS 0 for that traffic.
Step 7	UCSC(policy-mgr) /org/qos-policy/egress-policy # **set rate** {**line-rate** | *kbps*} **burst** *bytes*	Specifies the rate limit for egress traffic by defining the average traffic rate and burst size. The **line-rate** keyword sets the rate limit to the physical line rate. Rate limiting is supported only on vNICs on Cisco VIC 1240 and Cisco VIC 1280. M81KR supports rate limiting on both vNICs and vHBAs.
Step 8	UCSC(policy-mgr) /org/qos-policy/egress-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to:

- Create a QoS policy for vNIC traffic
- Assign the platinum system class
- Set the rate limit (traffic rate and burst size) for the egress policy

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create qos-policy VnicPolicy34
UCSC(policy-mgr) /org/qos-policy* # create egress-policy
UCSC(policy-mgr) /org/qos-policy/egress-policy* # set prio platinum
UCSC(policy-mgr) /org/qos-policy/egress-policy* # set rate 5000000 burst 65000
UCSC(policy-mgr) /org/qos-policy/egress-policy* # commit-buffer
UCSC(policy-mgr) /org/qos-policy/egress-policy #
```

The following example shows how to:

- Create a QoS policy for vHBA traffic
- Assign the FC (Fibre Channel) system class
- Set the rate limit (traffic rate and burst size) for the egress policy

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # create qos-policy VhbaPolicy12
UCSC(policy-mgr) /org/qos-policy* # create egress-policy
UCSC(policy-mgr) /org/qos-policy/egress-policy* # set prio fc
UCSC(policy-mgr) /org/qos-policy/egress-policy* # set rate 5000000 burst 65000
UCSC(policy-mgr) /org/qos-policy/egress-policy* # commit-buffer
UCSC(policy-mgr) /org/qos-policy/egress-policy #
```

What to Do Next

Include the QoS policy in a vNIC or vHBA template.

Deleting a QoS Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **delete qos-policy** *policy-name*	Deletes the specified QoS policy.
Step 4	UCSC(policy-mgr) /org # **commit-buffer**	Commits the transaction to the system configuration.

The following deletes the QoS policy named QosPolicy34:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org /
UCSC(policy-mgr) /org # delete qos-policy QosPolicy34
UCSC(policy-mgr) /org* # commit-buffer
UCSC(policy-mgr) /org #
```

VMQ Connection Policy

VMQ provides improved network performance to the entire management operating system. From Cisco UCS Central you can create a VMQ connection policy for a vNIC on a service profile. To configure the VMQ vNIC on a service profile for a server, at least one adapter in the server must support VMQ. Make sure the servers have at least one the following adapters installed:

- UCS-VIC-M82-8P
- UCSB-MLOM-40G-01
- UCSC-PCIE-CSC-02

You must have one of the following Operating systems to use VMQ:

- Windows 2012
- Windows 2012R2

When you select the vNIC connection policy for a service profile, make sure to select one of the three options such as Dynamic, usNIC or VMQ connection policy for the vNIC. You can apply only any one of the vNIC connection policies on a service profile at any one time.

When you have selected VMQ policy on the vNIC for a service profile, you must also have the following settings in the service profile:

- Select SRIOV in the BIOS policy.
- Select Windows in the Adapter policy.

Configuring a VMQ vNIC connection policy involves the following:

- Creating a VMQ connection policy
- Creating a static vNIC in a service profile
- Applying the VMQ connection policy to the vNIC

Configuring a VMQ Connection Policy

Procedure

	Command or Action	Purpose
Step 1	UCSC# **connect policy-mgr**	Enters policy manager mode.
Step 2	UCSC(policy-mgr) # **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 3	UCSC(policy-mgr) /org # **create vmq-conn-policy** *policy-name*	Creates a VMQ connection policy and enters organization VMQ connection policy mode.
Step 4	UCSC(policy-mgr) /org/vmq-conn-policy # **set queue-count** *queue-count*	Specifies the queue count for the VMQ connection policy.
Step 5	UCSC(policy-mgr) /org/vmq-conn-policy # **set interrupt-count** *interrupt-count*	Specifies the interrupt count for the VMQ connection policy.
Step 6	UCSC(policy-mgr) /org/vmq-conn-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a VMQ connection policy:

```
UCSC# connect policy-mgr
UCSC(policy-mgr)# scope org
UCSC(policy-mgr) /org # create vmq-conn-policy vmq-pol1
UCSC(policy-mgr) /org/vmq-conn-policy* # set descr "This is a VMQ connection policy example."
UCSC(policy-mgr) /org/vmq-conn-policy* # set queue-count 10
UCSC(policy-mgr) /org/vmq-conn-policy* # set interrupt-count 10
UCSC(policy-mgr) /org/vmq-conn-policy* # commit-buffer
UCSC(policy-mgr) /org/vmq-conn-policy #
```