Multi-tenancy

Feature Information for Multi-tenancy

Table 1: Feature Information for Multi-tenancy

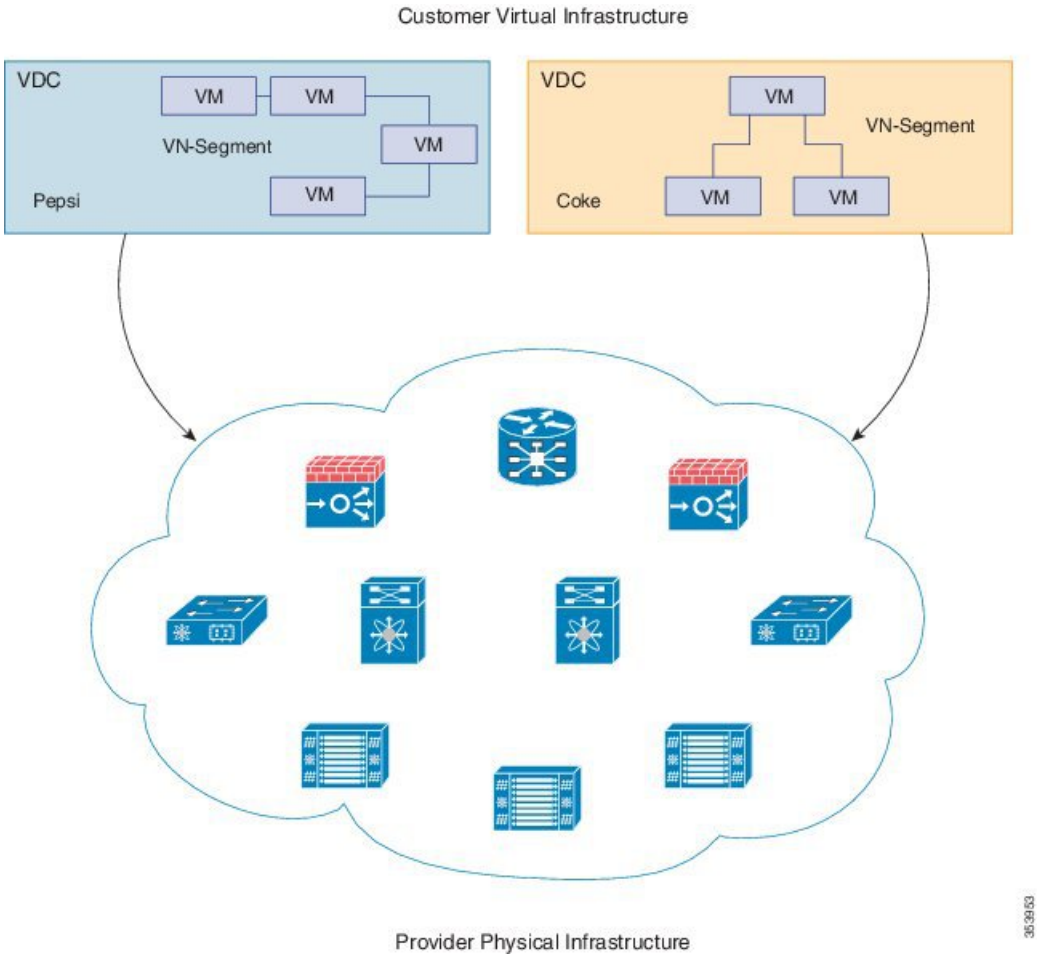
Feature	Releases	Feature Information
Multi-tenancy	7.2(0)D1(1)	Included a new chapter on *Multi-tenancy* . Multi-tenant data center handles the traffic segregation between different tenants.
Segment ID	7.2(0)D1(1)	Included a new section on *VN-Segment*. VN-Segment network can support up to 16 million virtual network segments.

Multi-tenancy

Multi-tenancy is a concept that refers to the logical isolation of shared virtual compute, storage, and network resources. In multi-tenant data center, tenants subscribe to virtual data center (VDC), and based on the services hosted by the tenants I within the virtual data center, each virtual data center can have multiple VN-Segments.

Multi-tenancy

Figure 1: Multi-tenant Data Center



The above figure depicts two virtual data centers assigned to different tenants. For example Coke and Pepsi, each virtual data center has virtual data center elements like virtual machines (VM), storage inter-connected by a VN-Segment.

Multi-tenant data center handles the traffic segregation between different tenants, and also within tenant traffic, for security and privacy. Data centers have deployed VLANs to isolate the machines of different tenants on a single Layer-2 network. This could be extended to the virtualized data centers by having the hypervisor encapsulate VM packets with a VLAN tag corresponding to the VM owners. This approach provides a Layer-2 abstraction to the tenants and, with VRF, it can completely virtualize the Layer-2 and Layer-3 address spaces. However, the VLAN is a 12-bit field in the VLAN header, limiting this to at most 4K tenants. Also, multi-tenant network should provide tenants with simple and flexible network abstractions, by completely and efficiently virtualizing the address space at both Layer-2 and Layer-3 for each tenant, without any restrictions on the tenant's choice of Layer-2 or Layer-3 addresses. Also, tenants might want to extend their IT services or storage network which uses non-IP protocols such as Fibre Channel over Ethernet (FCOE). These protocols may be important for tenants trying to move the existing applications into service provider data center (SPDC) and does not support in a network that has no Layer-2 abstraction. Similarly, these tenants will benefit from the SPDC that supports tenant-level broadcast or multicast trees. In order to maximize the benefits of resource sharing, which provides multiplexing to achieve better resource efficiency and cost saving, multi-tenant data centers must scale to larger size to accommodate more tenants and VMs. Maintaining such large multi-tenant data centers can be expensive and hence multi-tenant data centers require automated configuration and

management tools to reduce the cost. Also with the large scale Layer-2 multi-tenant data center needs high bi-sectional bandwidth and this can be achieved by using Layer-2 multi-pathing short path bridging technologies like FabricPath and TRILL, which also addresses the MAC address scale issues required for per-tenant Layer-2 abstraction.

Another important requirement for multi-tenant data center is to support the mobility of VMs within and across SPDC, and also into enterprise data centers. Mobility within SPDC allows for dynamic tenant growth and maximizes resource utilization and sharing. For instance, if a tenant needs to add a VM to the existing SPDC POD but all the servers are overloaded then the VM for the tenant can be accommodated on another SPDC POD, which has the capacity and is available in server. This means that the VN-Segment must be able to extend virtually anywhere within and across multi-tenant data center.

Bridge-Domain

Note This section is applicable only for multi-tenancy full version.

A bridge-domain is a generic object that represents a Layer-2 broadcast domain on a device. Either a VLAN or a bridge-domain with the same number can exist. The bridge-domain range needs to be carved out from the 4096 VLAN range. The reserved VLANs cannot be used as a bridge-domain. All the carved out bridge-domain can be used as user/tenant bridge-domain.

The following is an example to carve out the bridge-domain range:

```
system bridge-domain 10-3000
```

Given above is the entire set of bridge-domains that can be used on the switch. For bridge-domain to be used for different VRFs you need to define a fabric bridge-domain range. Out of this range of user bridge-domains, a subset of bridge-domains can be designated as fabric bridge-domains. The corresponding BDIs will be reserved as fabric BDIs.

The following example shows allocating fabric bridge-domains:

```
system fabric bridge-domain 2001-3000
```

This will designate bridge-domains 2001-3000 to be used as fabric bridge-domains. Fabric bridge-domains are used as part of applying the vrf-tenant-profile. The remaining bridge-domains (10-2000) are user bridge-domains. They will be used to map tenant VNIs on the switch.

Note Do not create, delete, or edit a bridge domain in the fabric bridge domain range. These are created whenever a new VRF is created and is removed when the VRF is removed.

A fabric-control bridge-domain is configured from the range of user bridge-domains only (in this case 10-2000). The fabric control bridge-domain/VLAN needs to be defined for control traffic to propagate. There can only be one fabric control bridge-domain or a VLAN in the system.

Note Use of VLAN 1 as fabric control is not allowed.

Configuring Bridge-Domain

SUMMARY STEPS

1. **configure terminal**
2. **[no] system bridge-domain { bd-list | add bd-list | all | except bd-list | none | remove bd-list }**
3. **[no] system fabric bridge-domain { bd-list | add bd-list | all | except bd-list | none | remove bd-list }**
4. **[no] bridge-domain {bd-id | bd-range}**
5. **[no] fabric-control**
6. **show bridge-domain summary**
7. **show bridge-domain id**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters configuration mode.
Step 2	**[no] system bridge-domain { bd-list | add bd-list | all | except bd-list | none | remove bd-list }** **Example:** `switch(config)# system bridge-domain add 100-200`	Identifies the IDs that are available for bridge-domain configurations. • The valid range for the ID argument is from 2 to 3967. • (Optional) The id keyword and argument combination identifies the last ID in a range of contiguous IDs. The hyphen (-) is mandatory. • (Optional) The arguments like add, remove, all, except, none can be used for adding, removing, adding all, adding all except and removing all respectively.
Step 3	**[no] system fabric bridge-domain { bd-list | add bd-list | all | except bd-list | none | remove bd-list }** **Example:** `switch(config)# system fabric bridge-domain 151-200`	Identifies the IDs that are available for fabric bridge-domain configuration. This command has same option as the previous command but the range it can act on is only the existing system bridge-domain carved out range.

Step 4	**[no] bridge-domain {bd-id | bd-range}**
	Example:
	`switch(config)# bridge-domain 100-110`
	`switch(config-bdomain)#`
		Enters bridge-domain configuration mode and configures a bridge-domain. The domain-ID argument is a unique identifier for the bridge-domain and underlying VLAN to be created. The valid range is defined by the system bridge-domain configuration.
		Note You can use the no form of this command to remove the bridge-domain configuration including port associations. Removing the bridge-domain configuration does remove the underlying VLAN and all the bridge-domain properties.

	Command or Action	Purpose

Step 5	**[no] fabric-control**
	Example:
	`switch(config)# bridge-domain 100`
	`switch(config-bdomain)# fabric-control`
		Make the bridge-domain as the fabric control bridge-domain. Only one bridge-domain or a VLAN can be configured as fabric control.

Step 6	**show bridge-domain summary** **Example:** `switch# show bridge-domain summary`	(Optional) To show the bridge-domain configuration. Similar to *show vlan summary*.
Step 7	**show bridge-domain id** **Example:** `switch# show bridge-domain 100`	(Optional) To show whether the bridge-domain is created or not. Also to show any bridge-domain property configured under it.
Step 8	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Example

The following example shows how to create a bridge-domain:

```
switch# configure terminal
switch(config)# system bridge-domain 100-200
switch(config)# bridge-domain 100
switch(config-bdomain)# name Cisco:tenant1
switch(config-bdomain)# no shutdown
switch(config-bdomain)# exit
switch(config)#
switch(config)# bridge-domain 101
switch(config-bdomain)# fabric-control
switch(config-bdomain)# name fabric-control_BD
switch(config-bdomain)# no shutdown
switch(config-bdomain)# exit
```

VN-Segment

VN-Segment network can support up to 16 million virtual network segments (also called Virtual Network Identifiers) and VN-Segment has global significance in Layer-2 network. In multi-tenant applications, tenant traffic can still be received as “Dot1Q” tagged that need to be classified to the VN-Segment assigned to those tenants. VN-Segment is the extension of VLANs – both need to coexist. VLAN range is from 1-4095 and VN-Segment (VNI) range is from 4096-16 Million.

Note For release 7.2(0)N1(1), to modify the VN-Segment of a VLAN, you must delete any existing VN-Segment mapping to add the new VN-Segment mapping.

Configuring VN-Segment

SUMMARY STEPS

1. **configure terminal**
2. **feature vni**
3. **vni <vni range>**
4. **shutdown/no shutdown vni**
5. **member vni <vni-range>**
6. **encapsulation profile vni <profile-name>**
7. **service instance vni**
8. **shutdown/no shutdown vsi**
9. **encapsulation profile vsi**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters configuration mode.
Step 2	**feature vni** **Example:** `switch(config)# feature vni`	Enables feature VNI or Segmentation.
Step 3	**vni <vni range>** **Example:** `switch(config)# vni 5000-5002, 5005`	Creates a range of VNIs.

Step 4	**shutdown/no shutdown vni**
	Example:
	`Switch(config)# vni 5000-5001`
	`switch(config-vni)# [no] shutdown`
		Shuts down a range of VNIs.

Step 5	**member vni <vni-range>**
	Example:
	`switch(config)# bridge-domain 10-12`
	`switch(config-bdomain)# [no] member vni 5000-5002`
		Configures VNIs as members under a range of bridge-domains.

	Command or Action	Purpose

Step 6	**encapsulation profile vni <profile-name>**
	Example:
	`switch(config)# [no] encapsulation profile vni cisco`
	`switch(config-vni-encap-prof)# [no] dot1q 20 vni 5000`
		Creates an encapsulation profile named *cisco* with dot1q 20 mapped to vni 5000.

Step 7	**service instance vni**
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)# service instance 1 vni`
	
	`switch(config)# interface ethernet 3/2`
	`switch(config-if)# service instance vni default`
		Creates a numbered VSI under parent port interface Ethernet 3/1 and 3/2.

Step 8	**shutdown/no shutdown vsi**
	Example:
	`switch(config-if)# service instance 1 vni`
	`switch(config-if-srv)# [no] shut`
		Shuts a numbered VSI.

Step 9	**encapsulation profile vsi**
	Example:
	`switch(config-if)# service instance 1 vni`
	`switch(config-if-srv-def)# encapsulation profile cisco default`
		Applies the encapsulation profile to a VSI.

Step 10	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Detailed Steps

The 'Feature vni' or segmentation can be enabled only when the virtual device context has been limited to F3.

```
switch(config)# feature vni
Feature vni requires F3 or newer linecards
switch(config)# vdc switch
switch(config-vdc)# limit-resource module-type f3
This will cause all ports of unallowed types to be removed from this vdc. Continue (y/n)?
[yes] yes
switch(config-vdc)# feature vni
```

There has to be 1:1 mapping between VNI and bridge-domain. VNI has global significance in the Layer-2 network while bridge-domains remain local to the virtual data center (switch). Bridge-domains would have VNIs as members.

Commands to create a VNI and adding the VNI under a bridge-domain.

```
switch(config)# [no] vni 5000-5002
switch(config-vni)# [no] shutdown
```

```
switch(config)# bridge-domain 50-52
switch(config-bdomain)# [no] member vni 5000-5002
```

Existing legacy IEEE 802.1Q switches and End-host/Servers, capable of sending dot1q tagged traffic, should be able to connect to VN-Segment supported network. This capability is provided by **VN-Segment Service Instance (VSI)**. VN-Segment Service Instance Ports on the VN-Segment capable switch allows to map the dot1q tagged frames received on that port uniquely to a VN-Segment (VNI).

An encapsulation profile like a template needs to be created to define the dot1q to VNI mappings.

Command to create an encapsulation profile template named ***cisco*** and add/delete a dot1q to VNI mapping under it.

```
switch(config)# [no] encapsulation profile vni cisco
switch(config-vni-encap-prof)# [no] dot1q 20 vni 5000
```

Command to create an encapsulation profile template named ***cisco*** and add/delete the untagged frame VNI mapping under it.

```
switch(config)# [no] encapsulation profile vni cisco
switch(config-vni-encap-prof)# [no] untagged vni 6000
```

There are two types of VSIs - Numbered VSI and Default VSI. VSIs can be created under a physical port or a port channel. Numbered VSI range is from 1-4094 while 4095 VSI ID is reserved for default VSI. The default VSIs are by default set to admin up always. Note that a default VSI and a numbered VSI cannot exist together under the same parent port. Multiple numbered VSIs can be created under same parent port.

Command to create a numbered VSI and apply encapsulation profile under it.

```
switch(config)# interface ethernet3/1
switch(config-if)# service instance 1 vni
switch(config-if-srv)# no shut
switch(config-if-srv)# encapsulation profile cisco default
```

Command to create a default VSI with ***cisco*** as the encapsulation profile.

```
switch(config)# interface ethernet3/2
switch(config-if)# service instance vni default
switch(config-if-srv-def)# encapsulation profile cisco default
```

Sample VNI & VSI configuration:

```
switch(config)# vni 5000-5002
switch(config-vni)# no shutdown
switch(config-vni)# exit
switch(config)# bridge-domain 50-52
switch(config-bdomain)# member vni 5000-5002
switch(config-bdomain)# exit
switch(config)# encapsulation profile vni cisco
switch(config-vni-encap-prof)# dot1q 20-22  vni  5000-5002
switch(config-vni-encap-prof)# exit
switch(config)# interface ethernet9/1
switch(config-if)# no shutdown
switch(config-if)# service instance vni default
switch(config-if-srv-def)# encapsulation profile cisco default
```

Bridge-Domain Interface

A bridge-domain interface (BDI), is a virtual routed interface that connects a bridge-domain on the device to the Layer-3 router engine on the same device. Only one BDI can be associated with a bridge-domain. You must configure a BDI for a bridge-domain only when you want to route between bridge-domains or to provide IP host connectivity to the device through a virtual routing and forwarding (VRF) instance that is not the management VRF.

- You must enable the VLAN network interface feature before you can configure it.
- You must configure the BDI in the same virtual device context as the bridge-domain.
- You must create the bridge-domain range in the virtual device context, and BDI can only be created for that range. The configurations under a BDI are same as that under VLAN interface.
- You can route across BDI to provide Layer-3 inter-bridge-domain routing by configuring a BDI for each bridge-domain that you want to route traffic to and assigning an IP address on the BDI.

Configuring Bridge-Domain Interface

Before you begin

- Ensure that you are in the correct virtual data center (or use the **switchto vdc** command)

SUMMARY STEPS

1. **configure terminal**
2. **feature interface-vlan**
3. **interface bdi**
4. **ip address**
5. **ipv6 address**
6. **show interface bdi**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters configuration mode.
Step 2	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables BDI mode.
Step 3	**interface bdi** **Example:**	Creates a BDI. The *number* range specified in system bridge-domain command.

	Command or Action	Purpose
	`switch(config)# interface bdi 10`	
Step 4	**ip address** **Example:** `switch(config-if)# ip address 192.0.2.1/8`	Configures an IP address for this BDI.
Step 5	**ipv6 address** **Example:** `switch(config-if)# ipv6 address 2001:0DB8::1/8`	Configures an IPv6 address for this BDI.
Step 6	**show interface bdi** **Example:** `switch(config-if)# show interface vlan 10`	(Optional) Displays the Layer-3 interface statistics.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Example

The following example shows how to create a BDI:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface bdi 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Configuring Multiple Leaf

The following example shows the multi-tenancy support at leaf using VRFs:

```
system bridge-domain 2-3967
system fabric bridge-domain 3001-3967

configure profile vrf-tenant-profile
 vni $vrfSegmentId
 bridge-domain $bridgeDomainId
  member vni $vrfSegmentId
 interface bdi $bridgeDomainId
  vrf member $vrfName
  ip forward
  ipv6 forward
  no shutdown
configure terminal

bridge-domain 2,10-11
```

```
bridge-domain 2
 fabric-control
bridge-domain 2,10-11
 member vni 5000,10010-10011

vrf context Cisco:vrf1
 vni 20000
 ipv6 pim ssm range ff30::/12
 rd auto
 address-family ipv4 unicast
  route-target both auto
 address-family ipv6 unicast
  route-target both auto

interface Bdi10
 no shutdown
 vrf member Cisco:vrf1
 ip address 100.1.1.1/24
 fabric forwarding mode anycast-gateway

interface Bdi11
 no shutdown
 vrf member Cisco:vrf1
 ip address 100.1.2.1/24
 fabric forwarding mode proxy-gateway
```