

A WHITE PAPER

Cisco Routing Techniques

Migrate from Software to Hardware Routers



Topic Summary:

- Software routers easy, inexpensive to setup but fundamentally sub-optimal
- Hardware routers require planning and configuration / programming
- VLANs provide additional protection but require additional planning
- Hands-on Use Cases: Migrate from Software to Hardware Routers

Table of Contents

1.0	Introduction.....	1
2.0	[ENCLAVE] Routing	1
2.1	Desired Logical Outcome.....	1
2.2	Simplified Current Physical Connectivity.....	2
2.3	Use Case #1: Migrate a Software Router to Hardware.....	3
2.3.1	Software Router As-Is	3
2.3.2	Setup VLAN Definitions.....	5
2.3.3	Setup VLAN Trunks.....	7
2.3.4	Current Router Information	9
2.3.5	Migrate Router from Software to Hardware.....	12
2.3.6	Verify Routing Migration.....	16
2.4	Use Case #2: Migrate “Access” Subnet.....	17
3.0	Some Final Thoughts... ..	22
	About the Author.....	22

Illustration Index

Figure 1: Desired Routing Outcome.....	2
Figure 2: Current Routing Infrastructure	3

Table Index

Table 1: Switch Update Table	18
------------------------------------	----

1.0 Introduction

This paper is the result of study and hands-on application of Cisco routing techniques based on the pursuit of the Cisco Certified Network Associate (CCNA) certification. The paper is geared towards system administration and security administration professionals who encounter network routing questions and concerns as part of an organization's security posture. An effective network security posture demands a strong router infrastructure; this set of Cisco-oriented whitepapers aims to provide detailed network administration documentation and use cases to aid entry- and mid-level network engineers.

In this paper, the reader follows along as a set of software routers are migrated to Cisco hardware routers. Software routers, while easy to setup and maintain, suffer from a number of performance and security issues:

- Non-purpose-built solution results in additional CPU processing and lower overall bandwidth
- Security patches are not specifically tailed to routing infrastructure needs
- Monitoring and management functions are – at best – substandard compared to dedicated hardware solution

This paper begins by outlining the current state within a small – but quite real and used – lab. The lab moreover is hardened to military standards and provides services to a number of consumers, but was initially constructed entirely with software routers. As dedicated hardware become available, this state of affairs required a change; namely, migrate the software routers to their dedicated hardware equivalent. Follow along to see how this was performed within a minimal impact on the lab's overall availability (not to mention confidentiality and integrity!).

2.0 [ENCLAVE] Routing

This document provides build information on how routing was setup in a real military test network enclave. (Referred to "[ENCLAVE]" throughout this whitepaper.) In this enclave, all routing was originally performed using software routers as documented due to hardware constraints; namely, the lack of available hardware! The hardware problem was finally overcome in Summer, 2013 and the Government leads directed all software routing functions to be migrated to hardware using newly-available Cisco 3560-G Layer 3 switches. This document leads the admin through the necessary steps to implement hardware routing, with the software router conversion process as a use case. In the event of a full environment rebuild, there is no need to rebuild the software routers as the steps outlined in this document cover the setup and configuration necessary for hardware routing to occur.

2.1 *Desired Logical Outcome*

The following simple diagram shows the logical routing outcome desired:

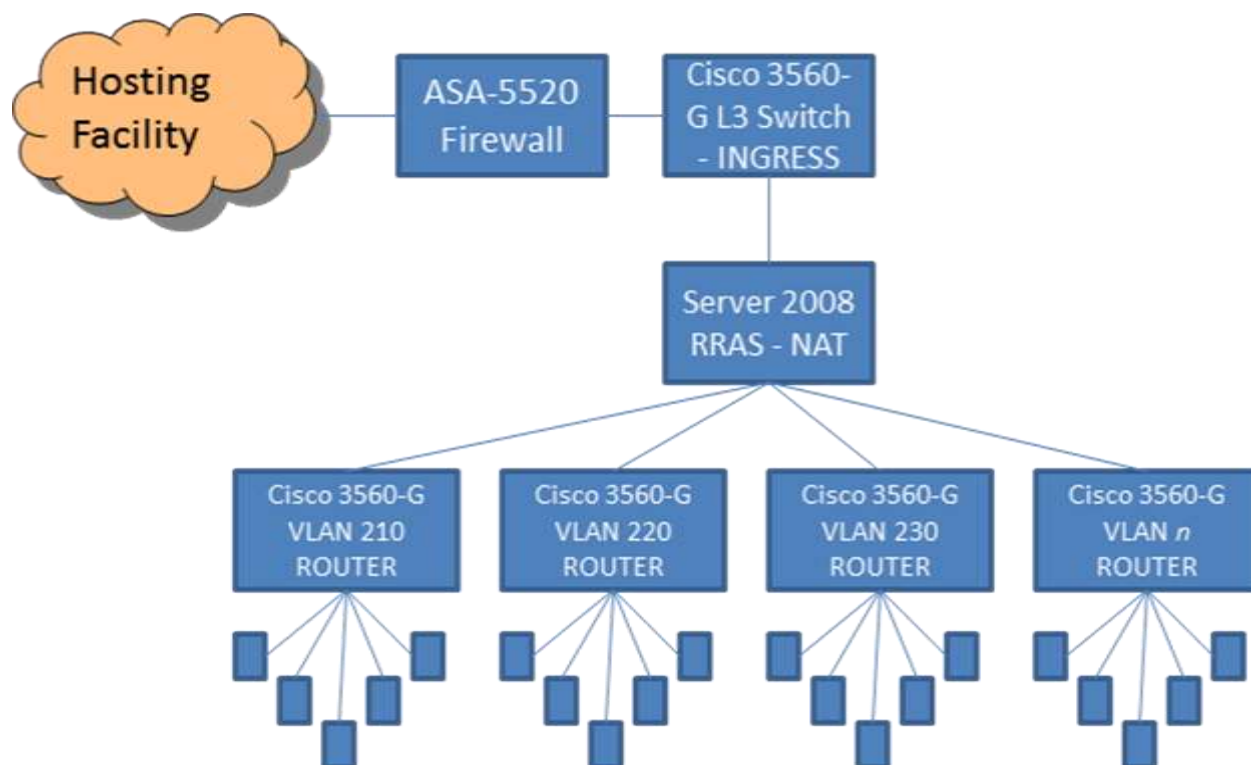


Figure 1: Desired Routing Outcome

Notice that the drawing still includes a single software router ("Server 2008 RRAS – NAT"). This is because the Cisco 3560-G Layer 3 switch does not support Network Address Translation (NAT). Therefore, for the [ENCLAVE] enclave to be private requires the single Windows software router that can perform the NAT function. A future upgrade to provide a true Cisco NAT'ing router (such as a Cisco 3640 router) will eliminate this need.

2.2 Simplified Current Physical Connectivity

The following drawing shows a simplified physical connection suitable for almost all routing. The astute reader will notice that this layout suffers from single-points-of-failure; individual switches are connected via a single port. If the port (or any switch) goes down, then network connectivity is lost. However, for this lab environment there is less concern over outages. (A full discussion of redundant switching and related spanning tree issues is beyond this routing guide.)

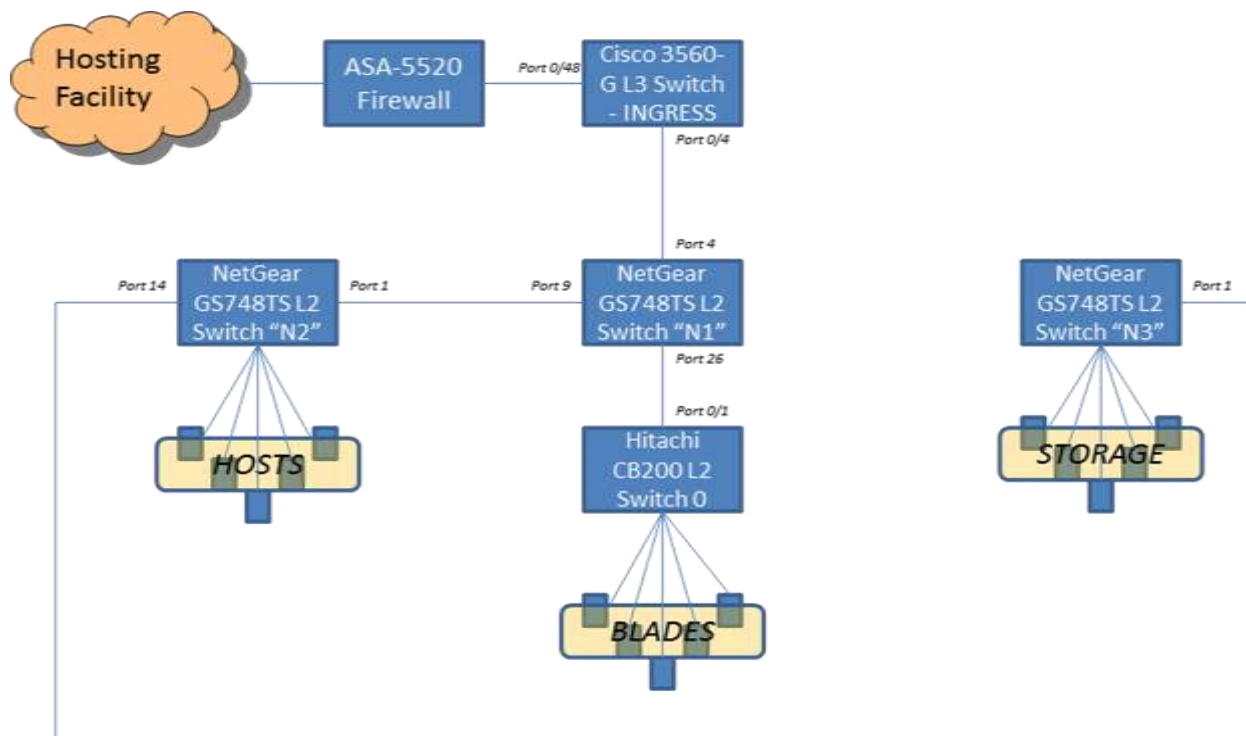


Figure 2: Current Routing Infrastructure

The physical connectivity differs from the desired logical outcome with the addition of the NetGear Layer 2 switches (one per rack of servers) in conjunction with the fact that only a *single* Cisco 3560-G Layer 3 switch exists. What this means is that the Cisco 3560-G switch will perform double-duty: both as an ingress point to the enclave as well as performing hardware routing functions inside the enclave. This approach is suboptimal; all enclave-specific processing needs to be performed using dedicated enclave hardware rather than “double-dipping” into edge hardware. However, the lack of another Cisco switch / router forces us to take this suboptimal approach.

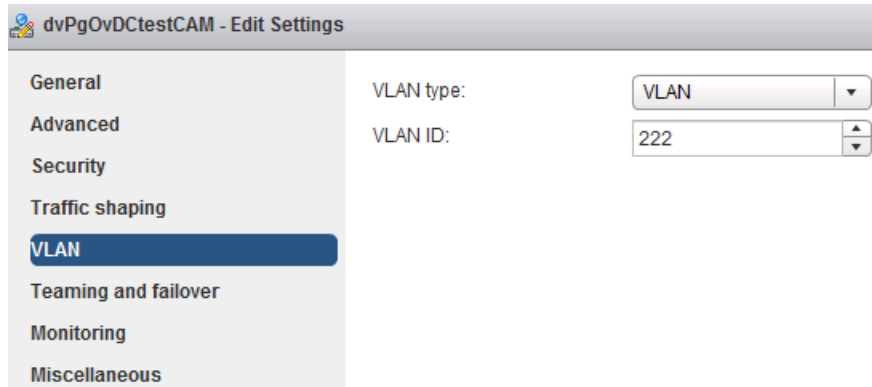
2.3 Use Case #1: Migrate a Software Router to Hardware

This use case is instructive in that it clearly identifies the configuration necessary on each device (Cisco, NetGear, and Hitachi switches). This use case also demonstrates how to use the Cisco 3560-G switch as a router and how to tie the [ENCLAVE] enclave infrastructure to use that router.

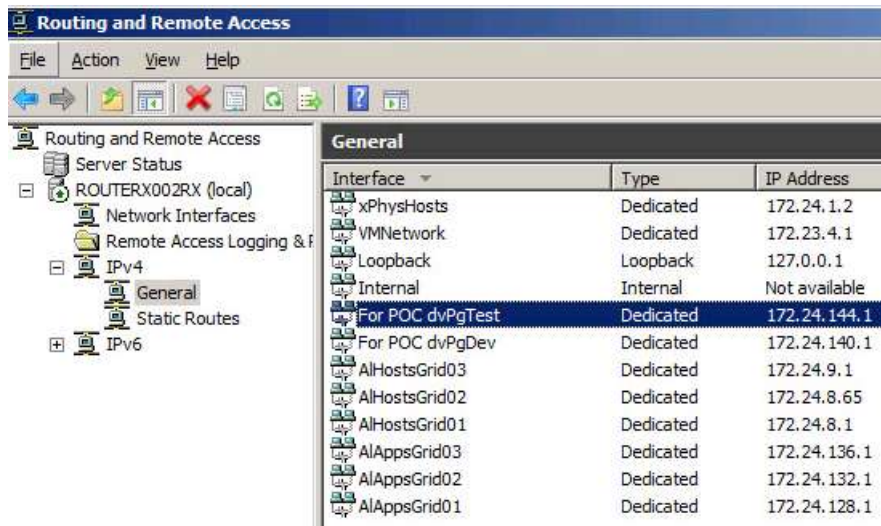
2.3.1 Software Router As-Is

This section briefly highlights the relevant pieces of the existing software routing infrastructure.

- VLAN is 222 – this is allocated VLAN for a local enclave Web site. Within the containing vSphere virtualized environment, a distributed portgroup dvPgOvDtestCAMos exists:



- The allocated subnet for this distributed portgroup is 172.24.144.0/22 (1022 addressable hosts). Gateway is 172.24.144.1 and is allocated to software router ROUTERX002RX (172.24.1.2 on the physical subnet):



- To add to the fun, the vSphere virtualized environment is further abstracted using the vFabric Application Director (vAD). Within the vAD, only a single deployed application uses the target subnet. The application has both a “DMZ” subnet (172.20.0.1/17, 32766 addressable hosts) which is used by a load balancer appliance and the 172.24.144.0/22 subnet which is used only for internal communications inside the vApp:



Node Name	Console	Logical Template	Cloud Template	Host Name	VC	Memory	Network Information	IP Address
OutSystems-DC		vAD_OutSystems_AgilePlat	Cat_Ready_OS_for_vAD		2	6144		
OutSystems-DC_2		vAD_OutSystems_AgilePlat	Cat_Ready_OS_for_vAD	WFE-1	2	6144	18C0vCloud_DMZvAD_2	172.24.145.5
LoadBalancer		vAD_CentOS5_BTIG	Ready_CentOS_for_vAD	TESTLB	1	1024	18C1ASDCloud_DMZvAD_1	172.20.60.4
							18C0vCloud_DMZvAD_2	172.24.145.5

All existing VLAN trunking is performed on existing physical Layer 2 switches.

2.3.2 Setup VLAN Definitions

This section provides guidance on setting up Virtual Local Area Networks (VLANs) on each of the Cisco 3560-G, NetGear GS748TS, and Hitachi CB2000 switches making up the target enclave. A VLAN helps to ensure network security as well as to optimize throughput by allowing the same physical network cable(s) to transport isolated network traffic; each frame placed on the wire has a “VLAN tag” that conforms to the IEEE 802.1q specification. VLANs restrict the “broadcast domain” of tagged network packets; destination stations (computers) not explicitly allowed to see a particular VLAN never receive the network packet.

VLANs are typically implemented at the network switch at Layer 2 and applied to a specific switchport. Switchports can be configured to accept untagged (“switchport mode access”) or tagged (“switchport mode trunk”) traffic; because the VLAN trunking is performed at the switch, connected stations have no ability to modify the type of network traffic that they receive.

Typical uses of VLANs are to divide large network infrastructures into multiple “broadcast domains” (subnets) to improve performance, as well as to ensure that specific types of traffic (such as management only, or database backend, or storage commands) is kept truly isolated from other traffic such as general purpose network communications.

1. **Cisco 3560-G.** This is the Layer 3 routable switch that will be used to replace the existing software router. Within this switch, use the `vlan` command as shown below for the 222 VLAN necessary for the use case. All other VLANs would be defined the same way:

```
vlan 222
name "222 - OvDCtestCAM"
state active
no shut
```

The same steps are performed for any other VLANs that will receive an IP address on the Cisco 3560-G. Note that the Cisco 3560-G does not require a VLAN to be defined if a switchport will only allow tagged traffic (this is different from either the NetGear or the Hitachi CB2000 switches – discussed below – which always require VLANs to be defined prior to use). Note that the Cisco 3560-G doesn’t store VLAN definitions in “running-config” but instead in a separate VLAN database; use the “`show vlan brief`” command to view the list of VLANs within the VLAN database:

```
ESD_Cloud_1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Gi0/2, Gi0/15, Gi0/27, Gi0/28, Gi0/49
2	cassatt	active	
5	inside1	active	
6	inside2	active	
7	inside3	active	
8	inside4	active	
9	inside5	active	
12	inside6	active	
13	inside7	active	
99	VLAN0099	active	
163	NUTANIX163	active	Gi0/5
192	VLAN0192	active	
198	VLAN0198	active	
201	201 - TESTING	active	
202	202 - TESTING	active	
203	203 - TESTING	active	
204	204 - TESTING	active	
205	205 - TESTING	active	
210	210 - Management	active	
211	211 - Access	active	
213	213 - vMotion	active	
220	220 - VMNetwork	active	
221	221 - OvDCdevCAM	active	

2. **NetGear GS748TS.** This is a Layer 2 switch which provides a “top-of-rack” solution within the enclave. For this switch, there is no command-line interface as with the Cisco switch; instead, use the Web-based user interface (UI) to create the necessary VLANs. The following screenshot shows not only the VLAN 222 created but also all other VLANs used in the [ENCLAVE] environment:

NETGEAR

Connect with Innovation™

System
Switching
QoS
Security
Monitoring
Maintenance

Ports
LAG
VLAN
STP
Multicast
Address Table

Basic
Advanced
VLAN Configuration
VLAN Membership
Port PVID Configuration

VLAN Configuration

VLAN Configuration

Select	VLAN ID	VLAN Name	Type
<input type="checkbox"/>			
<input type="checkbox"/>	1	default	Default
<input type="checkbox"/>	200	RESERVED	Static
<input type="checkbox"/>	201	TEST1	Static
<input type="checkbox"/>	202	TEST2	Static
<input type="checkbox"/>	203	TEST3	Static
<input type="checkbox"/>	204	TEST4	Static
<input type="checkbox"/>	205	TEST5	Static
<input type="checkbox"/>	210	Management	Static
<input type="checkbox"/>	211	Access	Static
<input type="checkbox"/>	213	vMotion	Static
<input type="checkbox"/>	220	VMNetwork	Static
<input type="checkbox"/>	221	OvDCdevCAM	Static
<input type="checkbox"/>	222	OvDCtestCAM	Static
<input type="checkbox"/>	230	DMZ	Static
<input type="checkbox"/>	240	iSCSI	Static

Note that a VLAN must be defined before it can trunked to a switchport. The above shot shows the 222 VLAN clearly.

- Hitachi CB200.** The target enclave uses Hitachi CB2000 blades for its compute backbone. The CB2000 comes with four embedded Layer 2 switches (more-or-less Cisco compatible but certainly not Cisco products). As with the NetGear switch, all VLANs must be defined on the Hitachi CB2000 before they can be trunked to a switchport. Unlike the Cisco 3560-G, the Hitachi CB2000 switch stores VLAN definitions as part of the “running-config”. Configuration is similar to the Cisco 3560-G except there is no need for “no shutdown” as part of the configuration.

```

vlan 222
state active
name "OvDCtestCAM"

```

Once necessary VLANs are created, continue to the next section.

2.3.3 Setup VLAN Trunks

In order for VLAN traffic to flow, the VLANs must be “trunked” to a switchport. What this means is that the

individual switchport within the switch will allow traffic only if it is tagged with a specified VLAN (or set of VLAN tags). To set this up for this enclave use case, the following systems must be configured:

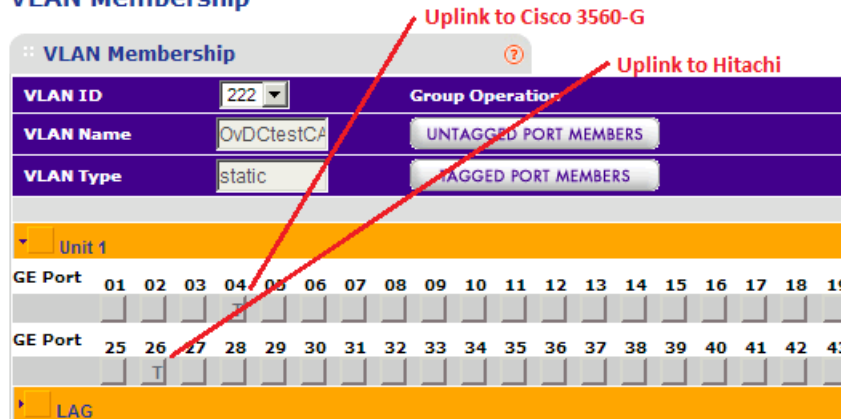
1. **Cisco 3560-G switchport g0/4.** The Cisco setup requires the switchport to be configured to permit tagged traffic. Do this by executing the following:

```
interface GigabitEthernet0/4
description Enclave routing interface
switchport trunk encapsulation dot1q
switchport trunk allowed vlan 201-205,210,211,213,220-222,230
switchport mode trunk
```

Note that the allowed VLANs include much more than the use case “222”; for a full set of VLANs within the [ENCLAVE] environment please refer to the “APG Cloud VMs.xlsx” document.

2. **NetGear GS748TS switchport 4.** The NetGear switch is different in that there is no command-line interface. To setup VLAN trunking on a switchport, use the VLAN Membership screen (from the Switching user interface menu). The following example shows how how defined VLAN 222 is associated to the switchport 4 that is connected to the Cisco 3560-G switchport g0/4:

VLAN Membership



VLAN Membership

VLAN ID: 222

VLAN Name: OvDCtestCA

VLAN Type: static

Group Operation: [UNTAGGED PORT MEMBERS] [TAGGED PORT MEMBERS]

Unit 1

GE Port	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
GE Port	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43

LAG

Be sure to apply the VLAN tagging to all necessary switchports; in the example above we tie both to the Cisco 3560-G switch (port 4 from the shot above) *and* the Hitachi CB2000 switch (port 26 from the shot above). It goes without saying that maintaining accurate network documentation is a must for this effort.

3. **Hitachi CB2000 switch.** This is a slightly different case in that there are two sets of switchports that must be configured. First, the uplink switchport from the CB2000 switch to the NetGear switch must be configured to support the necessary VLANs. Second, the internal switchports connecting the Hitachi blades to the CB2000 switch must be configured to permit the same VLAN tagged traffic. The following example shows how VLAN 222 from this use case would be assigned to both sets of switchports from enabled configuration mode:

```
! this configures the uplink from the CB2000 switch to the NetGear switch
int gig 0/1
description
```

```
switchport mode trunk
switchport trunk allowed vlan add 222
no shut
exit
```

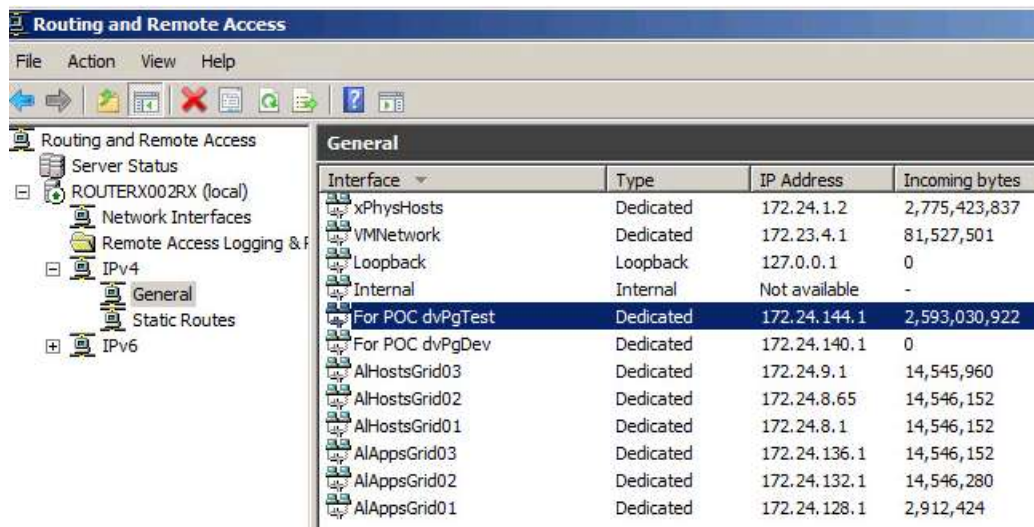
```
! this configures the internal links from the blades to the CB2000 switch
int range gig 0/5-11
switchport mode trunk
switchport trunk allowed vlan add 222
no shut
```

Note that there is no need (or support for) the Cisco command “switchport trunk encapsulation dot1q” – that command does not exist for the Hitachi CB2000 switch.

At this point VLAN trunking is setup for the initial use case.

2.3.4 Current Router Information

Routing for this use case currently occurs on a Windows Server 2008 R2 box with Routing and Remote Access Service (RRAS). RRAS is an easy-to-use server role within Windows Server and the existing 172.24.144.0/22 VLAN 222 subnet is routed as shown below:



Interface	Type	IP Address	Incoming bytes
xPhysHosts	Dedicated	172.24.1.2	2,775,423,837
VMNetwork	Dedicated	172.23.4.1	81,527,501
Loopback	Loopback	127.0.0.1	0
Internal	Internal	Not available	-
For POC dvPgTest	Dedicated	172.24.144.1	2,593,030,922
For POC dvPgDev	Dedicated	172.24.140.1	0
AlHostsGrid03	Dedicated	172.24.9.1	14,545,960
AlHostsGrid02	Dedicated	172.24.8.65	14,546,152
AlHostsGrid01	Dedicated	172.24.8.1	14,546,152
AlAppsGrid03	Dedicated	172.24.136.1	14,546,152
AlAppsGrid02	Dedicated	172.24.132.1	14,546,280
AlAppsGrid01	Dedicated	172.24.128.1	2,912,424

The above shot details which interfaces the Windows Server software routing function (RRAS) provides routes for. In other words, traffic between different subnets will not flow unless a router knows how to send network packets from one subnet to the other (which is the simplest definition of a routing function I can think of).

What the screenshot indicates is that a network interface card (NIC) named “For POC dvPgTest” exists on the Windows Server box. This NIC is created within the virtualized VMware vSphere environment as follows:

```
C:\Windows\system32>ipconfig -all

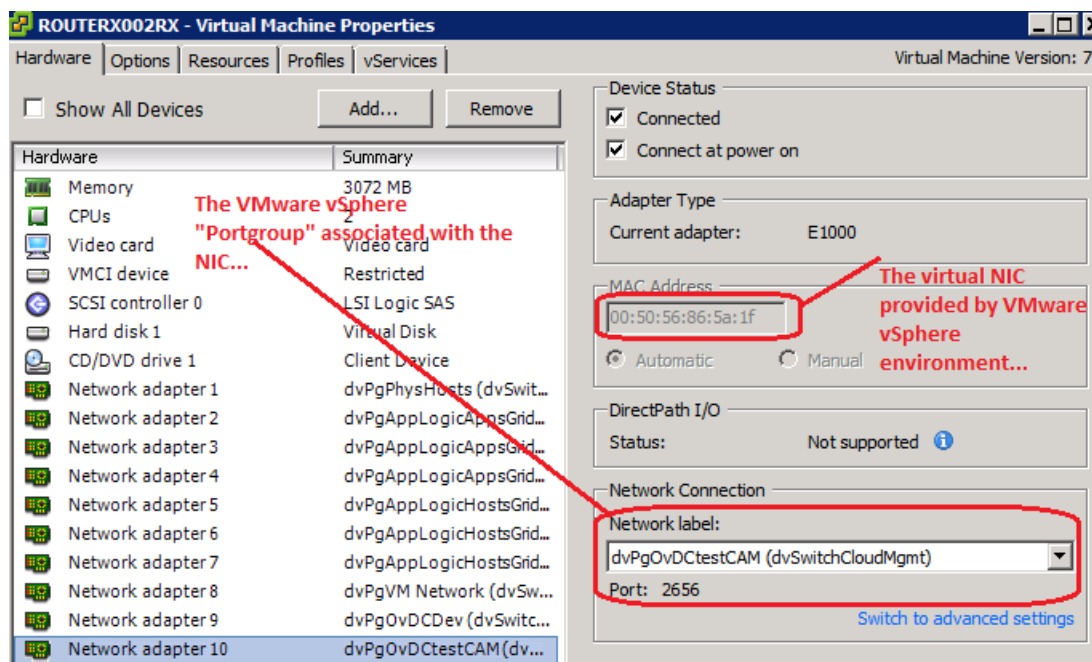
Windows IP Configuration

Host Name . . . . . : ROUTERX002RX
Primary Dns Suffix . . . . . : armycloud.cloud.army.mil
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : Yes
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : armycloud.cloud.army.mil

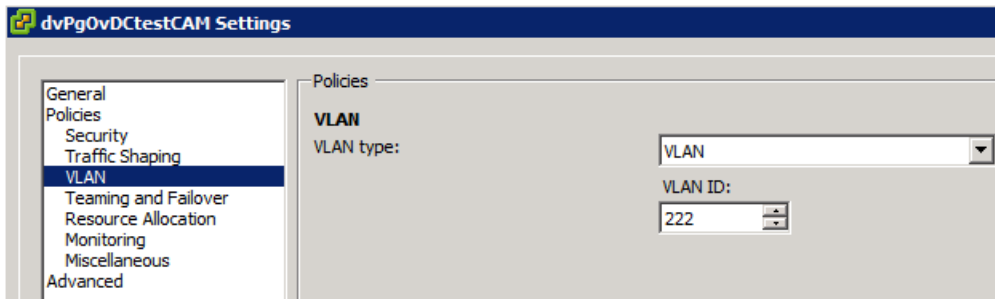
Ethernet adapter For POC dvPgTest:

Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) PRO/1000 MT Network Connection #
10
Physical Address. . . . . : 00-50-56-86-5A-1F
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv4 Address. . . . . : 172.24.144.1(Preferred)
Subnet Mask . . . . . : 255.255.252.0
Default Gateway . . . . . : 
NetBIOS over Tcpip. . . . . : Enabled
```

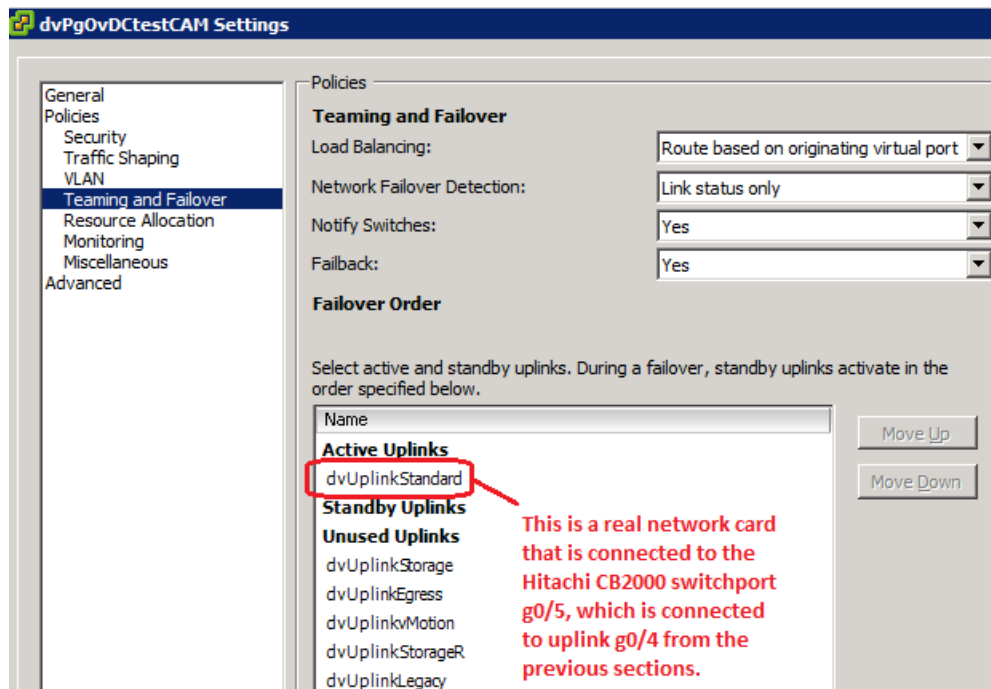
In the screenshot above, the “Ethernet adapter” (NIC) named “For POC dvPgTest” has a physical address of 00-50-56-86-5A-1F. That address is formally known as Machine Access Code (MAC) address and is provided by the vSphere environment as shown:



What the above shot means is that the NIC (“Ethernet adapter”) configured on the software router is associated with the VMware vSphere “portgroup” named dvPgOvDCtestCAM; one more trip up the food chain indicates that this portgroup is associated with the VLAN 222 we have looked at thus far:



Finally...at the end of the day a real, honest-to-goodness network card must be associated with this dvPgOvDCtestCAM portgroup. The following shows that the network card is one that is named “dvUplinkStandard”:



What all of this means is that for this use case we want to perform the following:

1. Make sure the Cisco 3560-G switch is VLAN trunked to the same switchport and – ultimately – the same NIC as the existing Windows Server software router.
2. Remove the Windows Server NIC (“Ethernet Adapter”) from the software routing function (RRAS).
3. Give the VLAN on the Cisco 3560-G switch the same IP address as is currently held by the Windows Server software router (172.24.144.1).
4. Update all “static routes” on the existing software routers such that current traffic magically works – the only downtime should be the time between disabling the NIC on the Windows Server and applying the same IP address (172.24.144.1) to the appropriate VLAN on the Cisco 3560-G switch.

The next section walks through these steps, and results in a successful migration from the current software router to the Cisco 3560-G based hardware router.

2.3.5 Migrate Router from Software to Hardware

This section covers the physical migration from the software router (currently IP 172.24.144.1) to the Cisco 3560-G switch. Follow these steps:

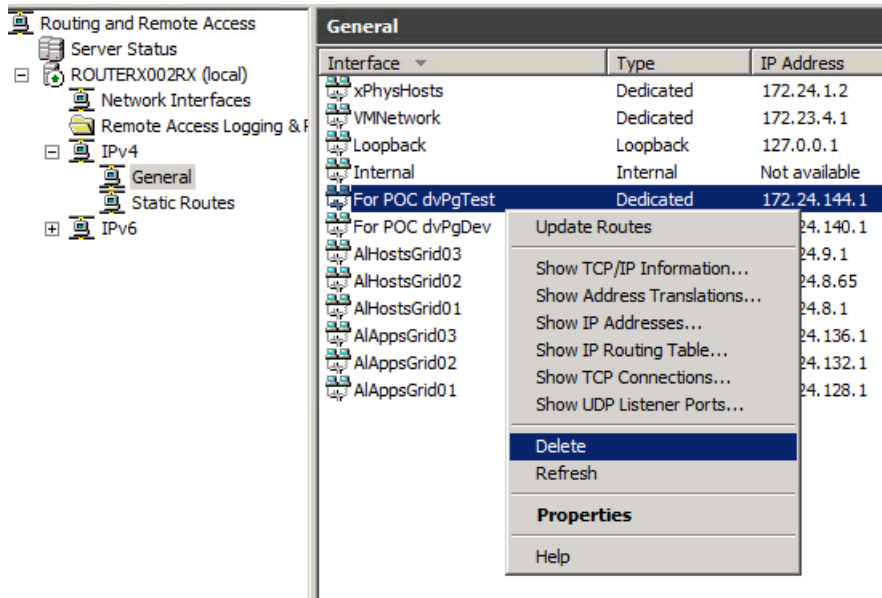
1. **Prepare Cisco 3560-G for routing.** This is easy; within the global configuration just use the following command:

```
ip routing
```

That's it...the Cisco 3560-G will now perform Layer 3 (Internet) routing for us. Of course, we have more work to do to actually migrate the current VLAN 222 172.24.144.0/22 routing function to the Cisco but we're getting there ☺
2. **Assign router (gateway) IP to the Cisco 3560-G.** Windows Server uses the term "gateway" instead of "router" but the two terms mean the same for our purposes. On the Cisco 3560-G switch, use the following commands from configuration mode:

```
interface Vlan222
description dvPgOvDCtestCAM
shutdown
ip address 172.24.144.1 255.255.252.0
no ip redirects
```

Quick notes on the above:
 - a. `interface Vlan222` – creates the "VLAN interface" to which an IP address can be assigned.
 - b. `description dvPgOvDCtestCAM` – This is simply a logical name for the VLAN interface.
 - c. `shutdown` – This is the command that makes it safe to assign the same IP address as is already in use by the Windows Server software router. Until we apply the "`no shutdown`" command to this VLAN interface, the Cisco switch will not listen on and attempt to respond to network traffic destined for 172.24.144.1.
 - d. `ip address 172.24.144.1 255.255.252.0` – This is a money shot...and probably confusing to you if you think like I do. Basically, aren't we reassigning an existing IP address (currently in use by the Windows Server software router) to the Cisco switch? The answer is "yes" but this is not a problem because of the "`shutdown`" command above. Also because of the "`ip routing`" command applied to the Cisco switch as a whole then – to a large degree – routing is already setup.
 - e. `no ip redirects` – This is a security measure; technically it prevents the Cisco switch from responding to ICMP requests with a more direct route to a destination. For our purposes, we want *all* network traffic always to go through the routes we specify explicitly.
3. **Remove interface routing on existing software router.** On the Windows Server software router, locate and right-click the interface being routed within RRAS management, then select to delete that interface:



Next, locate the network interface adapter and disable it:



At this point – you have loss of availability to that subnet:

```
C:\Users\andy.d.bruce>ping 172.24.144.1

Pinging 172.24.144.1 with 32 bytes of data:
Reply from 172.24.1.2: TTL expired in transit.
Reply from 172.24.1.2: TTL expired in transit.
Reply from 172.24.1.2: TTL expired in transit.
Reply from 172.24.1.2: TTL expired in transit.

Ping statistics for 172.24.144.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

4. **Enable IP address on Cisco hardware router.** Back on the Cisco 3560-G switch, use the “no shutdown” command for the physical interface (g0/4 in our use case) and the VLAN which has the router IP 172.24.144.1 assigned (Vlan222 in our use case):

```
ESD_Cloud_1(config-if)#int Vlan222
ESD_Cloud_1(config-if)#no shut
ESD_Cloud_1(config-if)#int g0/4
ESD_Cloud_1(config-if)#no shut
ESD_Cloud_1(config-if)#do show running-config interface g0/4
Building configuration...

Current configuration : 199 bytes
!
interface GigabitEthernet0/4
 description Enclave routing interface
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 201-205,210,211,213,220-222,230
 switchport mode trunk
end

ESD_Cloud_1(config-if)#do show running-config interface Vlan222
Building configuration...

Current configuration : 110 bytes
!
interface Vlan222
 description dvPgOvDCtestCAM
 ip address 172.24.144.1 255.255.252.0
 no ip redirects
end

ESD_Cloud_1(config-if)#do show interface Vlan222
Vlan222 is up, line protocol is up
 Hardware is EtherSVI, address is 001d.7195.14ca (bia 001d.7195.14ca)
 Description: dvPgOvDCtestCAM
 Internet address is 172.24.144.1/22
 MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
 reliability 255/255, txload 1/255, rxload 1/255
```

Both the gigabit interface 0/4
and the Vlan222 (which has the
assigned router IP of 172.24.144.1)
are up and functional.

5. **Configure routing tables on Cisco hardware router.** Still on the Cisco switch, ensure that the 172.24.144.0/22 network will be properly routed. Because the network is now directly connected, routing occurs automatically; basically, just ensure that no “ip route” command exists for this routed subnet.


```
no ip route 172.24.144.0 255.255.252.0 172.24.4.1
```

Next, ensure that the 172.24.144.0/22 subnet shows up as a direct connection using “show ip route” command:

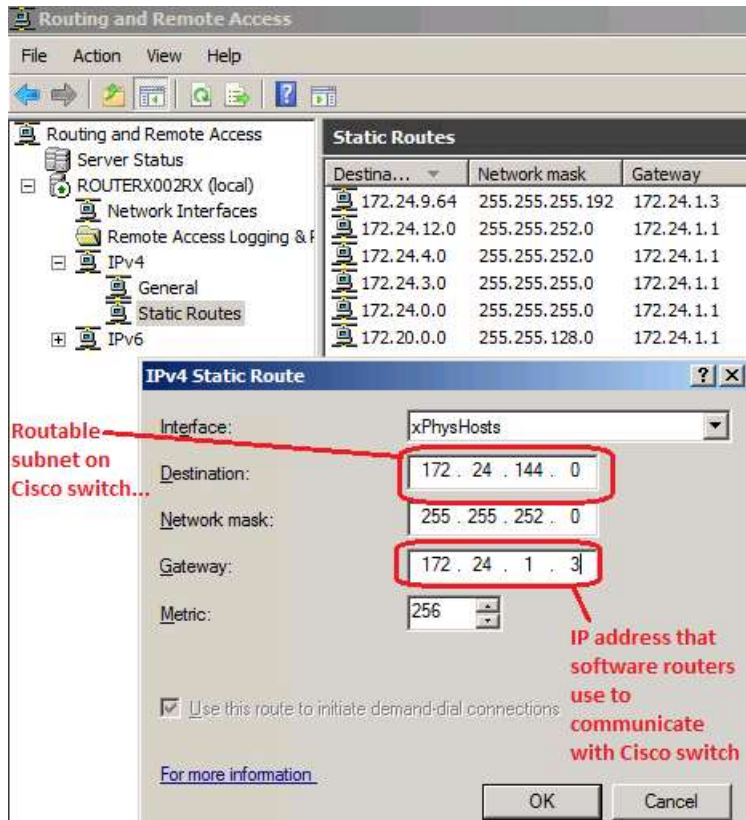
```
ESD_Cloud_1(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is [REDACTED] to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via [REDACTED]
[REDACTED] '16 is variably subnetted, 4 subnets, 3 masks
C [REDACTED] 0 is directly connected, GigabitEthernet0/48
L [REDACTED] 2 is directly connected, GigabitEthernet0/48
C [REDACTED] /25 is directly connected, Vlan998
L [REDACTED] /32 is directly connected, Vlan998
C [REDACTED] 172.24.144.0/22 is directly connected, Vlan222
L [REDACTED] 172.24.144.1/32 is directly connected, Vlan222
ESD_Cloud_1(config)#
```

The VLAN'ed interface we want to route...

6. **Update routing on other routers.** For this use case we are using “static” routing which means that any route updates must be applied manually. On the Windows Server software routers, applying route updates is easy (although static routing is only good for very small environments like our test lab):



The migration has now occurred and all routing can be verified.

2.3.6 Verify Routing Migration

Find a system running on the migrated subnet 172.24.144.0/22 – in the shot below, we set this up on a Windows server:

```
C:\Windows\system32>ipconfig
```

Windows IP Configuration

Ethernet adapter VLAN222-172.24.144.72-22:

Connection-specific DNS Suffix	:	:
IPv4 Address	:	172.24.144.72
Subnet Mask	:	255.255.252.0
Default Gateway	:	172.24.144.1

We can see that the machine with IP 172.24.144.72 has been setup on the 172.24.144.0/22 subnet. Next, from any other connected machine (preferably on a different subnet) using the `ping`, `tracert`, or `ipconfig` commands to verify routing:

```
C:\windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IPv4 Address. . . . . : 172.24.4.52
    Subnet Mask . . . . . : 255.255.252.0
    Default Gateway . . . . . : 172.24.4.1

C:\windows\system32>tracert -d 172.24.144.72

Tracing route to 172.24.144.72 over a maximum of 30 hops

  1  <1 ms    <1 ms    <1 ms    172.24.4.1
  2  <2 ms    <3 ms    <2 ms    172.24.1.3
  3  <1 ms    <1 ms    <1 ms    172.24.144.72

Trace complete.
```

IP address of the local machine from which we perform the routing test

IP address of the machine which is on the different subnet. Note that 172.24.1.3 is the address of the Cisco 3560-G switch on the routing subnet.

In the above shot, the local Windows machine 172.24.4.52 is on a completely different subnet than the 172.24.144.0/22 route we are migrating. We can see from the tracert command that routing is working correctly: the ICMP packet goes to the local gateway 172.24.4.1, which then routes to the Cisco 3560-G switch 172.24.1.3, and the Cisco switch finally completes the route to the test destination 172.24.144.72 (the other Windows server).

At this point the software router function has been migrated for this interface.

2.4 Use Case #2: Migrate “Access” Subnet

This next use case is more of the same, but concentrated to get just the steps necessary for a typical migration. Remember that setting up routing on the Cisco 3560-G switch involves almost all of these steps – it is only the removal of existing routing information that differentiates a routing “migration” from a brand-new routing setup.

1. **Identify VLAN tag and IP subnet.** The “Access” subnet is the out-of-band management network that allows admins to plug into authorized switchports, receive a DHCP IP address, and connect to internal boxes. For this use case the VLAN tag is 211 and the IP subnet is 172.24.3.0/24 with a gateway of 172.24.3.1. Currently, a Windows Server software router handles this function.
2. **Plan VLAN trunking at each switchport for each switch.** In our lab, we have the NetGear switches (top-of-rack solution), the Cisco 3560-G, and the Hitachi CB2000 switches. Here’s a handy table that can be used to track switches / switchports and ensure that VLAN trunking has been configured:

Table 1: Switch Update Table

Switch	Switchport	Function	Updated?
NetGear GS748TS	1/g4	Link to Cisco 3560-G g0/4 switchport	
NetGear GS748TS	1/g26	Link to Hitachi CB2000 switch 0 switchport g0/1	
Cisco 3560-G	g0/4	Link to NetGear 1/g4 switchport	
Hitachi CB2000 switch 0	g0/1	Link to NetGear 1/g4 switchport	

In the above very simple layout, you can see that the NetGear switchport is the glue connecting the Cisco 3560-G and the Hitachi CB2000 switches.

3. **Implement VLAN trunking for each switch.** This is a repeat of the previous use case so only the salient points highlighted here:
 - a. *Cisco 3560-G:* First, create the VLAN interface from configuration mode:


```

vlan 211
name "211 - dvPgAccess"
state active
no shut
          
```

 Second, ensure that the VLAN is trunked to the switchport using your table above:


```

int g0/4
switchport trunk allowed vlan add 211
          
```

 That should be it for the Cisco switch.
 - b. *Hitachi CB2000 switch 0:* Same steps as for the Cisco switch; create the VLAN and ensure it is tagged to the trunk switchport:


```

vlan 211
name "211 - dvPgAccess"
state active
          
```

 The trunk switchport config is below:


```

int g 0/1
switchport trunk allowed vlan add 211
          
```

 That configures the Hitachi switch.
 - c. *NetGear GS748TS switch:* This switch allows only Web configuration, the following shows the VLANs setup for the switch:

System | **Switching** | QoS | Security | Monitoring | Maintenance

Ports | LAG | **VLAN** | STP | Multicast | Address Table

Basic
» VLAN Configuration
» Advanced

VLAN Configuration

Select	VLAN ID	VLAN Name	Type
<input type="checkbox"/>			
<input type="checkbox"/>	1	default	Default
<input type="checkbox"/>	200	RESERVED	Static
<input type="checkbox"/>	201	TEST1	Static
<input type="checkbox"/>	202	TEST2	Static
<input type="checkbox"/>	203	TEST3	Static
<input type="checkbox"/>	204	TEST4	Static
<input type="checkbox"/>	205	TEST5	Static
<input type="checkbox"/>	210	Management	Static
<input type="checkbox"/>	211	Access	Static
<input type="checkbox"/>	212	Management	Static

The VLAN tag...

Also be sure that the appropriate switchports (ports 4 and 26 for this use case) are associated with that VLAN:

System | **Switching** | QoS | Security | Mon

Ports | LAG | **VLAN** | STP | Multicast | Address Table

Basic
» Advanced
» VLAN Configuration
» VLAN Membership
» Port PVID Configuration

VLAN Membership

VLAN ID: 211
VLAN Name: Access
VLAN Type: static

Unit 1

GE Port	01	02	03	04	05	06	07	08
				<input type="checkbox"/>				
GE Port	25	26	27	28	29	30	31	32
		<input type="checkbox"/>						

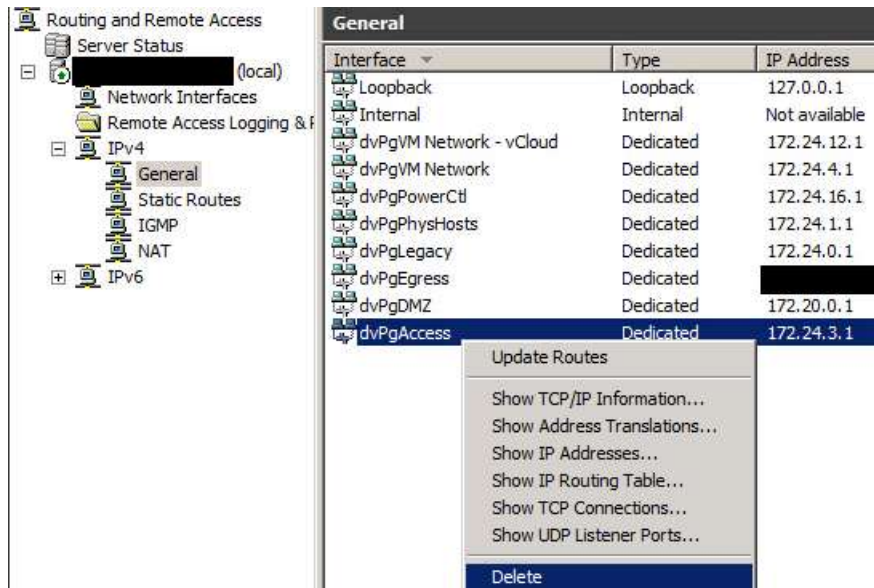
These ports *must* be setup to permit the VLAN tag...

4. **Setup the VLAN IP address on Cisco router.** Simply login to the router and set the interface information; keep the interface shutdown until you are ready to continue:

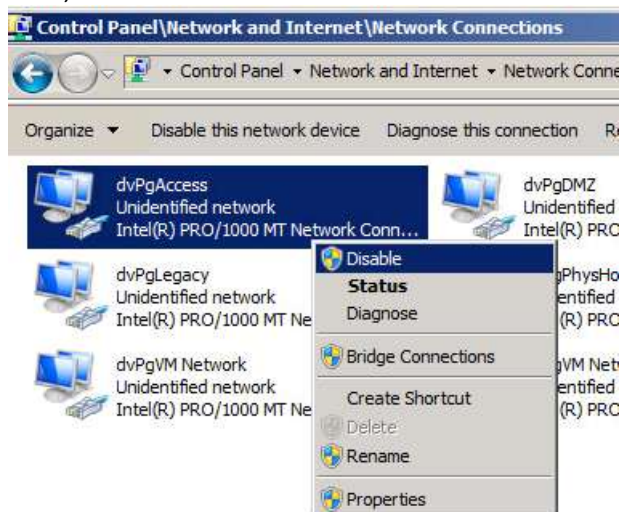
```
interface Vlan211
description dvPgAccess
shutdown
ip address 172.24.3.1 255.255.255.0
no ip redirects
```

The IP address is now ready to go.

5. **Disable existing routing.** Obviously this occurs only if you are migrating a routing function from an existing router. If you are setting up a brand-new route, you may skip this step. For our use case, we simply disable the routing function from the RRAS management screen:



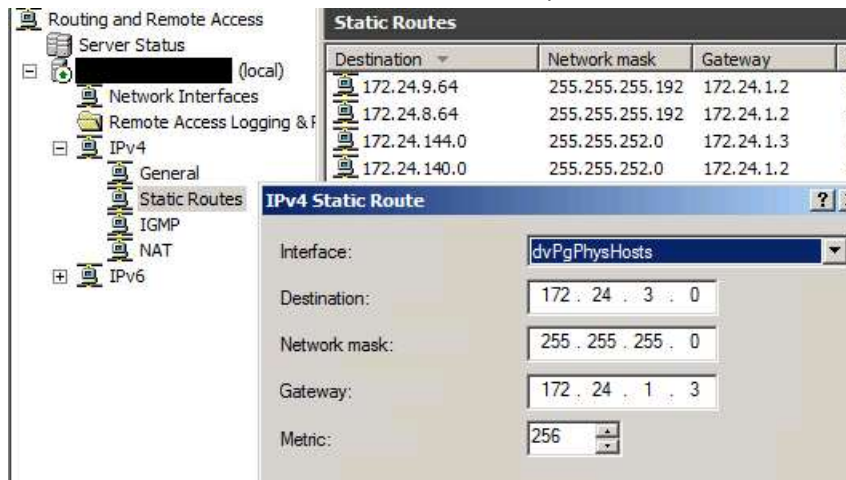
Next, we disable the network interface from that router:



As with the previous use case: we immediately lose routing to that subnet.

6. **Setup routes.** If you are using a dynamic routing protocol like Routing Internet Protocol (RIP) then this is done automatically. For our use case, we use static routing. All we have to do is go into the existing

software router that used to provide the 172.24.3.0/24 subnet routing function, and setup a static route to indicate that the Cisco 3560-G switch now provides that function.



Too easy!

- Verify correct operations.** From any other machine, verify that you can ping the gateway address provided by the Cisco 3560-G switch (172.24.3.1). In the shot below, we use one of the machines on the 172.24.3.0/24 subnet and verify we can ping a destination on another subnet.

```
C:\Users\andy.d.bruce>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : armycloud.cloud.army.mil
    IPv4 Address. . . . . : 172.24.3.151
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 172.24.3.1

C:\Users\andy.d.bruce>ping 172.24.4.52

Pinging 172.24.4.52 with 32 bytes of data:
Reply from 172.24.4.52: bytes=32 time<1ms TTL=126
Reply from 172.24.4.52: bytes=32 time=1ms TTL=126
Reply from 172.24.4.52: bytes=32 time<1ms TTL=126
Reply from 172.24.4.52: bytes=32 time=1ms TTL=126

Ping statistics for 172.24.4.52:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Remember that for ping to work that routing must be correct for both directions (source and destination).
Cisco - Software Router Migration Page 21

destination). Thus, since our ping worked to a host on another subnet, we know that routing is correct.

This second use case has repeated most of the steps from Use Case #1 above, but with much less discussion and more concentration on the steps. This same approach can be used to perform the route migration and verification for any subnet.

3.0 Some Final Thoughts...

This paper has covered very detailed routing steps from a real-world use case: assisting in the evolution of a fully-functional network lab to a more scalable and powerful infrastructure. It has led the reader through the specific steps required to create VLANs, configure trunked ports, setup routing infrastructure, and verify router functions using the test lab as an example.

This paper is designed to be one of many such hands-on and security-focused works that aim to provide a usable and workable network infrastructure environment for implementation by system administration and information assurance professionals. We hope this paper has been useful to you and that you can use it in your own network infrastructure. Be sure to visit the FITSI.org site for more hands-on and practical computer security techniques, and let us know what other papers we can provide.

About the Author

Andrew Bruce is a Lead Scientist for Computer Sciences Corporation (CSC) in the Army Programs group of the North American Public Sector. CSC provides professional services to the Federal Government and the Department of Defense, specializing in customizing and developing architecture and governance models that enable tight integration to the Army's enterprise portfolio management initiatives. Mr. Bruce's job responsibilities include: working directly with customers and partners for new business development, supporting proposal efforts, overseeing Army customers' network infrastructure, working with project managers to ensure project completion, managing software development efforts throughout the entire system life-cycle, and leading new technology research and proofs-of-concept. After a career spanning three decades in shrink-wrap, commercial, and corporate software development, Mr. Bruce is focusing on Information Assurance to achieve his goal of building and managing large data centers providing cloud computing utility services for commercial and Government customers. Mr. Bruce holds the CISSP, PMP, and FITSP-D certifications and has completed his Master's Degree in Information Assurance from Norwich University.