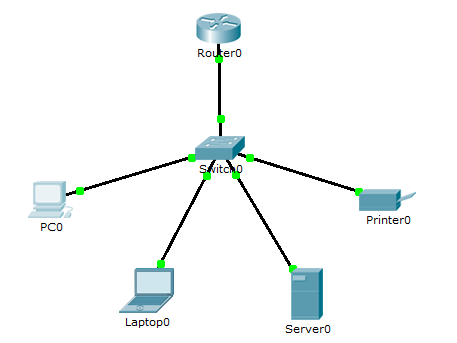
Networking Essentials

**Network**

Network is a collection of devices, inter-connected to each other. The process of sharing data/information between two or more devices is called **Networking.**



**Ethernet Cables:**

There are three types of cables  1.Ethernet (10mbps)  2.Fast Ethernet (100mbps)              3.Gigabit Ethernet (1000mbps)

**categories:**

CAT3 & CAT4 are **Ethernet Cables**.                                                                           CAT5 & CAT5E are **Fast Ethernet Cables.** CAT6 & CAT7 are **Gigabit Ethernet Cables**.

**Cables used in LAN:**

**1.Straight Cable:** It is a cable used to connect unlike devices.   Ex: switch to PC, PC to switch, etc. **2.Cross Cable:** It is a cable used to connect like devices.   Ex: switch to switch, PC to PC, etc.

**Types Of Networks**

1. LAN(Local Area Network) 2. MAN(Metro-politan Area Network) 3. WAN(Wide Area Network)

**1. LAN(Local Area Network): LAN** is a type of network which is built in a area within a small boundary. The protocol used in LAN is **ETHERNET**. Ex: Schools, colleges, etc.

**2. MAN(Metro-politan Area Network): MAN** is a type of network which is built within a city(approx.35-40kms). The protocol used in MAN is **Multiprotocol Label Switch (MPLS).** Ex: TV Cable Network, spread in a city.

**3. WAN(Wide Area Network):**  **WAN** is a type of network which is built to make connection between countries, continents, etc. It makes use of **Submarine Cables**, that is laid under sea bed. The protocol used in WAN is  **PPP Protocol (Point to Point).  Frame Relay Protocol.**

**4. CAN(Controlled Area Network): CAN** is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a **message-based protocol**, designed originally for multiplex electrical wiring within automobiles to save on copper, but is also used in many other contexts.

**5. PAN(Personal Area Network):**  **PAN** is a **computer** **network** used for data transmission amongst devices such as computers, telephones, tablets and personal digital assistants.

**OSI MODEL**

Before communication was possible only b/w devices of same manufactures. To solve this problem, OSI (**Open System Interconnect)** model was introduced. This model contained a list of same rules (protocols) for the manufactures to built a device. Therefore, communication was possible b/w devices of different manufactures. Also this OSI model was considered to be a reference model for the manufactures which resulted in the development of **TCP/IP model** There are 7 layers in OSI Model.

****A popular way to remember this table is to create a fun sentence with the first letters of each layer. For example: **A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing or layer 1 to layer 7: **P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way.

**Layer 7 – Application layer**

This is the closest layer to the end user. It provides the interface between the applications we use and the underlying layers. But notice that the programs you are using (like a web browser – IE, Firefox or Opera…) do not belong to Application layer. Telnet, FTP, email client (SMTP), Hyper Text Transfer Protocol (HTTP) are examples of Application layer.

**Layer 6 – Presentation layer**

This layer ensures the presentation of data, that the communications passing through are in the appropriate form for the recipient. In general, it acts as a translator of the network. For example, you want to send an email and the Presentation will format your data into email format. Or you want to send photos to your friend, the Presentation layer will format your data into GIF, JPG or PNG… format.

**Layer 5 – Session layer**

Layer 5 establishes, maintains and ends communication with the receiving device.

**Layer 4 – Transport layer**

This layer maintains flow control of data and provides for error checking and recovery of data between the devices. The most common example of Transport layer is Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

**Layer 3 – Network layer**

This layer provides logical addresses which routers will use to determine the path to the destination. In most cases, the logic addresses here means the IP addresses (including source & destination IP addresses).

**Layer 2 – Data Link Layer**

The Data Link layer formats the message into a *data frame*, and adds a header containing the hardware destination and source address to it. This header is responsible for finding the next destination device on a local network.

Notice that layer 3 is responsible for finding the path to the last destination (network) but it doesn’t care about who will be the next receiver. It is the Layer 2 that helps data to reach the next destination.

This layer is subdivide into 2 sub-layers: logical link control (LLC) and media access control (MAC).

The LLC functions include:

* Managing frames to upper and lower layers
* Error Control
* Flow control

The MAC sublayer carries the physical address of each device on the network. This address is more commonly called a device’s MAC address. MAC address is a 48 bits address which is burned into the NIC card on the device by its manufacturer.

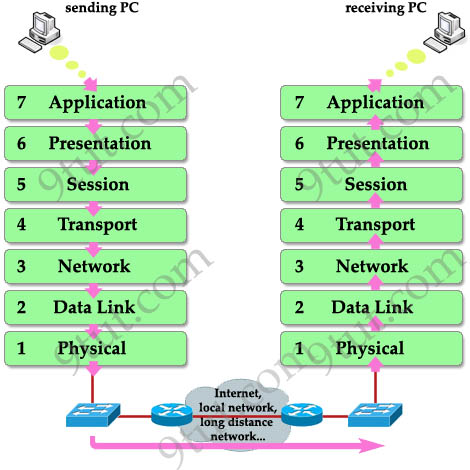
**Layer 1 – Physical layer**

The Physical Layer defines the physical characteristics of the network such as connections, voltage levels and timing.

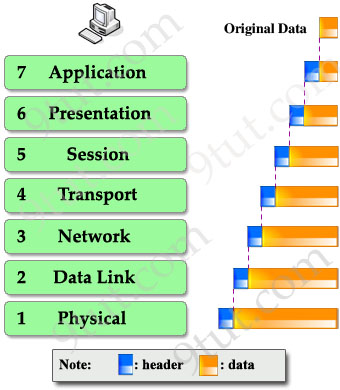
To help you remember the functions of each layer more easily, I created a fun story in which Henry (English) wants to send a document to Charles (French) to demonstrate how the OSI model works.

**Working:**

When a device wants to send information to another one, its data must go from top to bottom layer. But when a device receives this information, it must go from bottom to top to “decapsulate” it. In fact, the reverse action at the other end is very natural in our life. It is very similar when two people communicate via mail. First, the writer must write the letter, insert it into an envelope while the receiver must first open the envelope and then read the mail. The picture below shows the whole process of sending and receiving information.



When the information goes down through layers (from top to bottom), a header is added to it. This is called “encapsulation” because it is like wrapping an object in a capsule. Each header can be understood only by the corresponding layer at the receiving side. Other layers only see that layer’s header as a part of data.



At the receiving side, corresponding header is stripped off in the same layer it was attached. This process is called “decapsulation”.

**CRIMPING**

The metal has been deformed to pinch the **wire** and hold it in place. In order to **crimp** connectors onto a **wire.** This processis called **crimping.   Crimping Straight-through**

**side A**

White & Orange

Orange

White & Green

Blue

White & Blue

Green

White & Brown

Brown

**side B**

White & Orange

Orange

White & Green

Blue

White & Blue

Green

White & Brown

Brown

**Crimping Cross-over**

**side B**

White & Green

Green

White & Orange

Blue

White & Blue

Orange

White & Brown

Brown

**side A**

White & Orange

Orange

White & Green

Blue

White & Blue

Green

White & Brown

Brown

**IP Addressing (IPv4)**

* It is 32bit (4byte) IP address.
* It is divided into 4 **octets,** each octet is of 8 bits
* Each octet can hold any number from 0~255

**Index Value**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **27** | **26** | **25** | **24** | **23** | **22** | **21** | **20** |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |

0 means off

1 means on

For Example:

1.) (Bin) 00000000 🡪 (Dec) 0+0+0+0+0+0+0+0 = (Dec) 0

2.) (Bin) 11111111 🡪(Dec) 128+64+32+16+8+4+2+1 = (Dec) 255

3.) (Bin) 10110001 🡪 (Dec) 128+0+32+16+0+0+0+1 = (Dec) 177

**IP address 0~255 is divided into 5 classes**

1.Class A (0~127) 2.Class B (128~191)        3.Class C (192~223) 4.Class D (224~239) 5.Class E (240~255)

**1.Class A**

* Range 0~127
* First bit OFF
* Second bit ON
* Octet Representation: **NETWORK.HOST.HOST.HOST**
* Total Hosts: 16777214
* Usage: Global Companies (MNC's), Big ISP's

**2.Class B**

* Range 128~191
* First bit ON
* Second bit OFF
* Octet representation: **NETWORK.NETWORK.HOST.HOST**
* Total Hosts: 65534
* Usage: Medium-sized companies , mid-level ISP's

**3.Class C**

* Range 192~223
* First bit ON
* Second bit ON
* Third bit OFF
* Octet Representation: **NETWORK.NETWORK.NETWORK.HOST**
* Total Hosts: 254
* Usage: Schools, Colleges, Shops, etc.

**4.Class D**

* Range 224~239
* Usage: used for multicasting

**5.Class E**

* Range 240~255
* Usage: used for research & Science

**Cisco Router**

A **router** is a **networking** device that forwards data packets between computer **networks**. This process is called **Routing**. Routers perform the traffic directing functions on the Internet. A data packet is typically forwarded from one **router** to another **router** through the **networks** that constitute an internetwork until it reaches its destination node.



**Cisco Switch**

A network **switch** (also called **switching** hub, bridging hub, officially MAC bridge) is a **computer** networking device that connects devices together on a **computer** network by using packet **switching**, to receive, process, and forward data to the destination device.



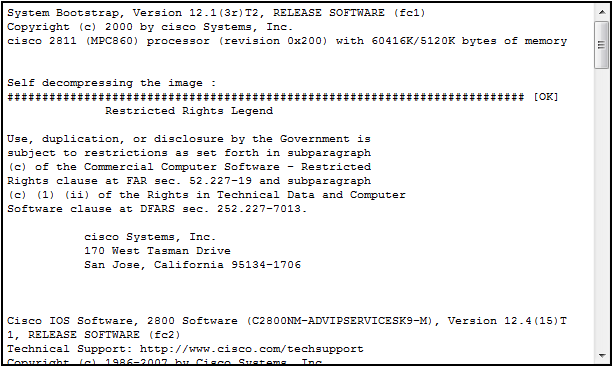
**Cisco IOS**

Cisco IOS (Internetwork Operating System) is a family of software used on most Cisco Systems routers and current Cisco network switches. IOS is a package of, routing, switching, internetworking and telecommunications functions integrated into a multitasking operating system.

A Local Area Network (LAN) relies on a software-driven Network Operating System (NOS) to function, in the same way an internetwork depends on a sophisticated operating system (IOS) to effectively connect users all over the world.

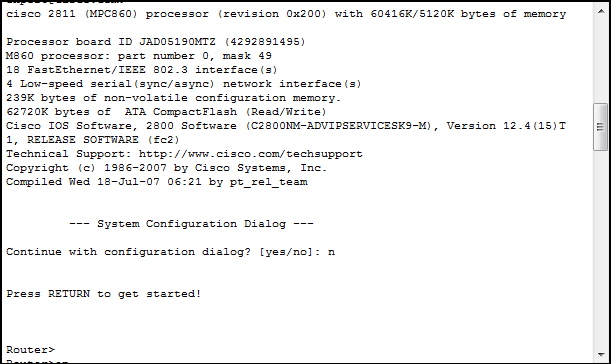
**Boot Up Process**

**Router Boot Process**  
1. The router is powered on.  
2. The router first runs Power-On Self Test (POST)  
3. The bootstrap checks the Configuration Register value to specify where to   load the IOS. By default (the default value of Configuration Register is 2102,    in hexadecimal), the router first looks for “boot system” commands in      startup-config file. If it finds these commands, it will run boot system      commands in order they appear in startup-config to locate the IOS. If not,      the IOS image is loaded from Flash . If the IOS is not found in Flash, the      bootstrap can try to load the IOS from TFTP server or from ROM (mini-IOS).  
4. After the IOS is found, it is loaded into RAM.  
5. The IOS attempts to load the configuration file (startup-config) from NVRAM      to RAM. If the startup-config is not found in NVRAM, the IOS attempts to      load a configuration file from TFTP. If no TFTP server responds, the router      enters Setup Mode (Initial Configuration Mode).

****

**Load & Extract IOS into RAM**

**POST**

****

**Loads IOS & start running router**

**Protocols**

Network protocols are formal standards and policies comprised of rules, procedures and formats that define communication between two or more devices over a network. Network protocols govern the end-to-end processes of timely, secure and managed data or network communication. Protocols are classified into two in **Cisco**,   1.Routed Protocol 2.Routing Protocol

**1.Routed Protocol:**

A routed protocol is a protocol by which data can be routed.

**TCP/IP, IPX-SPX** are protocols which are used in a Local Area Network (LAN) so computers can communicate between with each other and with other computers on the Internet. This protocol is what we call a "routed" protocol. The term "routed" refers to something which can be passed on from one place (network) to another. In this kind of protocols we require an addressing scheme and subnetting. Addressing scheme will be used to determine the network to which a host belongs and to identifying that host on that particular network.

**2.Routing Protocol**

A routing protocol specifies how routers communicate with each other, distributing information that enables them to select routes between any two nodes on a computer network.

Routing algorithms determine the specific choice of route. Each router has a prior knowledge only of networks attached to it directly. A routing protocol uses software and routing algorithms to determine optimal network data transfer and communication paths between network nodes. Routing protocols facilitate router communication and overall network topology understanding. A routing protocol is also known as a routing policy.

Ex: Static routing protocol, RIP, EIGRP,OSPF, BGP.

**DHCP(Dynamic Host Configuration Protocol)**

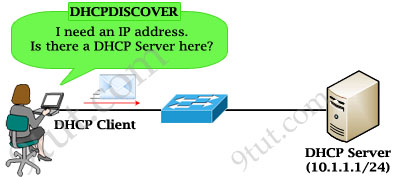
There are two ways of configuring an IP address on a device:  
+ Statically assign an IP address. This means we manually type an IP address for this computer  
+ Use a protocol so that the computer can obtain its IP address automatically (dynamically). The most popular protocol nowadays to do this task is **Dynamic Host Configuration Protocol (DHCP)** .

A big advantage of using DHCP is the ability to join a network without knowing detail about it. For example you go to a coffee shop, with DHCP enabled on your computer, you can go online without doing anything. Next day you go online at your school and you don’t have to configure anything either even though the networks of the coffee shop and your school are different (for example, the network of the coffee shop is 192.168.1.0/24 while that of your company is 10.0.0.0/8).

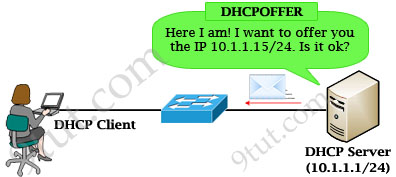
Without DHCP, you have to ask someone who knows about the networks at your location then manually choosing an IP address in that range. In bad situation, your chosen IP can be same as someone else who is also using that network and an address conflict may occur.

**Working:**

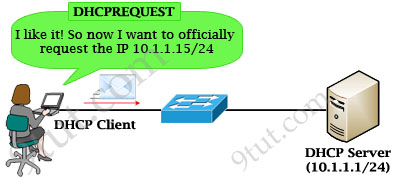
**1.** When a client boots up for the first time (or try to join a new network), it     needs to obtain     an IP address to communicate. So it first transmits a     **DHCPDISCOVER** message on its     local subnet. Because the client has no     way of knowing the subnet to which it belongs. The purpose of     DHCPDISCOVER message is to try to find out a DHCP Server (a server that     can assign IP addresses).

****

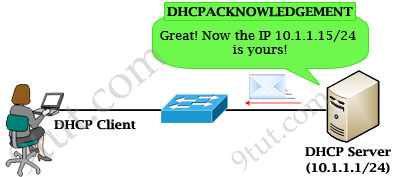
2.After receiving the discover message, the DHCP Server will dynamically pick     up an    unassigned IP address from its IP pool and broadcast a **DHCPOFFER**     message to the    client. DHCPOFFER message could contain other     information such as subnet mask,    default gateway, IP address lease time,     and domain name server (DNS).

****

3. If the client accepts the offer, it then broadcasts a **DHCPREQUEST** message     saying it will take this IP address. It is called request message because the     client might deny the offer by requesting another IP address. Notice that     DHCPREQUEST message is still a broadcast message because the DHCP client     has still not received an acknowledged IP. Also a DHCP Client can receive     DHCPOFFER messages from other DHCP Servers so sending broadcast     DHCPREQUEST message is also a way to inform other offers have been     rejected.



**4.** When the DHCP Server receives the DHCPREQUEST message from the client,      the DHCP Server accepts the request by sending the client a unicast      **DHCPACKNOWLEDGEMENT** message (DHCPACK).



     In conclusion there are four messages sent between the DHCP Client and           DHCP Server: DHCP**D**ISCOVER, DHCP**O**FFER, DHCP**R**EQUEST and     DHCP**A**CKNOWLEDGEMENT. This process are often abbreviated as **DORA** (for    Discover, Offer, Request, Acknowledgement).

    After receiving DHCPACKNOWLEDGEMENT, the IP address is leased to the     DHCP Client. A client will usually keep the same address by periodically     contacting the DHCP server to renew the lease before the lease expires.

|  |  |
| --- | --- |
| **Configuration** | **Description** |
| Router(config)#ipdhcp pool CLIENTS | Create a DHCP Pool named CLIENTS |
| Router(dhcp-config)#network 10.1.1.0 /24 | Specifies the subnet and mask of the DHCP address pool |
| Router(dhcp-config)#default-router 10.1.1.1 | Set the default gateway of the DHCP Clients |
| Router(dhcp-config)#dns-server 10.1.1.1 | Configure a Domain Name Server (DNS) |
| Router(dhcp-config)#domain-name nsg.com | Configure a domain-name |
| Router(dhcp-config)#lease 0 12 | Duration of the lease (the time during which a client computer can use an assigned IP address). The syntax is “**lease** {days[hours] [minutes] | infinite}”. In this case the lease is 12 hours. The default is a one-day lease. Before the lease expires, the client typically needs to renew its address lease assignment with the server |
| Router(dhcp-config)#exit |  |
| Router(config)# ipdhcp excluded-address 10.1.1.1 10.1.1.10 | The IP range that a DHCP Server should not assign to DHCP Clients. Notice this command is configured under global configuration mode |

**Feasible Distance (FD) and Administrative Distance (AD)**

**Administrative distance (AD):** the cost from the neighbor to the destination.

**Feasible distance (FD):** The sum of the AD plus the cost between the local router and the next-hop router

**Successor:** The primary route used to reach a destination. The successor route is kept in the routing table. Notice that successor is the best route to that destination.   
**Feasible successor:** The backup route. To be a feasible successor, the route must have an AD less than the FD of the current successor route

**Static Routing**

Static Routing is when you statically configure a router to send traffic for particular destinations in preconfigured directions.

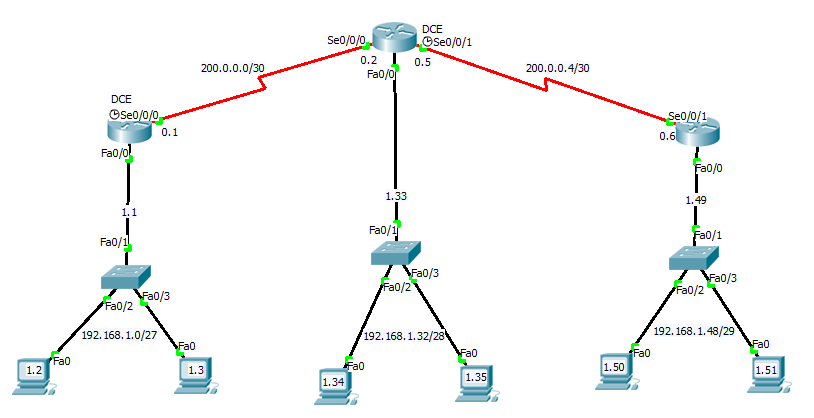
Static routing is not a routing protocol; instead, it is the manual configuration and selection of a network route, usually managed by the network administrator. It is employed in scenarios where the network parameters and environment are expected to remain constant.   
  
Static routing is only optimal in a few situations. Network degradation, latency and congestion are inevitable consequences of the non-flexible nature of static routing because there is no adjustment when the primary route is unavailable.

* Metric of Static Routing is 0.
* Administrative Distance(AD) value of Static Routing is 1.

**Syntax:**

#ip route (destination network IP address with SM) (next Hop gateway)

**Practical**

****

Router>enable Router#configure terminal     Router(config)#hostname R1 R1(config)#int f0/0 R1(config-if)#ip address 192.168.1.1 255.255.255.244 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#int s0/0/0 R1(config-if)#ip address 200.0.0.1 255.255.255.252 R1(config-if)#clock rate 64000 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#ip route 192.168.1.32 200.0.0.240 200.0.0.2 R1(config)# ip route 192.168.1.48 200.0.0.248 200.0.0.2 R1(config)#exit

Router>enable Router#configure terminal     Router(config)#hostname R2 R2(config)#int f0/0 R2(config-if)#ip address 192.168.1.33 255.255.255.240 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/0 R2(config-if)#ip address 200.0.0.2 255.255.255.252 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/1 R2(config-if)#ip address 200.0.0.5 255.255.255.252 R2(config-if)#clock rate 64000 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#ip route 192.168.1.0 200.0.0.224 200.0.0.1 R2(config)# ip route 192.168.1.48 200.0.0.248 200.0.0.6 R2(config)#exit

Router>enable Router#configure terminal     Router(config)#hostname R3 R3(config)#int f0/0 R3(config-if)#ip address 192.168.1.49 255.255.255.248 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#int s0/0/1 R3(config-if)#ip address 200.0.0.6 255.255.255.252 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#ip route 192.168.1.0 200.0.0.224 200.0.0.1 R3(config)# ip route 192.168.1.48 200.0.0.248 200.0.0.6 R3(config)#exit

**Dynamic Routing**

**Dynamic routing** is a networking technique that provides optimal data **routing**. Unlike static **routing**, **dynamic routing** enables routers to select paths according to real-time logical network layout changes. If a **router** on the route goes down the destination may become unreachable. **Dynamic routing** allows **routing** tables in **routers** to change as the possible routes change.

Dynamic routes, use a routing protocol to determine the best path and the routes can be changed depending on specific parameters (like bandwidth, delay, cost…).

**Types of Dynamic Routing**

1.RIP 2.EIGRP 3.OSPF

**1.Routing Information Protocol (RIP)**

Routing Information Protocol (RIP) is a distance-vector routing protocol. RIP sends the complete routing table out to all active interfaces every 30 seconds. RIP only uses hop count (the number of routers) to determine the best way to a remote network.

Distance vector protocols advertise routing information by sending messages, called routing updates, out the interfaces on a router

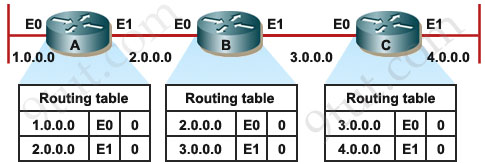
* It is the oldest type of Dynamic Routing.
* It works on the basis of **Neighbor shift Criteria.**
* Its Administrative Distance (AD) value is 120.
* Its Metric is **Hop Count**.
* Therefore, RIP is limited to **15 Hop counts.**

**Working**

Works on the basis of Neighbor shift Criteria.

A big problem with distance vector routing protocol is routing loop. Let’s take a look at how a routing loop occurs.

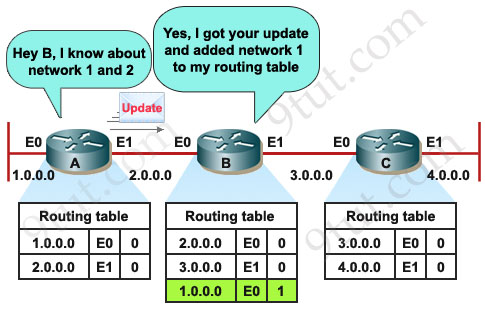
Here we have routers A, B and C. Notice that at the beginning (when a routing protocol is not turned on) there are only directly connected networks in the routing tables of these routers. For example, in the routing table of router A, **network 1.0.0.0** has already been known because it is directly connected through **interface E0** and the **metric** (of a directly connected network) is **0** (these 3 parameters are shown in the routing tables below).



Also B knows networks **2.0.0.0**&**3.0.0.0** with a **metric of 0**.  
Also C knows networks **3.0.0.0**&**4.0.0.0** with a **metric of 0**.

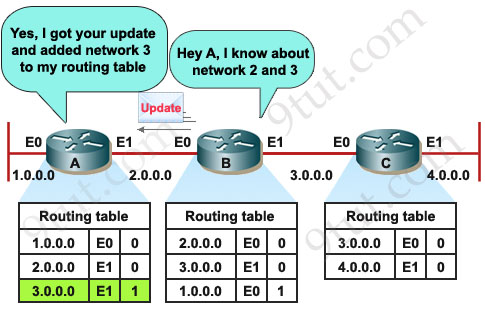
Now we turn on RIP on these router, we will call network 1.0.0.0 network 1, 2.0.0.0 network 2 and so on.

RIP sends update every 30 seconds so after 30 sec goes by, A sends a copy of its routing table to B, B already knew about network 2 but now B learns about network 1 as well. Notice the metric we have here for directly connected networks, since we’re using RIP, we’re using a metric of hop count. Remember a **Hop Count (or a hop) is how many routers that these packets will have to go through to reach the destination**. For example, from router A to network 1 & 2 (which are directly connected) it goes to 0 hop, router B has now learned about network 1 from A via E0 interface so the metric now will be 1 hop.

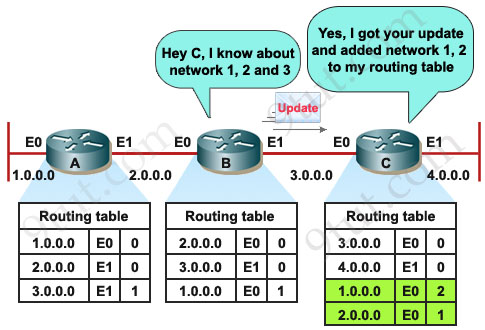


Each router receives a routing table from its direct neighbor. For example, Router B receives information from Router A about network 1 and 2. It then adds a distance vector metric (such as the number of hops), increasing the distance vector of these routes by 1.

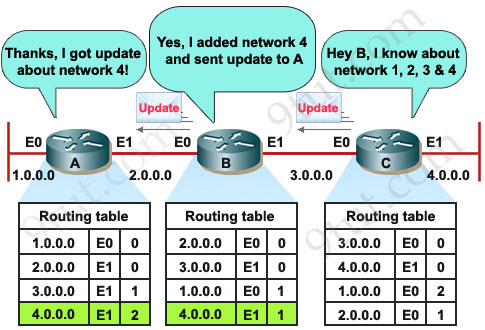
B also exchanges its routing table with A about network 2 and 3.



B then passes the routing table to its other neighbor, Router C.



C also sends its update to B and B sends it to A.



Now the network is converged.In the whole process each router is update of routing table in the network by enquiring with its neighbors, therefore, it works on **Neighbor Shift Criteria.**

RIP has two versions,

RIP v1 (Routing Information Protocol Version 1) RIP v2 (Routing Information Protocol Version 2)

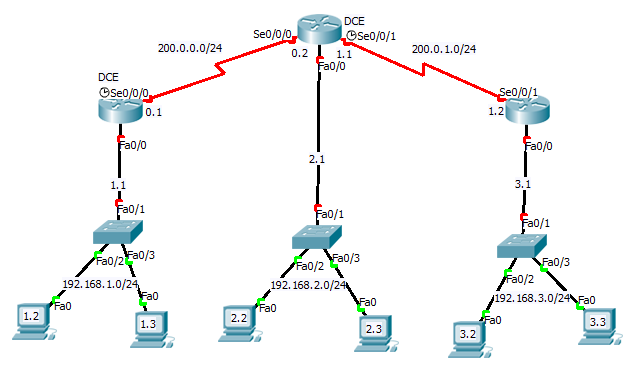
**RIP v1 (Routing Information Protocol Version 1)**

* It supports only Continuous Networks. Ex:192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24,so on.
* It supports only Class-full network ( Full Length Subnet Mask [FLSM] )i.e /24, /16, /8.

**Syntax(Routing Commands)**

Router(config)#router rip Router(config-router)# version [?] Router(config-router)# network [net.adress] Router(config-router)#exit

**Practical**

****

Router>enable Router#configure terminal     Router(config)#hostname R1 R1(config)#int f0/0 R1(config-if)#ip address 192.168.1.1 255.255.255.0 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#int s0/0/0 R1(config-if)#ip address 200.0.0.1 255.255.255.0 R1(config-if)#clock rate 64000 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#router rip R1(config-router)# version 1 R1(config-router)#network 192.168.1.0 R1(config-router)#network 200.0.0.0 R1(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R2 R2(config)#int f0/0 R2(config-if)#ip address 192.168.2.1 255.255.255.0 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/0 R2(config-if)#ip address 200.0.0.2 255.255.255.0 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/1 R2(config-if)#ip address 200.0.1.1 255.255.255.0 R2(config-if)#clock rate 64000 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#router rip R2(config-router)# version 1 R2(config-router)#network 192.168.2.0 R2(config-router)#network 200.0.1.0 R2(config-router)#network 200.0.0.0 R2(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R3 R3(config)#int f0/0 R3(config-if)#ip address 192.168.3.1 255.255.255.0 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#int s0/0/1 R3(config-if)#ip address 200.0.1.2 255.255.255.0 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#router rip R3(config-router)# version 1 R3(config-router)#network 192.168.3.0 R3(config-router)#network 200.0.1.0 R3(config-router)#exit

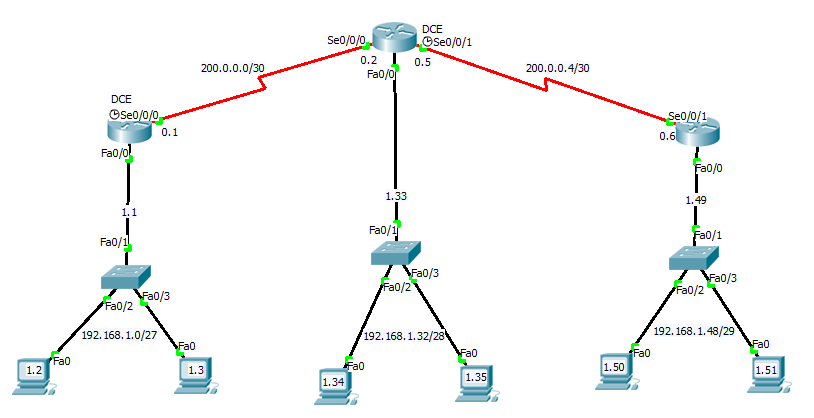
**RIPv2 (Routing Information Protocol Version 2)**

* It supports both Continuous & Dis-continuous Networks.
* It supports both Class-full network ( Full Length Subnet Mask [FLSM] ) & Class-less(Variable length Subnet Mask [VLSM] ).

**Syntax(Routing Commands)**

Router(config)#router rip Router(config-router)# version [?] Router(config-router)# network [net.adress] Router(config-router)#no auto-summary Router(config-router)#exit

**Practical**

****

Router>enable Router#configure terminal     Router(config)#hostname R1 R1(config)#int f0/0 R1(config-if)#ip address 192.168.1.1 255.255.255.244 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#int s0/0/0 R1(config-if)#ip address 200.0.0.1 255.255.255.252 R1(config-if)#clock rate 64000 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#router rip R1(config-router)# version 2 R1(config-router)#network 192.168.1.0

R1(config-router)#network 200.0.0.0 R1(config-router)#no auto-summary R1(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R2 R2(config)#int f0/0 R2(config-if)#ip address 192.168.1.33 255.255.255.240 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/0 R2(config-if)#ip address 200.0.0.2 255.255.255.252 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/1 R2(config-if)#ip address 200.0.0.5 255.255.255.252 R2(config-if)#clock rate 64000 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#router rip R2(config-router)# version 2 R2(config-router)#network 192.168.1.32 R2(config-router)#network 200.0.0.0 R2(config-router)#network 200.0.0.4 R2(config-router)#no auto-summary R2(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R3 R3(config)#int f0/0 R3(config-if)#ip address 192.168.1.49 255.255.255.248 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#int s0/0/1 R3(config-if)#ip address 200.0.0.6 255.255.255.252 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#router rip R3(config-router)# version 2 R3(config-router)#network 192.168.1.48 R3(config-router)#network 200.0.0.4 R3(config-router)#no auto-summary R3(config-router)#exit

**2.Enhanced Interior Gateway Routing Protocol (EIGRP)**

In the past, Enhanced Interior Gateway Routing Protocol (EIGRP) is a Cisco-proprietary routing protocol but from March-2013 Cisco opens up EIGRP as an open standard in order to help companies operate in a multi-vendor environment. EIGRP is a classless routing protocol, meaning that it sends the subnet mask of its interfaces in routing updates, which use a complex metric based on bandwidth and delay.

EIGRP is referred to as a **hybrid routing protocol** because it has the characteristics of both distance-vector and link-state protocols but now Cisco refers it as an advanced distance vector protocol.

**The main features,**

* Support **VLSM and discontiguous network**
* **Use Reliable Transport Protocol** (RTP) to delivery and reception of EIGRP packet
* Use the best path selection **Diffusing Update Algorithm (DUAL)**, guaranteeing loop-free paths and backup paths throughout the routing domain
* **Discover neighboring devices using periodic Hello messages** to discover and monitor connection status with its neighbors
* Exchange the full routing table at startup and send **partial\* triggered updates** thereafter (not full updates like distance-vector protocols) and the triggered updates are only sent to routers that need the information. This behavior is different from the link-state protocol in which an update will be sent to all the link-state routers within that area. For example, EIGRP will send updates when a new link comes up or a link becoming unavailable
* **Supports multiple protocols**: EIGRP can exchange routes for IPv4, IPv6, AppleTalk and IPX/SPX networks
* **Load balancing**: EIGRP supports unequal metric load balancing, which allows administrators to better distribute traffic flow in their networks.

EIGRP use metrics composed of bandwidth, delay, reliability, and load. By default, EIGRP uses only bandwidth and delay.

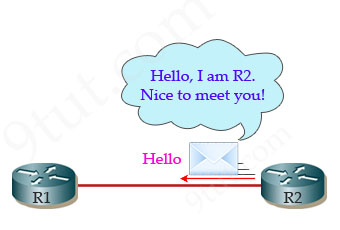
**EIGRP use five types of packets to communicate:**

* **Hello:** used to identify neighbors. They are sent as periodic multicasts
* **Update:** used to advertise routes, only sent as multicasts when something is changed
* **Ack:** acknowledges receipt of an update. In fact, Ack is Hello packet without data. It is always unicast and uses UDP.
* **Query:** used to find alternate paths when all paths to a destination have failed
* **Reply:** is sent in response to query packets to instruct the originator not to recompute the route because feasible successors exist. Reply packets are always unicast to the originator of the query

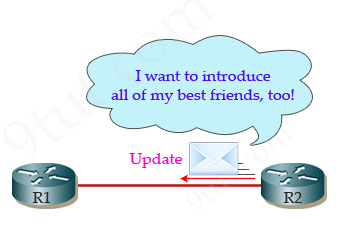
Firstly, the router will try to establish a neighboring relationships by sending **“Hello”** packets to others running EIGRP. The destination IP address is 224.0.0.10 which is the multicast address of EIGRP. By this way, other routers running EIGRP will receive and proceed these multicast packets. These packets are sent over TCP.



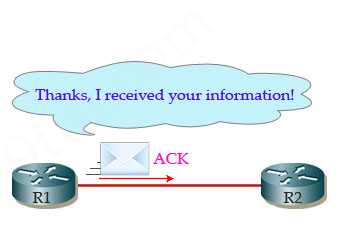
After hearing “Hello” from R1, R2 will respond with another “Hello” packet.



R2 will also send its routing table to R1 by **“Update”** packets. Remember that R2 will send its complete routing table for the first time.



R1 confirms it has received the Update packet by an **“ACK”** message.



R1 will also send to R2 all of its routing table for the first time



R2 sends a message saying it has received R1’s routing table.



Note: EIGRP sends every Query and Reply message using RTP, so every message is acknowledged using an EIGRP ACK message.

**To become a neighbor, the following conditions must be met:**

* The router must hear a Hello packet from a neighbor.
* The EIGRP autonomous system must be the same.
* K-values must be the same.

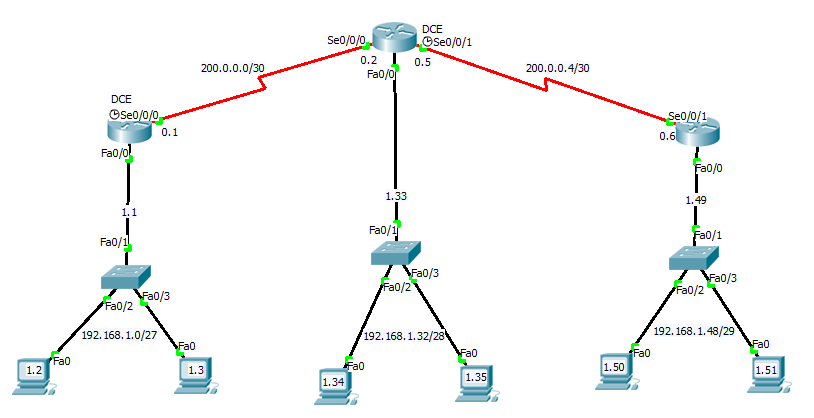
**EIGRP builds and maintains three tables:**

* Neighbor table: lists directly connected routers running EIGRP with which this router has an adjacency
* Topology table: lists all routes learned from each EIGRP neighbor
* Routing table: lists all best routes from the EIGRP topology table and other routing processes

**Syntax(Routing Commands)**

Router(config)#router eigrp [ASN] Router(config-router)# network [net.adress] Router(config-router)#no auto-summary Router(config-router)#exit

**Practical**

****

Router>enable Router#configure terminal     Router(config)#hostname R1 R1(config)#int f0/0 R1(config-if)#ip address 192.168.1.1 255.255.255.244 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#int s0/0/0 R1(config-if)#ip address 200.0.0.1 255.255.255.252 R1(config-if)#clock rate 64000 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#router eigrp 1 R1(config-router)#network 192.168.1.0 R1(config-router)#network 200.0.0.0 R1(config-router)#no auto-summary R1(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R2 R2(config)#int f0/0 R2(config-if)#ip address 192.168.1.33 255.255.255.240 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/0 R2(config-if)#ip address 200.0.0.2 255.255.255.252 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/1 R2(config-if)#ip address 200.0.0.5 255.255.255.252 R2(config-if)#clock rate 64000 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#router eigrp 1 R2(config-router)#network 192.168.1.32 R2(config-router)#network 200.0.0.0 R2(config-router)#network 200.0.0.4 R2(config-router)#no auto-summary R2(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R3 R3(config)#int f0/0 R3(config-if)#ip address 192.168.1.49 255.255.255.248 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#int s0/0/1 R3(config-if)#ip address 200.0.0.6 255.255.255.252 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#router eigrp 1 R3(config-router)#network 192.168.1.48 R3(config-router)#network 200.0.0.4 R3(config-router)#no auto-summary R3(config-router)#exit

**3.OSPF (Open Shortest Path First)**

Routers connect networks using the Internet Protocol (IP), and OSPF (Open Shortest Path First) is a [router](http://searchnetworking.techtarget.com/definition/router) [protocol](http://searchnetworking.techtarget.com/definition/protocol) used to find the best path for packets as they pass through a set of connected networks. OSPF is one of several Interior Gateway Protocols ([IGP](http://searchsecurity.techtarget.com/definition/IGP)s) -- that is, protocols aimed at traffic moving around within a larger [autonomous system](http://searchnetworking.techtarget.com/definition/autonomous-system) network like a single enterprise's network, which may in turn be made up of many separate local area networks linked through routers.

The OSPF routing protocol has largely replaced the older Routing Information Protocol (RIP) in corporate networks. Using OSPF, a router that learns of a change to a routing table (when it is reconfigured by network staff) or detects a change in the network, immediately [multicasts](http://searchnetworking.techtarget.com/definition/multicast) the information to all other OSPF hosts in the network so they will all have the same routing table information. Unlike RIP, which requires routers to send the entire routing table to neighbors every 30 seconds, OSPF sends only the part that has changed and only when a change has taken place. Rather than simply counting the number of router [hops](http://whatis.techtarget.com/definition/hop) between hosts on a network, as RIP does, OSPF bases its path choices on "link states" that take into account additional network information, including IT-assigned cost [metrics](http://whatis.techtarget.com/definition/metric) that give some paths higher assigned costs. For example, a satellite link may be assigned higher cost than a wireless WAN link, which in turn may be assigned higher cost than a metro Ethernet link.

**Important Terminologies**

**Router ID:** In OSPF this is a unique 32-bit number assigned to each router. This is chosen as the highest IP address on a router, and can be set large by configuring an address on a loopback interface of the chosen router.

**Neighbor Routers:** two routers with a common link that can talk to each other.

Adjacency: a two-way relationship between two neighbor routers. Neighbors don’t always form adjacencies.

**LSA:** Link State Advertisements are flooded; they describe routes within a given link.

**Hello Protocol:** this is how routers on a network determine their neighbors and form LSAs.

**Area:** a hierarchy. A set of routers that exchange LSAs, with others in the same area. An OSPF network can be divided into sub-domains called areas. An area is a logical collection of OSPF networks, routers, and links that have the same area identification. A router within an area must maintain a topological database for the area to which it belongs. The router does not have detailed information about network topology outside of its area. Therefore, routers in same area can communicate and routers with different areas cannot.

Areas limit the scope of route information distribution. It is not possible to do route update filtering within an area. The link-state database (LSDB) of routers within the same area must be synchronized and be exactly the same; however, route summarization and filtering is possible between different areas. The main benefit of creating areas is a reduction in the number of routes to propagate - by the filtering and the summarization of routes.

**Backbone Area:** An OSPF internetwork, whether or not it is subdivided into areas, always has at least one area called the backbone. The backbone has the reserved area ID of 0.0.0.0. The OSPF backbone area is also known as **area 0**.

The backbone acts as a hub for inter-area transit traffic and the distribution of routing information between areas. Inter-area traffic is routed to the backbone, then routed to the destination area, and finally routed to the destination host within the destination area.

**Wildcard Mask:**  A wildcard mask is a mask of bits that indicates which parts of an IP address are available for examination.

* Its Administrative Distance(AD) value is 110.
* It supports both VlSM & FLSM.
* Its Metric is COST( COST= bandwidth/speed of the interface)
* Wildcard Mask=255.255.255.255 - Subnet Mask of Network.

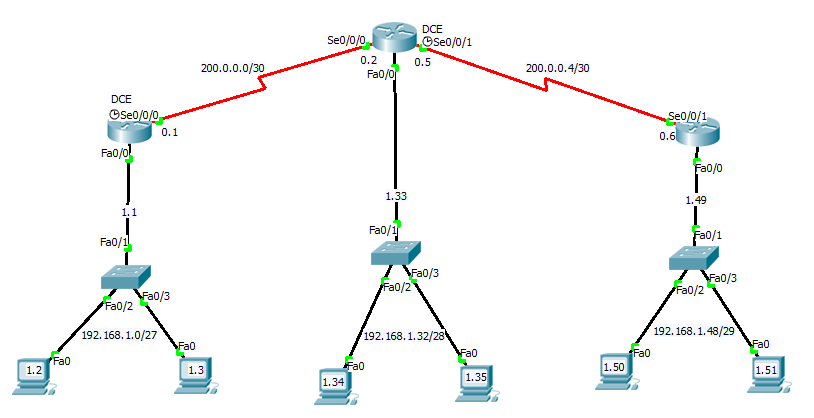
Ex: SM of **/27** is **255.255.255.224**

WM of **/27** is **0.0.0.31** (255.255.255.255 - 255.255.255.224)

**Syntax(routing commands):**

Router(config)#router ospf [ASN] Router(config-router)#router-id [id no]     Router(config-router)#network [net.adress] [wildcard mask] [area?] Router(config-router)#exit

**Practical**



Router>enable Router#configure terminal     Router(config)#hostname R1 R1(config)#int f0/0 R1(config-if)#ip address 192.168.1.1 255.255.255.244 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#int s0/0/0 R1(config-if)#ip address 200.0.0.1 255.255.255.252 R1(config-if)#clock rate 64000 R1(config-if)#no shutdown R1(config-if)#exit R1(config)#router ospf 1 R1(config-router)#router-id 1.1.1.1     R1(config-router)#network 192.168.1.0 0.0.0.31 area 0 R1(config-router)#network 200.0.0.0 0.0.0.3 area 0 R1(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R2 R2(config)#int f0/0 R2(config-if)#ip address 192.168.1.33 255.255.255.240 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/0 R2(config-if)#ip address 200.0.0.2 255.255.255.252 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#int s0/0/1 R2(config-if)#ip address 200.0.0.5 255.255.255.252 R2(config-if)#clock rate 64000 R2(config-if)#no shutdown R2(config-if)#exit R2(config)#router ospf 1 R2(config-router)#router-id 2.2.2.2     R2(config-router)#network 192.168.1.32 0.0.0.31 area 1 R2(config-router)#network 200.0.0.0 0.0.0.3 area 1 R2(config-router)#network 200.0.0.0 0.0.0.4 area 1 R2(config-router)#exit

Router>enable Router#configure terminal     Router(config)#hostname R3 R3(config)#int f0/0 R3(config-if)#ip address 192.168.1.49 255.255.255.248 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#int s0/0/1 R3(config-if)#ip address 200.0.0.6 255.255.255.252 R3(config-if)#no shutdown R3(config-if)#exit R3(config)#router ospf 1 R3(config-router)#router-id 3.3.3.3     R3(config-router)#network 192.168.1.48 0.0.0.31 area 1 R3(config-router)#network 200.0.0.4 0.0.0.3 area 1 R3(config-router)#exit

**Access Control List (ACL)**

Access control lists (ACLs) provide a means to filter packets by allowing a user to permit or deny IP packets from crossing specified interfaces. Just imagine you come to a fair and see the guardian checking tickets. He only allows people with suitable tickets to enter. Well, an access list’s function is same as that guardian.

Access lists filter network traffic by controlling whether packets are forwarded or blocked at the router’s interfaces based on the criteria you specified within the access list.

To use ACLs, the system administrator must first configure ACLs and then apply them to specific interfaces. There are 3 popular types of ACL: Standard, Extended and Named ACLs.

**Standard IP Access List**

Standard IP lists (1-99) only check source addresses of all IP packets.

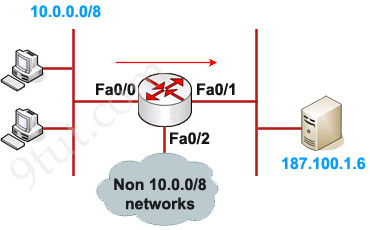
**Configuration Syntax**

|  |
| --- |
| **access-list** *access-list-number* {permit | deny} *source* {source-mask} |

Apply ACL to an interface

|  |
| --- |
| **ip access-group***access-list-number* {in | out} |

Example of Standard IP Access List



**Configuration:**

In this example we will define a standard access list that will only allow network 10.0.0.0/8 to access the server (located on the Fa0/1 interface)

**Define which source is allowed to pass:**

Router(config)#access-list 1 permit 10.0.0.0 0.255.255.255

(there is always an implicit deny all other traffic at the end of each ACL so we don’t need to define forbidden traffic)

**Apply this ACL to an interface:**

Router(config)#interface Fa0/1

Router(config-if)#ip access-group 1 out

The ACL 1 is applied to permit only packets from 10.0.0.0/8 to go out of Fa0/1 interface while deny all other traffic. So can we apply this ACL to other interface, Fa0/2 for example? Well we can but shouldn’t do it because users can access to the server from other interface (s0 interface, for example). So we can understand why an standard access list should be applied close to the destination.

Note: The “0.255.255.255” is the wildcard mask part of network “10.0.0.0”. We will learn how to use wildcard mask later.

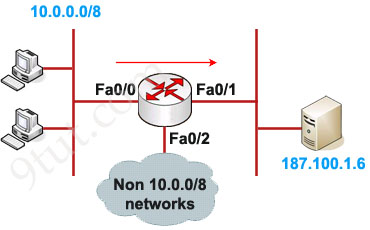
**Extended IP Access List**

Extended IP lists (100-199) check both source and destination addresses, specific UDP/TCP/IP protocols, and destination ports.

**Configuration Syntax**

|  |
| --- |
| **access-list***access-list-number* {permit | deny} *protocol* source {source-mask} destination {destination-mask} [eq destination-port] |

Example of Extended IP Access List



In this example we will create an extended ACL that will deny FTP traffic from network 10.0.0.0/8 but allow other traffic to go through.

Note: FTP uses TCP on port 20 & 21.

**Define which protocol, source, destination and port are denied:**

Router(config)#access-list 101 deny tcp 10.0.0.0 0.255.255.255 187.100.1.6 0.0.0.0 eq 21

Router(config)#access-list 101 deny tcp 10.0.0.0 0.255.255.255 187.100.1.6 0.0.0.0 eq 20

Router(config)#access-list 101 permit ip any any

**Apply this ACL to an interface:**

Router(config)#interface Fa0/1

Router(config-if)#ip access-group 101 out

Notice that we have to explicit allow other traffic (access-list 101 permit ip any any) as there is an “deny all” command at the end of each ACL.

As we can see, the destination of above access list is “187.100.1.6 0.0.0.0” which specifies a host. We can use “host 187.100.1.6” instead. We will discuss wildcard mask later.

In summary, below is the range of standard and extended access list

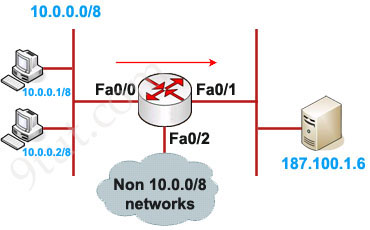
|  |  |
| --- | --- |
| **Access list type** | **Range** |
| Standard | 1-99, 1300-1999 |
| Extended | 100-199, 2000-2699 |

**Named IP Access List Configuration Syntax**

|  |
| --- |
| **ip access-list** {standard | extended} {name | number} |

Example of Named IP Access List

This is an example of the use of a named ACL in order to block all traffic except the Telnet connection from host 10.0.0.1/8 to host 187.100.1.6.



**Define the ACL:**

Router(config)#ip access-list extended in\_to\_out permit tcp host 10.0.0.1 host 187.100.1.6 eq telnet

(notice that we can use ‘telnet’ instead of port 23)

**Apply this ACL to an interface:**

Router(config)#interface Fa0/0

Router(config-if)#ip access-group in\_to\_out in

**Where to place access list?**

Standard IP access list should be placed close to destination.

Extended IP access lists should be placed close to the source.

**How many access lists can be used?**

You can have one access-list per protocol, per direction and per interface. For example, you cannotc have two access lists on the inbound direction of Fa0/0 interface. However you can have one inbound and one outbound access list applied on Fa0/0.

**How to use the wildcard mask?**

Wildcard masks are used with access lists to specify a host, network or part of a network.

The zeros and ones in a wildcard determine whether the corresponding bits in the IP address should be checked or ignored for ACL purposes. For example, we want to create a standard ACL which will only allow network 172.23.16.0/20 to pass through. We need to write an ACL, something like this:

**access-list 1 permit 172.23.16.0 255.255.240.0**

Of course we can’t write subnet mask in an ACL, we must convert it into wildcard mask by converting all bits 0 to 1 & all bits 1 to 0.

255 = 1111 1111 -> convert into 0000 0000

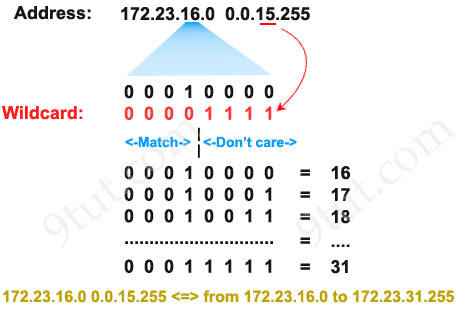
240 = 1111 0000 -> convert into 0000 1111

0 = 0000 0000 -> convert into 1111 1111

Therefore 255.255.240.0 can be written in wildcard mask as 00000000.00000000.00001111.11111111 = 0.0.15.255

Remember, for the wildcard mask, **1′s are I DON’T CARE, and 0′s are I CARE**. Now let’s analyze our wildcard mask.

Two first octets are all 0’s meaning that we care about the network **172**.**23**.x.x. The third octet, 15 (0000 1111 in binary), means that we care about first 4 bits but don’t care about last 4 bits so we allow the third octet in the form of 0001xxxx (minimum:0001**0000 =** 16; maximum: 000**1111** = 31).



The fourth octet is 255 (all 1 bits) that means I don’t care.

Therefore **network 172.23.16.0 0.0.15.255** ranges from **172.23.16.0** to **172.23.31.255**.

Some additional examples:

* Block TCP packets on port 30 from any source to any destination:

Router(config)#access-list 101 deny tcp any any eq 30

* Permit any IP packets in network 192.23.130.128 with subnet mask 255.255.255.248 to any network:

Router(config)#access-list 101 permit ip 192.23.130.128 0.0.0.7 any

Apply the access control list to an interface:

Router(config)#interface fastEthernet0/0

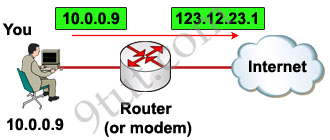
Router(config-if)#ip access-group 101 in .

**Network Address Translation (NAT)**

To go to the Internet we need to get an public IP address and it is unique all over the world. If each host in the world required a unique public IP address, we would have run out of IP address years ago. But by using Network Address Translation (NAT) we can save tons of IP addresses for later uses. We can understand NAT like this:

“NAT allows a host that does not have a valid registered IP address to communicate with other hosts through the Internet”

For example your computer is assigned a private IP address of 10.0.0.9 and of course this address cannot be routed on the internet but you can still access the internet. This is because your router (or modem) translates this address into a public IP address, 123.12.23.1 for example, before routing your data into the internet.



Of course when your router receives a reply packet destined for 123.12.23.1 it will convert back to your private IP 10.0.0.9 before sending that packet to you.

Maybe you will ask “hey, I don’t see any difference of using NAT to save tons of IP addresses because you still need a public IP address for each host to access the Internet and it doesn’t save you anything, why you need to use NAT?”

Ok, you are right :), in the above example we don’t see its usefulness but you now understand the fundamental of NAT!

Let’s take another example!

Suppose your company has 500 employees but your Internet Service Provider (ISP) only gives you 50 public IP addresses. It means that you can only allow 50 hosts to access the internet at the same time. Here NAT comes to save your life!

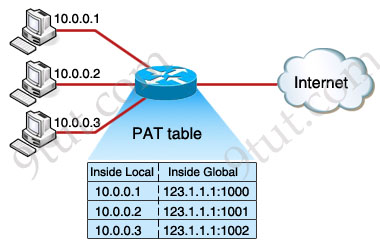
One thing you should notice that in real life, not all of your employees uses internet at the same time. Say, maybe 50 of them use internet to read newspaper at the morning; 50 others use internet at noon for checking mail… By using NAT you can dynamically assign these 50 public IP addresses to those who really need them at that time. This is called **dynamic NAT**.

But the above NAT solution does not solve our problem completely because in some days there can be more than 50 people surfing web at the morning. In this case, only the first 50 people can access internet, others must wait to their turns.

Another problem is, in fact, your ISP only gives you much lesser IP addresses than the number 50 because each public IP is very precious now.

To solve the two problems above, another feature of NAT can be used: **NAT Overload** or sometimes called **Port Address Translation** (PAT)

PAT permits multiple devices on a local area network (LAN) to be mapped to a single public IP address with different port numbers. Therefore, it’s also known as port address translation (PAT). When using PAT, the router maintains unique source port numbers on the **inside global** IP address to distinguish between translations. In the below example, each host is assigned to the same public IP address 123.1.1.1 1 but with different port numbers (from 1000 to 1002).



Note: Cisco uses the term **inside local** for the private IP addresses and **inside global** for the public IP addresses replaced by the router.

The outside host IP address can also be changed with NAT. The **outside global** address represents the outside host with a public IP address that can be used for routing in the public Internet.

The last term, **outside local** address, is a private address of an external device as it is referred to by devices on its local network. You can understand outside local address as the inside local address of the external device which lies at the other end of the Internet.

Maybe you will ask how many ports can we use for each IP? Well, because the port number ﬁeld has 16 bits, PAT can support about 216 ports, which is more than 64,000 connections using one public IP address.

Now you has learned all the most useful features of NAT but we should summary all features of NAT:

There are two types of NAT translation: dynamic and static.

**Static NAT**: Designed to allow one-to-one mapping between local and global addresses. This flavor requires you to have one real Internet IP address for every host on your network.

**Dynamic NAT:** Designed to map an unregistered IP address to a registered IP address from a pool of registered IP addresses. You don’t have to statically configure your router to map an inside to an outside address as in static NAT, but you do have to have enough real IP addresses for everyone who wants to send packets through the Internet. With dynamic NAT, you can configure the NAT router with more IP addresses in the inside local address list than in the inside global address pool. When being defined in the inside global address pool, the router allocates registered public IP addresses from the pool until all are allocated. If all the public IP addresses are already allocated, the router discards the packet that requires a public IP address.

**PAT (NAT Overloading):** is also a kind of dynamic NAT that maps multiple private IP addresses to a single public IP address (many-to-one) by using different ports. Static NAT and Dynamic NAT both require a one-to-one mapping from the inside local to the inside global address. By using PAT, you can have thousands of users connect to the Internet using only one real global IP address. PAT is the technology that helps us not run out of public IP address on the Internet. This is the most popular type of NAT.

Besides NAT gives you the option to advertise only a single address for your entire network to the outside world. Doing this effectively hides the internal network from the public world really well, giving you some additional security for your network.

**NAT terms:**

**\* Inside local address** – The IP address assigned to a host on the inside network. The address is usually not an IP address assigned by the Internet Network Information Center (InterNIC) or service provider. This address is likely to be an RFC 1918 private address.   
**\* Inside global address** – A legitimate IP address assigned by the InterNIC or service provider that represents one or more inside local IP addresses to the outside world.   
**\* Outside local address** – The IP address of an outside host as it is known to the hosts on the inside network.   
**\* Outside global address** – The IP address assigned to a host on the outside network. The owner of the host assigns this address.



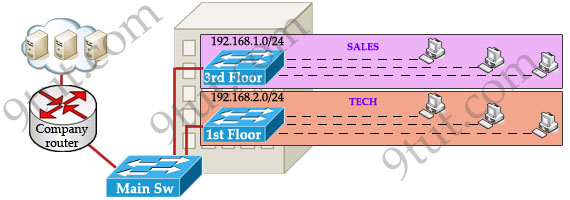
**Basics of SWITCH**

“A virtual LAN (VLAN) is a group of networking devices in the same broadcast domain, logically”

It means that the devices in the same VLAN may be widely separated in the network, both by geography and location. VLANs logically segment the network into different broadcast domains so that packets are only switched between ports that are designated for the same VLAN.

Let’s take an example to understand the benefits of VLAN. Suppose you are working in a big company with many departments, some of them are SALES and TECHNICAL departments. You are tasked to separate these departments so that each of them can only access specific resources in the company.

This task is really easy, you think. To complete this task, you just need to use different networks for these departments and use access-list to allow/deny that network to a specific resource. For example, you assign network 192.168.1.0/24 for SALES and 192.168.2.0/24 for TECH. At the “Company router” you apply an access-list to filter traffic from these networks. Below is the topology of your network without VLANs:



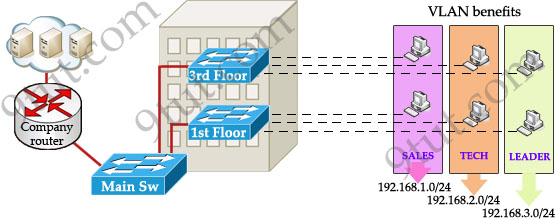
Everything looks good and you implement this design to your company. But after one month you receive many complaints from both your colleagues and leaders.

* First, your department leaders need to access to additional private resources which employees are not allowed.
* Second, the company has just recruited some new SALES employees but now the SALES room is full so they have to sit at the 1st floor (in the TECH area). They want to access to SALES resources but they can only access to the TECH resources because they are connecting to TECH switch.

To solve the first problem maybe you will create a new and more powerful network for your leaders. But notice that each leader sits at different floor so you will need to link all of them to a switch -> what a mess!

The second problem is more difficult than the first one. Maybe you have to create another network at the TECH area and apply the same policy as the SALES department for these hosts -> another mess in management!

Maybe you will be glad to know VLAN can solve all these problems. VLAN helps you group users together according to their function rather than their physical location. This means you can use the same network for hosts in different floors (of course they can communicate with each other).



In this design:

* you can logically create a new network with additional permissions for your leaders (LEADER network) by adding another VLAN.
* employees can sit anywhere to access the resources in their departments, provided that you allow them to do so.
* computers in the same department can communicate with each other although they are at different floors.

If these departments expand in the future you can still use the same network in any other floor. For example, SALES needs to have 40 more employees -> you can use 4th floor for this expansion without changing the current network.

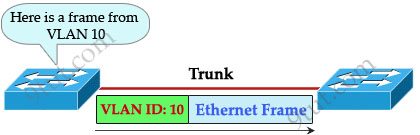
But wait… maybe you recognize something strange in the above design? How can 2 computers connecting to 2 different switches communicate? If one computer sends a broadcast packet will it be flooded to other departments as switch doesn’t break up broadcast domains?

The answer is “Yes, they can!” and it is the beauty of VLAN. Hosts in the same VLAN can communicate normally even they are connecting to 2 or more different switches. This makes the management much more simple.

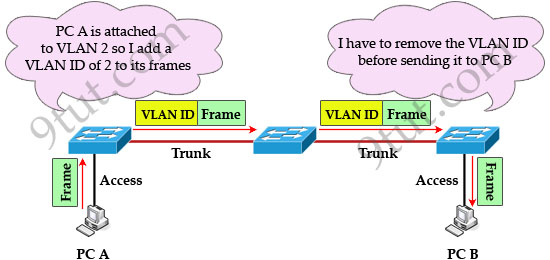
Although layer 2 switches can only break up collision domains but VLANs can be used to break up broadcast domains. So if a computer in SALES broadcasts, only computers in SALES will receive that frame.

So we don’t need a router, right? The answer is “we still need a router” to enable different VLANs to communicate with each other. Without a router, the computers within each VLAN can communicate with each other but not with any other computers in another VLAN. For example, we need a router to transfer file from LEADER to TECH. This is called “interVLAN routing”.

When using VLANs in networks that have multiple interconnected switches, you need to use **VLAN trunking between the switches**. With VLAN trunking, the switches tag each frame sent between switches so that the receiving switch knows which VLAN the frame belongs to. This tag is known as a VLAN ID. A VLAN ID is a number which is used to identify a VLAN.



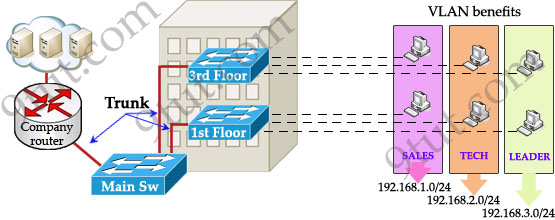
Notice that the tag is only added and removed by the switches when frames are sent out on the trunk links. Hosts don’t know about this tag because it is added on the first switch and removed on the last switch. The picture below describes the process of a frame sent from PC A to PC B.



Note: Trunk link does not belong to a specific VLAN, rather it is a conduit for VLANs between switches and routers.

To allow interVLAN routing you need to configure **trunking on the link between router and switch**.

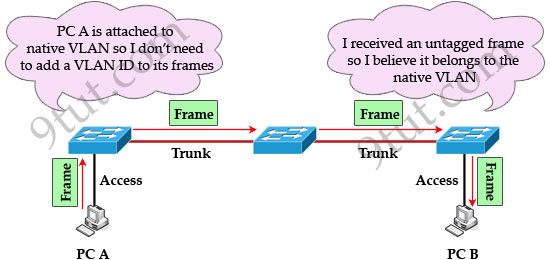
Therefore in our example we need to configure 3 links as “trunk”.



Cisco switches support two different trunking protocols, **Inter-Switch Link (ISL)** and **IEEE 802.1q**. Cisco created ISL before the IEEE standardized trunking protocol. Because ISL is Cisco proprietary, it can be used only between two Cisco switches -> 802.1q is usually used in practical.

In 802.1q encapsulation, there is a concept called native VLAN that was created for backward compatibility with old devices that don’t support VLANs. Native VLAN works as follows:

* Frame belonging to the native VLAN is not tagged when sent out on the trunk link
* Frame received untagged on the trunk link is set to the native VLAN.



So if an old switch doesn’t support VLAN it can still “understand” that frame and continue sending it (without dropping it).

Every port belongs to at least one VLAN. If a switch receives untagged frames on a trunkport, they are assumed to be part of the native vlan. By default, VLAN 1 is the default and native VLAN but this can be changed on a per port basis by configuration.

Cisco switches support two different trunking protocols, Inter-Switch Link (ISL) and IEEE 802.1q. In 802.1q, native VLAN frames are untagged.

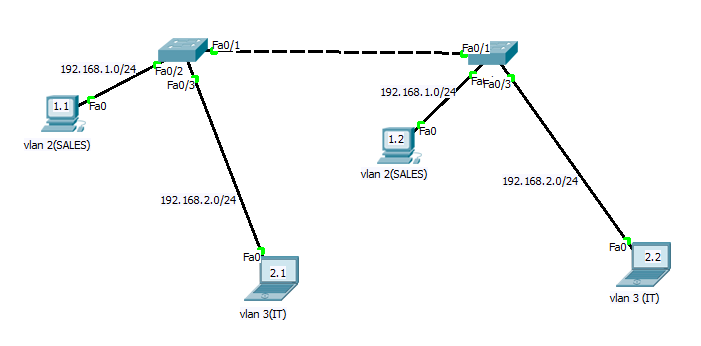
**The benefits of VLANs**

1. Segment networks into multiple smaller broadcast domains without Layer 3 network devices such as routers. VLANs make switched Ethernet networks more bandwidth-efficient through this segmentation of broadcast domains.

2. Group users together according to function rather than physical location. In a traditional network, users in a given work area are on the same network segment regardless of their job description or department. Using VLANs, however, you could have one salesperson in each work area of the building sitting next to engineers in their work area, yet on a separate logical network segment.

3. The ability to reconfigure ports logically without the need to unplug wires and move them around. If a user takes his or her computer to a new work area, no cables need to be swapped on the switch, just access the switch and issue commands to change the VLAN assignments for the old and new ports. VLANs thus simplify the process of adding, moving, and deleting users on the network. They also improve network security by avoiding cabling mishaps that can arise when users are moved in traditional Ethernet networks.

**Practical**



Switch>enable Switch#config t Switch(config)#hostname SW1 SW1(config)#vlan 2 SW1(config)#name SALES SW1(config)#vlan 3 SW1(config)#name IT SW1(config)#int f0/2 SW1(config)#switchport mode access SW1(config)#switchport access vlan 2 SW1(config)#int f0/3 SW1(config)#switchport mode access SW1(config)#switchport access vlan 3 SW1(config)#int f0/1 SW1(config)#switchport mode access SW1(config)#switchport mode trunk SW1(config)#exit

Switch>enable Switch#config t Switch(config)#hostname SW2 SW2(config)#vlan 2 SW2(config)#name SALES SW2(config)#vlan 3 SW2(config)#name IT SW2(config)#int f0/2 SW2(config)#switchport mode access SW2(config)#switchport access vlan 2 SW2(config)#int f0/3 SW2(config)#switchport mode access SW2(config)#switchport access vlan 3 SW2(config)#int f0/1 SW2(config)#switchport mode access SW2(config)#switchport mode trunk SW2(config)#exit

**VTP (VLAN Trunking Protocol)**

**“VTP allows a network manager to configure a switch so that it will propagate VLAN configurations to other switches in the network”**

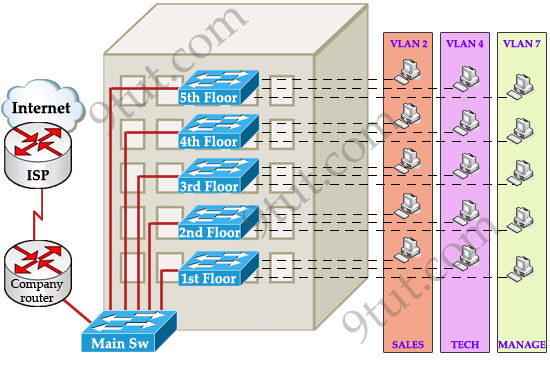
VTP minimizes misconfigurations and configuration inconsistencies that can cause problems, such as duplicate VLAN names or incorrect VLAN-type specifications. VTP helps you simplify management of the VLAN database across multiple switches.

VTP is a Cisco-proprietary protocol and is available on most of the Cisco switches.

**Why we need VTP?**

To answer this question, let’s discuss a real and popular network topology.

Suppose you are working in a medium company in a 5-floor office. You assigned each floor to a switch for easy management and of course they can be assigned to different VLANs. For example, your bosses can sit in any floor and still access Manage VLAN (VLAN 7). Your technical colleagues can sit anywhere on the floors to access Technical VLAN (VLAN 4). This is the best design because each person’s permission is not limited by the physical location.



Now let’s discuss about VTP role in this topology! Suppose VTP is not running on these switches. One day, your boss decides to add a new department to your office, the Support Department, and you are tasked to add a new SUPPORT VLAN for this department. How will you do that? Well, without VTP you have to go to each switch to enable this new VLAN. Fortunately your office only has 5 floors so you can finish this task in some hours :)

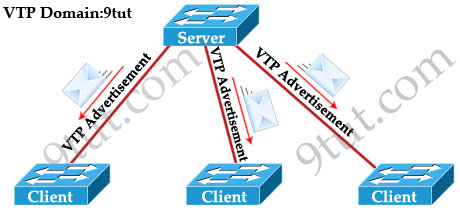
But just imagine if your company was bigger with 100-floor office and some VLANs needed to be added every month! Well, it will surely become a daunting task to add a new VLAN like this. Luckily, Cisco always “thinks big” to create a method for you to just sit at the “Main Sw”, adding your new VLANs and magically, other switches automatically learn about this VLAN, sweet, right? It is not a dream, it is what VTP does for you!

**How VTP Works**

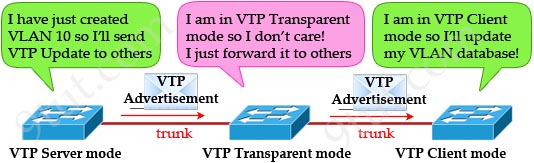
To make switches exchange their VLAN information with each other, they need to be configured in the same **VTP domain**. Only switches belonging to the same domain share their VLAN information. When a change is made to the VLAN database, it is propagated to all switches via **VTP advertisements**.

To maintain domain consistency, only one switch should be allowed to create (or delete, modify) new VLANs. This switch is like the “master” of the whole VTP domain and it is operated in **Server mode**. This is also the default mode.

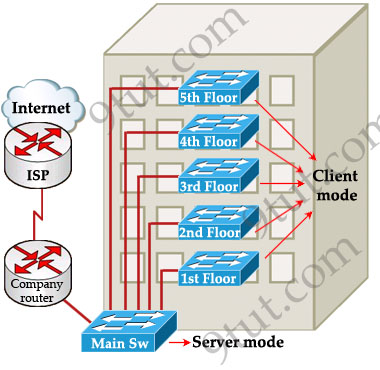
Other switches are only allowed to receive and forward updates from the “server” switch. They are operated in **Client mode**. Switches in this mode cannot create, delete or modify VLANs.



In some cases, the network manager doesn’t want a switch to learn VTP information from other switches. He can set it to **Transparent mode**. In this mode, a switch maintains its own VLAN database and never learn any VTP information from other switches (even from the switch in VTP server mode). However, it still forwards VTP advertisements from the server to other switches (but doesn’t read that advertisement). A transparent switch can add, delete and modify VLAN database locally.



Now return to the example above, we can configure any switches as the “server” but for our convenience, the “Main Sw” should be assigned this function and we should place it in a safe place.



As said above, VTP advertisements bring VLAN information to all the switches in a VTP domain. Each VTP advertisement is sent with a **Revision number**. This number is used in order to determine whether the VTP advertisement is more recent than the current version of that switch. Because each time you make a VLAN change in a switch, the configuration revision is incremented by one. So the higher the revision number, the better your VTP advertisement.

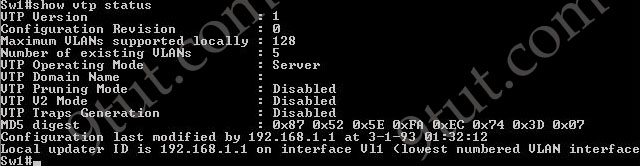
For example, the first time the Main Sw sends a VTP advertisement, its Revision number is 1. When you add a new VLAN to the Main Sw, it will send a VTP advertisement with the Revision number of 2. Client switches first receive the VTP advertisement with the Revision number of 1, which is bigger than its current Revision number (0) so it updates its VLAN database. Next it receives the VTP advertisement with the Revision number of 2, it continues comparing with its current Revision number (1) -> it continues update its VLAN database.

One important thing you must know is when a switch receives a better VTP advertisement, it deletes its whole VTP information and copy the new information from the better VTP advertisement to its VLAN database. A switch does not try to compare its own VLAN database with information from the received VTP advertisements to find out and update the difference!

Note: VTP advertisements are sent as multicast frames and all neighbors in that domain receive the frames.

**The “show vtp status” command analysis**

The most important command to view the status of VTP on Cisco switches that each CCNA learners must grasp is the “show vtp status” command. Let’s have a look at the output of this command:



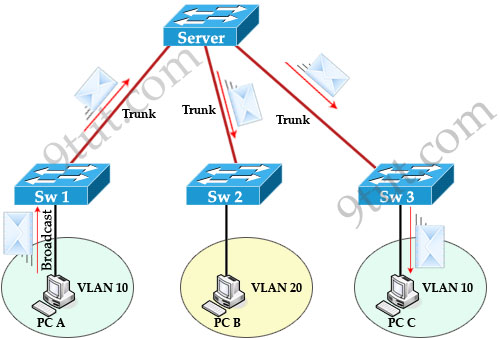
* **VTP Version**: displays the VTP version the switch is running. By default, the switch runs version 1 but can be set to version 2. Within a domain, the two VTP versions are not interoperable so make sure to configure the same VTP version on every switch in a domain.
* **Configuration Revision**: current Revision number on this switch
* **Maximum VLANs Supported Locally**: maximum number of VLANs supported locally.
* **Number of Existing VLANs**: Number of existing VLANs.
* **VTP Operating Mode**: can be server, client, or transparent.
* **VTP Domain Name**: name that identifies the administrative domain for the switch.

By default, a switch operates in VTP Server mode with a NULL (blank) domain name with no password configured (the password field is not listed in the output)

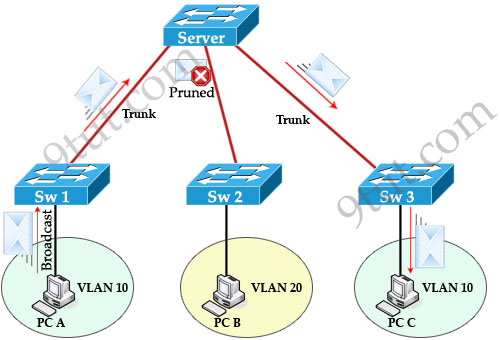
* **VTP Pruning Mode**: displays whether pruning is enabled or disabled. We will discuss about VTP Pruning later.
* **VTP V2 Mode**: displays if VTP version 2 mode is enabled. VTP version 2 is disabled by default.
* **VTP Traps Generation**: displays whether VTP traps are sent to a network management station.
* **MD5 Digest**: a 16-byte checksum of the VTP configuration
* **Configuration Last Modified**: date and time of the last configuration modification. Displays the IP address of the switch that caused the configuration change to the database.

**VTP Pruning**

To understand what VTP Pruning is, let’s see an example:



When PC A sends a broadcast frame on VLAN 10, it travels across all trunk links in the VTP domain. Switches Server, Sw2, and Sw3 all receive broadcast frames from PC A. But only Sw3 has user on VLAN 10 and it is a waste of bandwidth on Sw2. Moreover, that broadcast traffic also consumes processor time on Sw2. The link between switches Server and Sw2 does not carry any VLAN 10 traffic so it can be “pruned”.



VTP Pruning makes more efficient use of trunk bandwidth by forwarding broadcast and unknown unicast frames on a VLAN only if the switch on the receiving end of the trunk has ports in that VLAN. In the above example, Server switch doesn’t send broadcast frame to Sw2 because Sw2 doesn’t have ports in VLAN 10.

When a switch has a port associated with a VLAN, the switch sends an advertisement to its neighbors to inform that it has active ports on that VLAN. For example, Sw3 sends an advertisement to Server switch to inform that it has active port for VLAN 10. Sw2 has not advertised about VLAN 10 so Server switch will prune VLAN 10 on the trunk to Sw2.

You only need to enable pruning on one VTP server switch in the domain.

**VTP Configuration**

**Main Sw(config)#vtp version 2  
Main Sw(config)#vtp domain nsg  
Main Sw(config)#vtp mode server  
Main Sw(config)#vtp password keepitsecret**

On client switches

**Client(config)#vtp version 2  
Client(config)#vtp domain nsg  
Client(config)#vtp password keepitsecret  
Client(config)#vtp mode client**

Notice: Before configuring VTP make sure the links between your switches are trunk links. Your trunk link can automatically be formed if both of your switches are not 2960 or 3560 because ports on the 2960 and 3560 switches are set to dynamic auto by default. If both sides are set to dynamic auto, the link will remain in access mode. To configure trunk between these ports, use these commands:

**Client(config)#interface fa0/1** (or the interface on the link you want to be trunk)  
**Client(config-if)#switchport mode trunk**

These commands only need to be used on one of two switches to form the trunk.

**Below summaries important notes about VTP:**

* Whenever a change occurs in the VLAN database, the VTP server increments its configuration revision number and then advertises the new revision throughout the VTP domain via VTP advertisements.

**VTP modes:**

\* Server: The default mode. When you make a change to the VLAN configuration on a VTP server, the change is propagated to all switches in the VTP domain. VTP messages are transmitted out of all the trunk connections. In Server mode we can create, modify, delete VLANs.

\* Client: cannot make changes to the VLAN configuration when in this mode; however, a VTP client can send any VLANs currently listed in its database to other VTP switches. VTP client also forwards VTP advertisements (but cannot create VTP advertisements).

\* Transparent: When you make a change to the VLAN configuration in this mode, the change affects only the local switch and does not propagate to other switches in the VTP domain. VTP transparent mode does forward VTP advertisements that it receives within the domain.

VTP Pruning makes more efficient use of trunk bandwidth by forwarding broadcast and unknown unicast frames on a VLAN only if the switch on the receiving end of the trunk has ports in that VLAN.