

Technical Overview of Virtual Device Contexts

The Cisco® Nexus 7000 Series Switches introduce support for the Cisco NX-OS Software platform, a new class of operating system designed for data centers. Based on the Cisco MDS 9000 SAN-OS platform, Cisco NX-OS introduces support for virtual device contexts (VDCs), which allows the switches to be virtualized at the device level. Each configured VDC presents itself as a unique device to connected users within the framework of that physical switch. The VDC runs as a separate logical entity within the switch, maintaining its own unique set of running software processes, having its own configuration, and being managed by a separate administrator.

This document provides an insight into the support for VDCs on Cisco NX-OS.

1. Introduction to Cisco NX-OS and Virtual Device Contexts

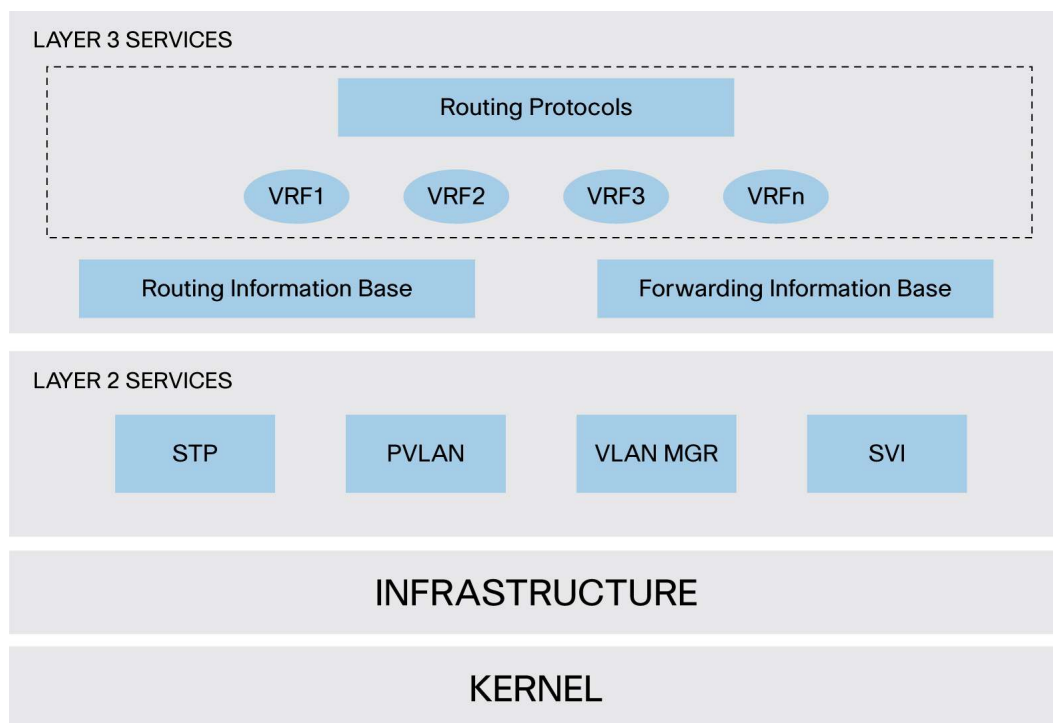
Cisco NX-OS is based on Cisco MDS 9000 SAN-OS and has been developed for data center deployments. It incorporates all of the essential Layer 2 and 3 protocols and other key features found in Cisco IOS® Software. Features such as Cisco In Service Software Upgrades (ISSU), superior fault detection, and isolation mechanisms such as Cisco Generic Online Diagnostics (GOLD) and Embedded Event Manager (EEM) as well as a traditional Cisco IOS Software look and feel for configuration purposes are all included. Cisco NX-OS is based on a modular architecture that supports the independent starting and stopping of processes, and supports multi-threaded processes that run in their own protected memory space.

The Cisco Nexus 7000 Series inherits a number of virtualization technologies present in Cisco IOS Software. From a Layer 2 perspective, virtual LANs (VLAN) virtualize bridge domains in the Nexus 7000 chassis. Virtualization support for Layer 3 is supported through the concept of virtual route forwarding instances (VRF). A VRF can be used to virtualize the Layer 3 forwarding and routing tables. The virtualization aspect of the Cisco NX-OS Software platform has been extended to support the notion of virtual device contexts (VDCs). A VDC can be used to virtualize the device itself, presenting the physical switch as multiple logical devices. Within that VDC it can contain its own unique and independent set of VLANs and VRFs. Each VDC can have assigned to it physical ports, thus allowing for the hardware data plane to be virtualized as well. Within each VDC, a separate management domain can manage the VDC itself, thus allowing the management plane itself to also be virtualized.

The definition of a switch control plane includes all those software functions that are processed by the switch CPU (found on the central supervisor). The control plane supports a number of crucial software processes such as the routing information base, the running of various Layer 2 and Layer 3 protocols, and more. All of these processes are important to the switches interaction with other network nodes. The control plane is also responsible for programming the data plane that enables many hardware-accelerated features.

In its default state, the switch control plane runs a single device context (called VDC 1) within which it will run approximately 80 processes. Some of these processes can have other threads spawned, resulting in as many as 250 processes actively running on the system at a time depending on the services configured. This single device context has a number of layer 2 and 3 services running on top of the infrastructure and kernel components of the OS as shown in the following diagram.

Figure 1. Default Operating Mode with Single Default VDC



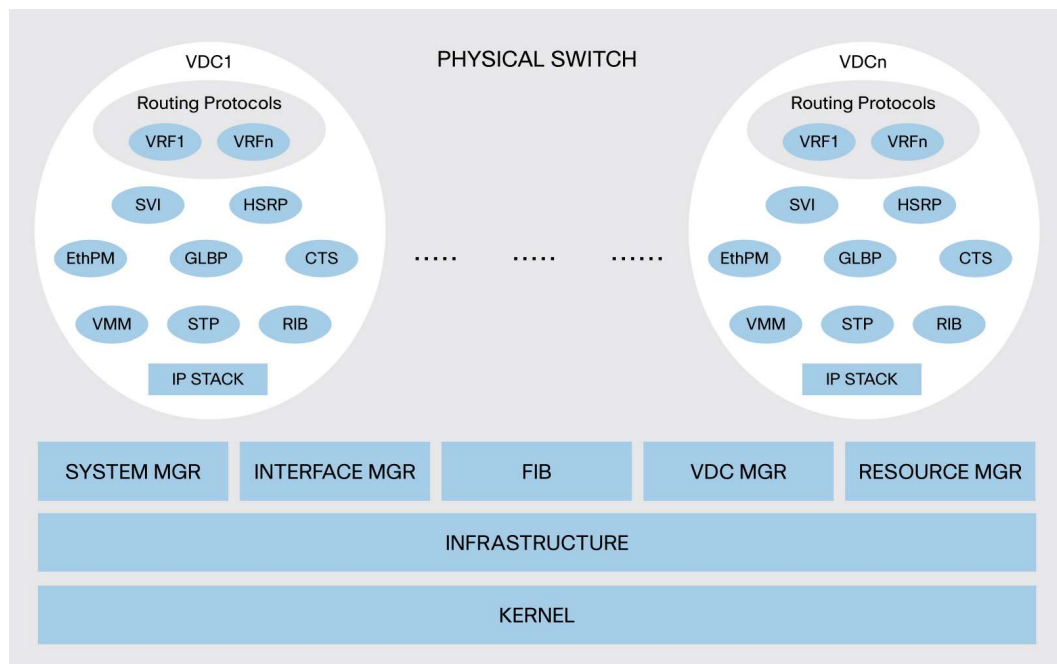
This collection of processes constitutes what is seen as the control plane for a single physical device (that being with no other virtual device contexts enabled). VDC 1 is always active, always enabled, and can never be deleted. It is important to reiterate that even in this default mode, virtualization support via VRF and VLAN is still applicable within the default VDC (or any VDC). This can also be viewed as virtualization nesting. At the higher level, you have the VDC. Within the VDC, you can have multiple VRFs and VDCs. In future software releases, you could potentially have further nested levels with features like MTR within a VRF.

The notion of enabling a subsequent (additional) VDC takes these processes and replicates it for each device context that exists in the switch. When this occurs, duplication of VRF names and VLAN IDs is possible. For example, you could have a VRF called manufacturing in one device context and the same “manufacturing” name applied to a VRF in another virtual device context. Hence, each VDC administrator essentially interfaces with its own set of processes and its own set of VRFs and VLANs, which in turn, represents its own logical (or virtual) switch context. This provides a clear delineation of management contexts and forms the basis for configuration separation and independence between VDCs.

Figure 2 depicts the major software elements that enable VDCs. In VDC mode, many benefits can be achieved such as per-VDC fault isolation, per-VDC administration, separation of data traffic,

and enhanced security. Hardware resources such as physical interfaces can also be divided between VDCs. Support for hardware partitioning is discussed later in this document.

Figure 2. VDC Mode



The use of VDC opens up a number of use cases that can provide added benefits for the administrators of this switch. Use cases for VDCs could include:

- Offering a secure network partition between different user departments traffic
- Provides empowered departments the ability to administer and maintain their own configurations
- One key use for VDC is to provide a device context for testing new configuration or connectivity options without impacting production systems
- Consolidation of multiple departments switch platforms into a single physical platform while still offering independence from the OS, administration and traffic perspective
- Use of a device context for network administrator and operator training purposes

2. VDC Architecture

The Cisco NX-OS Software platform provides the base upon which virtual device contexts are supported. The following sections provide more insight into the support for VDCs within this software platform.

2.1 VDC Architectural Layers

In analyzing the architectural diagram of the system running in VDC mode (see Figure 2 above), it becomes apparent that not all of the architectural elements of the platform are virtualized. Even though not all layers are virtualized, all of the major components have been built with the purpose to support the concept of VDCs.

At the heart of the OS is the kernel and infrastructure layer. The kernel is able to support all processes and all VDCs that run on the switch but only a single instance of the kernel will exist at any one point in time. The infrastructure layer provides an interface between the higher layer processes and the hardware resources of the physical switch (TCAM, etc.). Having a single instance of this layer reduces complexity (when managing the hardware resources). Having a single infrastructure layer also helps scale performance by avoiding duplication of this system's management process.

Working under control of the infrastructure layer is a number of other important system processes, which also exist as a unique entity. Of these, the VDC manager is a key process when it comes to supporting VDCs. The VDC manager is responsible for the creation and deletion of VDCs. More important, it provides VDC-related APIs for other infrastructure components such as the system manager and resource manager to perform their own related functions.

When a VDC is created, the system manager is responsible for launching all services required for VDC startup that run on a per-VDC basis. As new services are configured, the system manager will launch the appropriate process. For example, if OSPF were enabled in the VDC named Marketing, then the system manager would launch an OSPF process for that VDC. If a VDC is deleted, then the system manager is responsible for tearing down all related processes for that VDC.

The resources manager is responsible for managing the allocation and distribution of resources between VDCs. More about this topic is discussed later in this document, but resources such as VLANs, VRFs, PortChannels, and physical ports are examples of resources that are managed by the resource manager.

Sitting above the infrastructure layer and its associated managers are processes that run on a per-VDC basis. Each of these processes runs in its own set of protected memory space. Fault isolation is one of the main benefits derived from the use of a VDC. Should a process fail in one VDC, it will not have an effect on processes running in another VDC.

2.2 Virtual Device Context Resource Allocation

When a VDC is created, selected switch resources can be allocated to that VDC, helping ensure that it has exclusive use of that resource. A resource template controls the allocation of resources and defines how much of that resource can be allocated to a VDC. VDC administrators and users within the VDC cannot change this template. Only the super user is able to change this template. The super user is a user with the highest level of authority to invoke changes to the configuration in the switch. The super user exists within the domain of VDC 1 (the default VDC). Aside from being able to assign physical switch resources between device contexts, the super user has the ability to invoke change in any VDC configuration, create and delete VDCs, and create and delete administrators and users for each VDC. The template is administered separately from the switch configuration, which allows the super user to edit a template without affecting the resource allocation for the VDC that the template has previously been applied to. When the super user wants to apply an updated template to a VDC, that person will have to reapply the template so that it is activated, copied, and merged with the running configuration. For transparent backup purposes, merging templates with the configuration adds the benefit of being able to back up all configurations specific to a VDC by simply backing up the primary configuration. Most, but not all, of the switch resources can be allocated to a VDC. Table 1 lists the resources that can be allocated to VDCs and those that cannot be.

Table 1. Switch Resources

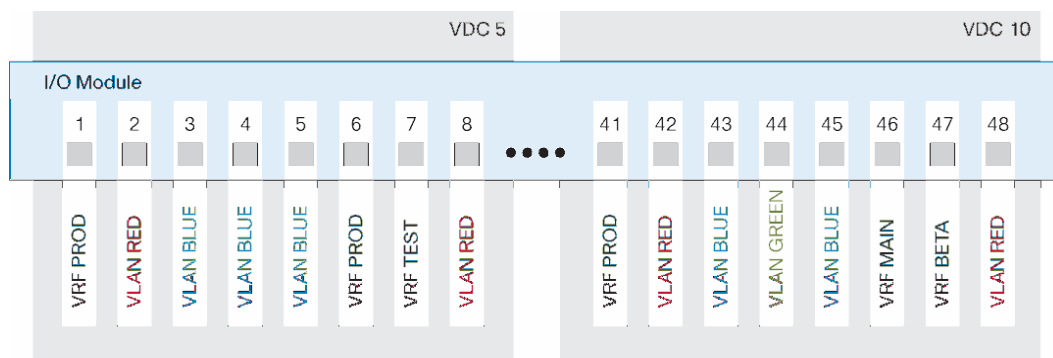
Switch Resources that Can Be Allocated to a VDC	Switch Resources that Cannot Be Allocated to a VDC
Physical Interfaces, PortChannels, Bridge Domains and VLANs, HSRP and GLBP Group IDs, and SPAN	CPU*, Memory*, TCAM Resources such as the FIB, QoS, and Security ACLs * Future releases may allow allocation of CPU or memory to a VDC.

For many of the resources that can be allocated on a per-VDC basis, it is important to note that the resource is typically managed globally. For example, there are 256 Cisco EtherChannel[®] link bundles per chassis that can be apportioned across the active VDCs. If those 256 Cisco EtherChannel link bundles are consumed by two VDCs, then there are no more link bundles available for any other VDC in the chassis. Configuring the load balancing option for Cisco EtherChannel is a global configuration and cannot be set on a per-VDC basis. Another example is SPAN. There are two SPAN sessions available for the switch. Both VDC A and VDC B can configure a SPAN session as “monitor session 1”; however, internal to the hardware they are recognized as different SPAN sessions. This is again as a direct result of SPAN being managed as a global resource. As with the link bundle example above, once these two sessions are consumed, there are no more SPAN sessions available for other VDCs.

The creation of a VDC builds a logical representation of the Cisco Nexus 7000 Series Switch, albeit initially with no physical interfaces assigned to it. Physical switch ports are resources that cannot be shared between VDCs. By default, all ports on the switch are assigned to the default VDC (VDC 1). When a new VDC is created, the super user is required to assign a set of physical ports from the default VDC to the newly created VDC, providing the new VDC with a means to communicate with other devices on the network. Once a physical port is assigned to a VDC, it is bound exclusively to that VDC, and no other VDC has access to that port. Inter-VDC communication is not facilitated from within the switch. A discrete external connection must be made between ports of different VDCs to allow communication between them.

Different port types can be assigned to a VDC. These include Layer 2 ports, Layer 3 ports, Layer 2 trunk ports, and PortChannel (Cisco EtherChannel) ports. Note that the ports on the 32 port 10G I/O Module (N7K-M132XP-12) must be allocated to a VDC in port groupings of four ports. The ports on the 48 port 10/100/1000 I/O Module (N7K-M148GT-11) can be allocated on a per-port basis. Logical interfaces such as SVIs that are associated with the same physical interface cannot be allocated to different VDCs in the current implementation. Thus, it is not possible to virtualize a physical interface and associate the resulting logical interfaces to different VDCs. However, it is possible to virtualize a physical interface and associate the resulting logical interfaces with different VRFs or VLANs. Thus, VDCs can be assigned physical interfaces, while VLANs and VRFs can be assigned logical and physical interfaces.

An example of this can be seen in Figure 3. Ports 1 through 8 belong to VDC 5 while ports 41 through 48 belong to VDC 10. Within each VDC, ports are further virtualized belonging to either a VLAN or VRF.

Figure 3. VDC Virtualization

After a port is allocated to a VDC by the root-admin at default-VDC level, it is up to the VDC-admin to manage (configure and use) the previously assigned port. Running other related commands such as a show interface command, for example, will allow the user to see only those interfaces that have been assigned to the VDC.

Each VDC maintains its own configuration file, reflecting the actual configuration of ports under the control of the VDC. In addition, the local configuration will contain any VDC specific configuration elements such as a VDC user role and the command scope allocated to that user. Having a separate configuration file per VDC also provides a level of security that protects this VDC from operation configuration changes that might be made on another VDC.

VLANs are another important resource that has been extended in Cisco NX-OS. Up to 16,384 VLANs, defined across multiple VDCs, are supported in a Cisco Nexus 7000 Series Switch. Each VDC supports a maximum of 4096 VLANs as per the IEEE 802.1q standard. The new extended VLAN support will allow an incoming VLAN to be mapped to a per-VDC VLAN. We will refer to this per-VDC VLAN as a bridge domain for clarity. In this manner, a VDC administrator can create VLANs numbered anywhere in the 802.1q VLAN ID range, and the VDC will map that newly created VLAN to one of the default 16,384 bridge domains available in the switch. This will allow VDCs on the same physical switch to reuse VLAN IDs in their configurations, while at the OS level the VLAN is in fact a unique bridge domain within the context of the physical switch. For example, VDC A and VDC B could both create VLAN 100, which in turn could be mapped to bridge domains 250 and 251, respectively. Each administrator would use show commands that reference VLAN 100 to monitor and manage that VLAN.

3. Scaling Nexus 7000 Switch Resources Using Virtual Device Contexts

Each line card uses a local hardware-forwarding engine to perform Layer 2 and Layer 3 forwarding in hardware. The use of VDCs enables this local forwarding engine to optimize the use of its resources for both Layer 2 and Layer 3 operations. The following section provides more detail on this topic.

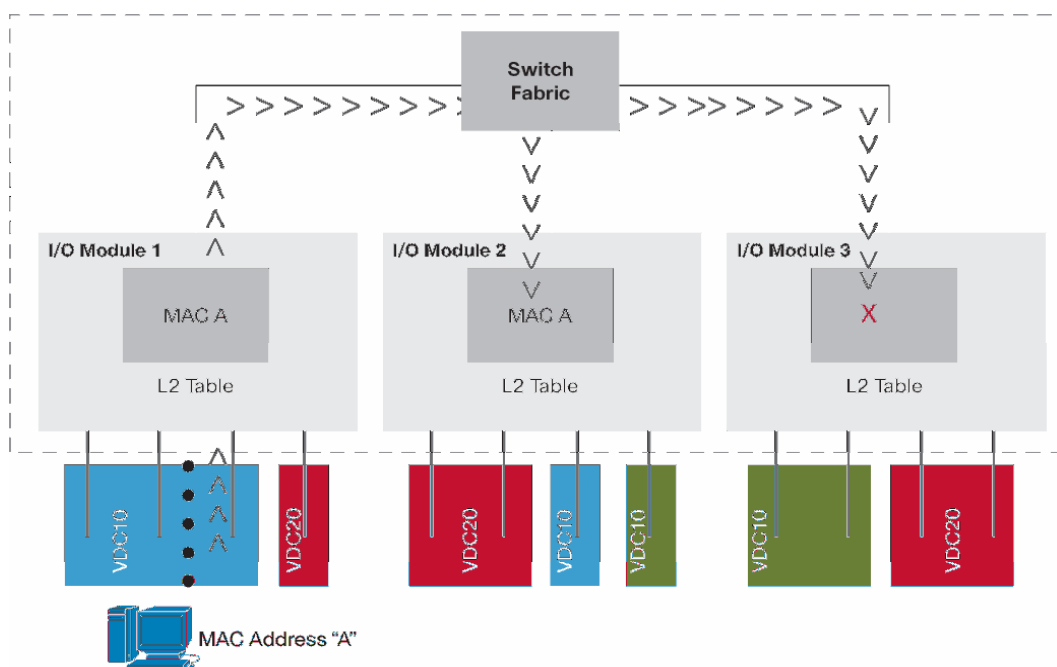
3.1 Layer 2 Address Learning with Virtual Device Contexts

The forwarding engine on each line card is responsible for Layer 2 address learning and will maintain a local copy of the Layer 2 forwarding table. The MAC address table on each line card supports 128,000 MAC addresses. When a new MAC address is learned by a line card, it will forward a copy of that MAC address to other line cards. This enables the Layer 2 address learning process to be synchronized across line cards.

Layer 2 learning is a VDC local process and as such has a direct effect on what addresses are placed into each line card.

Figure 4 shows how the distributed Layer 2 learning process is affected by the presence of VDCs. On line card 1, MAC address A is learned from port 1/2. This address is installed in the local Layer 2 forwarding table of line card 1. The MAC address is then forwarded to both line cards 2 and 3. As line card 3 has no ports that belong to VDC 10, it will not install any MAC addresses learnt from that VDC. Line card 2, however, does have a local port in VDC 10, so it will install MAC address A into its local forwarding tables.

Figure 4. MAC Address Learning



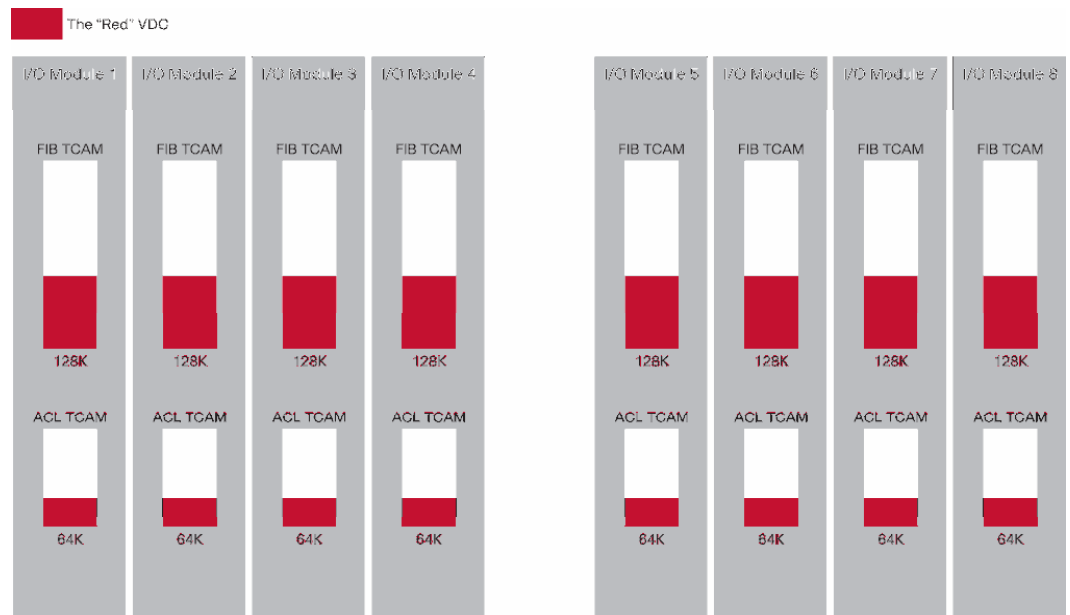
With this implementation of Layer 2 learning, the Cisco Nexus 7000 Series Switch offers a way to scale the use of the Layer 2 MAC address table more efficiently when VDCs are unique to line cards.

3.2 Layer 3 Resources and Virtual Device Contexts

The forwarding engine on each line card supports 128,000 entries in the forwarding information base (used to store forwarding prefixes), 64,000 access control lists, and 512,000 ingress and 512,000 egress NetFlow entries.

When the default VDC is the only active VDC, learnt routes and ACLs are loaded into each line card TCAM tables so that each line card has the necessary information local to it to make an informed forwarding decision. This can be seen in Figure 5, where the routes for the default "red" VDC are present in the FIB and ACL TCAMs.

Figure 5. Default Resource Allocation



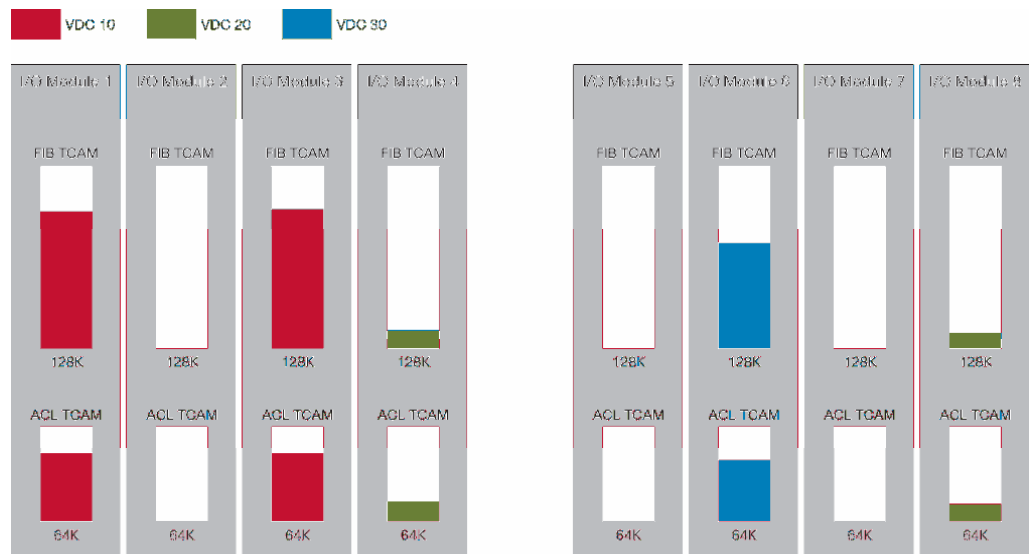
When physical port resources are split between VDCs, then only the line cards that are associated with that VDC are required to store forwarding information and associated ACLs. In this way, the resources can be scaled beyond the default system limits seen in the preceding example. An example is defined in Table 2.

Table 2. Resource Separation Example

VDC	Number of Routes	Number of Access Control Entries (ACE)	Allocated Line Cards
10	100,000	50,000	LC 1, LC 3
20	10,000	10,000	LC 4, LC 8
30	70,000	40,000	LC 6

In this example, the resources would be allocated as shown in Figure 6.

Figure 6. Resource Split



The effect of allocating a subset of ports to a given VDC results in the FIB and ACL TCAM for the respective line cards being primed with the forwarding information and ACLs for that VDC. This extends the use of those TCAM resources beyond the simple system limit described earlier. In the preceding example, a total of 180,000 forwarding entries have been installed in a switch that, without VDCs, would have a system limit of 128,000 forwarding entries. Likewise, a total of 100,000 ACE's have been installed where a single VDC would only allow 64,000 Access Control Entries. More important, FIB and ACL TCAM space on line cards 2, 5, and 7 is free for use by additional VDCs that might be created. This further extends the use of those resources well beyond the defined system limits noted here.

As with the TCAMs for FIB and ACLs, the use of the NetFlow TCAM is more granular when multiple VDCs are active. Let us assume the same setup as before, where we have a set of line cards, each of which belong to a different VDC.

When a flow is identified, a flow record will be created on the local NetFlow TCAM resident on that line card. Both ingress and egress NetFlow are performed on the ingress line card so it is this ingress line card's NetFlow TCAM where the flow will be stored. The collection and export of flows is always done on a per-VDC basis. No flow in VDC X will be exported to a collector that is part of VDC Y. Once the flow is created in a NetFlow TCAM on line card X, it will not be replicated to NetFlow TCAMs on other line cards that are part of the same VDC. In this manner, the use of the TCAM is optimized.

4. Effect of Virtual Device Contexts on Control Plane Processes

The previous sections have highlighted that some system resources have global significance and are managed at the global level while other system resources do not have that global significance and are managed at the VDC level. As with these resources, there are certain control plane processes that, when enabled, have global or per-VDC implications. The following section provides some insight into the main control plane processes that have VDC relevance.

4.1 Control Plane Policing (CoPP)

The switch control plane controls the operation of the switch, and compromising this entity could affect the operation of the switch. Control plane policing provides a protection mechanism for the control plane by rate limiting the number of packets sent to the control plane for processing.

Control plane policing is enabled from the default VDC and runs only in the default VDC. Its application, however, is system wide, and any packet from any VDC directed to the control plane is subject to the control plane policer in place. In other words, there is not any VDC awareness that can be used in the policy to police traffic differently depending on the VDC it came from.

4.2 Quality of Service (QoS)

Unlike control plane policing, quality of service has configuration implications that have either global or per-VDC significance.

Policers, which can be used to provide a rate limiting action on target traffic, are an example of a QoS resource specific to a VDC. Even though the policer is a systemwide resource, once configured it will have relevance only to the ports within that VDC. Any QoS configurations specific to a physical port also have significance only to the VDC that the port belongs to. An example of this type of QoS policy is Weighted RED (WRED), which is used to provide congestion management for port buffers.

Classification ACLs used to identify which traffic a QoS policy is going to be applied to are another example of a per-VDC resource. These ACLs are configured within a VDC and applied to ports in that VDC. More important, these QoS ACLs will be installed into the ACL TCAM only on those line cards that have ports in that VDC. In this instance, creating more than one VDC can have a positive effect on scaling TCAM resources beyond what is available for a single VDC.

Many of the QoS maps that are used to provide services such as CoS- and DCSP-to-queue mapping and markdown maps for exceed and violate traffic are examples of QoS configuration elements that have global significance. DSCP mutation maps, which offer a way to modify the ingress DSCP value, also have global significance. If these mutation- or CoS-to-queue maps are modified, they will affect packets entering all VDCs.

4.3 Embedded Event Manager (EEM)

Embedded Event Manager is an event-led subsystem that can automate actions based on a certain event occurring. When an event occurs, such as an OIR event or the generation of a certain syslog message, the system can invoke a user-written script (called an applet) that defines a preset list of actions to invoke.

An EEM policy (applet) is configured within the context of a VDC. While most events are seen by each VDC, there are some events that can be seen only from by the default VDC. An example of this event type is those events that are generated by the line cards themselves such as a Cisco GOLD diagnostic run result.

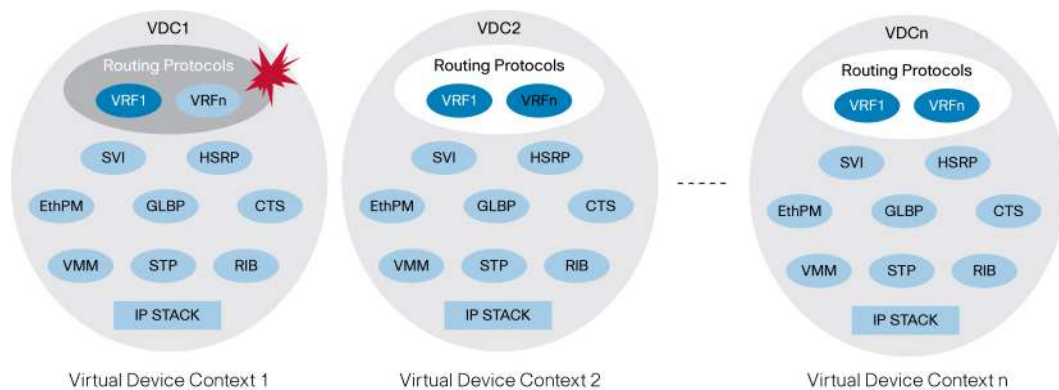
EEM maintains a log of EEM events and statistics, and this log is maintained on a per-VDC basis. When a policy is configured, only the VDC administrator can configure and administer that policy. VDC users are not able to administer EEM policies.

5. Virtual Device Context Fault Isolation

When multiple VDCs are created in a physical switch, inherently the architecture of the VDC provides a means to prevent failures within that VDC from affecting other VDCs. So, for instance, a spanning tree recalculation that might be started in one VDC is not going to affect the spanning tree domains of other VDCs in the same physical chassis. An OSPF process crash is another example where the fault is isolated locally to that VDC. Process isolation within a VDC thus plays an important role in fault isolation and serves as a major benefit for organizations that embrace the VDC concept.

As can be seen in Figure 7, a fault in a process running in VDC 1 does not affect any of the running processes in the other VDCs. Other equivalent processes will continue to run uninhibited by any problems associated with the faulty running process.

Figure 7. Per-VDC Fault Isolation



Fault isolation is enhanced with the ability to provide per-VDC debug commands. Per-VDC logging of messages via syslog is also another important characteristic of the VDC fault isolation capabilities. When combined, these two features provide a powerful tool for administrators in locate problems.

The creation of multiple VDCs also permits configuration isolation. Each VDC has its own unique configuration file that is stored separately in NVRAM. There are a number of resources in each VDC whose associated numbers and IDs can overlap between multiple VDCs without having an effect on another VDCs configuration. For example, the same VRF IDs, PortChannel numbers, VLAN IDs, and management IP address can exist on multiple VDCs. More important, configuration separation in this manner not only secures configurations between VDCs but also isolates a VDC from being affected by an erroneous configuration change in another VDC.

6. High Availability and Virtual Device Contexts

The Cisco NX-OS Software platform incorporates a high-availability feature set that helps ensure minimal or no effect on the data plane should the control plane fail. Different high-availability service levels are provided, from service restart to stateful supervisor switchover to ISSU without affecting data traffic.

Should a control plane failure occur, the administrator has a set of options that can be configured on a per-VDC basis defining what action will be taken regarding that VDC. There are three actions that can be configured: restart, bringdown, and reset. The restart option will delete the VDC and then re-create it with the running configuration. This configured action will occur regardless of whether there are dual supervisors or a single supervisor present in the chassis. The bringdown option will simply delete the VDC. The reset option will issue a reset for the active supervisor when there is only a single supervisor in the chassis. If dual supervisors are present, the reset option will force a supervisor switchover.

The default VDC always has a high-availability option of reset assigned to it. Subsequent VDCs created will have a default value of bringdown assigned to them. This value can be changed under configuration control.

Stateful switchover is supported with dual supervisors in the chassis. During the course of normal operation, the primary supervisor will constantly exchange and synchronize its state with the redundant supervisor. There is a software process (watchdog) that is used to monitor the responsiveness of the active (primary) supervisor. Should the primary supervisor fail, a fast switchover is enacted by the system. Failover occurs at both the control plane and data plane layers. At supervisor switchover, the data plane continues to use the Layer 2– and Layer 3–derived forwarding entries simply by maintaining the state written into the hardware. For the control plane,

the graceful restart process that is part of nonstop forwarding (NSF) is used to provide failover for Layer 3. For Layer 2, the control plane is maintained by locally stateful PSS mechanisms. This process provides for the following:

- Uninterrupted forwarding during a failover
- Rapid recovery from the failure to a stable operating state
- A nondisruptive recovery mechanism that will not render the network unstable during the recovery process

ISSU is another important aspect of high availability that has a direct effect on VDCs. ISSU allows the administrator to install and activate a new version of software in a chassis that is running two supervisors. The software upgrade can be applied to the backup supervisor, and then a switchover to that upgraded supervisor is invoked. The other supervisor is then upgraded with the same new set of software; all the while the system maintains data flow without interruption. At this juncture, ISSU cannot be applied on a per-VDC basis. The installed software on the chassis is applicable for all active VDCs.

First Hop Routing Protocols (FHRP) such as HSRP and GLBP are provided by the Cisco NX-OS Software platform. These services offer default gateway redundancy for attached hosts. Unlike ISSU, each FHRP service is available on a per-VDC basis. As part of creating the VDC, the administrator can define the number of FHRP groups that are available to each VDC.

7. Virtual Device Context Configuration

This section provides an overview of the steps involved in creating a VDC and assigning resources to it.

7.1 Initial VDC Setup

Configuration for VDC starts with the creation of a VDC. Up to four VDCs can exist in the system at one time. Given that there is always a default VDC active (VDC 1), this means up to three additional VDCs can be created from the CLI. Creating a VDC is done from configuration mode using the `vdc <name of vdc>` command as shown here:

```
switch# conf t
switch(config)# vdc production
switch(config-vdc)# show vdc
vdc_id  vdc_name                state                mac
-----  -
1       switch                        active               00:18:ba:d8:4c:3d
2       production                    active               00:18:ba:d8:4c:3e

switch(config-vdc)# show vdc detail
vdc id: 1
vdc name: switch
vdc state: active
vdc mac address: 00:18:ba:d8:4c:3d
vdc ha policy: RESET
```

```
vdc id: 2
vdc name: production
vdc state: active
vdc mac address: 00:18:ba:d8:4c:3e
vdc ha policy: BRINGDOWN
```

When the VDC has been created, the system places you into VDC configuration mode where further configuration options can be assigned to the VDC. A default set of configuration statements is assigned to the VDC when it is created as can be seen in the following output:

```
switch# show run | begin vdc
<snip>
vdc production id 2
  template default
  hap bringdown
  limit-resource vlan minimum 16 maximum 4094
  limit-resource span-ssn minimum 0 maximum 2
  limit-resource vrf minimum 16 maximum 8192
  limit-resource port-channel minimum 0 maximum 256
  limit-resource glbp_group minimum 0 maximum 4096
<snip>
```

These configuration statements provide a definition on the resource consumption that the VDC is allowed to work within. These resources include VLAN, VRF, SPAN, PortChannels, and GLBP group IDs. Resource limit assignments can, however, be changed via command line. An example of how a resource limit is changed is shown here:

```
switch(config)# vdc production
switch(config-vdc)# limit-resource vlan minimum 32 maximum 4094
switch(config-vdc)# show run | begin vdc
<snip>
vdc production id 2
  template default
  hap bringdown
  limit-resource vlan minimum 32 maximum 4094
  limit-resource span-ssn minimum 0 maximum 2
  limit-resource vrf minimum 16 maximum 8192
  limit-resource port-channel minimum 0 maximum 256
  limit-resource glbp_group minimum 0 maximum 4096
<snip>
```

This example shows how the minimum number of VLANs for the production VDC is changed from 16 to 32.

The use of resource templates provides an alternate way to assign resources to a VDC. Configuration of a resource template is performed in configuration mode as shown in the following example.

```
switch(config)# vdc resource template n7000switch
switch(config-vdc-template)# limit-resource vlan minimum 32 maximum 256
switch(config-vdc-template)# limit-resource vrf minimum 32 maximum 64
switch(config-vdc-template)# exit
```

Once the resource template is created it can be assigned to the VDC as shown in the following example.

```
switch(config)# vdc 2 template n7000switch
switch(config-vdc)# show vdc resource template
template::n7000switch
-----
Resource                Min           Max
-----
vrf                      32           64
vlan                     32           256

template ::default
-----
Resource                Min           Max
-----
glbp_group              0            4096
port-channel            0            256
span-ssn                0            2
vlan                    16           4094
vrf                     16           8192

switch(config-vdc)#
```

After the VDC is created, the administrator is placed into VDC configuration mode. The next task is to assign physical ports to this VDC. Ports on the physical line cards cannot be shared between different VDCs. By default, all physical ports belong to the default VDC. When a VDC is created, ports can be placed under the control of this VDC using the following CLI option:

```
switch(config)# show vdc membership
vdc_id: 1 vdc_name: switch interfaces:
Ethernet3/1            Ethernet3/2            Ethernet3/3
Ethernet3/4            Ethernet3/5            Ethernet3/6
Ethernet3/7            Ethernet3/8            Ethernet3/9
Ethernet3/10           Ethernet3/11           Ethernet3/12
Ethernet3/13           Ethernet3/14           Ethernet3/15
Ethernet3/16           Ethernet3/17           Ethernet3/1
Ethernet3/19           Ethernet3/20           Ethernet3/21
Ethernet3/22           Ethernet3/23           Ethernet3/24
Ethernet3/25           Ethernet3/26           Ethernet3/27
Ethernet3/28           Ethernet3/29           Ethernet3/30
Ethernet3/31           Ethernet3/32           Ethernet3/33
Ethernet3/34           Ethernet3/35           Ethernet3/36
Ethernet3/37           Ethernet3/38           Ethernet3/39
```

Ethernet3/40	Ethernet3/41	Ethernet3/42
Ethernet3/43	Ethernet3/44	Ethernet3/45
Ethernet3/46	Ethernet3/47	Ethernet3/48

```
vdc_id: 2 vdc_name: production interfaces:
```

```
switch(config)# vdc production
switch(config-vdc)# allocate interface ethernet 3/48
switch(config-vdc)# show vdc membership
```

```
vdc_id: 1 vdc_name: switch interfaces:
```

Ethernet3/1	Ethernet3/2	Ethernet3/3
Ethernet3/4	Ethernet3/5	Ethernet3/6
Ethernet3/7	Ethernet3/8	Ethernet3/9
Ethernet3/10	Ethernet3/11	Ethernet3/12
Ethernet3/13	Ethernet3/14	Ethernet3/15
Ethernet3/16	Ethernet3/17	Ethernet3/18
Ethernet3/19	Ethernet3/20	Ethernet3/21
Ethernet3/22	Ethernet3/23	Ethernet3/24
Ethernet3/25	Ethernet3/26	Ethernet3/27
Ethernet3/28	Ethernet3/29	Ethernet3/30
Ethernet3/31	Ethernet3/32	Ethernet3/33
Ethernet3/34	Ethernet3/35	Ethernet3/36
Ethernet3/37	Ethernet3/38	Ethernet3/39
Ethernet3/40	Ethernet3/41	Ethernet3/42
Ethernet3/43	Ethernet3/44	Ethernet3/45
Ethernet3/46	Ethernet3/47	

```
vdc_id: 2 vdc_name: production interfaces:
```

```
Ethernet3/48
```

The preceding example shows how a physical port (Ethernet 3/48) is moved under the control of the production VDC. Further configuration of this and other ports that are assigned to this VDC must now be completed from within the production VDC.

7.2 Switching between VDCs

After the VDC has been created, resource limits assigned and physical ports have been allocated, the administrator must session into, or switch to that VDC to perform additional configuration. From the default VDC CLI, the active VDCs can be seen using the following command:

```
switch# show vdc

vdc_id  vdc_name          state          mac
-----  -
1       switch             active         00:18:ba:d8:4c:3d
2       production         active         00:18:ba:d8:4c:3e
3       beta               active         00:18:ba:d8:4c:3f
```

The `switchto` command is used to allow the administrator to move between VDCs. When in the default VDC, the administrator can move to any VDC that is shown in the preceding VDC list. For example, switching to the engineering VDC is done as follows:

```
switch# switchto vdc ?
  production  VDC number 2
  beta        VDC number 3
  switch      VDC number 1
switch# switchto vdc production
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2007, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained herein are owned by
other third parties and are used and distributed under license.
Some parts of this software may be covered under the GNU Public
License or the GNU Lesser General Public License. A copy of
each such license is available at http://www.gnu.org/licenses/gpl.html
and http://www.gnu.org/licenses/lgpl.html
switch(vdc)# show vdc current-vdc
Current vdc is 2
switch(vdc)#
```

When the administrator switches from the default VDC to another VDC, the CLI prompt will change to reflect that the user has moved into a new VDC. Once in a VDC that is not the default VDC, restrictions will be applied regarding the capability to move between VDCs. Using the preceding example, from within the production VDC, the `switchto` command does not provide the same view as that seen from the default VDC. This is shown here:

```
switch(vdc)# show vdc current-vdc
Current vdc is 2
switch(vdc)# switchto vdc ?
  production  VDC number 2
switch(vdc)#
```

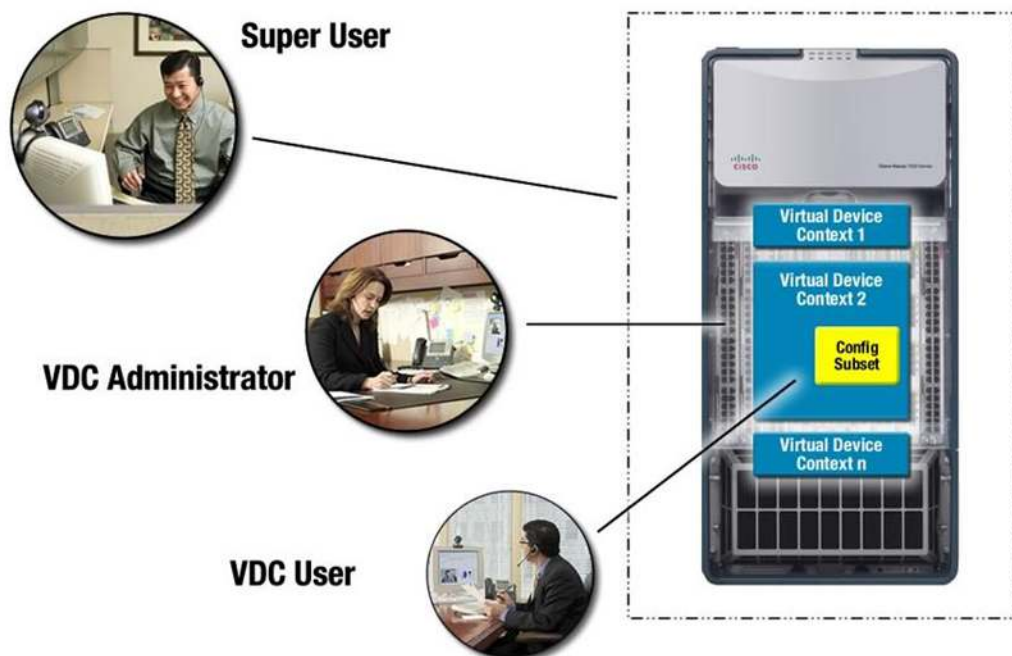
In the preceding output, it can be seen that a user within the production VDC has no `switchto` rights to move to another VDC, even though there are other active VDCs in the system. Once a user has switched into the VDC, they continue with the configuration to meet their requirements.

8. Virtual Device Context Administration

The VDC architecture defines a number of administrator levels that can be used to administer the physical switch and the VDCs. Each level defines access rights to configuration commands that can be invoked on the switch at both a global level and within a VDC. Commands outside the scope of a given user role are either hidden from the user's view or can return an error if the command is invoked.

There are three user levels as seen in the following figure—super user, VDC administrator and VDC user. Each of these user roles is discussed in more detail in the following sections.

Figure 8. VDC User Roles



Super user

At the top of the administrator tree is the super user role. When a Cisco Nexus 7000 Series Switch is booted for the first time, it will have a default VDC (VDC number 1) enabled. The administrator level used to configure the switch in this mode is, in essence, the same level afforded to the role of a super user. It is the super user that has the authority to create and delete additional VDCs. This level of administrator can invoke all global commands as well as assign physical switch ports to a nominated VDC. The super user has the authority to invoke other commands that have global scope. These commands can affect the operation of any VDC on the switch and include the ability to reload the entire system, modify global IP addresses (such as the management IP address), and configure boot image locations.

VDC Administrator

When a VDC is created, the super user would, in parallel, create a VDC administrator for that VDC. The VDC administrator is the second user type that can exist on the switch. Within the confines of the VDC, this user can make configuration changes to that VDC's configuration and save the VDC configuration independently of other VDCs. A VDC administrator can also have an administration scope across multiple VDCs. This user type cannot, however, perform any of the global configuration or physical resource allocation options that the super user role has access to. This user type also cannot create or delete a VDC, including the VDC that it administers.

VDC User

Within the boundary of the specific VDC, the VDC administrator can create a third level of user, the VDC user. A VDC user can log into the switch and invoke a subset of the configuration commands as defined by the VDC administrator. The VDC administrator defines the subset of allowed commands as part of a role. A VDC user will inherit the rights of the roles assigned by the VDC administrator. In total, up to 256 roles can be defined and active on the switch at any one time.

When multiple VDCs exist on a physical switch, the super user can, in fact, allow a VDC administrator and VDC user to access more than one VDC. Switching between VDCs is a supported feature of Cisco NX-OS and will allow valid users to use a switch to command to move between device contexts. Security is a crucial component of this feature and will require the switching user to re-authenticate when moving to a new VDC for the first time; subsequent moves would not require re-authentication as the state is maintained. Super users are not required to authenticate when moving between VDCs. Access rights to a VDC should not be confused with the broader role-based access control (RBAC). RBAC provides a means to define security permissions and associate them with a role. Users are given one or more roles, and as user data traverses the network, it is assigned a tag that allows the network device to determine the access rights of that user.

9. Summary

VDCs in the Cisco NX-OS Software platform extend virtualization support to enable true device virtualization. Virtualization of the physical switch offers a number of operational benefits such as greater fault isolation, leading to improved availability. Furthermore, traffic isolation within the confines of a VDC leads to greater security for user data. Numerous switch contexts within a physical switch can help scale physical switch resources across multiple logical groups within an organization. This can lead to both administrative efficiencies and lower operational costs.




Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

 Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)