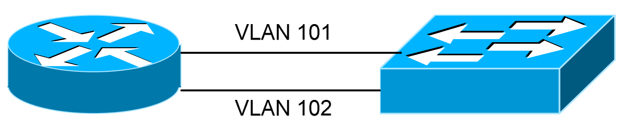
- Multilayer Switching -

Routing Between VLANs

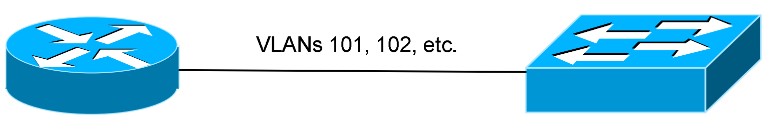
VLANs separate a Layer-2 switch into multiple broadcast domains. Each VLAN becomes its own individual broadcast domain (or IP subnet). Only interfaces belonging to the same VLAN can communicate without an intervening device. Interfaces assigned to separate VLANS require a **router** to communicate.

Routing between VLANs can be accomplished one of three ways:

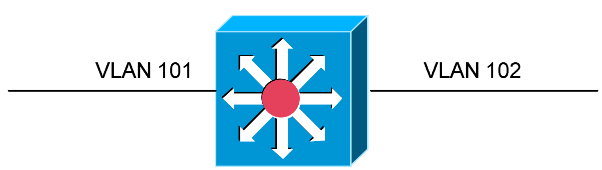
- Using an external router that has an interface to each VLAN. This is the *least scalable* solution, and completely impractical in environments with a large number of VLANs:



- Using an external router that has a *single* link into the switch, over which all VLANs can be routed. The router must understand either 802.1Q or ISL trunking encapsulations, and the switch port must be configured as a trunk. This method is known as **router-on-a-stick:**

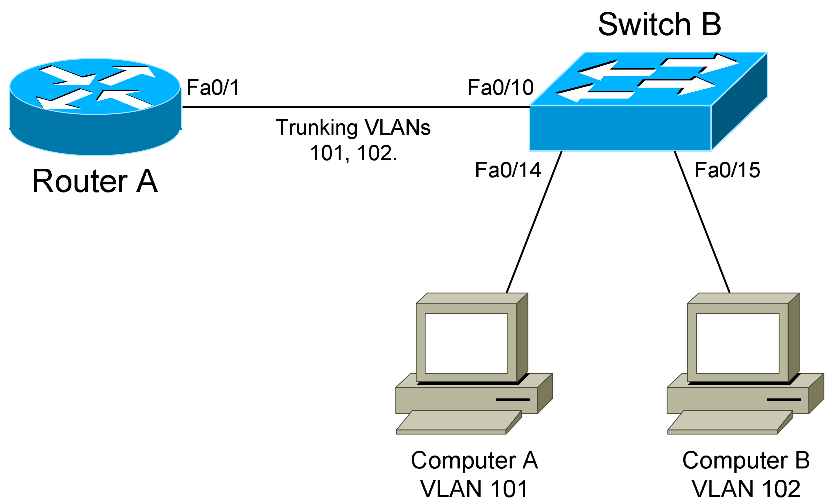


- Using a Multilayer switch with a built-in routing processor:



This guide will demonstrate the function and configuration of router-on-a-stick and Multilayer switching.

Configuring Router on a Stick



Consider the above router-on-a-stick example. To enable inter-VLAN communication, three elements must be configured:

- Interface fa0/10 on Switch B must be configured as a trunk port.
- Interfaces fa0/14 and fa0/15 on Switch B must be assigned to their respective VLANs.
- Interface fa0/1 on the Router A must be split into separate **subinterfaces** for each VLAN. Each subinterface must support the frame-tagging protocol used by the switch's trunk port.

Configuration on *Switch B* would be as follows:

Switch(config)# *interface fa0/10*
Switch(config-if)# *switchport mode trunk*
Switch(config-if)# *switchport trunk encapsulation dot1q*

Switch(config)# *interface fa0/14*
Switch(config-if)# *switchport access vlan 101*
Switch(config)# *interface fa0/15*
Switch(config-if)# *switchport access vlan 102*

Configuration on the *Router A* would be as follows:

Router(config)# *interface fa0/1*
Router(config-if)# *no shut*

Router(config)# *interface fa0/1.101*
Router(config-subif)# *encapsulation dot1q 101*
Router(config-subif)# *ip address 172.16.1.1 255.255.0.0*

Router(config)# *interface fa0/1.102*
Router(config-subif)# *encapsulation dot1q 102*
Router(config-subif)# *ip address 10.1.1.1 255.255.0.0*

Host devices in each VLAN will point to their respective subinterface on Router A. For example, Computer A's default gateway would be 172.16.1.1, and Computer B's would be 10.1.1.1. This will allow Router A to perform all inter-VLAN communication on behalf of Switch B.

Multilayer Switch Port Types

Multilayer switches support both Layer-2 (*switching*) and Layer-3 (*routing*) functions. Three **port types** can exist on Multilayer switches:

- **Switchports –** Layer-2 ports on which MAC addresses are learned.
- **Layer-3 Ports** – Essentially *routing* ports on multi-layer switches.
- **Switched Virtual Interfaces (SVI)** – A VLAN virtual interface where an IP address can be assigned to the VLAN itself.

The port type for each interface can be modified. By default, on Catalyst 2950's and 3550's, all interfaces are **switchports.**

To configure a port as a switchport:

Switch(config)# *interface fa0/10*
Switch(config-if)# *switchport*

To configure a port as a Layer-3 (*routing*) port, and assign an IP address:

Switch(config)# *interface fa0/11*
Switch(config-if)# *no switchport*
Switch(config-if)# *ip address 192.168.1.1 255.255.0.0*
Switch(config-if)# *no shut*

To assign an IP address to an SVI (virtual VLAN interface):

Switch(config)# *interface vlan 101*
Switch(config-if)# *ip address 192.168.1.1 255.255.0.0*
Switch(config-if)# *no shut*

Note that the VLAN itself is treated as an interface, and supports most IOS interface commands. To view the port type of a particular interface:

Switch# *show int fa0/10 switchport*

```
Name:              Fa0/10
Switchport:        Enabled

<snip>
```

A Layer-3 interface would display the following output:

Switch# *show int fa0/10 switchport*

```
Name:              Fa0/10
Switchport:        Disabled

<snip>
```

Multilayer Switching Methods

Multilayer switches contain both a switching and routing engine. A packet must first be *routed*, allowing the switching engine to cache the IP traffic flow. After this cache is created, subsequent packets destined for that flow can be *switched* and not routed, reducing latency.

This concept is often referred to as **route once, switch many.** Cisco implemented this type of Multilayer switching as **NetFlow switching** or **route-cache switching**.

As is their habit, Cisco replaced NetFlow multilayer switching with a more advanced method called **Cisco Express Forwarding (CEF)**, to address some of the disadvantages of route-cache switching:

- CEF is less intensive than Netflow for the multilayer switch CPU.
- CEF does not cache routes, thus there is no danger of having stale routes in the cache if the routing topology changes.

CEF contains two basic components:

- **Layer-3 Engine –** Builds the routing table and then *routes* data
- **Layer-3 Forwarding Engine –** *Switches* data based on the FIB.

The Layer-3 Engine builds the routing table using standard methods:

- Static routes.
- Dynamically via a routing protocol (such as RIP or OSPF).

The routing table is then reorganized into a more efficient table called the **Forward Information Base (FIB)**. The *most specific routes* are placed at the *top* of the FIB. The Layer-3 Forwarding Engine utilizes the FIB to then *switch* data in hardware, as opposed to *routing* it through the Layer-3 Engine's routing table.

Additionally, CEF maintains an **Adjacency Table**, containing the hardware address of the next-hop for each entry in the FIB. Entries in the adjacency table are populated as new neighboring routers are discovered, using ARP. This is referred to as **gleaning** the next-hop hardware address.

Creating an adjacency table eliminates latency from ARP lookups for next-hop information when data is actually routed/switched.

(Reference: http://www.cisco.com/en/US/docs/ios/12_1/switch/configuration/guide/xcdcef.html)

CEF Configuration

CEF is **enabled by default** on all Catalyst multi-layer switches that support CEF. CEF cannot even be disabled on Catalyst 3550, 4500 and 6500 switches.

To manually enable CEF:

Switch(config)# *ip cef*

To disable CEF on a specific interface:

Switch(config)# *interface fa0/24*
Switch(config-if)# *no ip route-cache cef*

To view the CEF Forward Information Base (FIB) table:

Switch# *show ip cef*

```
Prefix              Next Hop     Interface

172.16.1.0/24       10.5.1.1     Vlan100
172.16.2.0/24       10.5.1.2     Vlan100
172.16.0.0/16       10.5.1.2     Vlan100
0.0.0.0/0           10.1.1.1     Vlan42
```

Note that the FIB contains the following information:

- The destination prefix (and mask)
- The next-hop address
- The interface the next-hop device exists off of

The most specific routes are placed at the top of the FIB. To view the CEF Adjacency table:

Switch# *show adjacency*

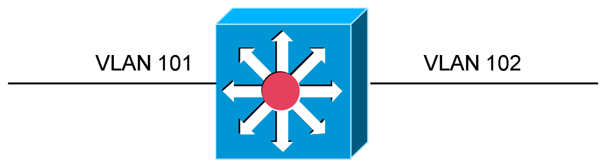
```
Protocol    Interface             Address
IP          Vlan100               10.5.1.1(6)
                                  0 packets, 0 bytes
                                  0001234567891112abcdef120800
                                  ARP 01:42:69

Protocol    Interface             Address
IP          Vlan100               10.5.1.2(6)
                                  0 packets, 0 bytes
                                  000C765412421112abcdef120800
                                  ARP 01:42:69
```

Multilayer Switching vs. Router on a Stick

The configuration of *router-on-a-stick* was demonstrated earlier in this section. Unfortunately, there are inherent disadvantages to router-on-a-stick:

- There may be insufficient bandwidth for each VLAN, as all routed traffic will need to share the same router interface.
- There will be an increased load on the router processor, to support the ISL or DOT1Q encapsulation taking place.

A more efficient (though often more expensive) alternative is to use a multilayer switch.



Configuration of inter-VLAN routing on a multilayer switch is simple. First, create the required VLANs:

Switch(config)# *vlan 101*
Switch(config-vlan)# *name VLAN101*
Switch(config)# *vlan 102*
Switch(config-vlan)# *name VLAN102*

Then, routing must be globally enabled on the multilayer switch:

Switch(config)# *ip routing*

Next, each VLAN SVI is assigned an IP address:

Switch(config)# *interface vlan 101*
Switch(config-if)# *ip address 192.168.1.1 255.255.0.0*
Switch(config-if)# *no shut*

Switch(config)# *interface vlan 102*
Switch(config-if)# *ip address 10.1.1.1 255.255.0.0*
Switch(config-if)# *no shut*

These IP addresses will serve as the default gateways for the clients on each VLAN. By adding an IP address to a VLAN, those networks will be added to the routing table as directly connected routes, allowing routing to occur.

Fallback Bridging

The Catalyst 3550 only supports IP when using CEF multilayer switching. If other protocols (IPX, Appletalk, SNA) need to be routed between VLANs, **fallback bridging** can be used.

To configure fallback bridging, a *bridge-group* must first be created. Then specific VLANs can be assigned to that *bridge-group*. A maximum of 31 bridge-groups can be created.

Switch(config)# *bridge-group 1 protocol vlan-bridge*

Switch(config)# *interface vlan 100*
Switch(config-if)# *bridge-group 1*

Switch(config)# *interface vlan 101*
Switch(config-if)# *bridge-group 1*

The first command creates the *bridge-group*. The next command place VLANs 100 and 101 in *bridge-group 1*. If protocols other than IP utilize these VLANs, they will be **transparently bridged** across the VLANs.

To view information about all configured bridge groups:

Switch# *show bridge group*