Available Categories

Adobe

Macromedia

Programming

SQL

Server Administration

Networking

Microsoft Products

Mac OS

Linux systems

Mobile devices

XML

Certification

Misc

see other chapters

Available Tutorials

Lan switching fundamentals

Router firewall security

Wireless lan security

Integrated cisco and unix network architectures

Lan switching first-step

Mpls VPN security

Beginner's guide to wi-fi wireless networking

802.11 security. wi-fi protected access and 802.11i

Wimax Technology for broadband wireless access

Wireless community networks

Network security assessment

Network security hacks

Network Management

Wireless networks first-step

LAN switching first-step

CCSP Cisco Certified Security Professional Certification

Check Point FireWall

MPLS and VPN Architectures

VLAN Configurations

Today, two dominant VLAN tagging methods exist: the Cisco proprietary Inter-Switch Link (ISL) approach and the standardized IEEE 802.1Q method. Note also that VLAN trunks are special ports or interfaces that are capable of delivering multi-VLAN traffic to a directly connected trunk port or interface.

Cisco has developed a proprietary protocol (VTP, or VLAN Trunking Protocol) to distribute VLAN information through a vast switched network without the need to configure VLANs on every switch. The only task left to do for the administrator is to configure a VTP domain and its participants and to assign ports to specific VLANs distributed via VTP. A rather young open standard for that is available as well (see IEEE GVRP - Generic VLAN Registration Protocol).

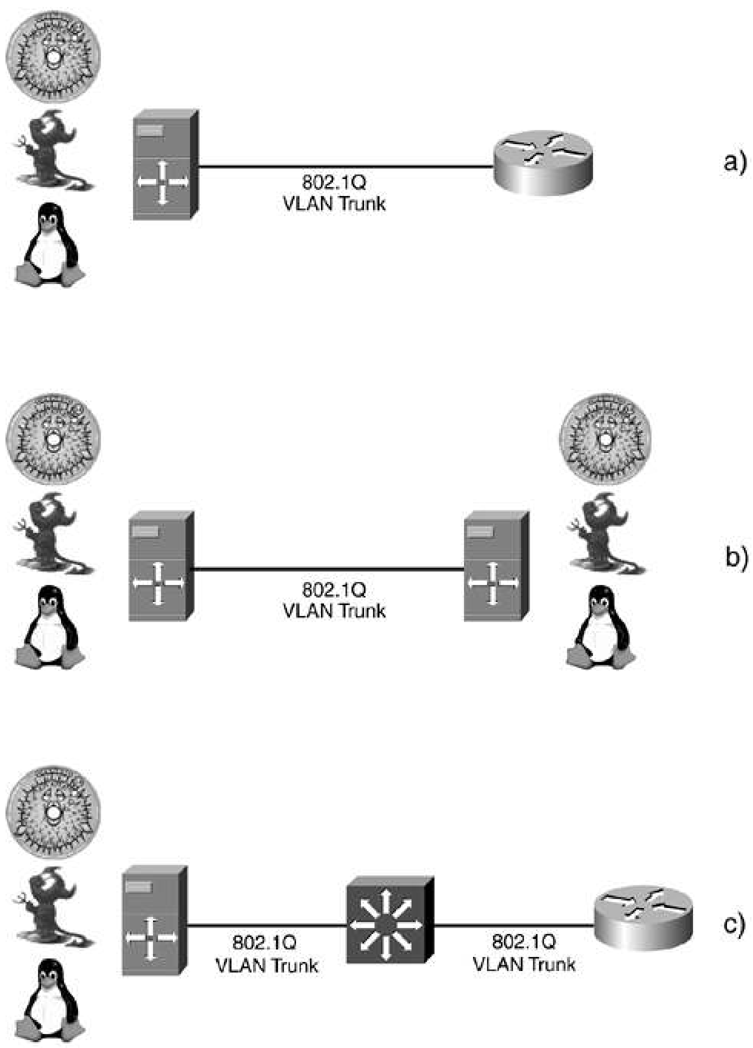
VLAN setup itself is not difficult to configure, if you adhere to the following:

- To ensure that everything is working, I recommend verifying proper operation with the **arp, netstat**, and **ifconfig/ip** commands.
- In addition, check MTU issues with large IP datagrams such as FTP transfers or handcrafted ping packets. 802.1Q VLAN tagging adds 4 overhead octets between the frame header and the payload that need to be accounted for. Therefore, adjust the interface MTU size to 1496 in case that is not done automatically (as is done on BSD systems).
- Keep in mind that the MTU throughout of your subnet should be consistent as well (for example, 1496 octets). Depending on the protocols involved, it might even become necessary to further decrease the MTU. This might be necessary for both the VLAN and parent interfaces.
- Some NICs, such as the Intel FastEtherPro, support large frames and VLAN demultiplexing natively (in firmware) and operate well with the default MTU of 1500. Sometimes patching the drivers also helps.
- Adding alias interfaces to VLAN interfaces works perfectly fine, too, exactly as with physical interfaces.
- All represented platforms have no problem with the Cisco native VLAN1. The alias and VLAN limits of a platform usually can be derived only when investigating the sources.

  Linux imposes a VLAN limit of 4096 VLANs per interface on 2.4.x kernels.
- Remember to restart your firewall when adding/deleting interfaces! Unfortunately, a lot of ill-configured firewall gateways nowadays break two-way-path MTU discovery. Ensure that you allow the proper Internet Control Message Protocol (ICMP) packets through in both directions. (ICMP type 3/code 4 = "fragmentation needed but do not fragment bit set" in combination with the probing IP packets with DF-bit set.)
- Adding frame overhead such as with Multiprotocol Label Switching (MPLS) shim headers or VLAN tagging represents in the view of many switches a so-called giant or jumbo frame, which usually is silently discarded on regular switch ports. If you encounter problems that appear to affect only large frames, check the giant counters of your switch. Most modern switches and IOS/CatOS versions can deal with this issue. As a workaround, you could configure a VLAN trunk.

Figure 5-1 shows the three VLAN topologies discussed in this chapter. Example 5-5 shows the switch VLAN configuration, and Example 5-6 shows the corresponding ARP output. Example 5-7 presents the analogous configuration for the router involved. Example 5-8 provides status information, and Example 5-9 shows the router's ARP table.

Figure 5-1. VLAN Lab Topologies



Example 5-5. Ethernet Switch VLAN Configuration (IOS)

```
Switch# show running-config
!
ip subnet-zero
!
interface FastEthernet0/1
 switchport mode trunk
!
interface FastEthernet0/2
 switchport mode trunk
!
interface FastEthernet0/3
 switchport mode trunk
!
interface VLAN1
 ip address 192.168.7.8 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
!
```

```
interface VLAN8
 ip address 192.168.80.2 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
!
ip default-gateway 192.168.7.7
!
```

Example 5-6. VLAN-Related Switch ARP Table

```
Switch# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.80.1           10   5254.05e3.e488  ARPA   VLAN8
Internet  192.168.80.2            -   0006.5258.5d40  ARPA   VLAN8
Internet  192.168.7.8             -   0006.5258.5d40  ARPA   VLAN1
Internet  192.168.80.254          8   0008.e34d.be81  ARPA   VLAN8
```

Example 5-7. Router VLAN Configuration

```
mufasa# show running-config
...
!
interface FastEthernet0/1
 description *** 802.1Q Trunk ***
 no ip address
 no ip mroute-cache
 duplex auto
 speed auto
!
interface FastEthernet0/1.1
 encapsulation dot1Q 1 native
 ip address 192.168.7.254 255.255.255.0
!
interface FastEthernet0/1.8
 encapsulation dot1Q 8
 ip address 192.168.80.254 255.255.255.0
!
...
```

Example 5-8. Router VLAN Status

```
mufasa# show vlans
```

```
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   FastEthernet0/1.1

 This is configured as native Vlan for the following interface(s) :

FastEthernet0/1

   Protocols Configured:   Address:              Received:        Transmitted:

           IP              192.168.7.254                0                   19

Virtual LAN ID:  8 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   FastEthernet0/1.8

   Protocols Configured:   Address:              Received:        Transmitted:

           IP              192.168.80.254              20                   27
```

Example 5-9. Router VLAN-Related ARP Table

```
mufasa# show arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Internet  192.168.80.1            7   5254.05e3.e488  ARPA   FastEthernet0/1.8

Internet  192.168.80.2            5   0006.5258.5d40  ARPA   FastEthernet0/1.8

Internet  192.168.7.254           -   0008.e34d.be81  ARPA   FastEthernet0/1.1

Internet  192.168.80.254          -   0008.e34d.be81  ARPA   FastEthernet0/1.8
```

The following two subsections elaborate on VLAN capabilities of FreeBSD, OpenBSD, and Linux and discuss differences and similarities in setup and behavior.

Linux VLAN Capabilities

Late 2.4.x kernels provide 802.1Q VLAN capabilities as a native kernel module. However, one still needs to retrieve the vconfig VLAN administration utility from http://www.candelatech.com/~greear/vlan.html#setup. Most up-to-date Linux distributions already include this utility.

Recently, the capability to define MAC-based VLANs was added via the macvlan_config utility, which is included in the vconfig archive. You still have to apply a kernel patch for that extension, however. Example 5-10 shows the configuration sequence for Linux VLAN interfaces, Example 5-11 shows the resulting status, and Example 5-12 shows additional monitoring information. The shaded text emphasizes the previously mentioned warnings about MTU.

Example 5-10. Linux VLAN Interface Configuration

```
[root@callisto:~#] vconfig add eth0 8

[root@callisto:~#] ifconfig vlan8 192.168.80.3/24 mtu 1496
```

Example 5-11. Linux Interface Status After VLAN Configuration

```
[root@callisto:~#] ifconfig

eth0      Link encap:Ethernet  HWaddr 00:10:5A:D7:93:60

          inet addr:192.168.14.1  Bcast:192.168.14.255  Mask:255.255.255.0

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:124 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:9246 (9.0 Kb)  TX bytes:2478 (2.4 Kb)
          Interrupt:5 Base address:0xd800

eth1      Link encap:Ethernet  HWaddr 52:54:05:E3:51:87
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9007 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5240 errors:0 dropped:0 overruns:0 carrier:0
          collisions:37 txqueuelen:100
          RX bytes:1891927 (1.8 Mb)  TX bytes:497578 (485.9 Kb)
          Interrupt:9 Base address:0xd400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:84 errors:0 dropped:0 overruns:0 frame:0
          TX packets:84 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6308 (6.1 Kb)  TX bytes:6308 (6.1 Kb)

vlan1     Link encap:Ethernet  HWaddr 00:10:5A:D7:93:60
          inet addr:192.168.7.10  Bcast:192.168.7.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1496  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0  (0.0 b)  TX bytes:0  (0.0 b)

vlan8     Link encap:Ethernet  HWaddr 00:10:5A:D7:93:60
          inet addr:192.168.80.3  Bcast:192.168.80.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1496  Metric:1
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1764 (1.7 Kb)  TX bytes:2168 (2.1 Kb)
```

Example 5-12. Linux VLAN-Related Status Information

```
[root@callisto:~#] netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
```

```
192.168.7.0     0.0.0.0         255.255.255.0   U        40 0          0 vlan1
192.168.80.0    0.0.0.0         255.255.255.0   U        40 0          0 vlan8
192.168.1.0     0.0.0.0         255.255.255.0   U        40 0          0 eth1
192.168.14.0    0.0.0.0         255.255.255.0   U        40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
0.0.0.0         192.168.1.254   0.0.0.0         UG       40 0          0 eth1

[root@callisto:~#] arp -an
? (192.168.1.2) at 08:00:46:64:74:1B [ether] on eth1
? (192.168.1.254) at 48:54:E8:8C:0A:3F [ether] on eth1
? (192.168.80.1) at 52:54:05:E3:E4:88 [ether] on vlan8

[root@callisto:~#] less /proc/net/vlan/config
VLAN Dev name    | VLAN ID
Name-Type: VLAN_NAME_TYPE_PLUS_VID_NO_PAD
vlan1          | 1  | eth0
vlan8          | 8  | eth0

[root@callisto:~#] less /proc/net/vlan/vlan8
vlan8  VID: 8    REORDER_HDR: 1  dev->priv_flags: 1
         total frames received:           21
          total bytes received:         1764
      Broadcast/Multicast Rcvd:            0

      total frames transmitted:           24
       total bytes transmitted:         2168
            total headroom inc:            0
           total encap on xmit:           24
Device: eth0
INGRESS priority mappings: 0:0  1:0  2:0  3:0  4:0  5:0  6:0 7:0
EGRESSS priority Mappings:

[root@callisto:~#] less /proc/net/vlan/vlan1
vlan1  VID: 1    REORDER_HDR: 1  dev->priv_flags: 1
         total frames received:            0
          total bytes received:            0
      Broadcast/Multicast Rcvd:            0

      total frames transmitted:            0
       total bytes transmitted:            0
            total headroom inc:            0
           total encap on xmit:            0
```

```
Device: eth0

INGRESS priority mappings: 0:0  1:0  2:0  3:0  4:0  5:0  6:0 7:0

EGRESSS priority Mappings:
```

FreeBSD/OpenBSD VLAN Capabilities

FreeBSD/OpenBSD setup is straightforward and works the same way for both operating systems. The MTU size is adjusted automatically during setup of the VLAN interfaces.

Consult the BSD vlan(4) and ifconfig(8) man pages for further details about these platforms. Example 5-13 shows the configuration steps for FreeBSD VLAN setup, Example 5-14 shows the resulting interface status, and Example 5-15 provides additional status information.

Example 5-13. BSD VLAN Configuration

```
[root@castor:~#] ifconfig vlan8 create

[root@castor:~#] ifconfig vlan8 vlan 8 vlandev ed0

[root@castor:~#] ifconfig vlan8 192.168.80.1/24
```

Example 5-14. FreeBSD Interface Status After VLAN Configuration

```
[root@castor:~#] ifconfig -a

xl0: flags=8b43<UP,BROADCAST,RUNNING,PROMISC,ALLMULTI,SIMPLEX,MULTICAST> mtu 1500

        options=3<rxcsum,txcsum>

        inet 192.168.2.7 netmask 0xffffff00 broadcast 192.168.2.255

        inet6 fe80::210:5aff:fec4:2c04%xl0 prefixlen 64 scopeid 0x1

        ether 00:10:5a:c4:2c:04

        media: Ethernet autoselect (10baseT/UTP)

        status: active

ed0: flags=8a43<UP,BROADCAST,RUNNING,ALLMULTI,SIMPLEX,MULTICAST> mtu 1500

        inet 192.168.7.7 netmask 0xffffff00 broadcast 192.168.7.255

        inet6 fe80::5054:5ff:fee3:e488%ed0 prefixlen 64 scopeid 0x2

        ether 52:54:05:e3:e4:88

lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384

        inet6 ::1 prefixlen 128

        inet6 fe80::1%lo0 prefixlen 64 scopeid 0xb

        inet 127.0.0.1 netmask 0xff000000

vlan8: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1496

        inet6 fe80::210:5aff:fec4:2c04%vlan8 prefixlen 64 scopeid 0xe

        inet 192.168.80.1 netmask 0xffffff00 broadcast 255.255.255.0

        ether 52:54:05:e3:e4:88

        vlan: 8 parent interface: ed0

...
```

Example 5-15. FreeBSD VLAN Status Information

```
[root@castor:~#] netstat -rn -f inet
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif Expire
default            192.168.2.254      UGSc        5     3826    xl0
127.0.0.1          127.0.0.1          UH          0        0    lo0
192.168.2          link#1             UC          1        0    xl0
192.168.2.254      52:54:05:e3:e4:2f  UHLW        5        0    xl0    694
192.168.7          link#2             UC          1        0    ed0
192.168.7.7        52:54:05:e3:e4:88  UHLW        0        4    lo0
192.168.80         link#14            UC          1        0  vlan8
192.168.80.1       52.54.5.e3.e4.88   UHLW        0        4    lo0

[root@castor:~#] arp -an
? (192.168.2.254) at 52:54:05:e3:e4:2f on xl0 [ethernet]
? (192.168.7.7) at 52:54:05:e3:e4:88 on ed0 permanent [ethernet]
? (192.168.80.1) at 52:54:05:e3:e4:88 on vlan8 permanent [vlan]
? (192.168.80.2) at (incomplete) on vlan8 [vlan]
? (192.168.80.3) at 00:10:5a:d7:93:60 on vlan8 [vlan]
```

0