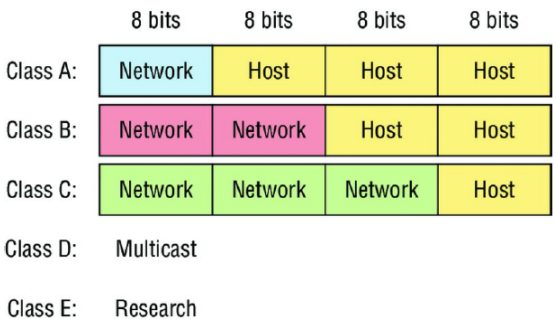
IPv4 Subnetting:

An IP address consists of 32 bits of information. These bits are divided into four sections, referred to as octets or bytes, with each containing 1 byte (8 bits). You can depict an IP address using one of three methods:
Dotted-decimal, as in 172.16.30.56
Binary, as in 10101100.00010000.00011110.00111000
Hexadecimal, as in AC.10.1E.38
The network address (which can also be called the network number) uniquely identifies each network.



The node/host IP address is assigned to, and uniquely identifies, each machine on a network.

Starting Octet shows which class the IP belongs to:

Class A: 0 - 126
Loopbak: 127
Class B: 128 - 191
Class C: 192 - 223
Class D: 224 - 239
Class E: 240 – 255

Address Class	Reserved Address Space
Class A	10.0.0.0 through 10.255.255.255
Class B	172.16.0.0 through 172.31.255.255
Class C	192.168.0.0 through 192.168.255.255

Class	Format	Default Subnet Mask
A	*network.node.node.node*	255.0.0.0
B	*network.network.node.node*	255.255.0.0
C	*network.network.network.node*	255.255.255.0

$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$
$2^9 = 512$
$2^{10} = 1,024$
$2^{11} = 2,048$
$2^{12} = 4,096$
$2^{13} = 8,192$
$2^{14} = 16,384$

Loopback (localhost) Used to test the IP stack on the local computer. Can be any address from 127.0.0.1 through 127.255.255.254.
Broadcasts (layer 3) These are sent to all nodes on the network.
Unicast This is an address for a single interface, and these are used to send packets to a single destination host.
Multicast These are packets sent from a single source and transmitted to many devices on different networks. Referred to as “one-to-many.”

Classless inter-domain routing (CIDR) is a set of Internet protocol (IP) standards that is used to create unique identifiers for networks and individual devices. The IP addresses allow particular information packets to be sent to specific computers. Classless interdomain routing (CIDR) helps reduce the size of routing tables by aggregating routes, and Network Address Translation (NAT), which reduces the number of required public IP addresses used by each organization or company.
The primary goal of CIDR is to improve the scalability of Internet routers’ routing tables. Imagine the implications of an Internet router being burdened by carrying a route to every class A, B, and C network on the planet.

Subnet Mask	CIDR Value
255.0.0.0	/8
255.128.0.0	/9
255.192.0.0	/10
255.224.0.0	/11
255.240.0.0	/12
255.248.0.0	/13
255.252.0.0	/14
255.254.0.0	/15
255.255.0.0	/16
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30

CIDR uses both technical tools and administrative strategies to reduce the size of the Internet routing tables. Technically, CIDR uses route summarization, but with Internet scale in mind. For example, CIDR might be used to allow a large ISP to control a range of IP addresses from 198.0.0.0 to 198.255.255.255.
ISPs 2, 3, and 4 need only one route (198.0.0.0/8) in their routing tables to be able to forward packets to all destinations that begin with 198. Note that this summary actually summarizes multiple class C networks—a typical feature of CIDR. ISP 1’s routers contain more detailed routing entries for addresses beginning with 198, based on where they allocate IP addresses for

their customers. ISP 1 would reduce its routing tables similarly with large ranges used by the other ISPs.
CIDR attacks the problem of large routing tables through administrative means as well. As shown in Figure 4-5 , ISPs are assigned contiguous blocks of addresses to use when assigning addresses for their customers. Likewise, regional authorities are assigned large address blocks, so when individual companies ask for registered public IP addresses, they ask their regional registry to assign them an address block. As a result, addresses assigned by the regional agency will at least be aggregatable into one large geographic region of
the world.
CIDR refers to the administrative assignment of large address blocks, and the related summarized routes, for the purpose of reducing the size of the Internet routing tables.
Note: Because CIDR defines how to combine routes for multiple classful networks into a single route, some people think of this process as being the opposite of subnetting. As a result, many people refer to CIDR's summarization results as supernetting.

Private Addressing:

One of the issues with Internet growth was the assignment of all possible network numbers to a small number of companies or organizations. Private IP addressing helps to mitigate this problem by allowing computers that will never be directly connected to the Internet to not use public, Internet-routable addresses. For IP hosts that will purposefully have no direct Internet connectivity, you can use several reserved network numbers. In other words, any organization can use these network numbers. However, no organization is allowed to advertise these networks using a routing protocol on the Internet. Furthermore, all Internet routers should be configured to reject these routes.

Table 4-12 *RFC 1918 Private Address Space*

Range of IP Addresses	Class of Networks	Number of Networks
10.0.0.0 to 10.255.255.255	A	1
172.16.0.0 to 172.31.255.255	B	16
192.168.0.0 to 192.168.255.255	C	256

Class C Subnettng:

Binary	Decimal	CIDR
00000000	= 255.255.255.0	/24
10000000	= 255.255.255.128	/25
11000000	= 255.255.255.192	/26
11100000	= 255.255.255.224	/27
11110000	= 255.255.255.240	/28
11111000	= 255.255.255.248	/29
11111100	= 255.255.255.252	/30

Subnetting Steps:

1. How many subnets 2^x?
2. How many hosts per subnet 2^y - 2?
3. What are the valid subnets (256 - subnet_mask = each subnet block size)?
4. What is the broadcast address of each subnet?
5. What are valid hosts?

Example1:

192.168.10.0 = Network address
255.255.255.128 = Subnet mask /25
1. 2^1 = 2 subnets.
2. 2^7 - 2 = 126 hosts per subnet.
3. 256 - 128 = 128 block size

192.168.10.0 and 192.168.10.128 are two subnets.

4. 192.168.10.127 and 192.168.10.255 are broadcast addresses.
5. 192.168.10.1-126 and 192.168.10.129-254 are valid hosts.

Example2:

192.168.10.0 = Network address
255.255.255.252 = Subnet mask /30

1. 2^6 = 64 subnets.
2. 2^2 - 2 = 2 hosts per subnet.
3. 256 - 252 = 4 block size.

Subnet	0	4	8	12	...	240	244	248	252
First host	1	5	9	13	...	241	245	249	253
Last host	2	6	10	14	...	242	246	250	254
Broadcast	3	7	11	15	...	243	247	251	255

/30 is usually used by Point-to-point WAN links.

Class B Subnetting:

255.255.0.0 (/16)
255.255.128.0 (/17) 255.255.255.0 (/24)
255.255.192.0 (/18) 255.255.255.128 (/25)
255.255.224.0 (/19) 255.255.255.192 (/26)
255.255.240.0 (/20) 255.255.255.224 (/27)
255.255.248.0 (/21) 255.255.255.240 (/28)
255.255.252.0 (/22) 255.255.255.248 (/29)
255.255.254.0 (/23) 255.255.255.252 (/30)

Example1:

172.16.0.0 = Network address
255.255.254.0 = Subnet mask /23

1. 2^7 = 128 subnets.
2. 2^9 - 2 = 510 hosts per subnet.
3. 256 - 254 = 2 block size.

valid subnets are 0, 2, 4, 6, 8, etc., up to 254.
First five subnets:

Subnet	0.0	2.0	4.0	6.0	8.0
First host	0.1	2.1	4.1	6.1	8.1
Last host	1.254	3.254	5.254	7.254	9.254
Broadcast	1.255	3.255	5.255	7.255	9.255

Example2:

172.16.0.0 = Network address
255.255.255.0 = Subnet mask /24

1. 2^8 = 256 subnets.
2. 2^8 - 2 = 256 hosts per subnet.
3. 256 - 255 = 1 block size.

Valid subnets are 0, 1, 2, 3, etc., all the way to 255.

Subnet	0.0	1.0	2.0	3.0	...	254.0	255.0
First host	0.1	1.1	2.1	3.1	...	254.1	255.1
Last host	0.254	1.254	2.254	3.254	...	254.254	255.254
Broadcast	0.255	1.255	2.255	3.255	...	254.255	255.255

Example3:

172.16.0.0 = Network address
255.255.255.224 = Subnet mask /27

1. 2^11 = 2048.
2. 2^5 - 2 = 30 hosts per subnet.
3. 256 - 224 = 32 block size.

Valid subnets are 0, 32, 64, 96, 128, 160, 192, 224.

Subnet	0.0	0.32	0.64	0.96	0.128	0.160	0.192	0.224
First host	0.1	0.33	0.65	0.97	0.129	0.161	0.193	0.225
Last host	0.30	0.62	0.94	0.126	0.158	0.190	0.222	0.254
Broadcast	0.31	0.63	0.95	0.127	0.159	0.191	0.223	0.255

Subnet	255.0	255.32	255.64	255.96	255.128	255.160	255.192	255.224
First host	255.1	255.33	255.65	255.97	255.129	255.161	255.193	255.225
Last host	255.30	255.62	255.94	255.126	255.158	255.190	255.222	255.254
Broadcast	255.31	255.63	255.95	255.127	255.159	255.191	255.223	255.255

Class A Subnetting:

255.0.0.0 (/8)
255.128.0.0 (/9) 255.255.240.0 (/20)
255.192.0.0 (/10) 255.255.248.0 (/21)
255.224.0.0 (/11) 255.255.252.0 (/22)
255.240.0.0 (/12) 255.255.254.0 (/23)
255.248.0.0 (/13) 255.255.255.0 (/24)
255.252.0.0 (/14) 255.255.255.128 (/25)
255.254.0.0 (/15) 255.255.255.192 (/26)
255.255.0.0 (/16) 255.255.255.224 (/27)
255.255.128.0 (/17) 255.255.255.240 (/28)
255.255.192.0 (/18) 255.255.255.248 (/29)
255.255.224.0 (/19) 255.255.255.252 (/30)

Example1:

10.0.0.0
255.255.0.0 (/16)
1. Subnets? 28 = 256.
2. Hosts? 216 – 2 = 65,534.
3. Valid subnets? What is the interesting octet? 256 – 255 = 1. 0, 1, 2, 3, etc. (all in the second octet). The subnets would be 10.0.0.0, 10.1.0.0, 10.2.0.0, 10.3.0.0, etc., up to 10.255.0.0.

Subnet	10.0.0.0	10.1.0.0	...	10.254.0.0	10.255.0.0
First host	10.0.0.1	10.1.0.1	...	10.254.0.1	10.255.0.1
Last host	10.0.255.254	10.1.255.254	...	10.254.255.254	10.255.255.254
Broadcast	10.0.255.255	10.1.255.255	...	10.254.255.255	10.255.255.255

Example2:

10.0.0.0
255.255.240.0 (/20)
1. Subnets? 212 = 4096.
2. Hosts? 212 – 2 = 4094.
3. Valid subnets? What is your interesting octet? 256 – 240 = 16. The subnets in the second octet are a block size of 1 and the subnets in the third octet are 0, 16, 32, etc.

Subnet	10.0.0.0	10.0.16.0	10.0.32.0	...	10.255.240.0
First host	10.0.0.1	10.0.16.1	10.0.32.1	...	10.255.240.1
Last host	10.0.15.254	10.0.31.254	10.0.47.254	...	10.255.255.254
Broadcast	10.0.15.255	10.0.31.255	10.0.47.255	...	10.255.255.255

Example3:

10.0.0.0
255.255.255.192 (/26)
1. Subnets? 218 = 262,144.
2. Hosts? 26 – 2 = 62.
3. Valid subnets? In the second and third octet, the block size is 1, and in the fourth octet, the block size is 64.

Subnet	10.0.0.0	10.0.0.64	10.0.0.128	10.0.0.192
First host	10.0.0.1	10.0.0.65	10.0.0.129	10.0.0.193
Last host	10.0.0.62	10.0.0.126	10.0.0.190	10.0.0.254
Broadcast	10.0.0.63	10.0.0.127	10.0.0.191	10.0.0.255

Subnet	10.255.255.0	10.255.255.64	10.255.255.128	10.255.255.192
First host	10.255.255.1	10.255.255.65	10.255.255.129	10.255.255.193
Last host	10.255.255.62	10.255.255.126	10.255.255.190	10.255.255.254
Broadcast	10.255.255.63	10.255.255.127	10.255.255.191	10.255.255.255

VLSM:

Create many networks from a large single network using subnet masks of different lengths in various kinds of network designs is called VLSM (Variable Length Subnet Mask) networking.

192.168.10.0 = Network
255.255.255.240 (/28) = Mask

Prefix	Mask	Hosts	Block Size
/25	128	126	128
/26	192	62	64
/27	224	30	32
/28	240	14	16
/29	248	6	8
/30	252	2	4

Example1:

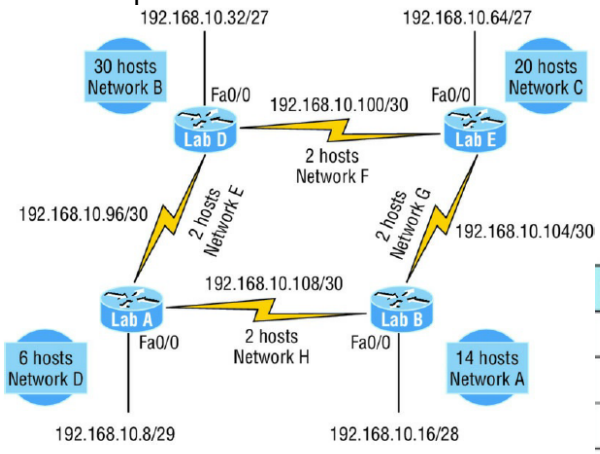
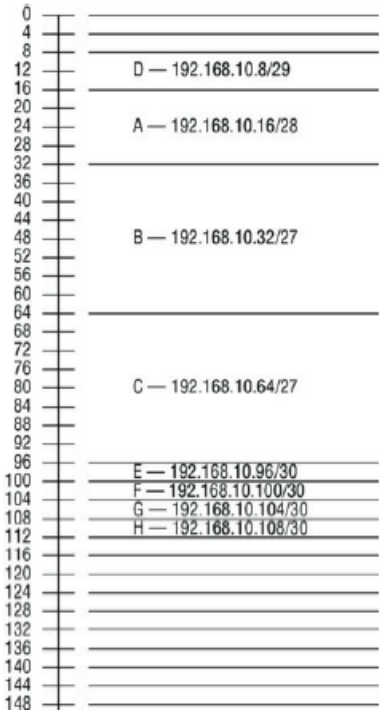


Figure 5.4 VLSM network example 1

Subnet	Mask	Subnets	Hosts	Block
/25	128	2	126	128
/26	192	4	62	64
/27	224	8	30	32
/28	240	16	14	16
/29	248	32	6	8
/30	252	64	2	4

Network	Hosts	Block	Subnet	Mask
A	14	16	/28	240
B	30	32	/27	224
C	20	32	/27	224
D	6	8	/29	248
E	2	4	/30	252
F	2	4	/30	252
G	2	4	/30	252
H	2	4	/30	252



Example2:

A: /27
B: /28
C: /28
D: /30
E: /30
F: /30
G: /28
H: /26
I: /28
J: /26
K: /28

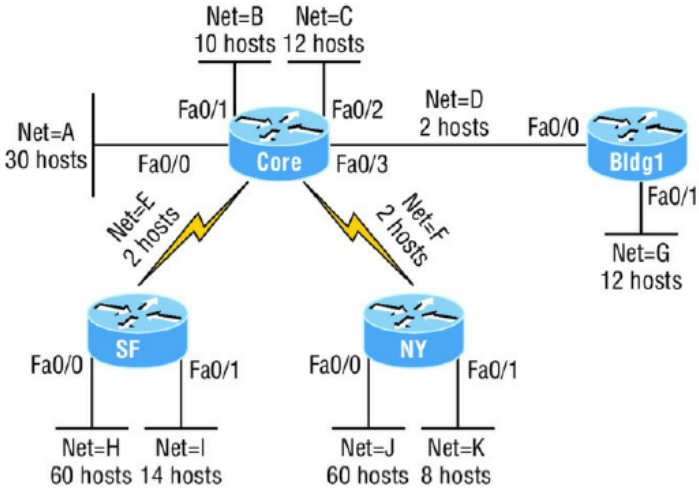
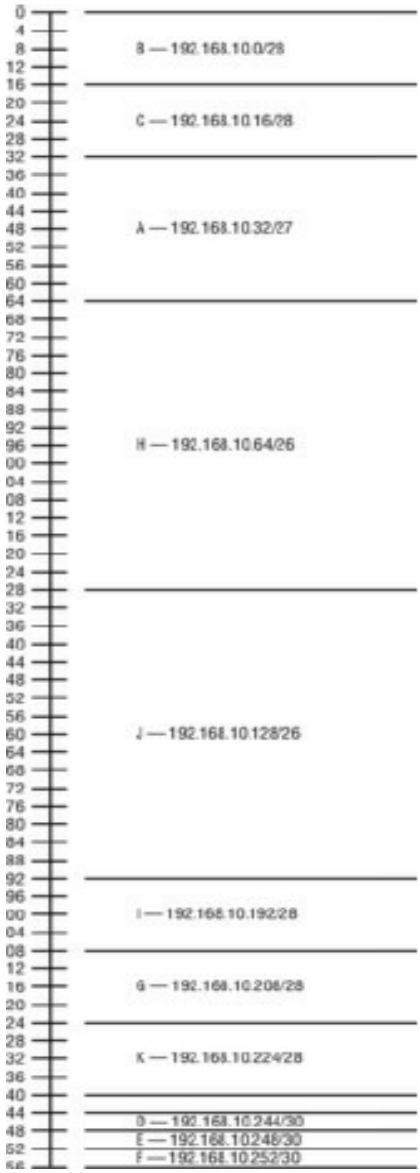


Figure 5.6 VLSM network example 2



192.168.10.0/28 - B
192.168.10.16/28 - C
192.168.10.32/27 - A
192.168.10.64/26 - H
192.168.10.128/26 - J
192.168.10.192/28 - I
192.168.10.208/28 - G
192.168.10.224/28 - K
192.168.10.244/30 - D
192.168.10.248/30 - E
192.168.10.252/30 - F

Route summarization (also called route aggregation or supernetting):

Example1:

Let's say we want to create the most optimal summary for the following 4 networks:
192.168.0.0 / 24 subnet mask 255.255.255.0
192.168.1.0 / 24 subnet mask 255.255.255.0
192.168.2.0 / 24 subnet mask 255.255.255.0
192.168.3.0 / 24 subnet mask 255.255.255.0
Let's convert these network addresses to binary:

192.168.0.0	11000000	10101000	00000000	00000000
192.168.1.0	11000000	10101000	00000001	00000000
192.168.2.0	11000000	10101000	00000010	00000000
192.168.3.0	11000000	10101000	00000011	00000000

Now we have to look how many bits these network addresses have in common. The first and second octets are the same, so that's 16 bits.
Let's zoom in on the third octet:
00000000
00000001
00000010
00000011
Our summary address will be 192.168.0.0 /22 (subnet mask 255.255.252.0).

Example2:

Let's look at another example. Let's say we want to summarize the following networks:
172.16.0.0 / 16 subnet mask 255.255.0.0
172.17.0.0 / 16 subnet mask 255.255.0.0
172.18.0.0 / 16 subnet mask 255.255.0.0
172.19.0.0 / 16 subnet mask 255.255.0.0
172.20.0.0 / 16 subnet mask 255.255.0.0
172.21.0.0 / 16 subnet mask 255.255.0.0
172.22.0.0 / 16 subnet mask 255.255.0.0
172.23.0.0 / 16 subnet mask 255.255.0.0
Let's look at it in binary first. I'll write down the second octet since the first one is the same for all network addresses:

16	00010000
17	00010001
18	00010010
19	00010011
20	00010100
21	00010101
22	00010110
23	00010111

The first 5 bits for all these addresses are the same. The first octet had 8 similar bits so that's 8 + 5 = 13 bits.
The summary address will be 172.16.0.0 /13 (subnet mask will be 255.248.0.0).

Router Modes:

Router>: User mode = Limited to basic monitoring commands
Router#: Privileged mode (exec-level mode) = Provides access to all other router commands
Router(config)#: global configuration mode = Commands that affect the entire system
Router(config-if)#: interface mode = Commands that affect interfaces
Router(config-subif)#: subinterface mode = Commands that affect subinterfaces
Router(config-line)#: line mode = Commands that affect in lines modes (console, vty, aux…)
Router(config-router)#: router configuration mode

Interface Configuration:

Router(config)#default int range fa 0/0 - 1 !(to clear all int config back to default)
Router(config)#default int range fa 0/0 – 1, fa 0/4 - 5
Router(config)#int fa 0/0
Router(config-if)#mac-address 0000.1111.1111 !(hard code a mac address for ease of use)
Router(config-if)#ip address 192.168.1.1 255.255.255.0 !(or 'ip address dhcp')
sh int bri
sh int status
sh int fa0/1
sh run int fa0/1

enable, disable, conf t, end, exit(exec), exit(global), logout commands:

Router(config)# do sh run !(use do to run any show command from any mode)
Router> enable
Password: <letmein>
Router# disable
Router>
Router# configure terminal
Router(config)# interface serial 1:1
Router(config-if)# alps ascu 4B
Router(config-alps-ascu)# end
Router# show interface serial 1:1
Router(config)# exit
Router# disable
Router> exit
Router(config-subif)# exit
Router(config-if)#
Router(config)# exit
Router# disable
Router> logout

Saving config to NVRAM:

Router#copy running-config startup-config !(copy run start)
Router# write mem !(wr)

Keyboard shortcuts:

Ctrl+A !(move to the start of the line)
Ctrl+E !(move to the end of the line)
Ctrl+shift+6 !(stop traceroute or ping)

Terminal command:

terminal length 24
terminal history size 256 !(can be a global command or under line console/vty)
show history

Verifying commands:

sh version
sh flash:
sh run !(default commands don't show in run)
sh start
sh post
sh arp
sh hosts
sh users !(to see the users who are loggedin to vty)
clear line vty 0 !(to clear the loggedin user on a specific line)
sh processes cpu
sh processes cpu sort !(more like top command in linux)
sh process cpu history !(utilization over time)
sh ip int bri

Pipe parameter:

sh run | include hostname
sh run | include revision| modified
sh run | section fa 0/1
sh run | begin line con 0
sh run | exclude interface

Resetting switch config:

delete flash:vlan.dat
erase start
reload !('reload in 5' command will reload in 5 mins)

Configuring switch/router to use SSH:

SW1(config)#hostname SW1
SW1(config)# ip domain-name example.com !(encryption keys are associated to FQDN domain names)
SW1(config)# username admin password cisco
SW1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
SW1(config)# ip ssh version 2
SW1(config)# line vty 0 4
SW1(config-line)# login local !(ssh required username and password both. You must create local username and password prior to setting this command)
SW1(config-line)# transport input telnet ssh
SW1# show crypto key mypubkey rsa

Aliases:

SW1(config)# alias exec c configure terminal
SW1(config)# alias exec s show ip interface brief
SW1(config)# alias exec sr show running-config

Capturing wireshark packets on a router and export to tftp as pcap:

R1#monitor capture 1 int fa0/0 both !(options: both | in | out)
R1#monitor capture 1 match any
R1#monitor capture 1 start
R1#monitor capture 1 export tftp://10.0.0.100/r1.marking.pcap
R1#no monitor capture 1

Test or verification by generating http traffic converting a router into a http server:

R1(config)#username ali password cisco
R1(config)#username privilege 15
R1(config)#ip http server
R1(config)#ip http authentication local
R1(config)#ip http path bootflash:
R2#copy http://ali:cisco@R1_IP/FILE_NAME null:

ping, traceroute, ssh and telnet:

R1(config)#ip telnet tos B8
R1#ping !(extended ping)
R1#ping 8.8.8.8 source fa0/0
R1#traceroute !(extended traceroute)
R1#traceroute 8.8.8.8
R1#telnet bing.com 80
R1#telnet 192.168.1.1 /source-interface loopback0
R1#ssh -l root 192.168.1.1

mac address table:

sh mac-address-table
sh mac-address-table | include 9fe7
sh mac-address-table interface fastEthernet 1/0/48
sh mac-address-table count
clear mac address-table dynamic
debug matm all !(to see everything mac address table management switch does)

ARP:

arp !('arp -a' for windows CLI)
clear ip arp
arp –a

using ACL with a debug command for tshoot:

R#access-list 1 permit host 10.0.0.2
R#debug ip packet 1 detail

Basic switch/router setup commands:

SW#setup
Switch(config)# hostname SW1
SW1(config)# enable secret cisco !(MD5 hash)
SW1(config)# enable password notcisco !(Clear text)
SW1(config)# line console 0
SW1(config-line)# password cisco
SW1(config-line)# login
SW1(config)# line vty 0 4
SW1(config-line)# password cisco
SW1(config-line)# login
SW1(config)# service password-encryption !(to encrypt all the passwords in the config, but is weak type 7 encryption and can be cracked easily)
SW1(config)# banner motd $
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
UNAUTHORIZED ACCESS IS PROHIBITED
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
$
SW1(config)# interface vlan 1

SW1(config-if)# ip address 172.16.1.11 255.255.255.0 !(or DHCP)
SW1(config-if)# no shutdown
SW1(config)# ip default-gateway 172.16.1.1
SW1# copy running-config startup-config
SW1# wr
SW1(config)# no ip domain-lookup !(turn dns lookups off)
SW1(config)# line vty 0 4
SW1(config-line)# history size 15
SW1(config-line)# exec-timeout 0 0 !(or 'no exec-timeout')
SW1(config-line)# logging synchronous

Description, mdix speed and duplex:

SW1(config)# interface fastEthernet 0/1
SW1(config-if)# description LINK TO INTERNET ROUTER
SW1(config-if)# speed 100 !(Options: 10, 100, auto)
SW1(config)# interface range fastEthernet 0/5 - 10
SW1(config-if-range)# duplex full !(options: half, full, auto)!(usually late collisions mean half-duplex)
SW1(config-if)# mdix auto
SW1(config-if)# no mdix auto
!(it the pc/server is hard-coded 100-full-duplex and if the switch is set to auto then switch can fall-back to 100-half-duplex. So always hard-code on the switch side too)
sh int fa0/1

Password recovery:

(0x2142: skip startup config / 0x2102: normal boot process)
1. Press Ctrl+Break while router is powering up for router to go into ROMmon.
2. rommon 1>confreg 0x2142 and rommon 1>reset
3. no to the initial setup script
4. R1#copy start run
5. R1(config)#enable secret cisco
6. R1(config)#config-register 0x2102 !(default is 0x2102 i.e. reads the startup config from nvram)
7. R1#copy run start

To boot your router from the flash device:

R1(config)#boot system flash c3640-i-mz.120-7.T.bin

To boot the system from the TFTP server:

R1(config)#boot system tftp://192.168.1.1/IOS/3640/c3640-is-mz.120-7.T.bin

CCP (Cisco Configuration Professional) pre-config:

R6(config)#ip http server
R6(config)#ip http secure-server
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
R6(config)#ip http authentication local
R6(config)#username user1 privilege 15 password cisco
R6(config)#interface fastEthernet 0/0
R6(config-if)#ip address 20.0.0.1 255.0.0.0
R6(config-if)#no shutdown

Resetting switch config (Factory Default):

Reset Catalyst Switches Running CatOS:

Cat5k> (enable) clear config all

Reset Catalyst Switches Running Cisco IOS Software:

Cat2950# write erase
Erasing the nvram filesystem will remove all files! Continue? [confirm]y[OK]
Erase of nvram: complete
Cat2950# reload

Reset VLAN Information:

Cat2950# delete flash:vlan.dat
Cat2950# reload

Backup and restore:

!(Flash(IOS)/RAM(Running config)/NVRAM(Startup config)/HTTP/FTP/TFTP)

Backup IOS from the flash:

Router#copy flash tftp:
Source filename []? c1841-advipservicesk9-mz.124-15.T1.bin
Address or name of remote host []? 192.168.2.2
Destination filename [c1841-advipservicesk9-mz.124-15.T1.bin]?
Writing c1841-advipservicesk9-mz.124-15.T1.bin...!!!!!!!!!!!!!!!!!!!!!!!!![OK - 33591768 bytes]
33591768 bytes copied in 0.554 secs (6366420 bytes/sec)

Restoring the IOS from ROMmon:

!(if IOS is corrupted and the router goes to ROMmon)
rommon 1 > IP_ADDRESS=192.168.2.1
rommon 2 > IP_SUBNET_MASK=255.255.255.0
rommon 3 > DEFAULT_GATEWAY=192.168.2.2
rommon 4 > TFTP_SERVER=192.168.2.2
rommon 5 > TFTP_FILE=c1841-advipservicesk9-mz.124-15.T1.bin
rommon 6 > TFTP_CHACKSUM=0
rommon 7 > tftpdnld
!(yes to continue)
rommon 10 > reset

Backup Running or Startup config to tftp:

R1#copy run start
R1#copy startup-config tftp
Address or name of remote host []? 1.0.0.2
Destination filename [R1-confg]? R1-config
Writing startup-config...!!
[OK - 552 bytes]
552 bytes copied in 0.001 secs (552000 bytes/sec)

Restore config from tftp to Running or Startup config:

!(when you do it to running config it merges so better do it to startup config and reload)
Router#copy tftp running-config
Address or name of remote host []? 1.0.0.2
Source filename []? R1-config
Destination filename [running-config]?
Accessing tftp://1.0.0.2/R1-config....
Loading R1-config from 1.0.0.2: !
[OK - 552 bytes]
552 bytes copied in 3.003 secs (183 bytes/sec)

Static and Default Routes:

Static Route:

R(config)#ip route <destination network> <subnet-mask> <next hop address>

Headquarters(config)#ip route 2.2.2.0 255.255.255.0 192.168.12.2
sh ip route
sh ip route static
ipconfig
ipconfig /all

Floating static route:

Headquarters(config)#ip route 2.2.2.0 255.255.255.0 192.168.13.2 10
!(backup route with higher AD if it is the same AD then it load-balances equally)

Default Route:

R(config)#ip route 0.0.0.0 0.0.0.0 <next hop address>
Headquarters(config)#ip route 0.0.0.0 0.0.0.0 1.2.3.1

Port-Security:

SW(config)#interface fa0/1
SW(config-if)#switchport mode access !(works only on access ports not on dynamic port)
!(it can be configured on a trunk port, but not a good idea as the max MACs need to be set)
SW(config-if)#switchport port-security !(turn ON port security)
SW(config-if)#switchport port-security violation shutdown !(options: shutdown | protect | restrict)
!(default is shutdown)
SW(config-if)#switchport port-security maximum 1 !(allows max 1 MAC address on the port)!
(default max is 1)!(you might need max 2 MAC allowed if PC connected to Iphone and Iphone connected to switch)
SW(config-if)#switchport port-security mac-address aaaa.bbbb.cccc !(can hard code the allowed MAC)
OR
SW(config-if)#switchport port-security mac-address sticky !(or to get the MACs the switch sees instead of manually adding them, based on max MACs value set)
sh port-security
sh port-security interface fa0/1 !(important)
sh port-security address
sh interfaces fa0/1
sh int status err-disable
sh err-disable recovery
sh mac-address-table
sh mac-address-table count
sh run

To bring the port manually up when it is in err-disable state, otherwise it will stay in it forever:

SW(config)#interface fa0/1
SW(config-if)#shutdown
SW(config-if)#no shutdown

To automatically bring the port up when it is in err-disable state:

SW(config)#errdisable recovery cause psecure-violation !(only when port security violation occurs)
SW(config)#interface fa0/1
SW(config-if)#switchport port-security aging time 10 !(in mins)!(default is 5 mins)

VLANs:

VLAN Creation:

!(this creates mac-address-table and stp instance straight away)
Switch(config)# vlan 100
Switch(config-vlan)# name Engineering

!(This method is the only way to configure extended range VLANs as opposed to database mode)
!(Normal VLAN 1-1005. Extended VLAN(1006-4094) transparent mode or V3.Internal 1002-1005)
sh vlan

VLAN database mode (is being deprecated):

Switch#vlan database
Switch(vlan)#vlan 4 name sales
Switch(vlan)#apply
Switch(vlan)#exit

Access Port Configuration (Assigning a port to an access VLAN):

Switch(config-if)# switchport mode access !(can belong only to one VLAN. Will not send DTP)
!(It is good security measure to disable DTP/trunk negotiation on unused ports)
Switch(config-if)# switchport access vlan 100
Switch(config-if)# switchport voice vlan 150 !(options: vlan-id | dot1p | untagged | none)
!(You can configure the switch port, which is connected to an IP Phone, to use one VLAN for voice traffic and another VLAN for data traffic originating from a device that is connected to the access port of the IP Phone)

Trunk (tagged) Port Configuration:

!(Trunk port can be connected to a server, switch or a router)
Switch(config-if)# switchport trunk encapsulation dot1q !(do this first before making it a trunk)
OR
Switch(config-if)# switchport trunk encapsulation isl !(not all switches support this anymore)
Switch(config-if)# switchport mode trunk !(transmits DTP messages as courtesy)
Switch(config-if)# switchport nonegotiate !(will not send DTP messages even it is a trunk port)
Switch(config-if)# switchport trunk native vlan 10
!(any traffic that is not a part of any vlan on a trunk is put under native vlan e.g. cdp traffic)
!(it is a good security measure to change the native vlan to something other than VLAN 1)
sh int fa0/1 switchport

Allowed VLANs on the trunk:

Switch(config-if)# switchport trunk allowed vlan 10,20-30 !(these are the only allowed. Careful!)
Switch(config-if)#switchport trunk allowed vlan remove 1- 4094
Switch(config-if)#switchport trunk allowed vlan add 1-50 !(adds to the previous ones)
Switch(config-if)#switchport trunk allowed vlan none
Switch(config-if)#switchport trunk allowed vlan all !(default so won't see in show run)

Trunk Negotiation (DTP Negotiation):

1. dynamic auto and dynamic desirable.
Switch(config-if)#switchport mode dynamic auto
OR
Switch(config-if)#switchport mode dynamic desirable
sh vlan !(only shows interfaces in access mode and no trunk interfaces)
sh vlan bri
sh ip int bri
sh vlan bri vlan 10
sh int trunk !(shows trunk interfaces in use and allowed vlans)
sh int fa0/14 trunk !(shows allowed vlans on a trunk. Also shows native vlan)
sh int fa0/1 switchport !(see the operational and admin modes of a port. Also trunk encapsulation)
sh vlan id 2
sh vlan name sales
sh int vlan 1

sh int status !(shows up/down state and access/trunk state)
sh int switchport
sh vlan-switch !(used in gns3)
sh dtp !(displays the operating state of DTP globally and on individual ports)
sh dtp interface fa0/12

VTP:

Switch(config)# vtp mode server !(options: server | client | transparent)
Switch(config)# vtp domain CBTNuggets
Switch(config)# vtp password MyPassword !(must be the same on all the switches)
Switch(config)# vtp v2-mode !(options: 1 | 2 | 3)
OR
Switch(config)# vtp version 2 !(options: 1 | 2 | 3) !(must be the same on all the switches)
sh vtp status !(to see almost everything vtp)
sh int fa0/14 pruning !(to see pruned vlans)
sh vtp counters
sh vtp password
debug sw-vlan vtp events
debug sw-vlan vtp pruning

VTP version3:

Switch(config)#vtp domain CBT
Switch(config)# vtp mode server
Switch(config)#vtp version 3
Switch(config)#vtp primary !(this will be the only one to make changes and advertise)
Switch(config)#vtp password cisco hidden !(hashed password, more like service password)
Switch(config)#vtp password <32 chars length hash> secret

VTP Pruning (Dynamic Pruning) (VLAN 2 - 1001 prune eligible):

Switch(config)# vtp pruning !(send VTP prune message and not VTP Join message)
Switch(config-if)#switchport trunk pruning vlan remove 4,20-30 !(Removes VLANs 4 and 20-30)
Switch(config-if)#switchport trunk pruning vlan except 40-50 !(All VLANs are added to the pruning list except for 40-50)

InterVLAN Routing (Router-on-a-stick) (ROAS)(each sub-interface share the same mac address):

Switch(config)#int fa0/3
Switch(config-if)#switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20
R1(config)#interface fastEthernet 0/0
R1(config-if)#no shutdown
R1(config-if)#no ip address
R1(config)#interface fastEthernet 0/0.10
R1(config-subif)#encapsulation dot1Q 10 !(do this first before assigning an IP)
R1(config-subif)#ip address 192.168.10.254 255.255.255.0
R1(config)#interface fastEthernet 0/0.20
R1(config-subif)#encapsulation dot1Q 20
R1(config-subif)#ip address 192.168.20.254 255.255.255.0

SVI(Switch Virtual Interface)/Inter-VLAN Routing/L3 Switching/MultiLayer Switch Config:

!(SVI (Using MultiLayer Switch for routing) (each SVI interface has different a mac address))
!(Logical layer3 VLAN interface (Switch routing capabilities. Config SVI for each VLAN and put an IP address on it, used by computers as their default gateway))

Switch(config)#ip routing !(turns ON the layer 3 engine)
Switch(config)#int vlan 10
Switch(config-if)#no shut
Switch(config-if)#ip address 192.168.10.254 255.255.255.0
Switch(config)#int vlan 20
Switch(config-if)#no shut
Switch(config-if)#ip address 192.168.20.254 255.255.255.0
sh ip int bri

PPP(Point to Point Protocol) and HDLC (High-Level Data Link Control):

R1(config)#interface serial 0/0
R1(config-if)#encapsulation ppp !(options: ppp | hdlc)
!(same config on the other end)
R1(config)#interface serial 0/0
R1(config-if)#ip address 192.168.12.1 255.255.255.0
R1(config-if)#clock rate 64000 !(ISP DCE side instead of DTE)
R1(config-if)#no shut
!(same and opposite IP config on the other end)
R1(config)#username Skull password MYSECRET
!(same and opposite config on the other end)
R1(config)#interface serial 0/0
R1(config-if)#ppp authentication chap
sh controllers serial 0/0 !(shows if DCE or DTE side)
sh interfaces serial 0/0 !(shows the encapsulation in use)
debug ppp authentication !("O" that stands for outgoing and the "I" for incoming)
debug ppp negotiation
!(First you will see the "LCP" messages that are setting up the link. Once LCP is done you see "IPCP" and "CDPCP" messages. This is NCP making sure that we can send IP and CDP traffic over our PPP link)

CDP (Cisco Discovery Protocol)/LLDP (Link Local Discover Protocol):

!(CDP is enabled by default on cisco devices, but LLDP is not)
R(config)#cdp run
R(config)#cdp timer 5 !(5 secs instead of default 60 secs)
R(config)#cdp holdtime 35 !(35 secs instead of default 180 secs)
R(config)#no cdp run
R(config-if)#cdp enable
R(config-if)#no cdp run !(turn it off on ports it is not needed | security measure)
!(lldp has same commands as cdp, just use lldp instead of cdp in the same commands)
R(config)#lldp run
R(config-if)#lldp receive !(receive only)
R(config-if)#lldp transmit !(transmit only)
R(config-if)#no lldp receive
R(config-if)#no lldp transmit
!(always check both cdp and lldp)(lldp commands contains more information)
sh cdp
sh cdp neigh
sh cdp neigh detail
sh cdp neigh f0/0 detial
sh cdp entry *

sh cdp entry SW*
sh cdp int
sh cdp int fa0/0 !(shows set timers)
sh cdp traffic

STP:

!(STP(802.1d)/PVST+/RSTP(802.1w)/RPVST/MST)
!([BPDU:BridgePriority(32768)+MAC])
!(STP: RootPorts->DesignatedPorts->BlockingPorts)
!(All RootBridgePorts are DP and One DP per link)
!(ElectRoot->RootPorts(LowestCost/LowestBrdigeID/LowestPortNumber)->BlockTheRest)
!(Costs: 100Mbps=19/1Gbps=4/10Gbps=2)
!(STP Timers:Block20sec/Listen15sec/Learn15sec/Forward)
!(RSTP: RootPorts->DesignatedPorts->AlternatePorts)

Change BPDU default timers:

SwitchA(config)#spanning-tree vlan 10 hello-time 1 !(1sec/default 2sec)
SwitchB(config)#spanning-tree vlan 20 max-age 6 !(6sec/default 20sec)
SwitchC(config)#spanning-tree vlan 30 forward-time 4 !(4sec/default 15sec)

Change Root Bridge:

SwitchA(config)#spanning-tree vlan 1 root primary !(hard code a root bridge)
!(This is a macro that looks at the current priority of the root bridge and changes your running-config to lower your own priority. Based on VLAN number)
OR
SwitchA(config)#spanning-tree vlan 1 priority 4096 !(hard code priority. multiple of 4096)

Change Root Port/Non-Designated Port:

SwitchB(config)#interface fa0/14
SwitchB(config-if)#spanning-tree cost 500
!(can be used to change the cost of root port to get a different root port chosen)
OR
SwitchA(config)#interface fa0/14
SwitchA(config-if)#spanning-tree port-priority 16
!(can be used to change the port priority to get a different root port chosen)

PortFast:

!(To avoid spanning tree calculations and disable STP for connections to PCs)
SwitchB(config)interface fa0/2
SwitchB(config-if)#spanning-tree portfast !(configured on access ports)
SwitchB(config)#spanning-tree portfast default !(can be enabled globally for all access mode ports)

Enable Rapid-PVST:

SwitchA(config)#spanning-tree mode rapid-pvst !(to enable rapid spanning-tree protocol)
sh spa !(important)
sh spa | begin int
sh spa summary !(stp mode, portfast, bpdu guard/filter, loopguard, uplink/backbone fast)
sh spa detail !(to see number of topology changes and the int it is happening on)
sh run | inc priority
sh spa vlan 10
sh mac address-table dynamic !(to verify traffic path. Port on which mac was learnt)
sh mac address-table aging-time
sh spa inconsistentports !(to see when root guard is working)
debug spa

debug spa events !(to see topology change notifications)
debug spa backbonefast detail
debug spa bpdu

ACLs:

sh access-lists
sh ip access-lists
sh access-lists 1
sh ip access-lists 1
sh access-lists ACL_NAME1
sh ip access-lists ACL_NAME1
sh ip access-lists interface fa0/0
sh ip int fa0/0
sh time-range
sh time-range ACL_NAME1
sh run | inc access-list

Standard Access List:

!(Standard Access List close to the destination are best)
R1(config)#access-list 1 permit 10.0.0.0 0.255.255.255
R1(config)#access-list 1 deny host 10.0.0.1 log
R1(config)#access-list 1 permit any any !(don't forget this as there is a default deny at the end)

Extended Access list:

!(Extended Access lists closer to the source are best)
R1(config)#access-list 101 permit tcp 10.0.0.0 0.255.255.255 187.100.1.6 0.0.0.0 eq 20
!(187.100.1.6 0.0.0.0 is the same as host 187.100.1.6)
R1(config)#access-list 101 deny tcp any eq 22 host 10.0.0.1 range 22 23
R1(config)#access-list 101 permit ip any any dscp cs2

Apply this ACL to an interface:

R1(config)#interface Fa0/1
R1(config-if)#ip access-group 1 out
OR
R1(config)#interface Fa0/0
R1(config-if)#ip access-group 1 in

Named ACLs:

R(config)#ip access-list extended MyACL
R(config-ext-nacl)#100 permit ip host 1.1.1.1 any
Edit and Insert Lines in ACLs:
R(config)#ip access-list extended MyACL
R(config-ext-nacl)#no 500
R(config-ext-nacl)#500 permit ip any host 5.5.5.5
R(config-ext-nacl)#510 permit ip any host 6.6.6.6

Time-based ACLs:

R(config)#time-range TR_WORKDAYS
R(config-time-range)#periodic weekdays 08:00 to 19:00
!(Don't configure NTP unless mentioned in the LAB)
R(config)#ip access-list extended 100
R(config-ext-nacl)#27 permit tcp any any eq www time-range TR_WORKDAYS

Block pings with acls:

access-list 100 deny icmp any any echo

access-list 100 deny icmp any any echo-reply
access-list 100 permit ip any any
OR
access-list 101 deny icmp host 192.168.1.51 host 192.168.1.34 echo
access-list 100 permit ip any any

ACL log keyword:

R1(config)# ip access-list extended Block_SSH
R1(config-ext-nacl)# no 10
R1(config-ext-nacl)# 10 deny tcp any any eq 22 log

Vty lines ACL (block/allow telnet/ssh):

R1(config)#ip access-list extended VTY_ACCESS
R1(config-ext-nacl)#20 permit tcp 10.1.1.0 0.0.0.255 any eq 22
R1(config-ext-nacl)#30 permit tcp 10.1.1.0 0.0.0.255 any eq 23
R1(config-ext-nacl)#500 deny ip any any log
R1(config-ext-nacl)#line vty 0 4
R1(config-line)#access-class VTY_ACCESS in

NAT:

1. ACL to identify the addresses to be translated
2. Set inside and outside interfaces
3. Define nat operations

sh nat translations
netstat
sh run | inc ip nat
sh ip nat trans verbose
sh ip nat stat
clear ip nat translation *
debug ip nat

Define the traffic to match:

R(config)#access-list 10 permit 10.0.0.0 0.0.255.255 !(ACL to match)

Static NAT: (Note: two way NAT. Just like a port-forward so be careful)

R(config)#ip nat inside source static 10.0.0.19 192.0.2.1
OR
R(config)#ip nat outside source static 192.0.2.1 10.0.0.19
R(config)#ip nat inside source static tcp 10.0.0.3 80 192.0.2.1 80
R(config)#ip nat inside source static tcp 10.0.0.3 443 192.0.2.1 443

Dynamic NAT:

R(config)#ip nat pool MyPool 192.0.2.1 192.0.2.254 prefix-length 24 !(Public IP pools)
R(config)#ip nat inside source list 10 pool MyPool !(actual NAT rule)
R(config)#ip nat inside source static 10.0.0.42 192.0.2.42 !(can be combined with static)

PAT:

R(config)#ip nat inside source static tcp 10.0.0.3 80 192.0.2.1 80
R(config)#ip nat inside source static tcp 10.0.0.3 443 192.0.2.1 443
R(config)#ip nat inside source static tcp 10.0.0.10 3389 192.0.2.1 3389
R(config)#ip nat inside source static tcp 10.0.0.11 3389 192.0.2.1 3390
R(config)#ip nat inside source list 10 pool MyPool overload
OR
R(config)#ip nat inside source list 10 interface FastEthernet 0/0 overload

Identify interfaces: (Note: should be the last step, especially in a production network)

R(config)#int fa0/0
R(config-if)#ip address dhcp !(ISP assigned IP for external interface)
R(config-if)#ip nat outside
R(config)#int fa0/1
R(config-if)#ip address 192.168.1.1
R(config-if)#ip nat nat inside

Syslog (splunk/kiwi)(port udp 514):

Router(config)#ntp server pool.ntp.org
Router(config)#no service timestamps !(you can disable timestamps and use sequence numbers)
Router(config)#service sequence-numbers
Router(config)#logging console errors !(severity level 3 and lower)
!(This logging information is saved in the RAM of your device. Once you reboot it you will lose this logging history)
Router(config)#terminal monitor !(if you want to see syslog messages on vty sessiosn when your on vty)
Router(config)#logging buffered 4096 !(buffer size in bytes)
Router(config)#logging 192.168.1.100 !(all logging sent to the syslog server except level 7)
Router(config)#logging trap 7 !(this will sent also the debug info to the syslog server)
R1(config)# logging host 192.168.1.25 !(sending it to a syslog server e.g. kiwi syslog server)
R1(config)# logging source-interface Loopback0
R1(config)# logging trap notifications
R1(config)# no logging console
sh logging !(to see logging information)
sh logging history !(to see the set severity level)
sh ntp status !(to see if clock is synchronized)

NTP:

Clock commands:

R1# clock set 14:12:00 10 feb 2005
R1(config)# clock timezone ARIZONA -7
sh clock

NTP config:

R1(config)# ntp server 1.gr.pool.ntp.org
sh ntp associations

Use the router as an NTP server:

R1(config)# ntp master

SNMP (Simple Network Management Protocol):

!(PRTG is well-known)
Router(config)#snmp-server community MY_STRING ro 10 !(options: ro | rw)
Router(config)#access-list 10 permit host 192.168.1.2
Router(config)#snmp-server location Amsterdam
Router(config)#snmp-server contact info@gns3vault.com

Netflow:

Router(config)#interface fastEthernet 0/0 !(the direction to monitor)
Router(config-if)#ip flow ingress
Router(config-if)#ip flow egress
Router(config)#ip flow-export version 9 !(netflow version)
Router(config)#ip flow-export destination 192.168.1.2 100 !(netflow collector)

sh ip cache flow !(shows you netflow is working)

DHCP (Dynamic Host configuration protocol):

R1(config)# service dhcp
R1(config)# ip dhcp pool NET-POOL
R1(dhcp-config)# network 192.168.1.0 255.255.255.0 !(you can use prefix notation with it)
R1(dhcp-config)# default-router 192.168.1.1
R1(dhcp-config)# dns-server 8.8.8.8 8.8.4.4
R1(dhcp-config)# domain-name company1.com
R1(dhcp-config)# lease 9 !(9 days/default is 1 day)
OR
R1(dhcp-config)# lease 0 4 30 !(set the lease time for 4 hours and 30 minutes)
R1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.5
R1(config)# ip dhcp excluded-address 192.168.1.10
sh ip dhcp binding
ipconfig /release
ipconfig /renew

Cisco DHCP Option 150 Configuration:

R1(config)# ip dhcp pool NET-POOL
R1(dhcp-config)# option 150 ip 10.10.22.99 10.10.22.100 !(Configure 2 IPs)

DHCP Relay:

R1(config)#int fa0/0.52
R1(config-subif)#ip helper-address 10.1.51.200

RIP (AD is 120 and metric is hop-count):

!(Distance Vector(DV): 1. Max Distance(16hops)/2.RoutePoison/3.TriggeredUpdates/
4.SplitHorizon/5.HoldDownTimers)
R1(config)#router rip
R1(config-router)#network 192.168.12.0 !(network command tells what network to advertise and what interfaces to send advertisements out)
R1(config-router)#network 172.16.1.0
R1(config-router)#version 2
R1(config-router)#no auto-summary
R2(config)#router rip
R2(config-router)#network 192.168.12.0
R2(config-router)#network 192.168.23.0
R2(config-router)#version 2
R2(config-router)#no auto-summary
R3(config)#router rip
R3(config-router)#network 172.16.0.0
R3(config-router)#network 192.168.23.0
R3(config-router)#version 2
R3(config-router)#no auto-summary
sh ip route
sh ip route rip !(R NETWORK [AD/METRIC] via NEXT_HOP, TIME_LEARNT, EXIT_INTERFACE)
sh ip protocols
sh ip rip database
debug ip rip
sh ipv6 route
sh ipv6 route rip

sh ipv6 protocols
sh ipv6 rip database
debug ipv6 rip

FHRP (First Hop Redundancy Protocols):

!(Virtual IP and Virtual MAC)

HSRP(Hot Standby Routing Protocol):

!(Hello 3sec/Hold 10sec)
!(Active-Standby)
SwitchA(config)#interface fa0/17 !(internal interface)
SwitchA(config-if)#no switchport
SwitchA(config-if)#ip address 192.168.1.1 255.255.255.0
SwitchA(config)#interface fa0/19 !(external interface)
SwitchA(config-if)#no switchport
SwitchA(config-if)#ip address 192.168.14.1 255.255.255.0
SwitchA(config)#ip routing
SwitchA(config)#ip route 0.0.0.0 0.0.0.0 192.168.14.4
SwitchA(config)#interface fa0/17
SwitchA(config-if)#standby 1 ip 192.168.1.3 !(create a standby group and assign an IP)
!(Same and opposite config on SwitchB)
!(PC is configured with a default gateway of 192.168.1.3 i.e. Virtual Gateway's IP)
!(HSRPv1 uses the 0000.0c07.acXX MAC address where XX is the HSRP group number e.g. 0000.0c07.ac01 for group 1)(HSRPv2 uses 0000.0C9F.FXXX MAC)
!(By default the switch with the highest priority will become the active HSRP device. If the priority is the same then the highest IP address)
SwitchA(config)#interface fa0/17
SwitchA(config-if)#standby 1 priority 150 !(default priority is 100)
SwitchA(config-if)#standby 1 preempt !(could become active if router goes down & then up)
SwitchA(config-if)#standby 1 preempt delay minimum 60
OR
SwitchA(config-if)#standby 1 preempt delay reload 60 !(If a router reboots it might need some time to “converge”)
SwitchA(config-if)#standby 1 mac-address 0000.1111.5555 !(instead of using default hsrp MAC)
SwitchA(config-if)#standby 1 version 2 !(version must be same on both devices)
SwitchA(config-if)#standby 1 name my-hsrp-name !(name of the group for ease)
SwitchA(config-if)#standby 1 authentication md5 key-string md5pass !(prevent rogue hsrp router)
!(can also use a key chain the same way you would use in eigrp or anywhere else)
SwitchA(config-if)#standby 1 timers msec 100 msec 300 !(default hello 3 secs & hold 10 secs)

Interface tracking (object tracking):

1. Select an interface to track and if it fails decrease the priority so that another device can become the active router. Tracking checks the line protocol status, but can also be configured to check if the route exists or with SLA probes.

SwitchA(config)#interface fa0/17 !(internal interface)
SwitchA(config-if)#standby 1 preempt !(preemption is imp otherwise tracking is useless)
SwitchB(config)#interface fa0/19 !(internal interface)
SwitchB(config-if)#standby 1 preempt !(preemption is imp otherwise tracking is useless)
SwitchA(config-if)#standby 1 track fastEthernet 0/19 !(external interface)
SwitchA(config)#interface fa0/19
SwitchA(config-if)#shutdown

!(Verify)
SwitchA#show standby | include Priority
Priority 140 (configured 150) !(by default decrements the priority by 10)
SwitchA(config)#interface fa0/17
SwitchA(config-if)#standby 1 track fastEthernet 0/19 60
OR
SwitchA(config-if)#standby 1 track 1 decrement 60
!(If the links goes down again the priority will become 90 reducing it by 60 so the other device with default priority 100 will then become active)
!(Interface tracking will only check the state of any interface not if anything fails upstream)

IP SLA with object tracking:

!(IP SLA can be used for many things. One of them is to generate a ping to a destination every X seconds and we can combine this with object tracking)
SwitchA(config)#interface fa0/17
SwitchA(config-if)#no standby 1 track fastEthernet 0/19 60
SwitchA(config)#ip sla 1
SwitchA(config-ip-sla)#icmp-echo 192.168.14.4
SwitchA(config)#ip sla schedule 1 start-time now life forever
SwitchA(config)#track 1 rtr 1 reachability
SwitchA(config)#interface fa0/19
SwitchA(config-if)#standby 1 track 1 decrement 60
!(Test)
RouterA(config)#interface fa0/13
RouterA(config-if)#shutdown

VRRP (Virtual Router Redundancy Protocol):

!(Hello 1sec/Hold 3sec)
!(Master-Backup)
1. Exactly same config as HSRP except we use vrrp instead of standby
2. VRRP uses the 0000.5e00.01XX MAC address where XX is the VRRP group number.
3. VRRP can have the same IP as that of the real interface as opposed to HSRP. The device that has got the real IP as the virtual IP becomes the master, priority is not looked at then because the priority of the master is set to the highest value of 255.

GLBP (Gateway Load Balancing Protocol):

!(Hello 3sec/Hold 10sec)
!(Active-Active Load Balancing)
!(AVG(Active Virtual Gateway)(Manager: Generates MACs for AVFs) and AVF(Active Virtual Forwarder))
R1(config)#interface fa1/0
R1(config-if)#glbp 1 ip 192.168.1.3
R1(config-if)#glbp 1 preempt
R1(config-if)#glbp 1 authentication md5 key-string mypass
!(Same and opposite config on SwitchB)
R1(config-if)#glbp 1 priority 150
!(The virtual MAC address that GLBP uses is 0007.b400.XXYY (where X = GLBP group number and Y = AVF number))

GLBP interface tracking:

!(Interface tracking works differently for GLBP compared to HSRP or VRRP. It has a weighting mechanism whic is used to determine if a device can be AVF or not)

SwitchB#show glbp | include Weighting
Weighting 100 (default 100)
SwitchB(config)#track 16 interface fastEthernet 0/16 line-protocol
SwitchB(config)#track 17 interface fastEthernet 0/17 line-protocol
SwitchB(config)#interface fa0/19
SwitchB(config-if)#glbp 1 weighting track 16 decrement 20
SwitchB(config-if)#glbp 1 weighting track 17 decrement 20
SwitchB(config-if)#glbp 1 weighting 100 lower 70 upper 90
!(Verify)
SwitchB#show glbp | include Weighting
Weighting 100 (configured 100), thresholds: lower 70, upper 90
!(Test and verify)
!(Now shutdown int fa0/16)
!(Weighting 80 (configured 100), thresholds: lower 70, upper 90)
!(Now shutdown int fa0/17)
!(Weighting 60, low (configured 100), thresholds: lower 70, upper 90)
!(Now 'no shutdown' int fa0/16
!(Weighting 80, low (configured 100), thresholds: lower 70, upper 90)
!(Now 'no shutdown' int fa0/17)
!(Weighting 100, low (configured 100), thresholds: lower 70, upper 90)

GLBP weighted Load-Balancing:

!(default I round-robin. If you have a beefier router you can use weighted load-balancing)
!(configured on AVG)
SwitchB(config)#interface fa0/19
SwitchB(config-if)#glbp 1 load-balancing weighted
SwitchB(config-if)#glbp 1 weighting 200 !(this switch will have double the traffic going through)
sh standby
sh standby bri
sh standby fa0/17
sh standby | inc priority
sh standby | inc time
sh standby vlan 1
sh standby all
debug standby events
debug standby errors
debug standby packets
debug standby events track
!(Note: Exactly same commands for VRRP and GLBP, but use vrrp and glbp instead of stanby)
sh ip arp !(verification of mac on the switch)
c:\>ipconfig !(to verify the gateway configured on the PC)
c:\>arp -a !(to verify the mac learned on the PC)
c:\>tracert 8.8.8.8 !(Note: tracert shows the original active router IP instead of the VIP)
sh glbp bri !(To see the active AVG & AVF. Also to verify the mac addresses of the AVF i.e. which one the PC if going through, as it will keep doing round-robin. Clear arp cache of the PC and verify again)

EIGRP:

!(BackupRoutes(FastConvergence/DUAL)/FelxibleSummrization/UnequalCostLB)
!(NeighbourTable(OnlyNeighbours)/TopologyTable(LearnedFromneighbours)/RoutingTable(Best))
(FeasibleDistance(HowFarFromYou)/AdvertisedDistance(HowFarFromNeighbour)/Successor(Rout

ingTable)/FeasibleSuccessor(TopologyTable)/ActiveRoute(ActivelySearching)/PassiveRoute)
!(To be a Feasible Successor, the AD must be less than the FD of the Successor)
!(Hello/Update/Query/Reply/Ack)
!(Bandwidth (k1) and Delay (k3) by default/MTU/Load/Reliability)
Metric = bandwidth (slowest link) + delay (sum of delays)
1. Bandwidth: [10^7 / minimum bandwidth in the path] * 256.
2. Delay: sums of delays in the path multiplied by 256 (in tens of microseconds).
So the formula looks like:
Metric = (10^7 / minimum bandwidth) * 256 + (sum of delays) * 256

EIGRP Config:

```
R1(config)#router eigrp 1 !(1 is AS number)
R1(config-router)#no auto-summary
R1(config-router)#network 1.1.1.0 0.0.0.255
R1(config-router)#network 192.168.12.0
R1(config-router)#exit
R2(config)#router eigrp 1
R2(config-router)#no auto-summary
R2(config-router)#network 2.2.2.0 0.0.0.255
R2(config-router)#network 192.168.12.0
```

Named EIGRP:

```
!(Since IOS 15, EIGRP has a new method of configuration called named EIGRP)
R1(config)#router eigrp MY_NAME
R1(config-router)#address-family ipv4 autonomous-system 12
R1(config-router-af)#network 192.168.12.0
R1(config-router-af)#af-interface FastEthernet 0/0
```

EIGRP Authentication:

```
R1(config)#key chain MYCHAIN
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string BANANA
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip authentication mode eigrp 1 md5
R1(config-if)#ip authentication key-chain eigrp 1 MYCHAIN
sh ip route !(to see the routing table)
sh ip router eigrp !(you will also see feasible successor if load balanced)
!(D NETWORK [AD/METRIC] via NEXT_HOP, TIME_LEARNT, EXIT_INTERFACE)
sh ip eigrp interfaces
sh ip eigrp neighbors
sh ip eigrp topology
!(P NETWORK, 1 successors, FD is 158720 via NEXT_HOP (FD/AD), EXIT_INTERFACE)
sh ip eigrp traffic
clear ip eigrp neighbor !(reset the EIGRP neighbor adjacency on a router)
debug ip eigrp packet
debug ip eigrp neighbors
debug eigrp packets hello
debug eigrp packets update
debug eigrp packets ack
debug eigrp packets query
debug eigrp packets
```

sh ip protocols !(It will show you for which networks you are routing, passive interfaces and the administrative distance)!(Also shows K values)

EtherChannel:

!(PagP Modes: On/Desirable/Auto/Off)

!(LACP Modes: On/Active/Passive/Off)

EtherChannel L2 Configuration:

SwitchA(config)#default int range fa0/13 - 14
SwitchA(config)#int range fa0/13 – 14
SwitchA(config-if)#shut
SwitchA(config-if)#channel-group 1 mode desirable !(options(LACP/PAgp): active/desirable,passive/auto, on)
SwitchA(config-if)#no shut
SwitchA(config)#interface port-channel 1
SwitchA(config-if)#switchport trunk encapsulation dot1q
SwitchA(config-if)#switchport mode trunk
!(same config when using LACP except for modes are different)

EtherChannel L3 Configuration:

SW1(config)# interface port-channel 1
SW1(config-if)# no switchport
SW1(config-if)# ip address 172.16.1.11 255.255.255.0
SW1(config)# deafult int range fastethernet0/1 - 2
SW1(config)# int range fastethernet0/1 – 2
SW1(config-if-range)# no switchport
SW1(config-if-range)# no ip address
SW1(config-if-range)# channel-group 1 mode desirable
!(same an opposite config on SW2)

OSPF:

Configuration:

R1(config)#router ospf 1
R1(config-router)#network 192.168.23.0 0.0.0.255 area 0

Changing Router ID:

R1(config)#router ospf
R1(config-router)#router-id 3.3.3.3
R1#clear ip ospf process

Manually change the cost:

R1(config)#interface fastEthernet 1/0
R1(config-if)#ip ospf cost 50 !(manually changing cost)

Advertising Default Route:

R1(config)#router ospf 1
R1(config-router)#default-information originate always !(advertising default route)

Plain-text Authentication:

R1(config)#interface fastEthernet 1/0
R1(config-if)#ip ospf authentication
R1(config-if)#ip ospf authentication-key secret

MD5 authentication:

R1(config)#interface fastEthernet 1/0
R1(config-if)#ip ospf authentication message-digest
R1(config-if)#ip ospf message-digest-key 1 md5 mykey

Change the OSPF timers:

```
R1(config-if)#interface fastEthernet 1/0
R1(config-if)#ip ospf hello-interval 5
R1(config-if)#ip ospf dead-interval 15
```

Multi-area OSPF:

```
R1(config)#router ospf 1
R1(config-router)#network 1.1.1.0 0.0.0.255 area 0
R1(config-router)#network 192.168.12.0 0.0.0.255 area 0
R2(config)#router ospf 1
R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
R2(config-router)#network 192.168.23.0 0.0.0.255 area 1
sh ip ospf neighbor !(to see the full state of neighbor and DR/BDR/DROTHER)
sh ip protocols !(to see the router id)
sh ip route ospf
!(O NETWORK_LEARNT [AD/METRIC] via NEXT_HOP, TIME, EXIT_INTERFACE)
sh ip ospf interface fa1/0 !(to check the cost of an interface and timers)
sh ip ospf database !(shows the LSDB/to see the LSAs)
debug ip ospf packet
debug ip ospf adj !(to see 7 states of adjacency)
clear ip ospf process !(to restart the ospf process)
```

IPv6:

```
R1(config)#ip routing  !(this is used for ipv4 which is ON by default)
R1(config)#ipv6 unicast-routing
R1(config)#int fa1/0
R1(config-if)#ipv6 address 2001:210:10:1::1/64
R1(config-if)#ipv6 address autoconfig   !(allows PCs to auto generate IPv6 addresses)
R1(config-if)#no shut
R1(config)#ipv6 route 2001:56::0/64 2001:210:10:1::2
sh ipv6 int bri
sh ipv6 int
sh ipv6 route
sh ipv6 neighbors
R2#ping 2001:210:10:1::1
R2#ping ipv6   !(extended ipv6 ping)
```

IPv6 ACLs:

```
!(same concept as ipv4 ACL)
R1(config)#ipv6 access-list CBTACL
R1(config-ipv6-acl)#permit tcp any any eq 23
R1(config)#int fa0/0
R1(config-if)#ipv6 traffic-filter CBTACL in
sh ipv6 access-list
```

IPv6 EIGRP:

```
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router eigrp 100
R1(config-rtr)#no shut
R1(config)#int fa0/0
R1(config-if)#ipv6 eigrp 100
sh ipv6 protocols
```

sh ipv6 eigrp neighbors

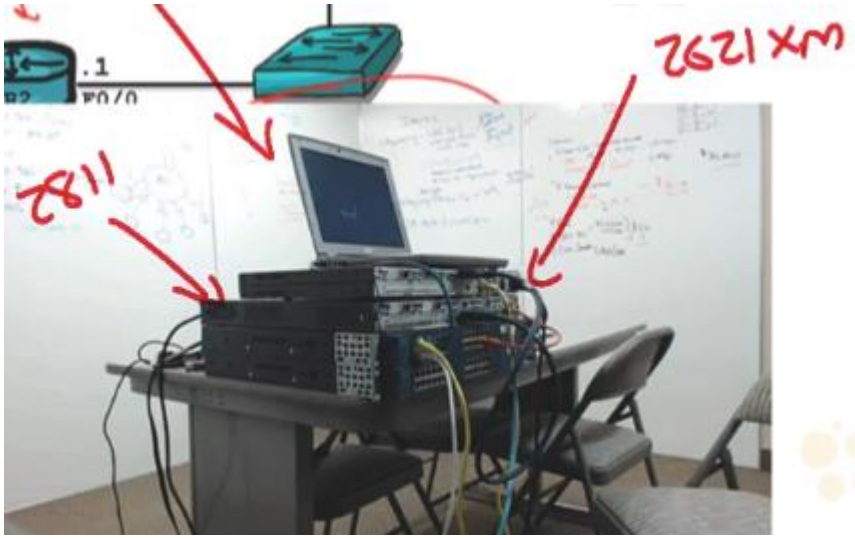
IPv6 OSPF:

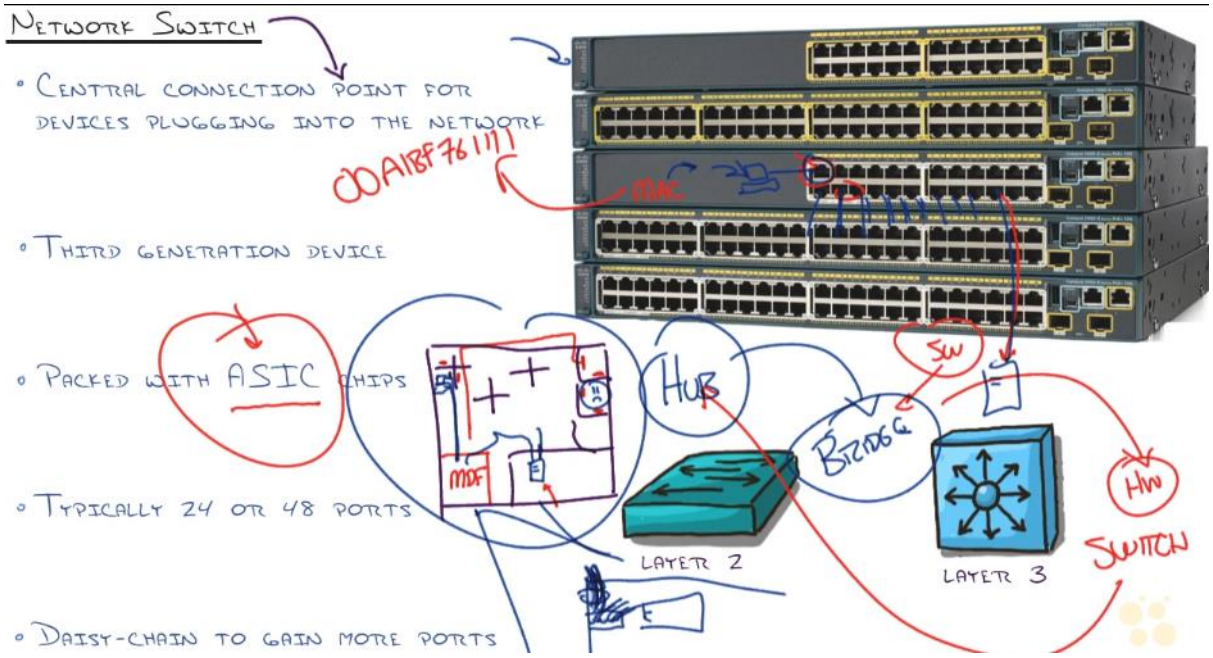
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config)#int fa0/0
R1(config-if)#ipv6 ospf 1 area 0
sh ipv6 ospf neighbors
sh ipv6 ospf int

CCNA RnS Notes

ICND (Internetworking Cisco Networking Devices)

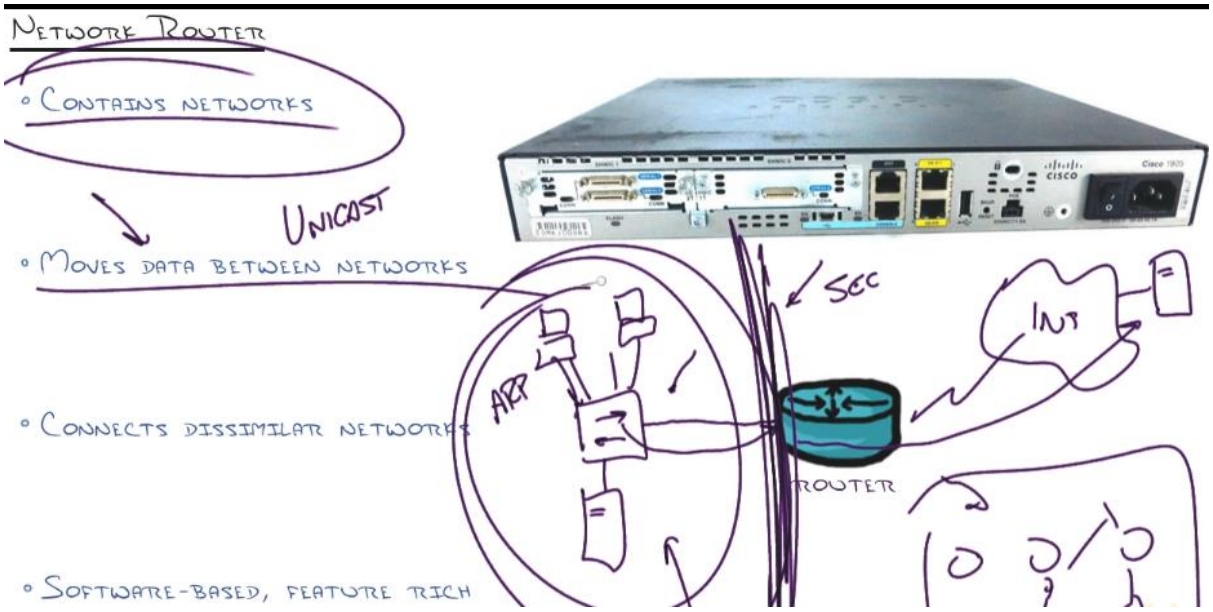
Cisco serial crossover cables
(You will need USB to serial converter)
GNS3 (Emulator), VIRL (Emulator) and Packet tracer (Simulator) can be used for Labs.
MAC address is 12 digit Hexadecimal and 48bits.
ASIC: Application Specific Integrated Circuitry
SFP: Small Form Pluggable
CAM: Content Addressable Memory (CAM table or MAC address table)
StackWise Switches: Cisco StackWise technology unites up to nine individual Cisco Catalyst 3750 switches into a single logical unit, using special stack interconnect cables and stacking software. The stack behaves as a single switching unit that is managed by a master switch elected from one of the member switches.



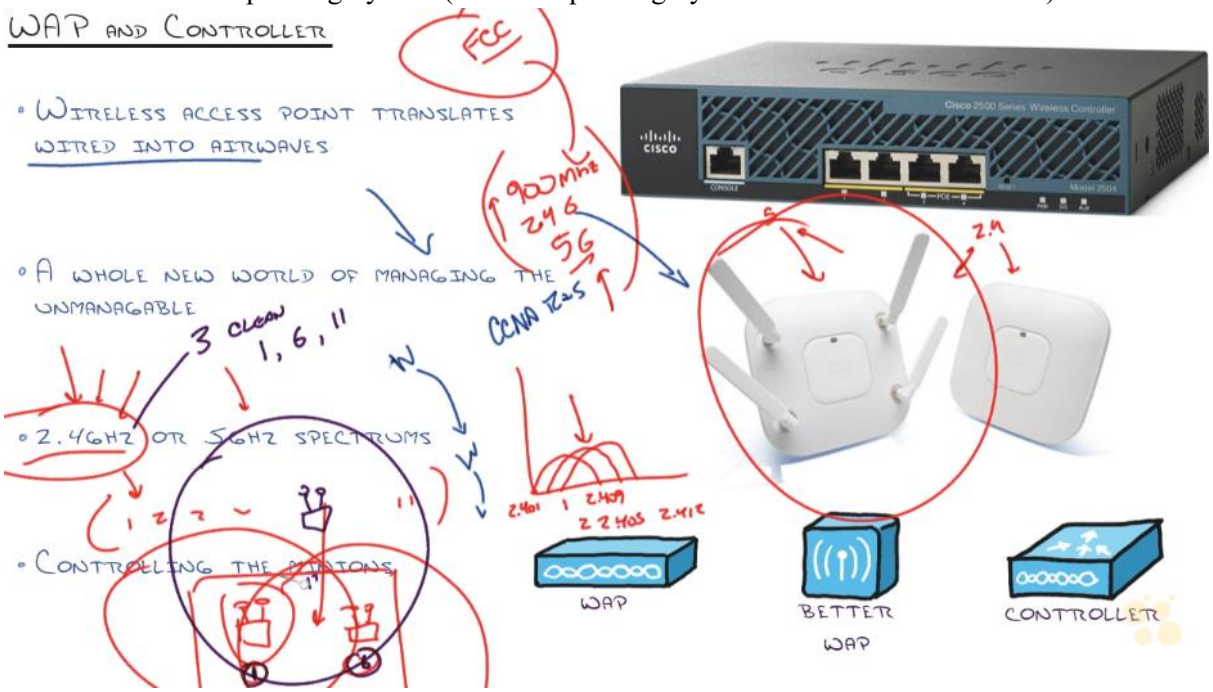


Cisco StackWise technology unites up to nine individual **Cisco** Catalyst **3750**switches into a single logical unit, using special stack interconnect cables and stacking software. The stack behaves as a single switching unit that is managed by a master switch elected from one of the member switches.

ASICs belong to switches

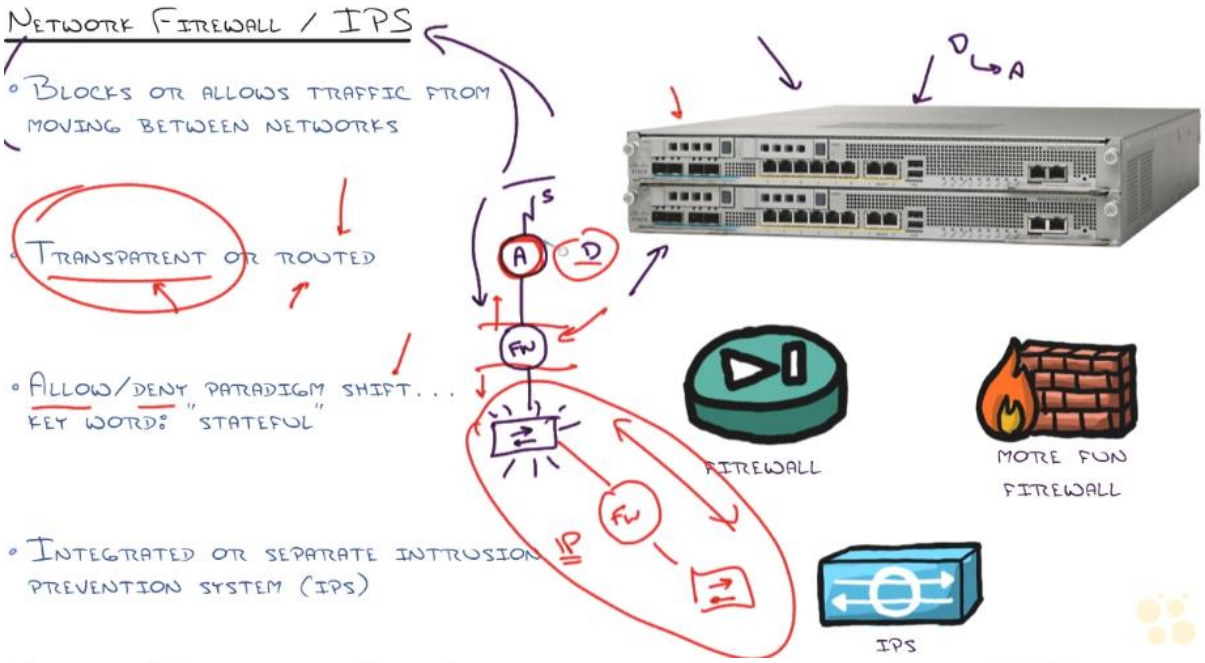


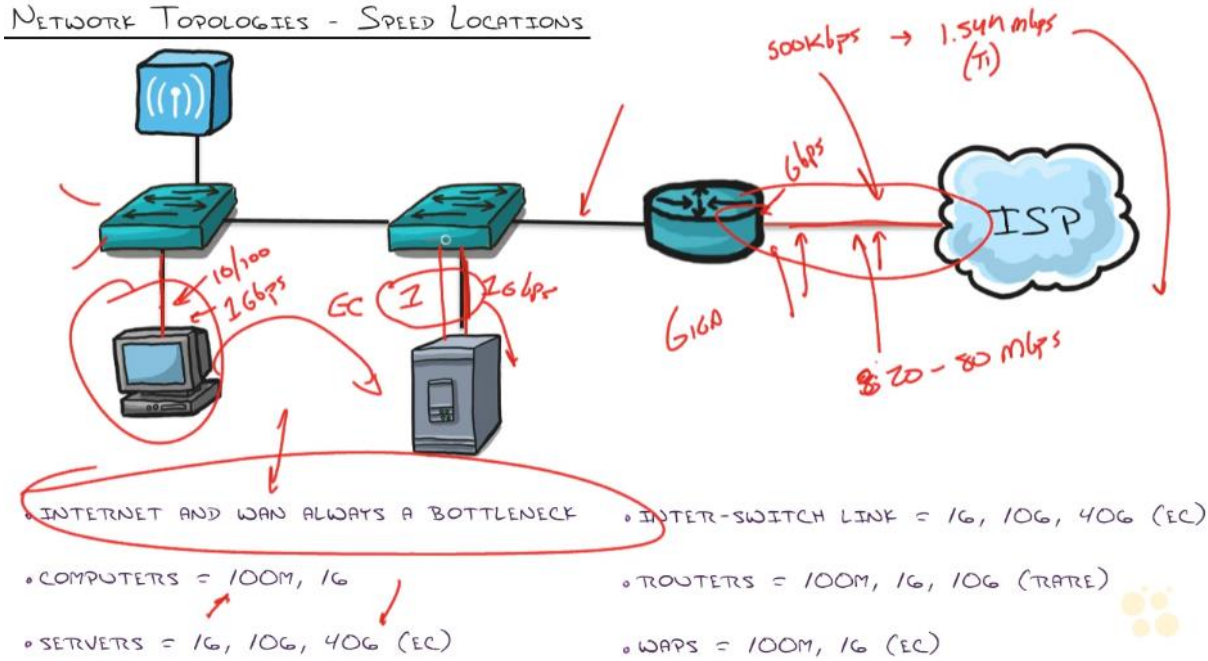
IOS: Internetwork Operating System (Cisco's Operating System on Router and Switches)



STATELESS Firewall: Stateless firewalls watch network traffic, and restrict or block packets based on source and destination addresses or other static values. They are not 'aware' of traffic patterns or data flows. A stateless firewall uses simple rule-sets that do not account for the possibility that a packet might be received by the firewall 'pretending' to be something you asked for.
STATEFUL Firewall: Stateful firewalls can watch traffic streams from end to end. It keeps track of connections and allows only the return traffic. They are are aware of communication paths and can implement various IP Security (IPsec) functions such as tunnels and encryption. In technical terms, this means that stateful firewalls can tell what stage a TCP connection is in (open, open sent, synchronized, synchronization acknowledge or established), it can tell if the MTU has changed, whether packets have fragmented etc.
IPS: Intrusion prevention System (Signature based or Anomaly based)

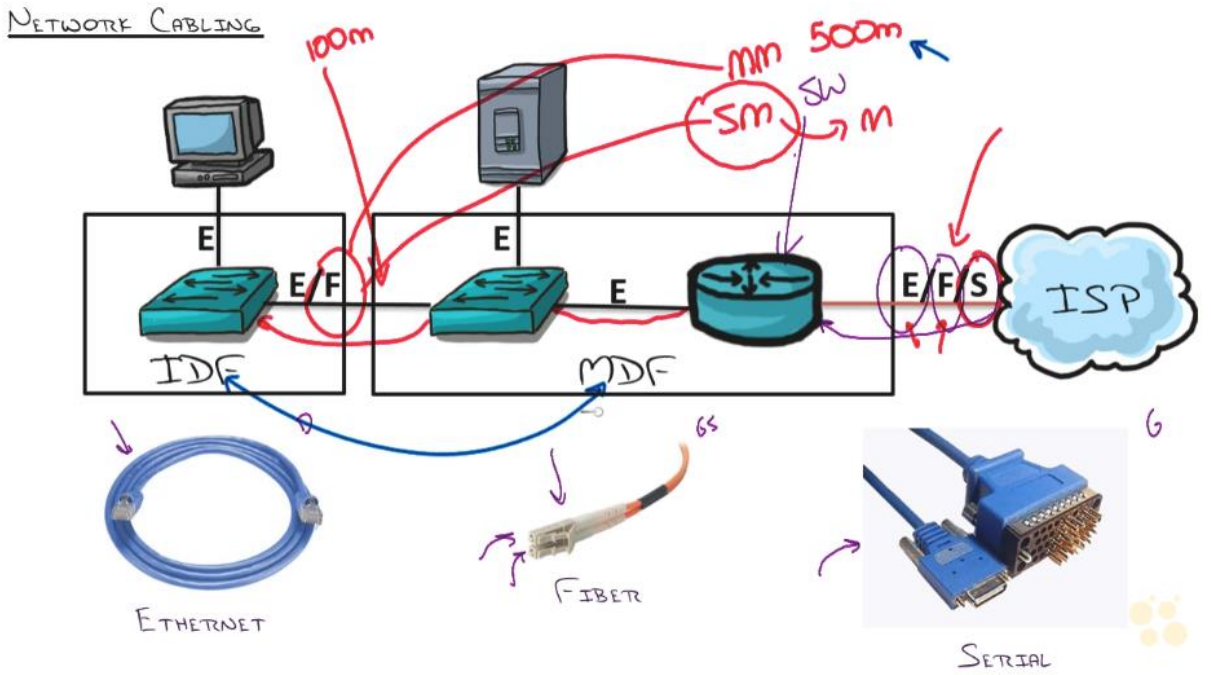
Higher security zone (100) (inside) can reach lower security zone (0 or 50) (outside and DMZ) but not the other way around.

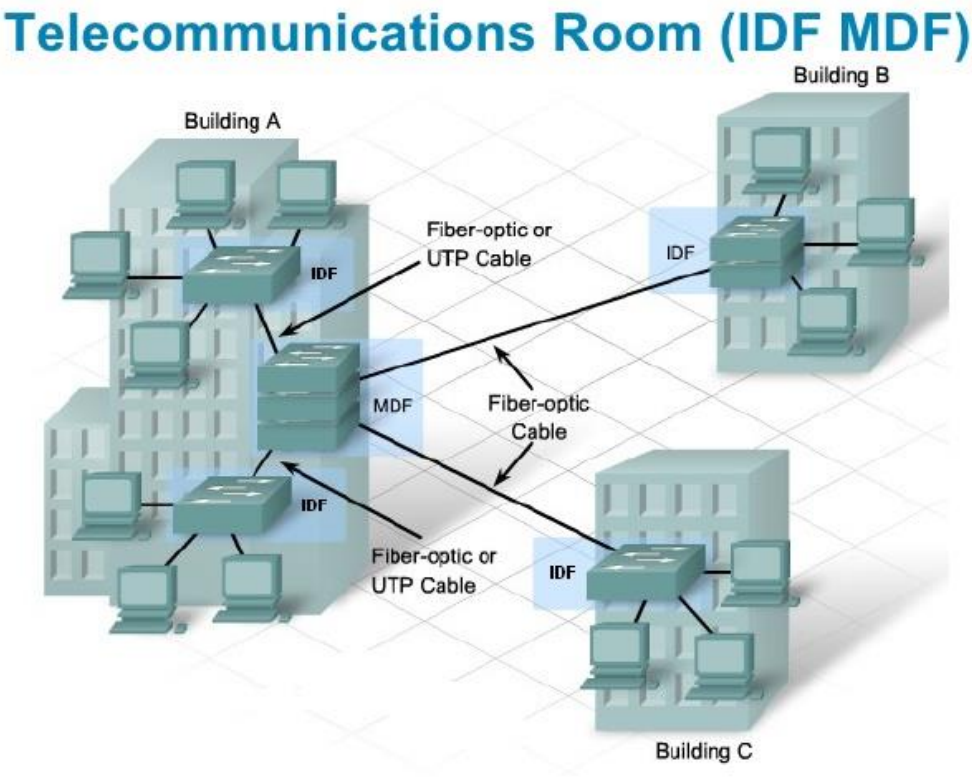




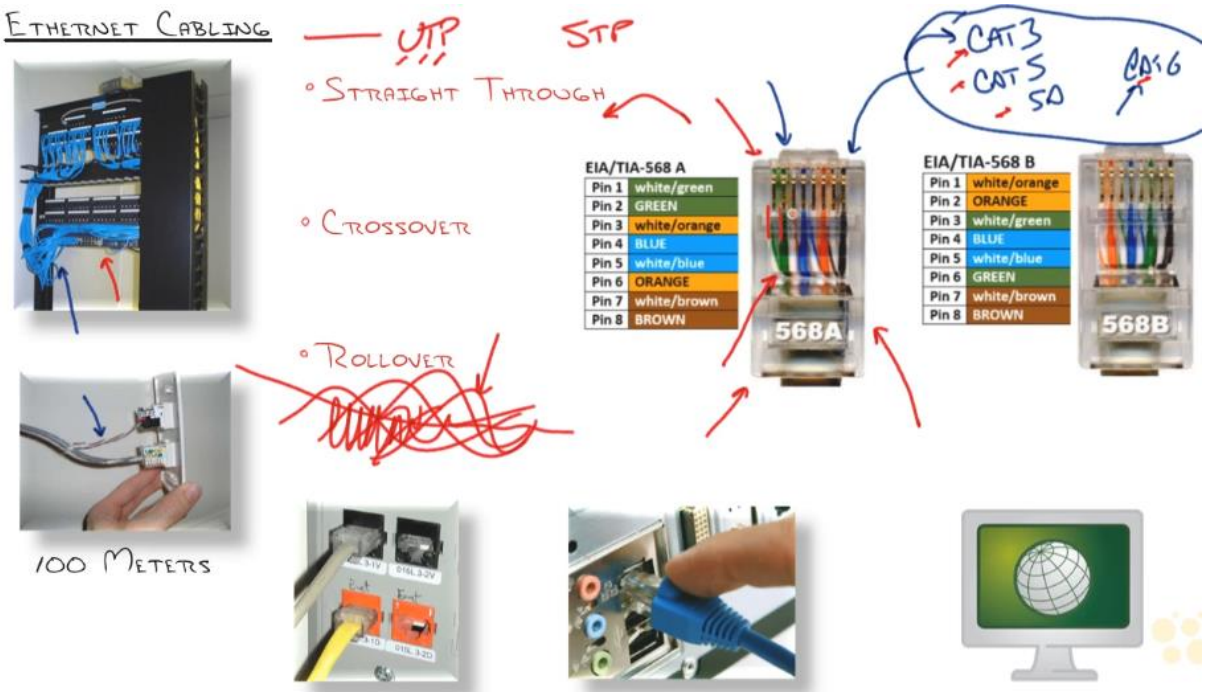
EtherChannel: Bundle upto 8 links for combined bandwidth (i.e. LoadBalanced).
CSMA/CD: Carrier Sense Multiple Access/Collision Detection (used on hub wired network)
CSMA/CA: Carrier Sense Multiple Access/Collision Avoidance (used on wireless network)

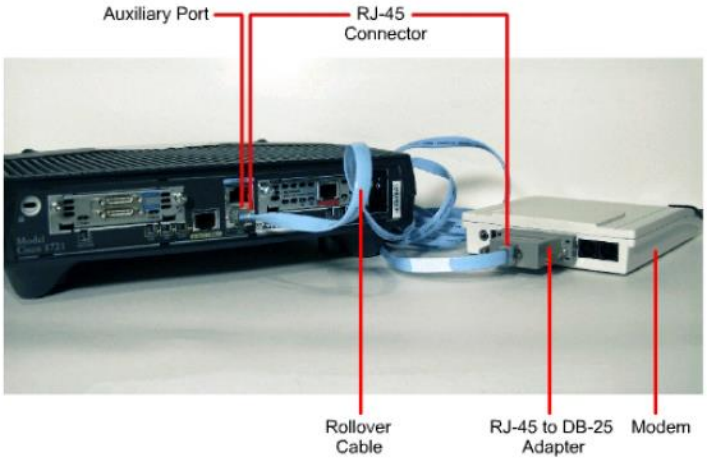
MDF: Main Distribution Frame
IDF: Intermediate Distribution Frame
Ethernet: 100m
MultiMode (MM) Fibre: around 500m
SingleMode (SM) Fibre: more than 2Kms





UTP: Unshielded Twisted Pair cable
STP: Shielded twisted pair cable
Auto-MDIX: Media Dependent Interface Crossover

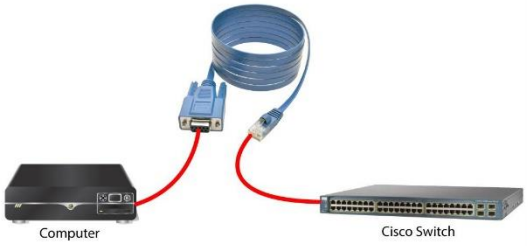




Cisco Serial Console Rollover Cable, RJ45 to DB9 (M/F), 6 ft.

Tripp Lite Model: P430-006

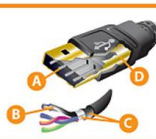
Connect a computer's DB9 serial connector to an RJ45 Cisco console port.

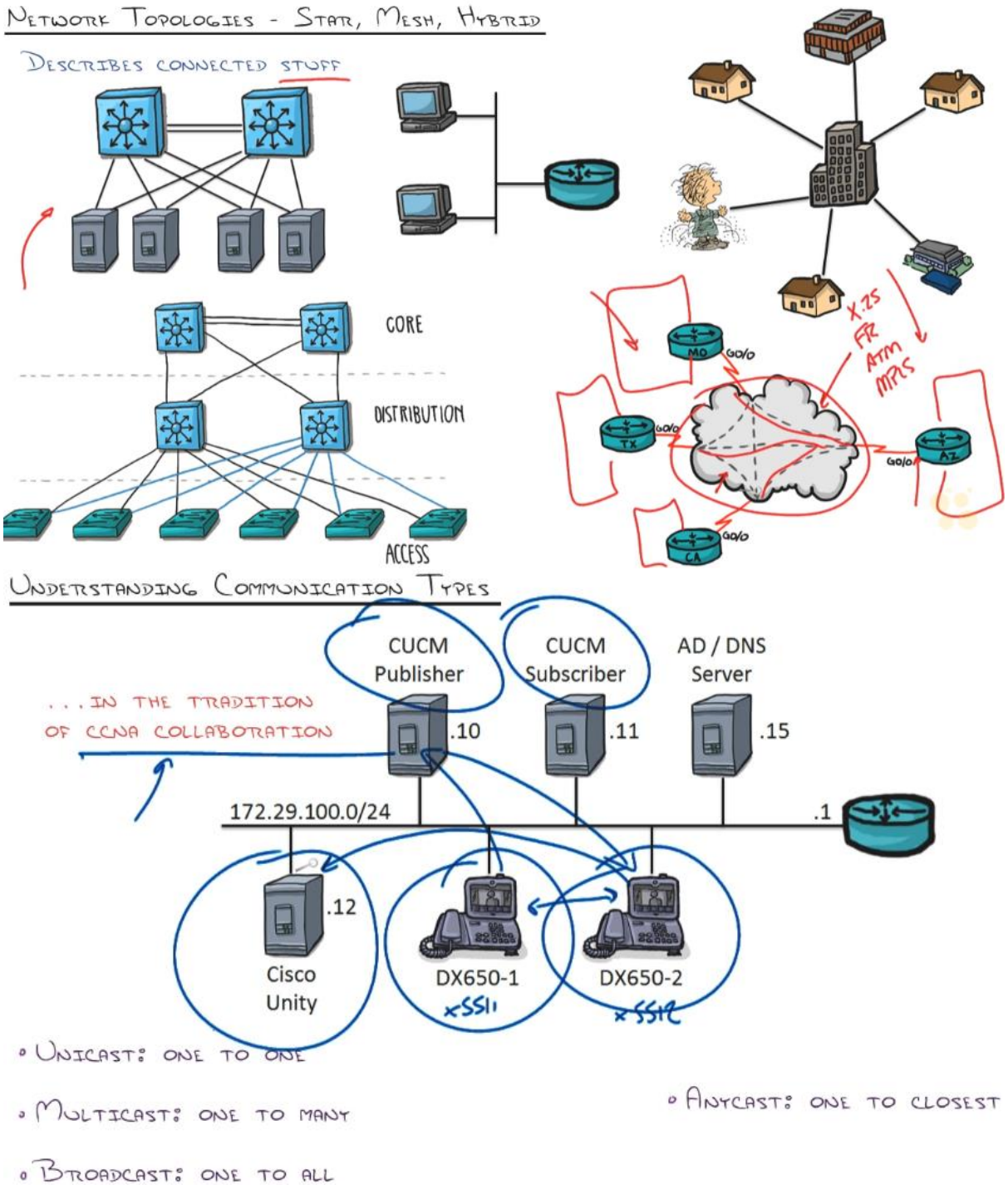


USB to Serial Adapter

Tripp Lite Model: U209-000-R

- A Gold-plated copper contacts
- B Impedance-matched conductors
- C Braid-and-foil shield
- D Undermold Shield

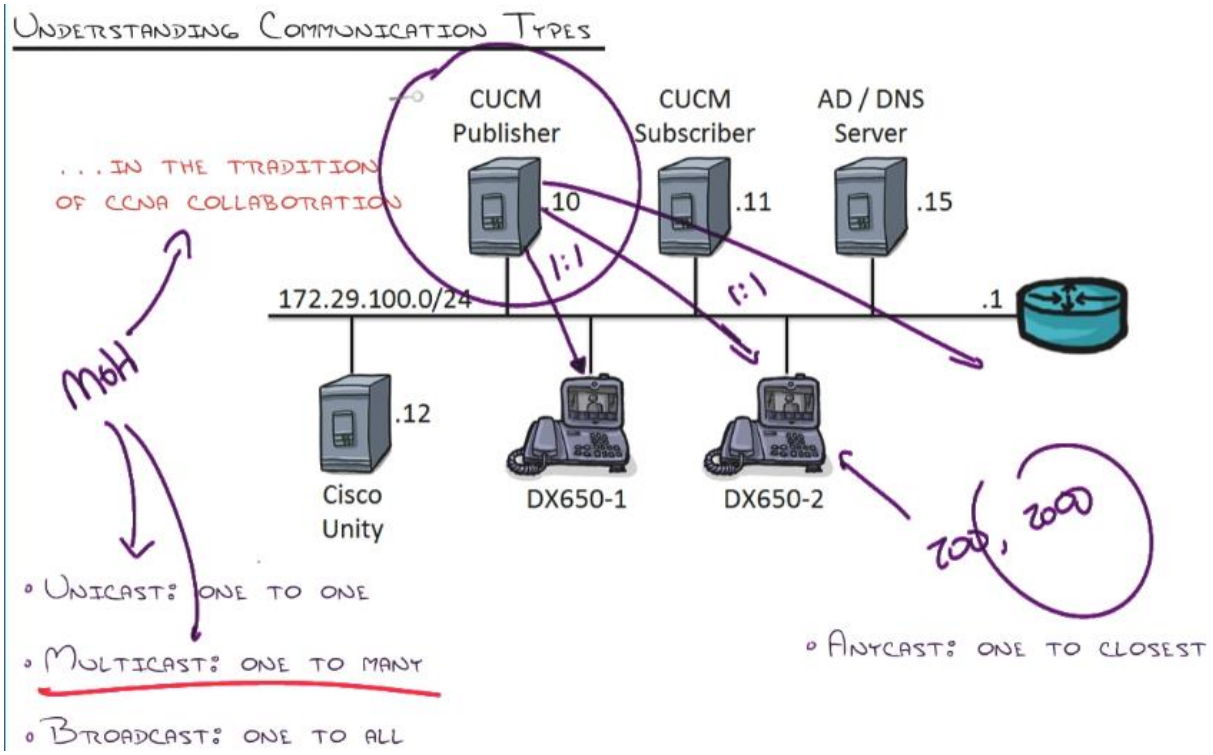


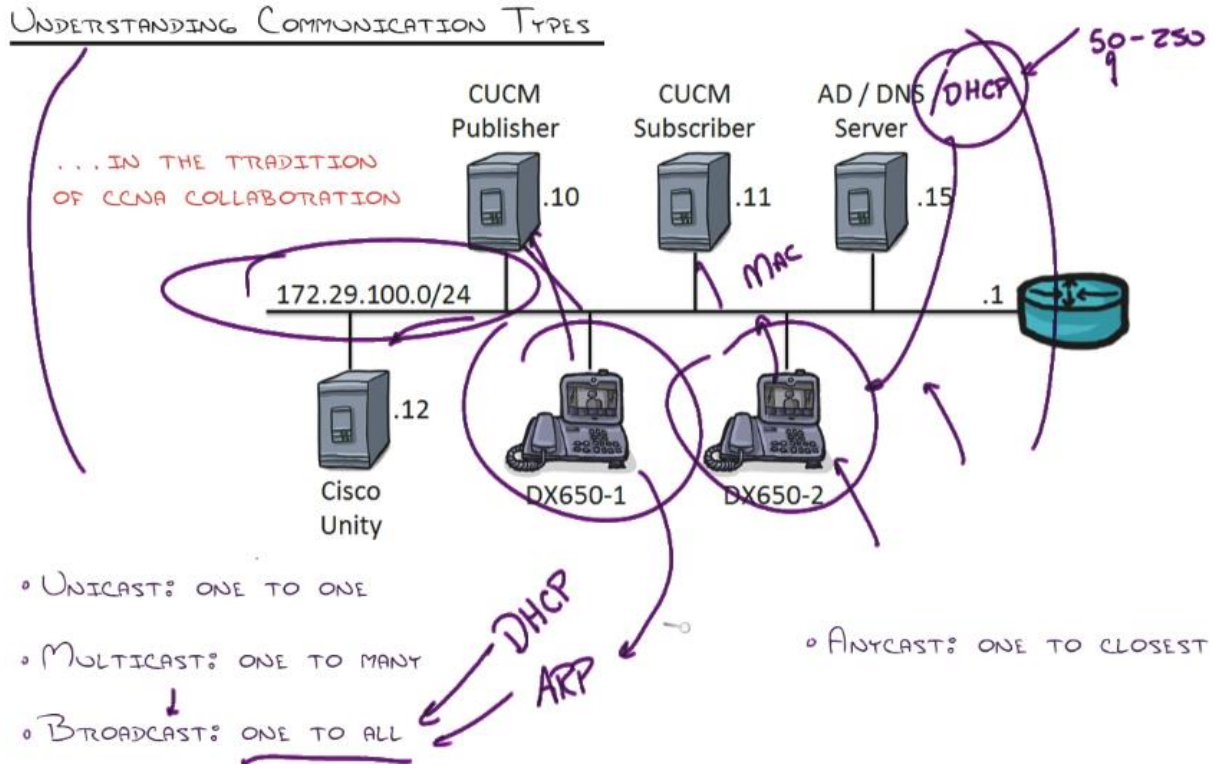


CUCM (Cisco Unified Communication Manager) is what makes the call between IP phones. The brains of the voice network.

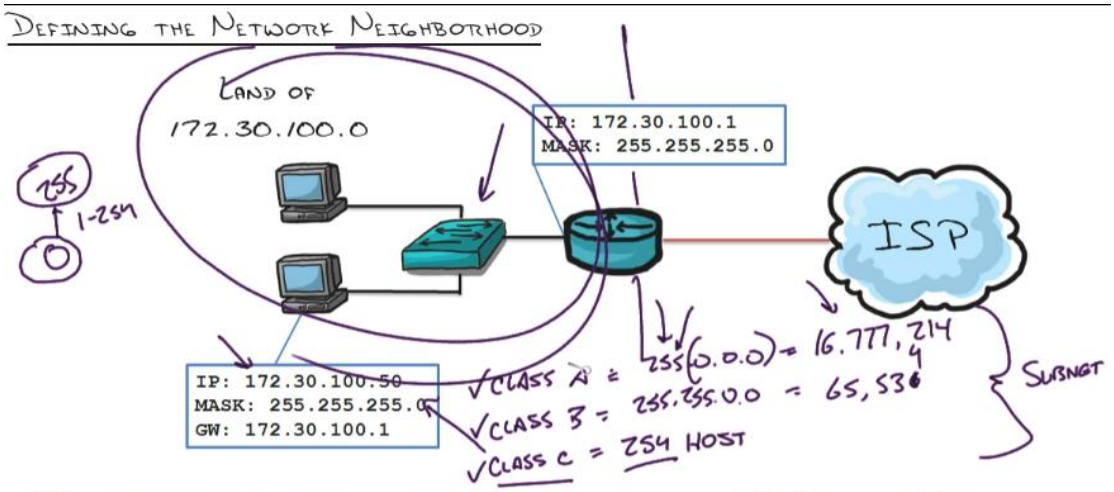
IP phones know how to communicate to the CUCM through dhcp option 66.

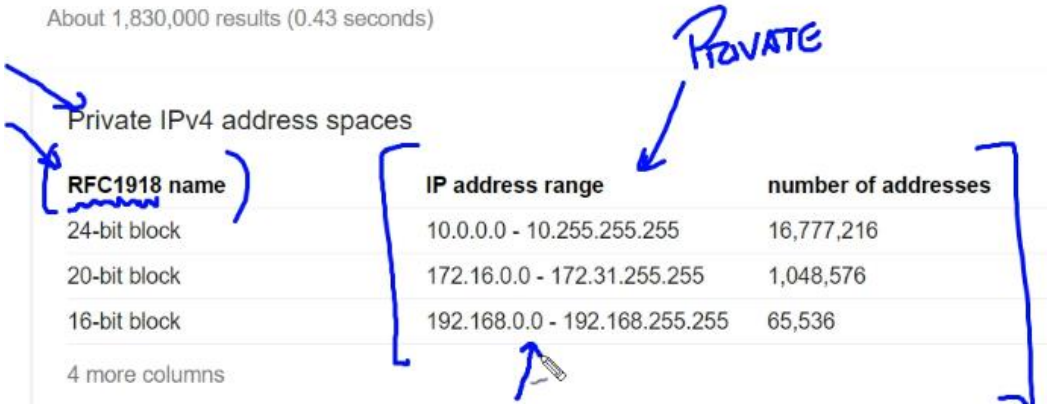
Cisco unity is the voice mail server where the CUCM send the voice mails to if ip phone hasn't accepted the call.



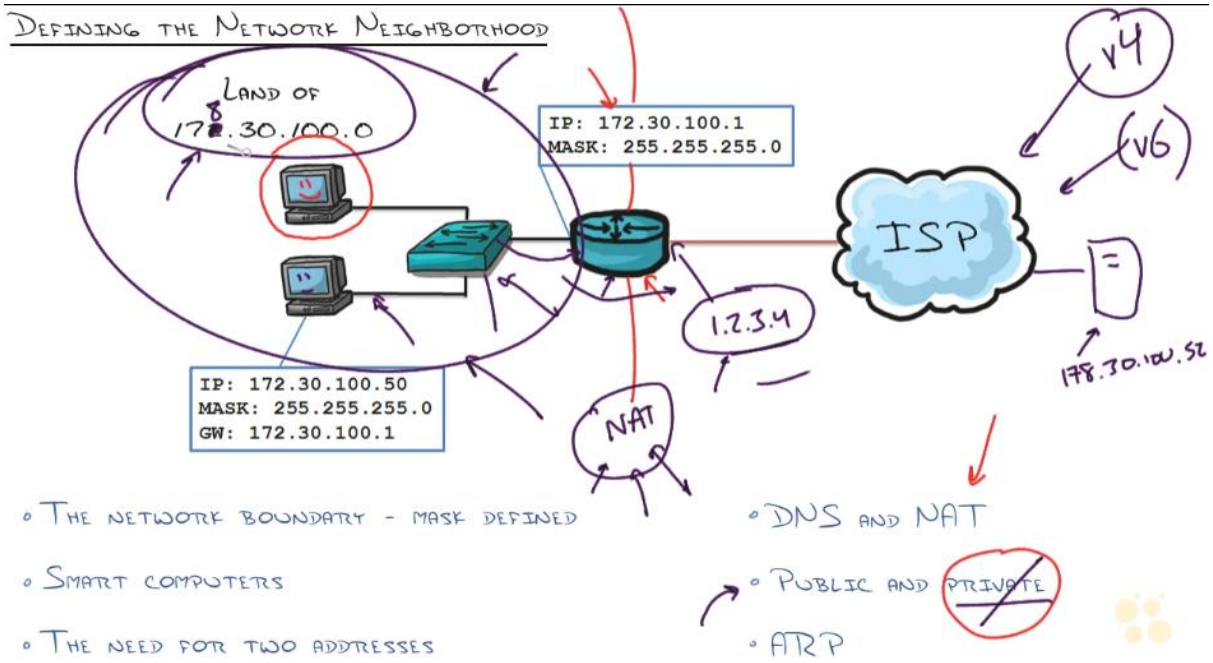


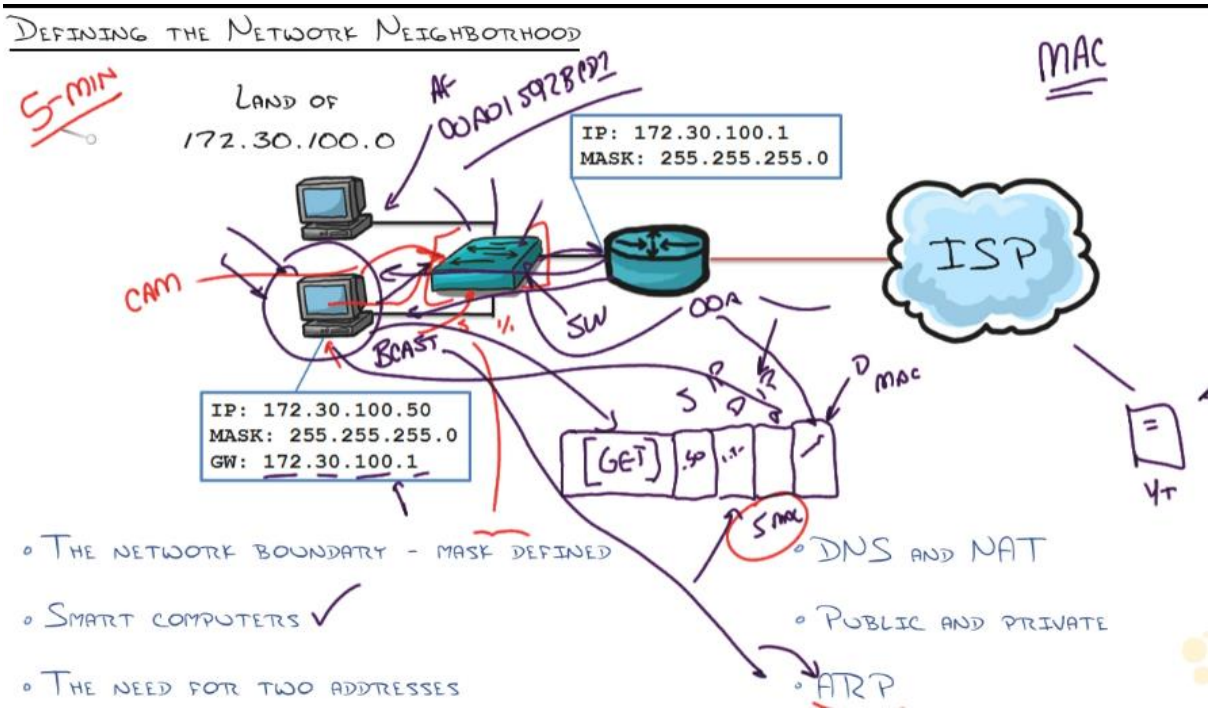
Anycast is used in IPv6 and can be used in IPv4 for DNS.





Private IP addresses are not routable on the internet because ISPs block them.





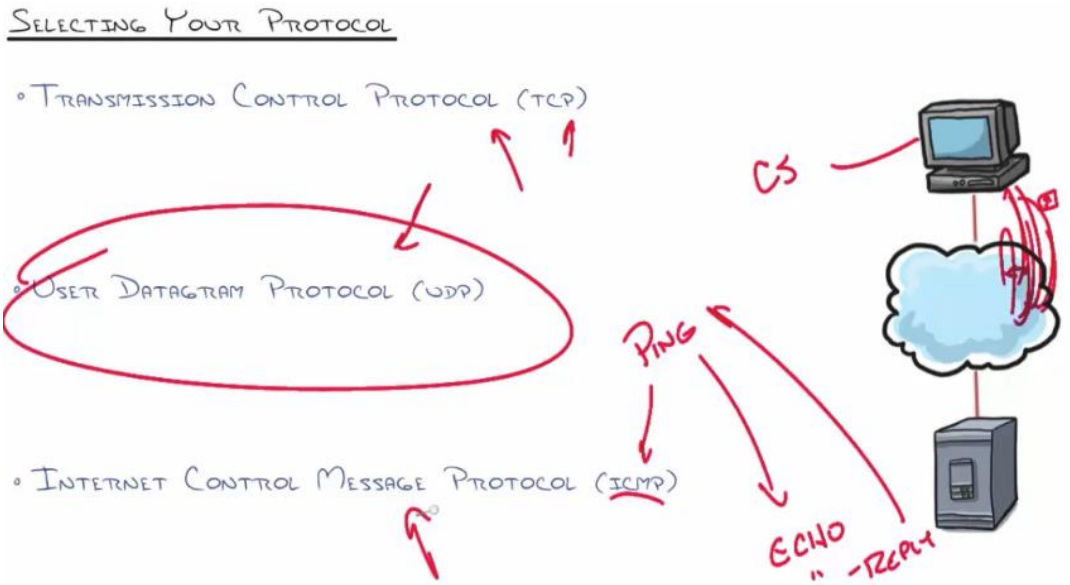
Select C:\WINDOWS\system32\cmd.exe

```
C:\Users\Jeremy Cioara>arp -a

Interface: 192.168.157.1 --- 0xc
  Internet Address      Physical Address      Type
  192.168.157.254       00-50-56-fc-d2-67     dynamic
  192.168.157.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

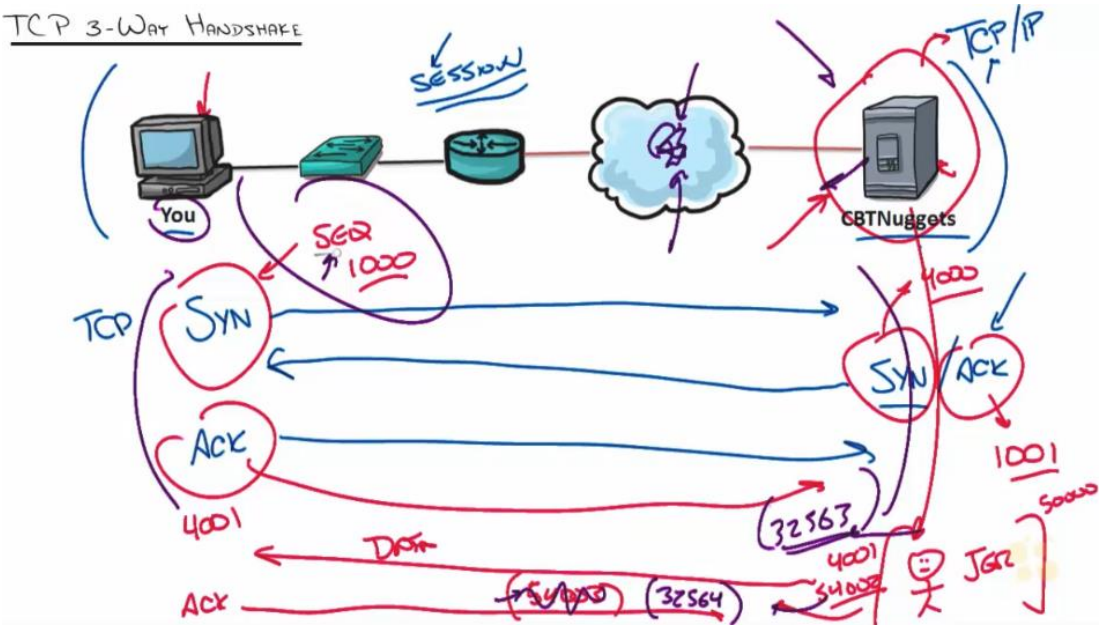
```
C:\Users\Jeremy Cioara>arp -d *
The ARP entry deletion failed: The requested operation requires elevation.


C:\Users\Jeremy Cioara>arp -a

Interface: 192.168.157.1 --- 0xc
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16     static

Interface: 10.225.1.132 --- 0x12
  Internet Address      Physical Address      Type
  10.225.1.252          a8-48-1c-b8-80-30     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
```

Arp cache is by default 5 mins.

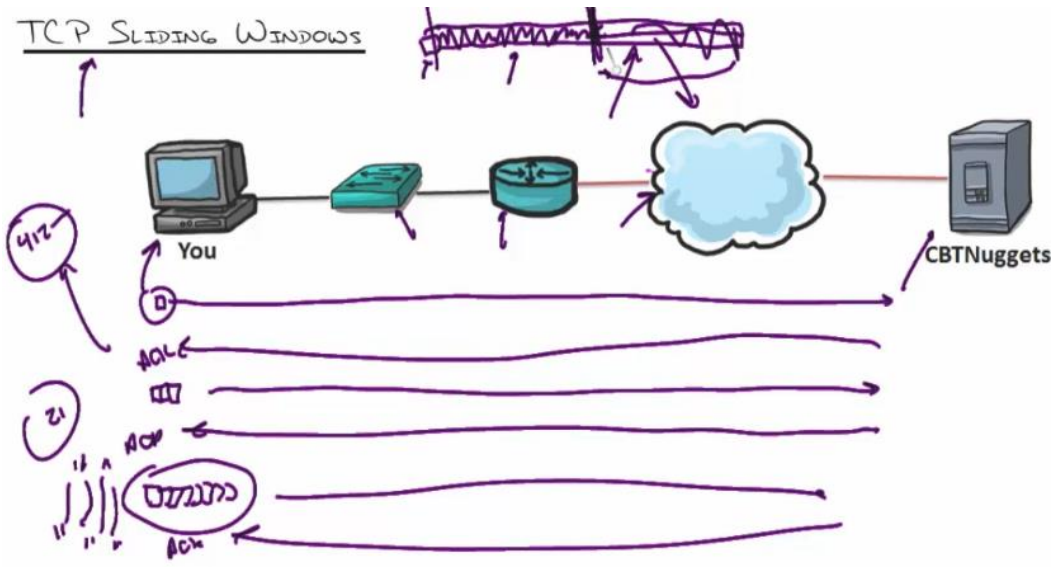
Selecting Your Protocol

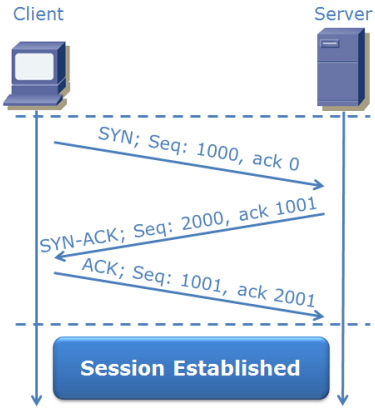


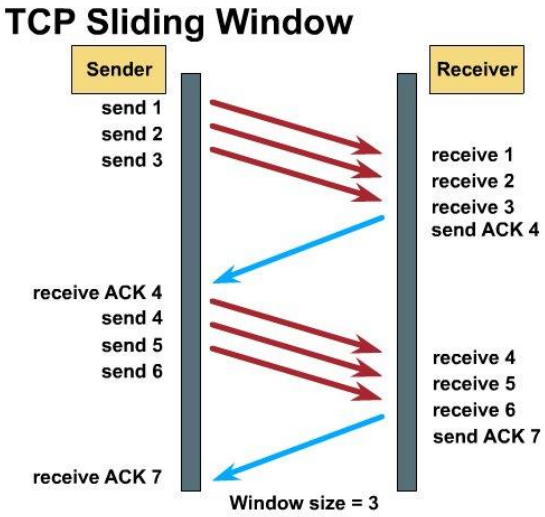
TCP 3-Way Handshake

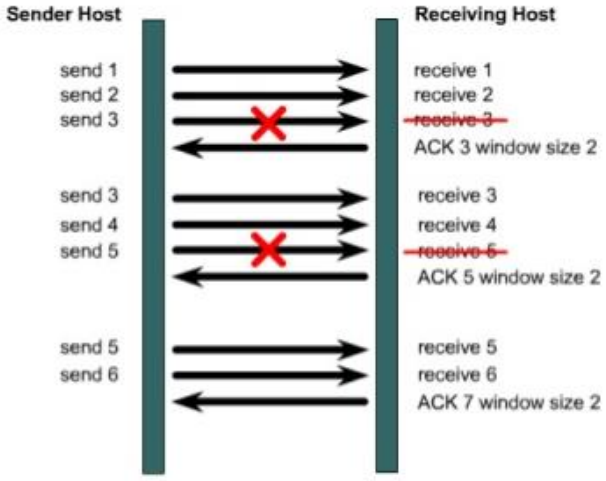


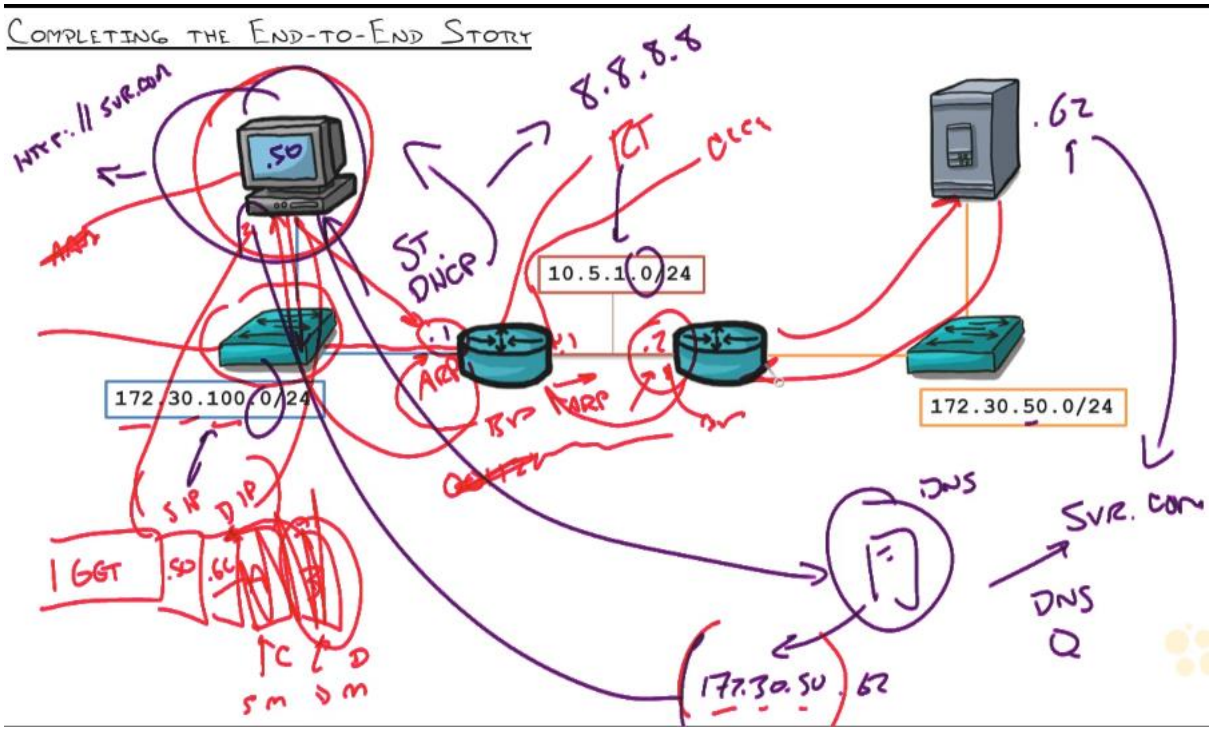
TCP Sliding Windows

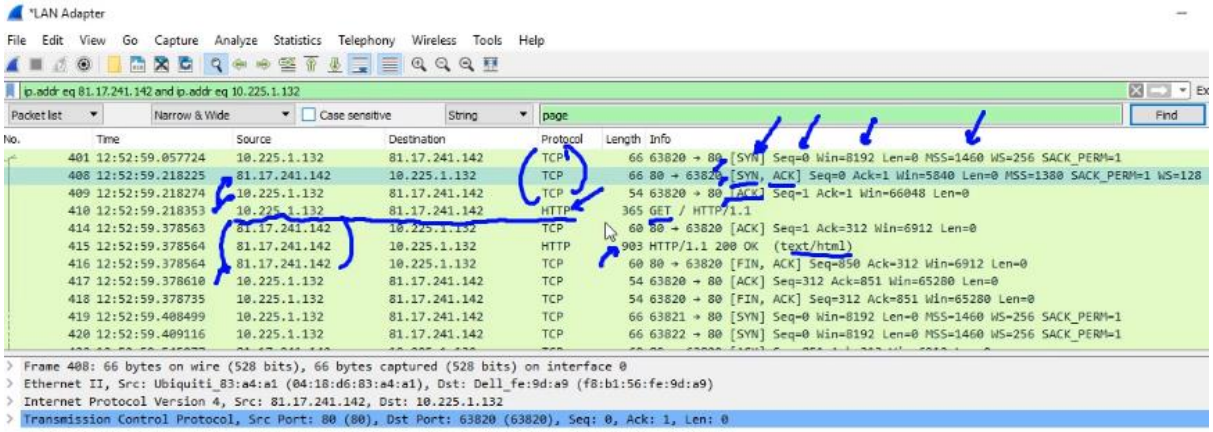



Client
Server
SYN; Seq: 1000, ack 0
SYN-ACK; Seq: 2000, ack 1001
ACK; Seq: 1001, ack 2001
Session Established


TCP Sliding Window
Sender
Receiver
send 1
send 2
send 3
receive 1
receive 2
receive 3
send ACK 4
receive ACK 4
send 4
send 5
send 6
receive 4
receive 5
receive 6
send ACK 7
receive ACK 7
Window size = 3

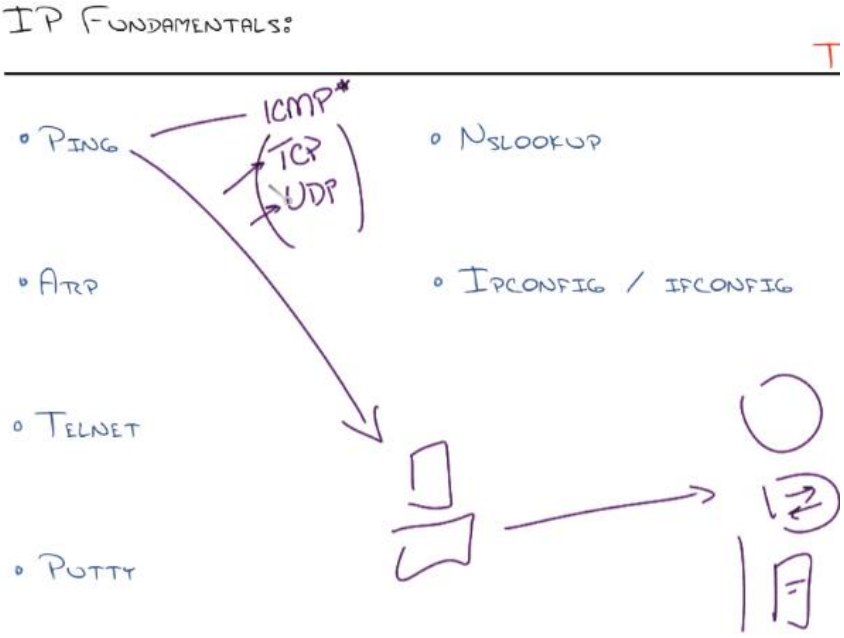

Sender Host
Receiving Host
send 1
send 2
send 3
receive 1
receive 2
receive 3
ACK 3 window size 2
send 3
send 4
send 5
receive 3
receive 4
receive 5
ACK 5 window size 2
send 5
send 6
receive 5
receive 6
ACK 7 window size 2

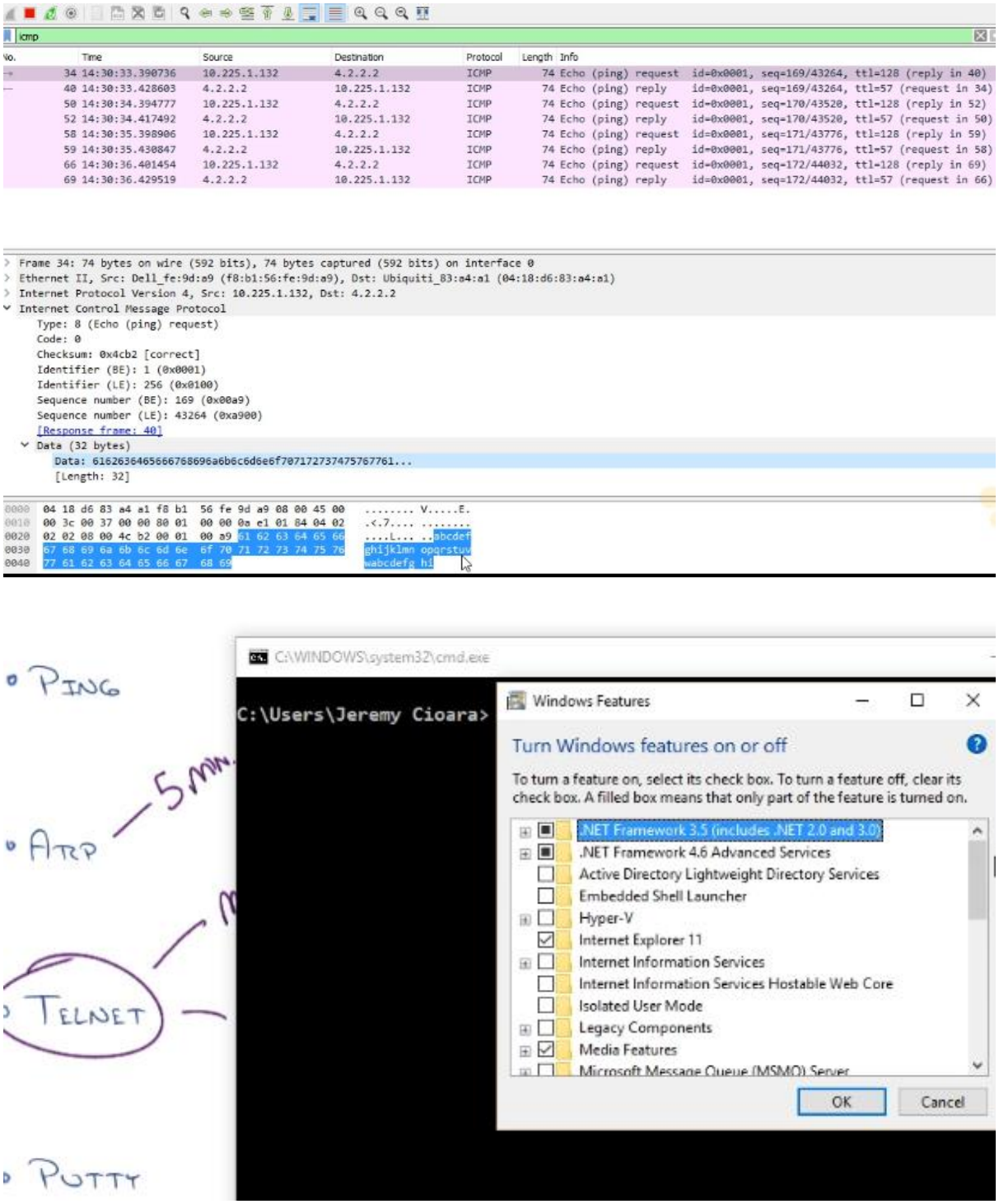

COMPLETING THE END-TO-END STORY
8.8.8.8
10.5.1.0/24
172.30.100.0/24
172.30.50.0/24
DNS
SVR.COM
DNS Q



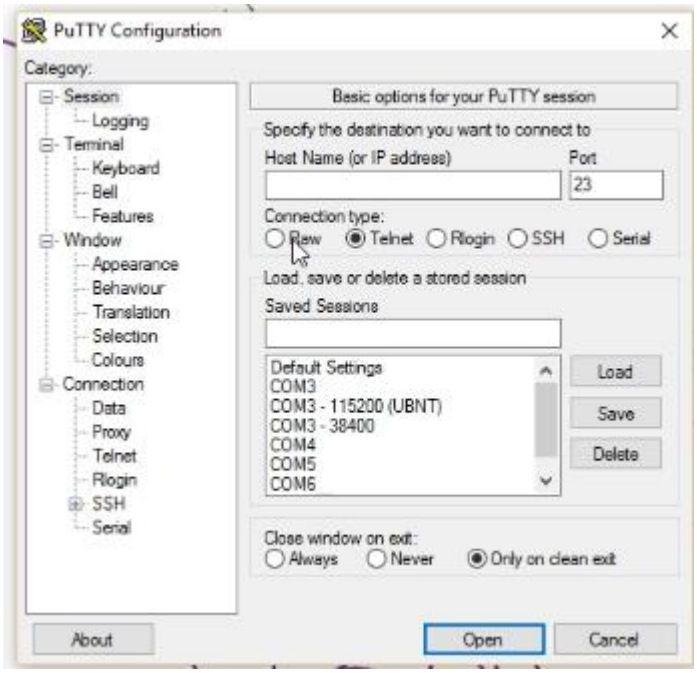

```
C:\Users\Jeremy Cioara>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.225.1.132:50090     server25102:5938       ESTABLISHED
  TCP    10.225.1.132:50141     bbus3001-03:61521      ESTABLISHED
  TCP    10.225.1.132:53216     msnbot-65-52-108-230:https  ESTABLISHED
  TCP    10.225.1.132:63834     64.4.54.254:https      TIME_WAIT
```

Well-known port numbers: 0-1024



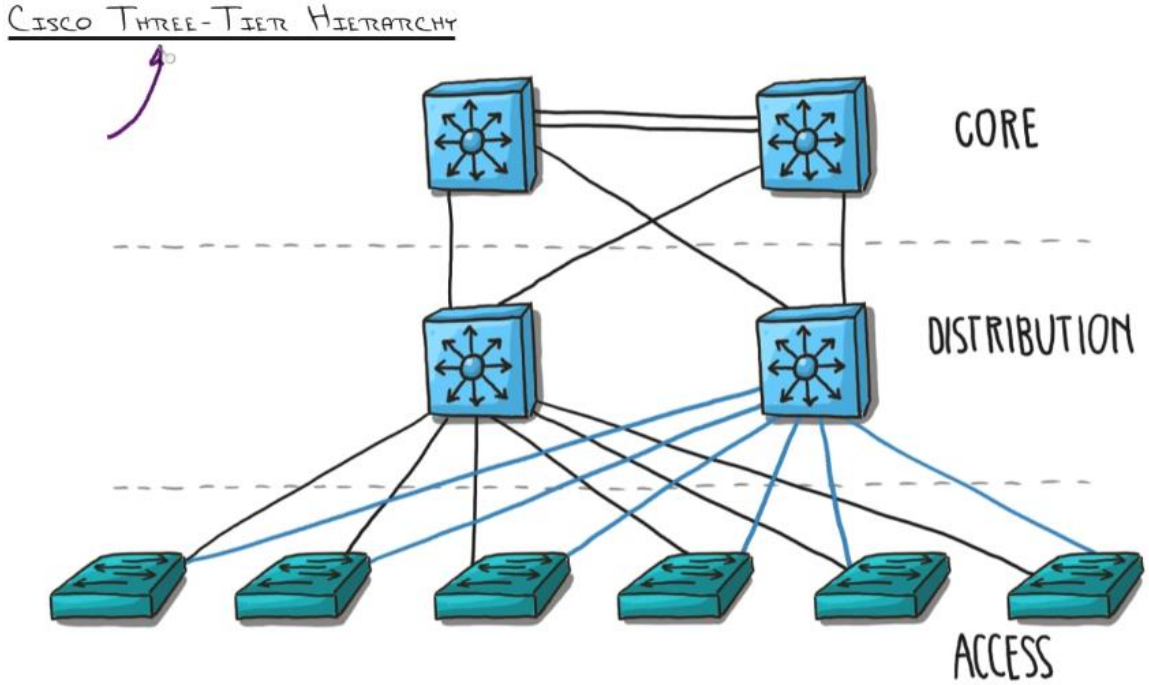


Default telnet port is 23, but you can telnet to any port. Not used for management anymore.
telnet 1.1.1.1
telnet 1.1.1.1 80
telnet google.com 80
telnet google.com 443

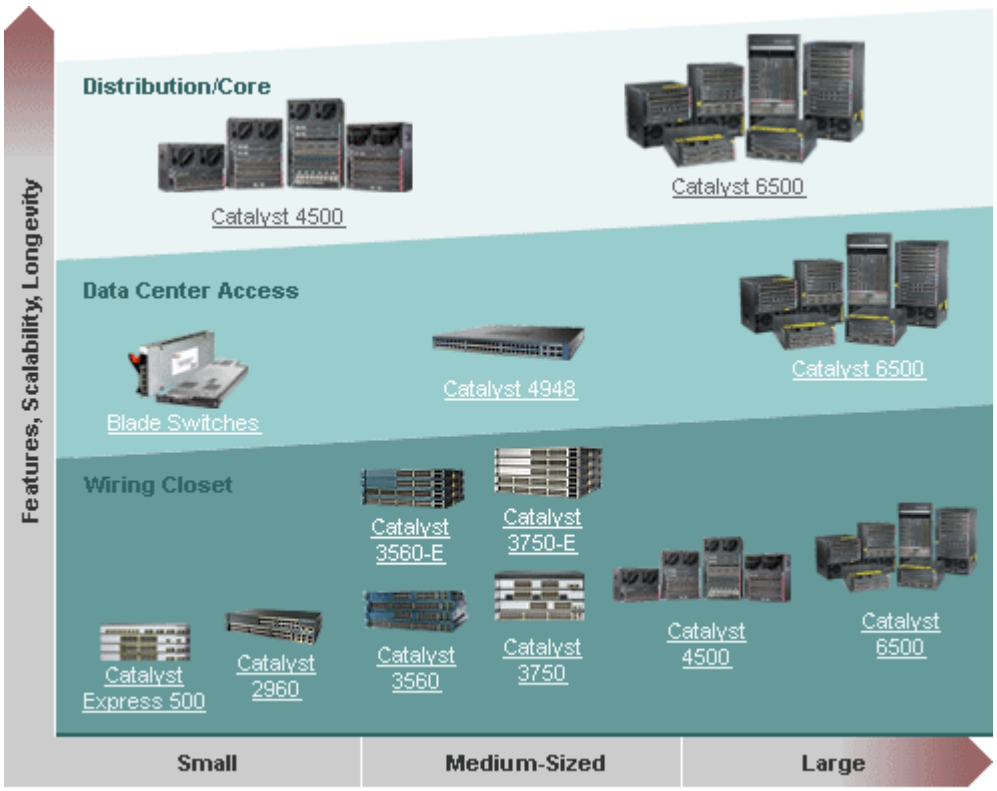


Youcan use securecrt instead of telnet. Putty is free.
ipconfig
ipconfig/all
ipconfig/release
ipconfig/renew
ipconfig/flushdns

```
C:\Users\Jeremy Cioara>tracert -d 4.2.2.2

Tracing route to 4.2.2.2 over a maximum of 30 hops

  1     *        *        *     Request timed out.
  2    13 ms    13 ms    16 ms  10.35.64.1
  3    10 ms    12 ms    25 ms  100.127.65.160
  4    13 ms    13 ms    12 ms  70.169.73.90
  5    20 ms    13 ms    14 ms  4.31.188.61
  6    21 ms    19 ms    22 ms  4.69.144.201
  7    22 ms    24 ms    34 ms  4.69.144.201
  8    22 ms    20 ms    20 ms  4.2.2.2

Trace complete.
```



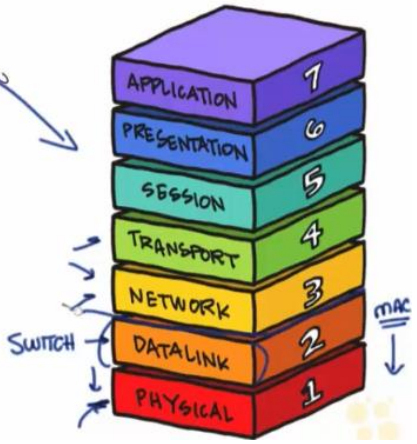
Collapsed core means both the core and the distribution are the same.


Features, Scalability, Longevity
Distribution/Core
Catalyst 4500
Catalyst 6500
Data Center Access
Blade Switches
Catalyst 4948
Catalyst 6500
Wiring Closet
Catalyst 3560-E
Catalyst 3750-E
Catalyst Express 500
Catalyst 2960
Catalyst 3560
Catalyst 3750
Catalyst 4500
Catalyst 6500
Small
Medium-Sized
Large
Organization Size/Density

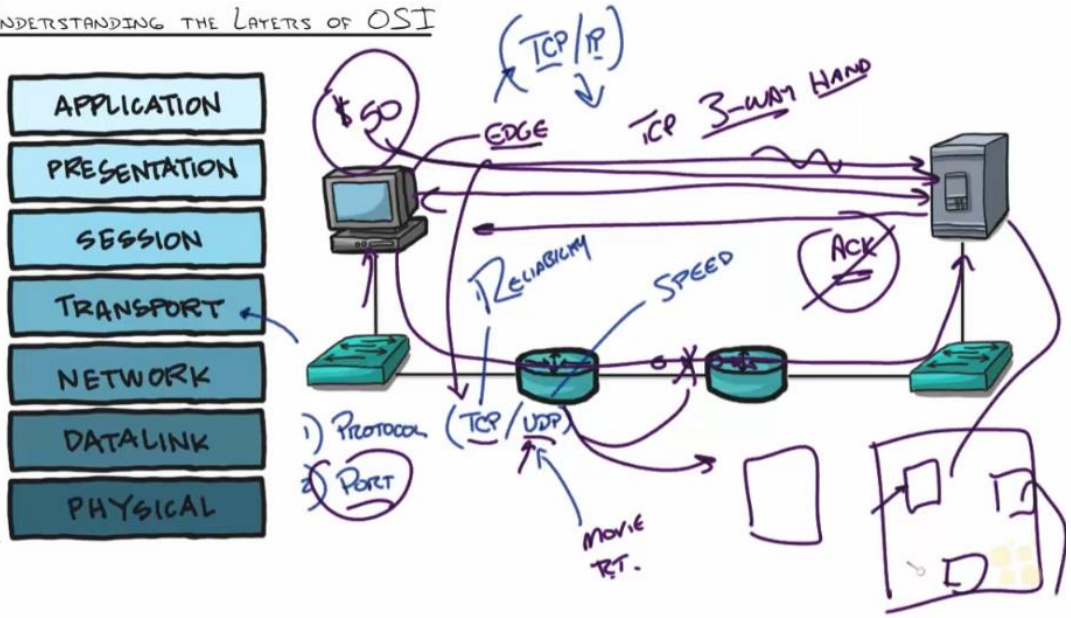

3550
4000
6500
2950

What is the OSI Model?

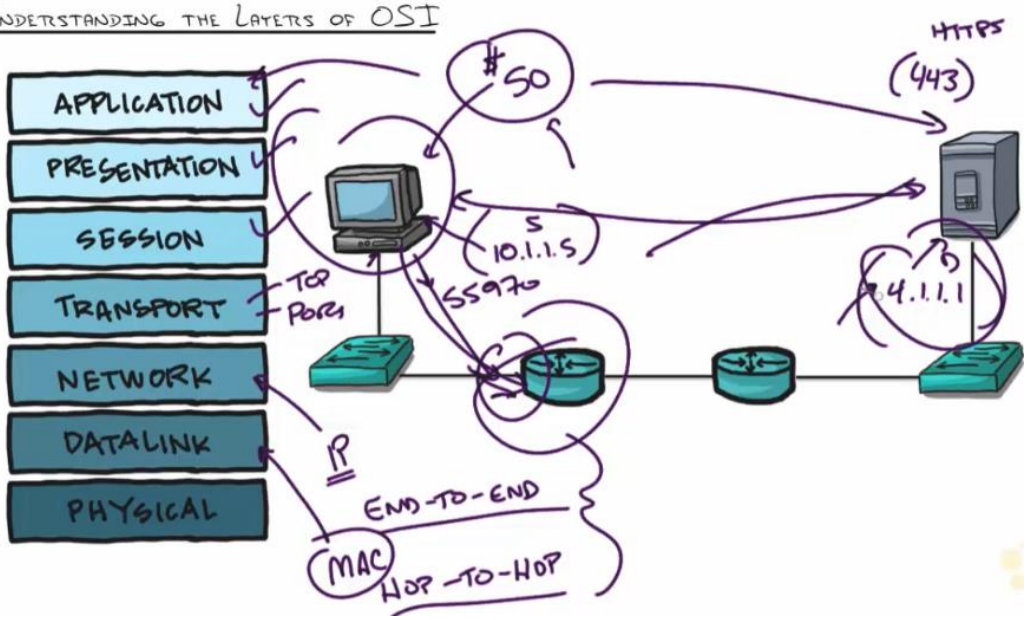
- A standardized architecture defining network communication
- A system to "break down" network communication
- (A standard to create standards)
- A competing protocol to TCP/IP

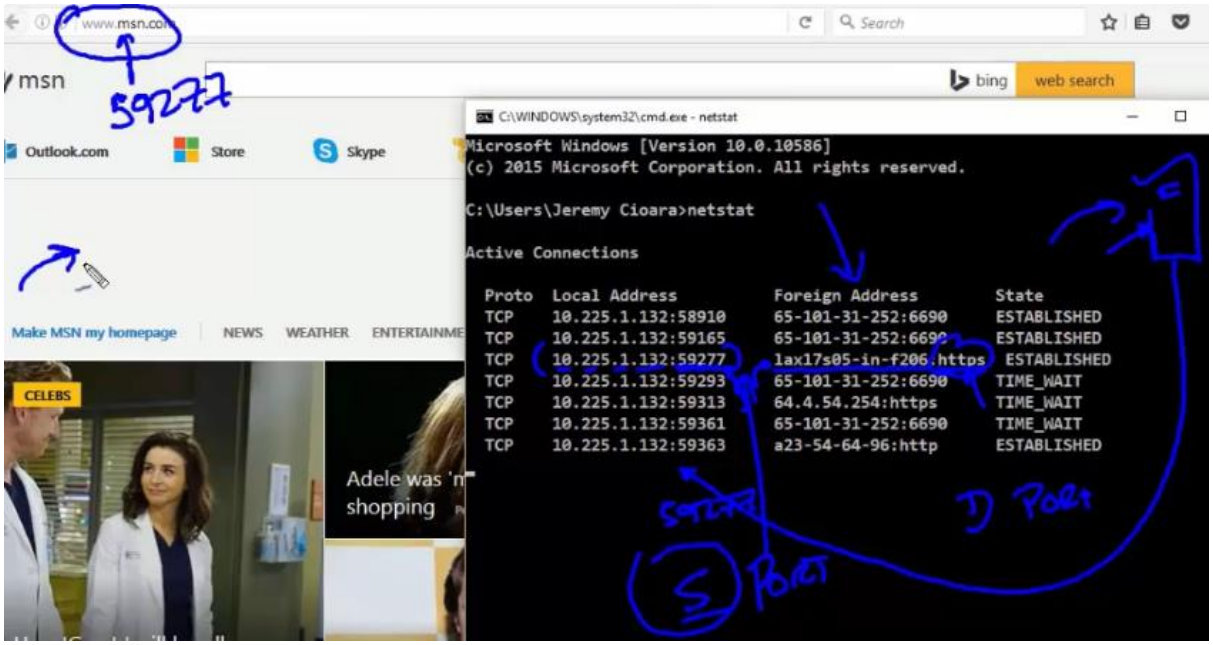


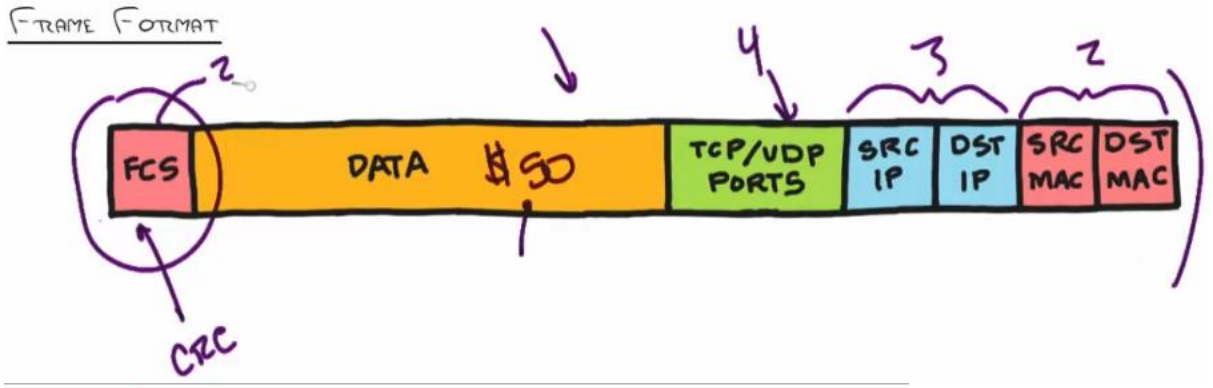
Understanding the Layers of OSI

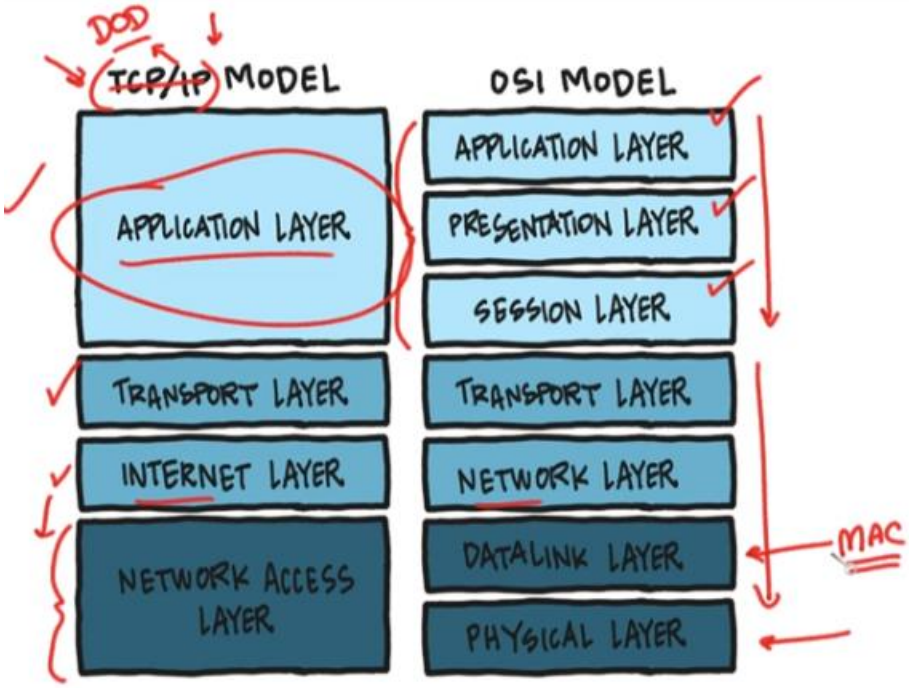


Understanding the Layers of OSI

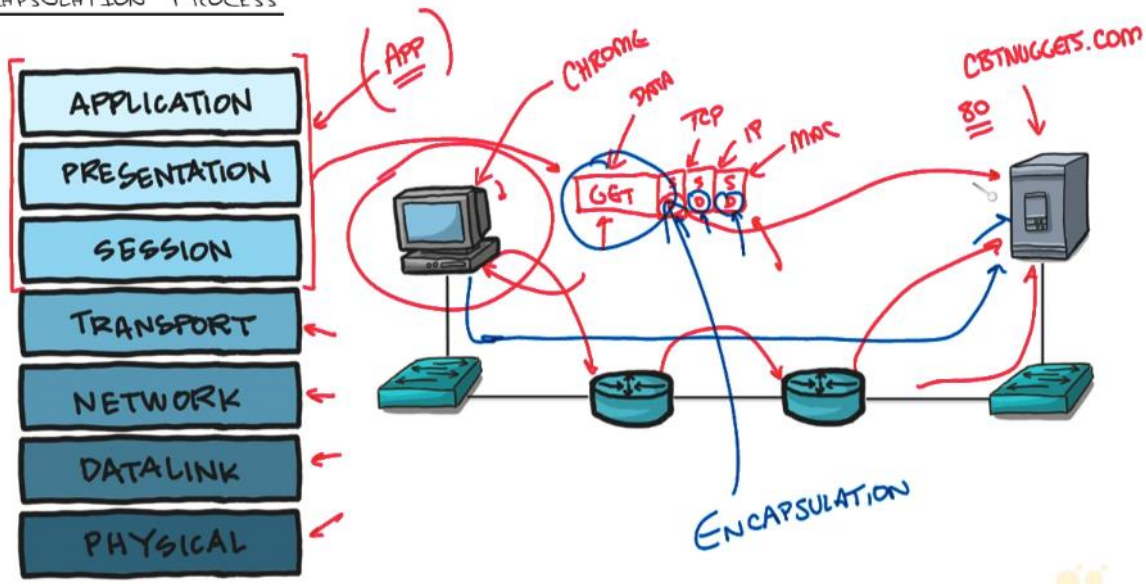


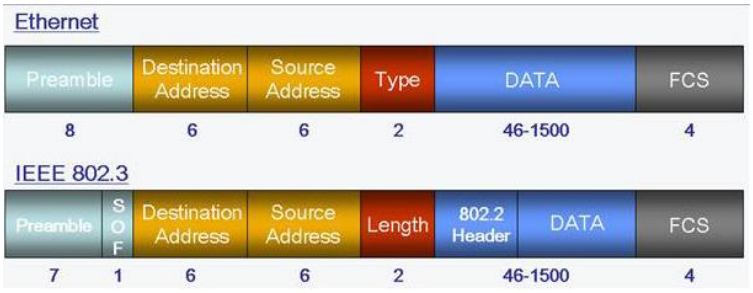

www.msn.com
59277
msn
bing
web search
Outlook.com
Store
Skype
Make MSN my homepage
NEWS
WEATHER
CELEBS
Adele was 'n
shopping
C:\WINDOWS\system32\cmd.exe - netstat
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Jeremy Cioara>netstat
Active Connections
Proto Local Address Foreign Address State
TCP 10.225.1.132:58910 65-101-31-252:6690 ESTABLISHED
TCP 10.225.1.132:59165 65-101-31-252:6690 ESTABLISHED
TCP 10.225.1.132:59277 lax17s05-in-f206.https ESTABLISHED
TCP 10.225.1.132:59293 65-101-31-252:6690 TIME_WAIT
TCP 10.225.1.132:59313 64.4.54.254:https TIME_WAIT
TCP 10.225.1.132:59361 65-101-31-252:6690 TIME_WAIT
TCP 10.225.1.132:59363 a23-54-64-96:http ESTABLISHED

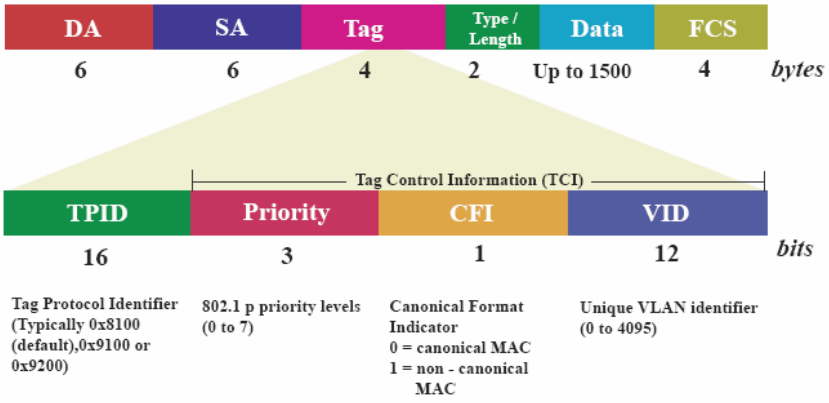

FRAME FORMAT
FCS
DATA
TCP/UDP PORTS
SRC IP
DST IP
SRC MAC
DST MAC
CRC

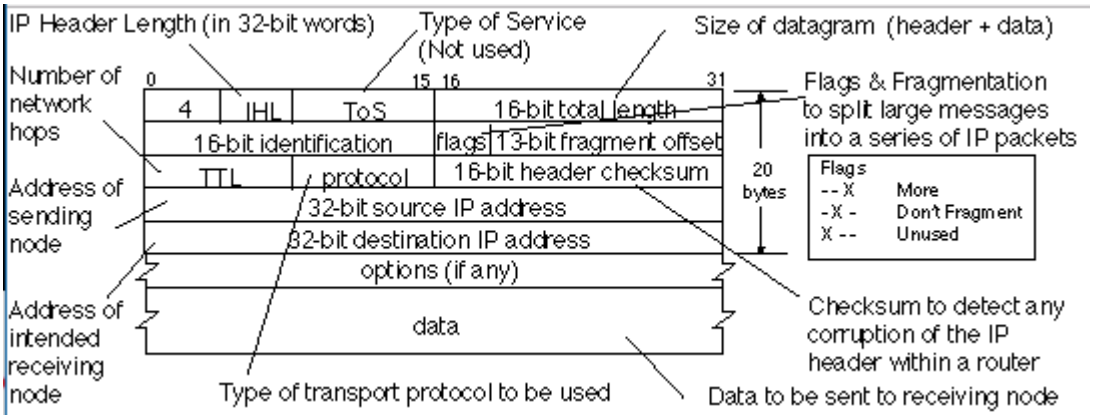

DOD
TCP/IP MODEL
OSI MODEL
APPLICATION LAYER
TRANSPORT LAYER
INTERNET LAYER
NETWORK ACCESS LAYER
APPLICATION LAYER
PRESENTATION LAYER
SESSION LAYER
TRANSPORT LAYER
NETWORK LAYER
DATALINK LAYER
PHYSICAL LAYER
MAC

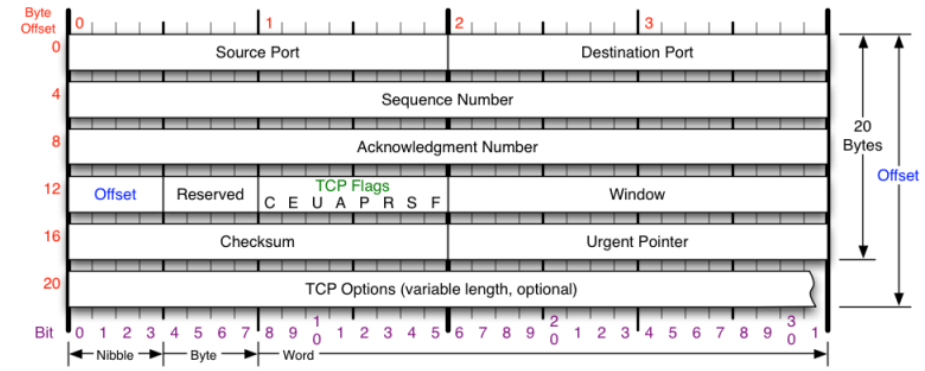
Encapsulation Process











TCP Flags

C E U A P R S F

Congestion Window
C 0x80 Reduced (CWR)
E 0x40 ECN Echo (ECE)
U 0x20 Urgent
A 0x10 Ack
P 0x08 Push
R 0x04 Reset
S 0x02 Syn
F 0x01 Fin

Congestion Notification

ECN (Explicit Congestion Notification). See RFC 3168 for full details, valid states below.

Packet State	DSB	ECN bits
Syn	0 0	1 1
Syn-Ack	0 0	0 1
Ack	0 1	0 0
No Congestion	0 1	0 0
No Congestion	1 0	0 0
Congestion	1 1	0 0
Receiver Response	1 1	0 1
Sender Response	1 1	1 1

TCP Options

0 End of Options List
1 No Operation (NOP, Pad)
2 Maximum segment size
3 Window Scale
4 Selective ACK ok
8 Timestamp

Checksum

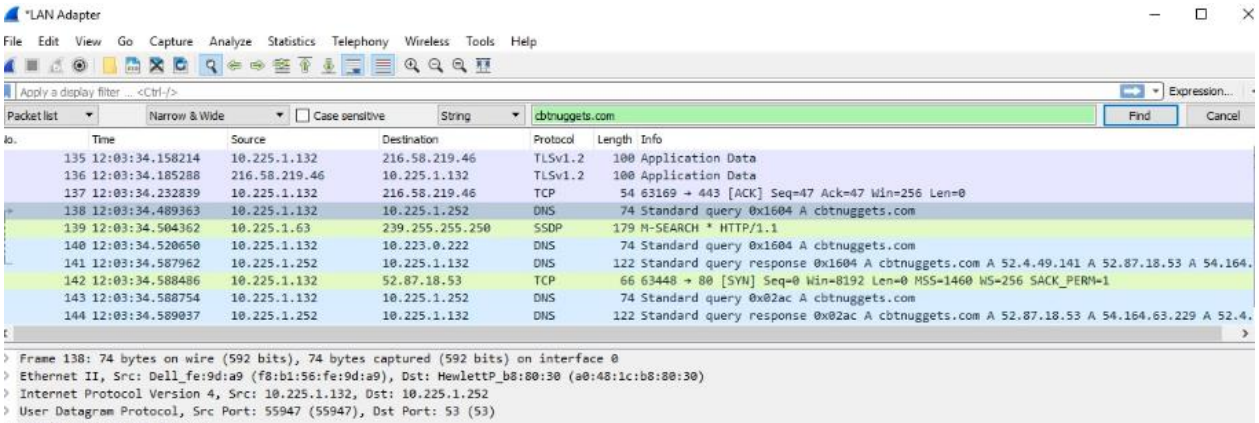
Checksum of entire TCP segment and pseudo header (parts of IP header)

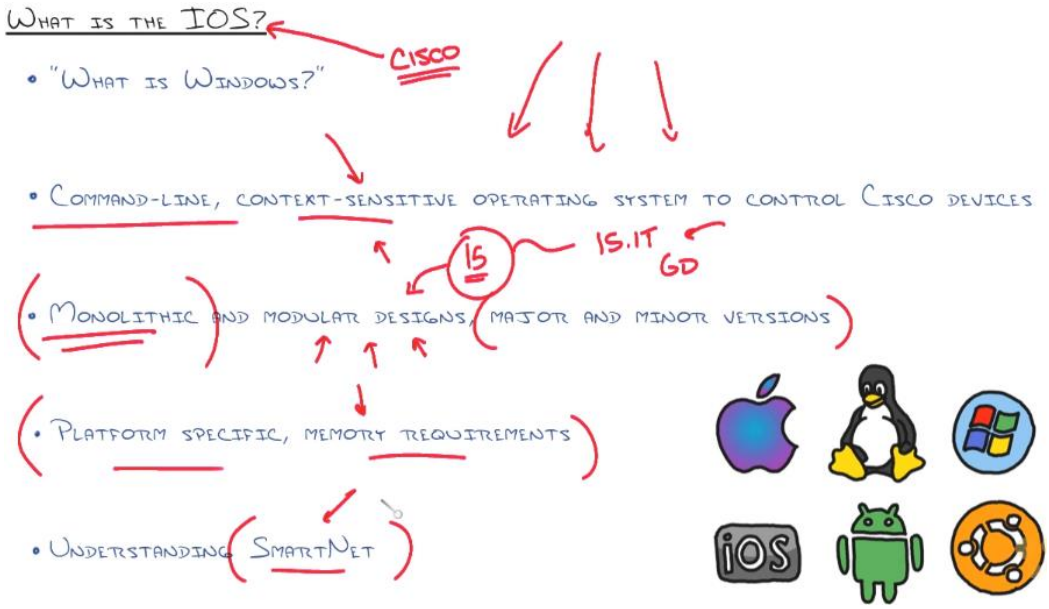
Offset

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

RFC 793

Please refer to RFC 793 for the complete Transmission Control Protocol (TCP) Specification.



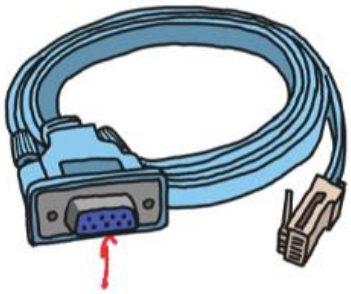


Command-Line Benefits

- Impress your friends, look smart
- Troubleshooting and diagnostics far beyond GUI (GUI)
- Tuning and tweaking configurations far beyond GUI
- Remote accessibility, OS / browser independence
- Configuration speed, scriptability, automation ✓

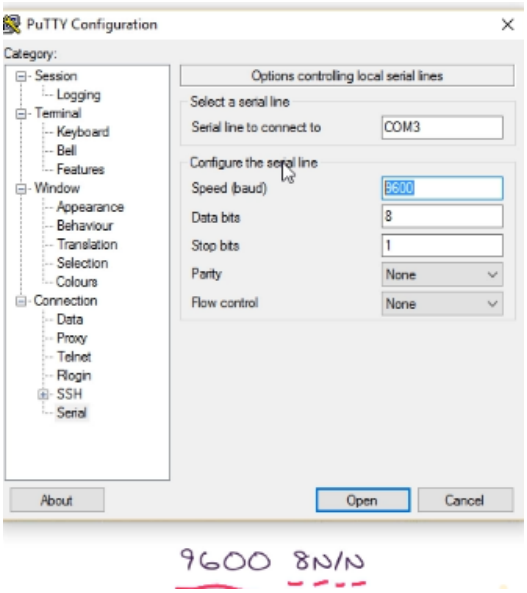
How Do I Get to the IOS?

- Console port ✓ (GUI)



- Telnet
- SSH

Console Connections and Rollover Cables



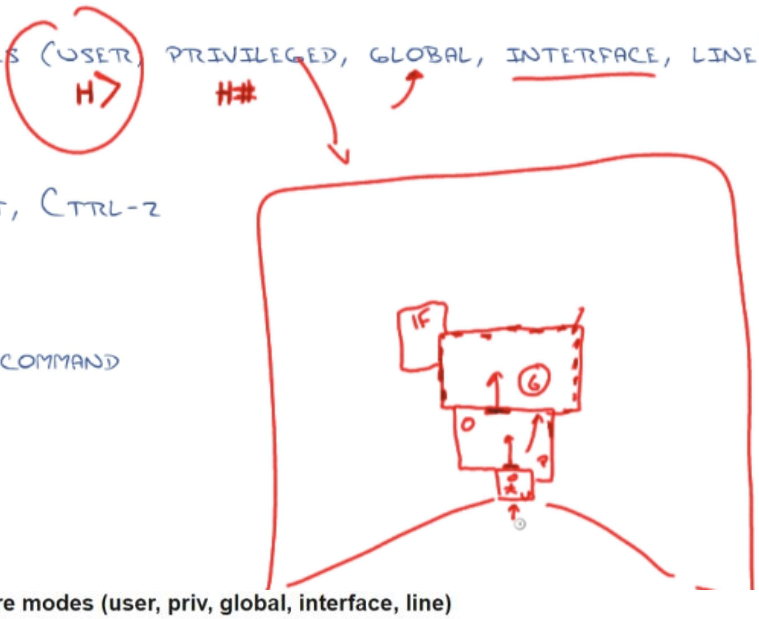
9600 8N/N

Putty and SecureCRT are usually used.

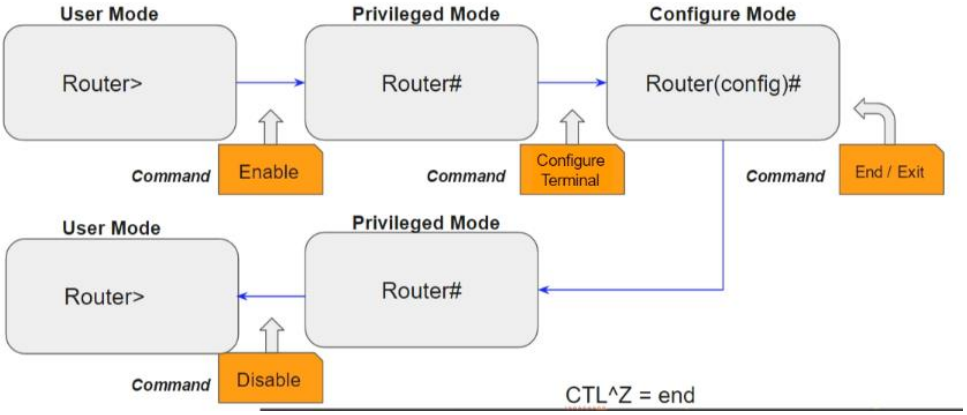
IOS Fundamentals:

IOS Modes an

- Core modes (user, privileged, global, interface, line)
- End, Exit, Ctrl-z
- The "no" command



Core modes (user, priv, global, interface, line)

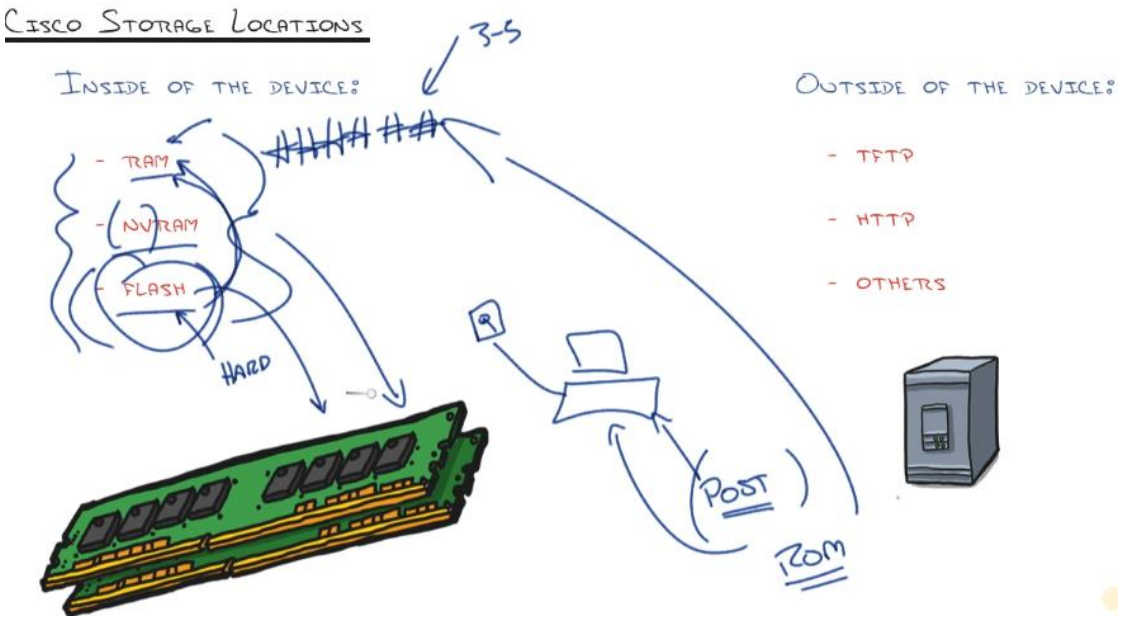


IOS Fundamentals:

IOS Navigation Lab

- Lab steps:

1. **Configure a console connection and connect via Putty**
2. **Discover what interfaces your device has from privileged mode**
3. **Navigate to the configuration mode for the first interface**
4. **Back out to privileged mode with one command**
5. **Adjust the command history buffer to 80 commands**
6. **Change the hostname of your device to Noodles**
7. **Move back to user mode without logging out of the router/switch**
8. **BONUS: Use the ? command to figure out how to set the router/switch clock from privileged mode. HINT: the command starts with "clock" ;o)**

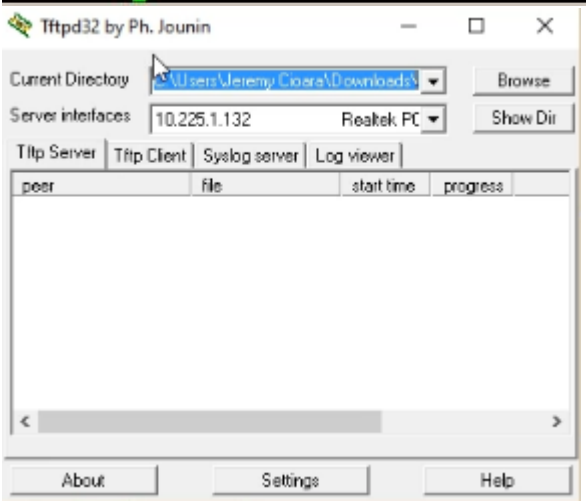


R1#copy running-config startup-config

R1#erase startup-config !(remove configuration)

R1#reload

```
CBTRouter#show flash

System flash directory:
File  Length   Name/status
  1   29715256  c2600-adventerprisek9-mz.124-13b.bin
  2   1642     BackupCFG [deleted]
  3   1642     cbtrouter-confg
[29718736 bytes used, 3311404 available, 33030140 total]
32768K bytes of processor board System flash (Read/Write)

CBTRouter#copy run
CBTRouter#copy running-config flash
Destination filename [cbtrouter-confg]? 2016different-cfg
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0x8060)
899 bytes copied in 6.044 secs (149 bytes/sec)
CBTRouter#
```



```
CBTRouter#copy running-config tftp
Address or name of remote host []? 10.225.1.132
Destination filename [cbtrouter-confg]? routerback.txt
%Error opening tftp://10.225.1.132/routerback.txt (Socket error)
CBTRouter#copy tftp run
CBTRouter#copy tftp running-config
Address or name of remote host []?
```

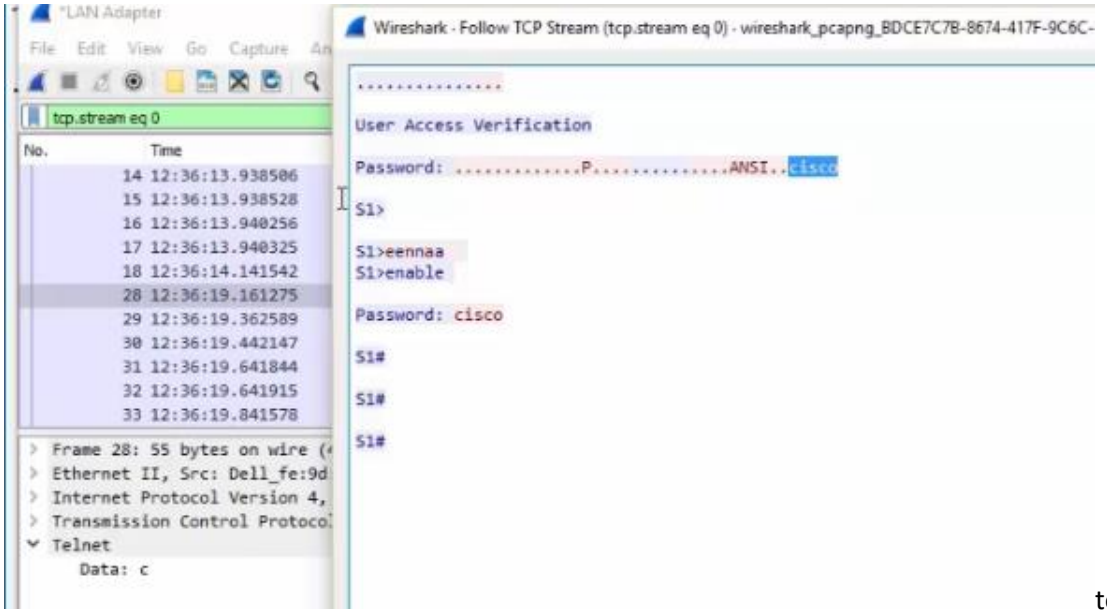
When you copy from startup to running config it merges the config and messes things up further so the best thing is to reboot. You can try using configure replace instead. Anything copied into ram will merge.

'configure replace' can be used to load a configuration file from any supported filesystem, not just NVRAM/Flash.

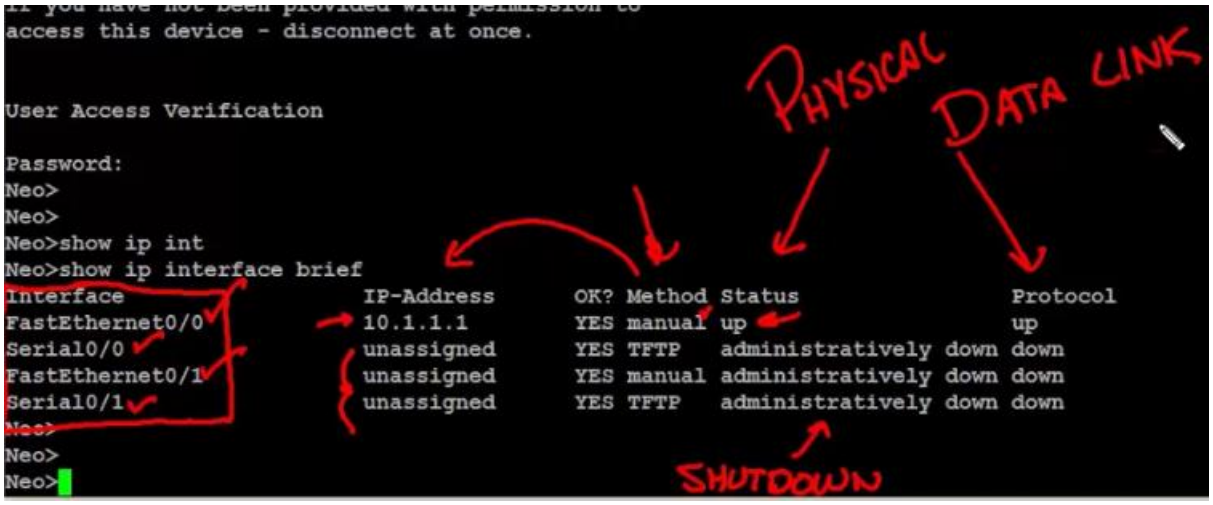
```
Router# configure replace nvram:startup-config
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
*Mar  1 00:22:03.095: Rollback:Acquired Configuration lock.
*Mar  1 00:22:06.619: %PARSER-6-EXPOSEDLOCKRELEASED: Exclusive configuration lock released fr
*Mar  1 00:22:08.627: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:22:09.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, change
The rollback configlet from the last pass is listed below:
********

!List of Commands:
line vty 0 4
 no login
end
********

Rollback aborted after 5 passes
Router#
*Mar  1 00:22:14.995: %PARSER-3-CONFIGNOTLOCKED: Unlock requested by process '193'. Configura
Router#
```

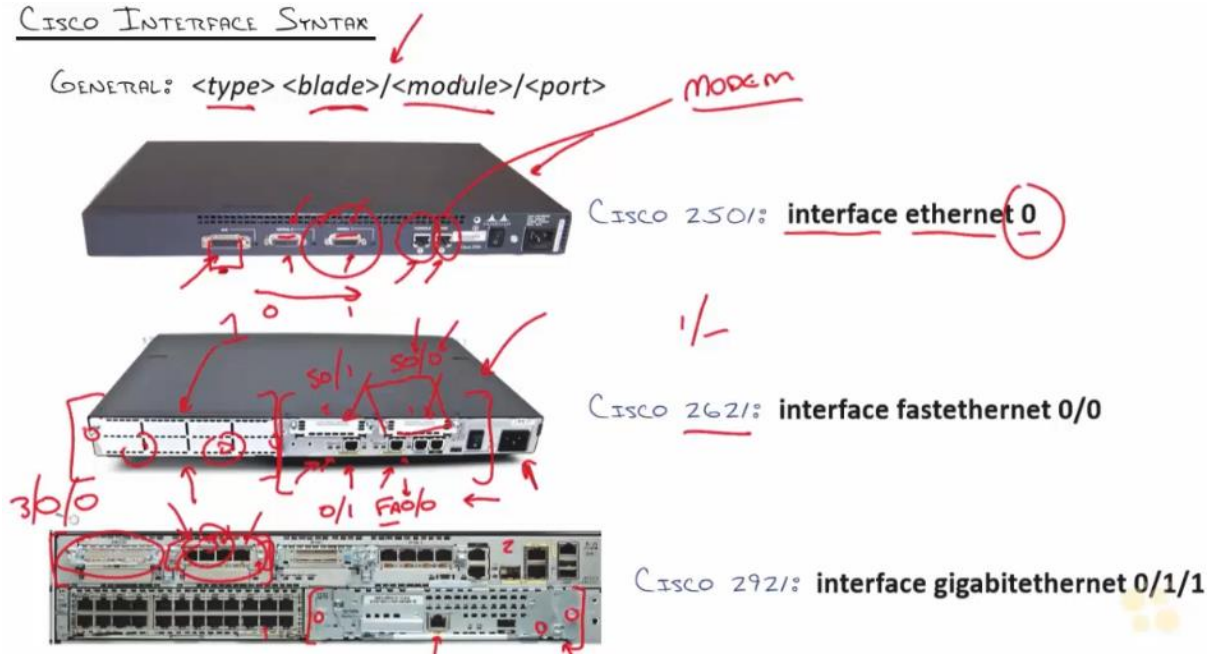
IOS Base config:

1. Hostname
2. Banner
3. Logging synchronous
4. Console password
5. Vty password
6. Ip address/activate interface
7. No ip domain-lookup
8. Service password-encryption
9. Enable password/secret
10. Copy run start



telnet is not secure for remote management and is not used anymore. It is a best practise to disable telnet and use ssh





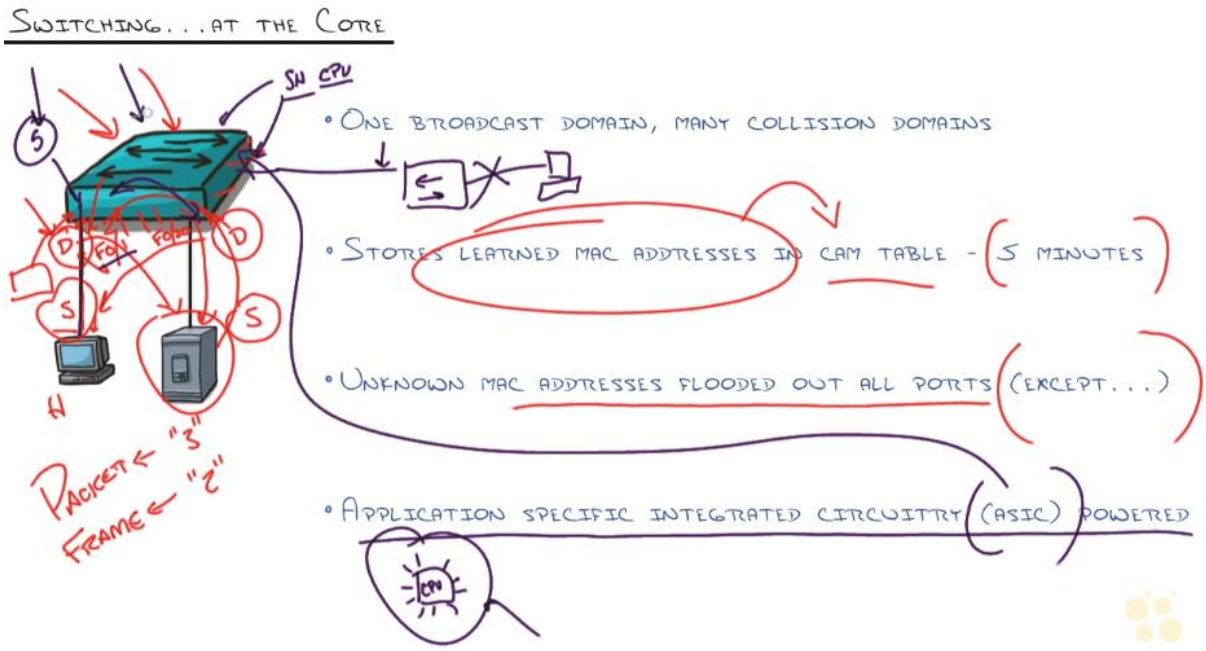
With 3750 stackwise switches you combine the management plane of the switches into one.

IOS Fundamentals:

IOS F

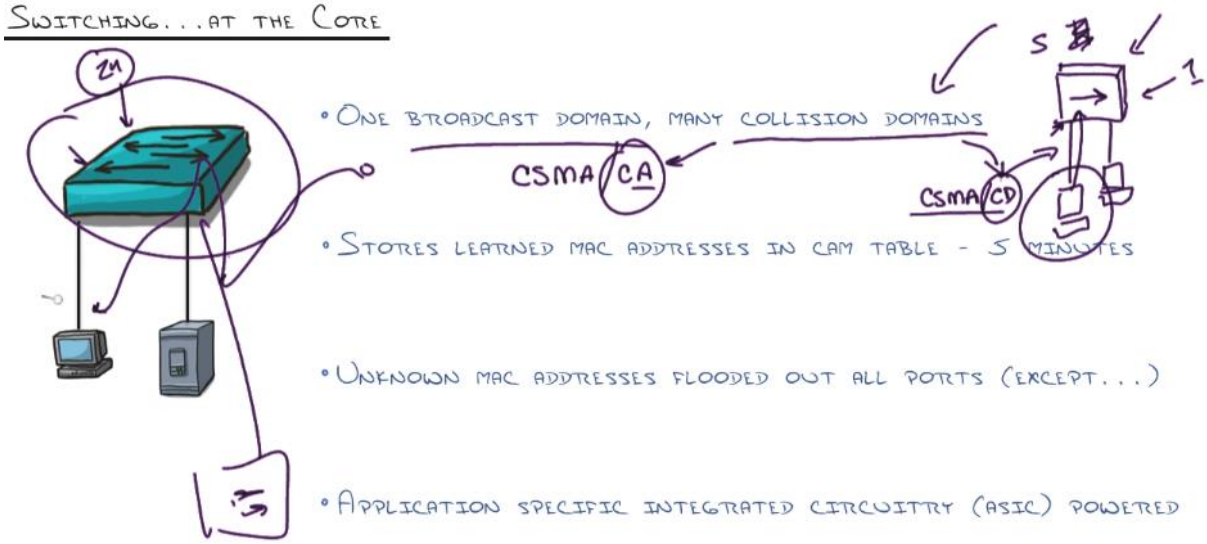
° Lab steps:

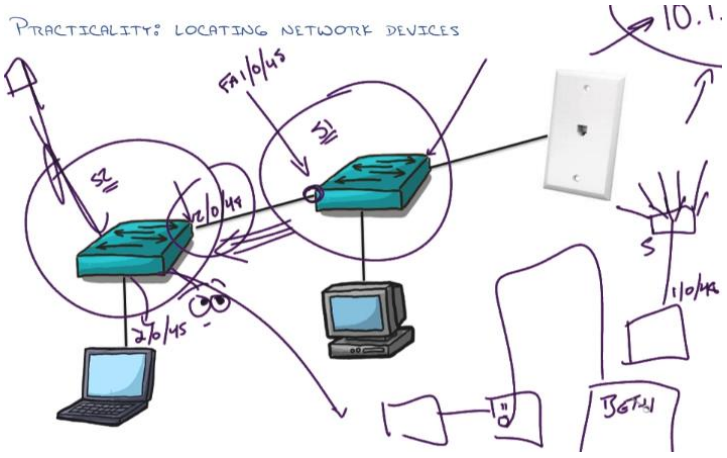
1. Using a console connection, log into your Cisco device
2. Assign your Cisco device an IP address from the 10.2.1.0/24 network
3. Assign your computer an IP address from the same subnet
4. Set the telnet and enable password to "ninja" using the most secure method possible.
5. Telnet to the device and set a logon banner for all future sessions (creativity counts!)
6. Change the hostname of your device to your first name
7. Prevent your device from looking up mistyped privileged mode commands via DNS
8. BONUS: Download tftpd32 to your computer and backup your running configuration via TFTP server



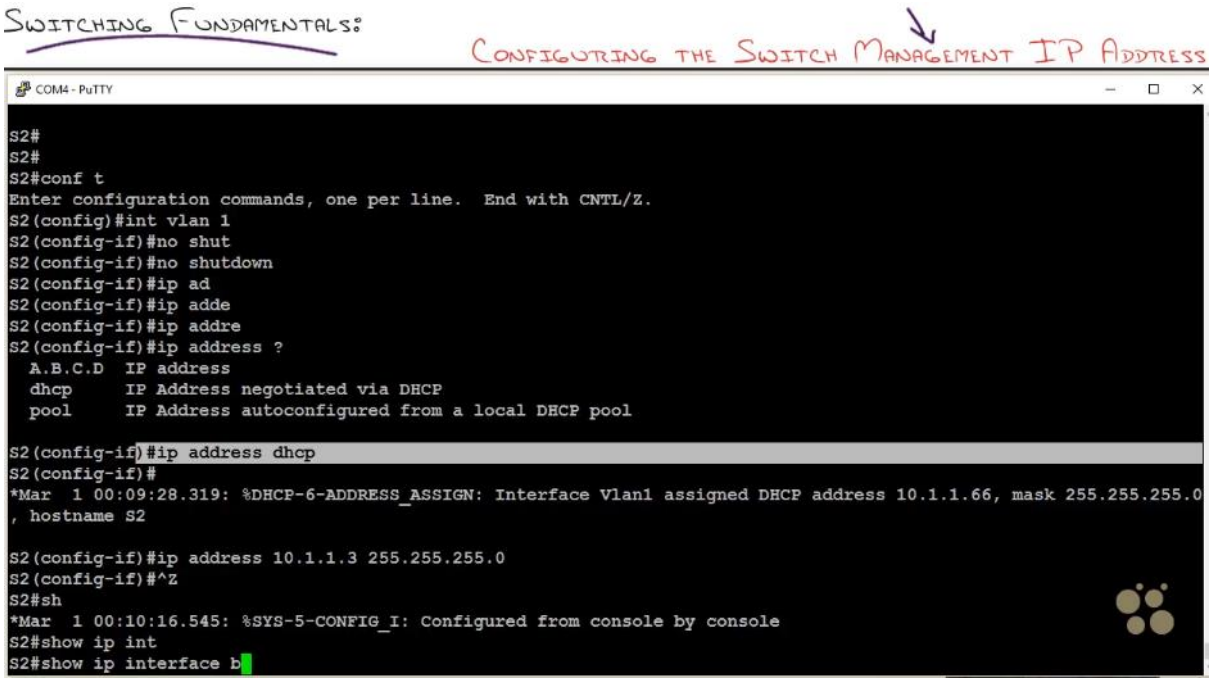

```
b4-a4-e3-28-9f-e7
b4a4.e328.9fe7
```

First six digits represent the vendor.

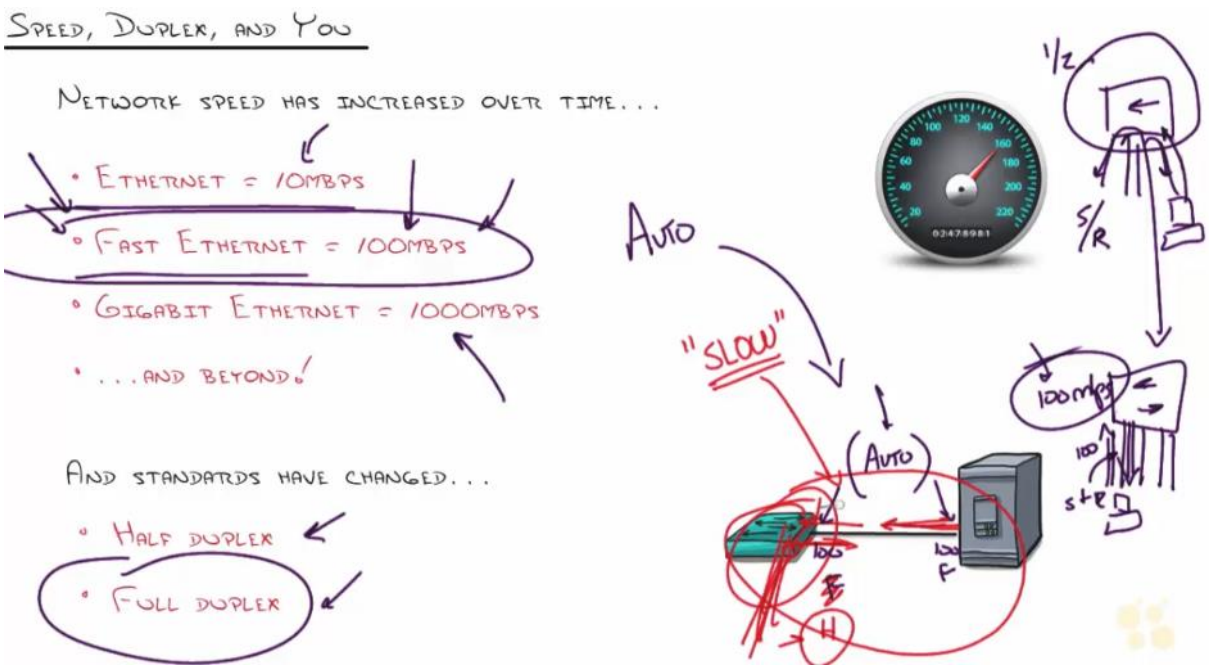




Use mac-address tables, cdp and cable maps to locate a device.



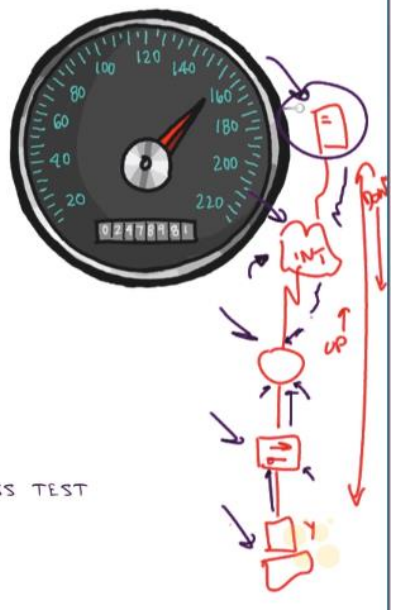
Rest is the same config for telnet/ssh

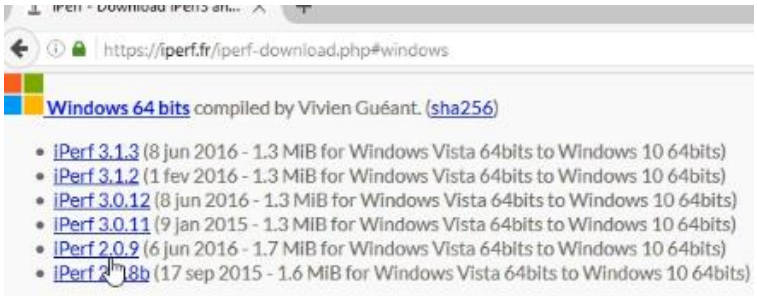


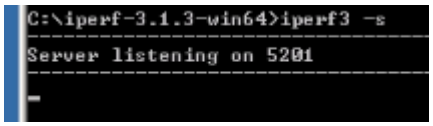
It is recommended to hard code the speed and duplex especially when you have 100Mbps devices otherwise the standard says that if the other side doesn't reply back then fall-back to half duplex.

How to Test if the Network is Slow

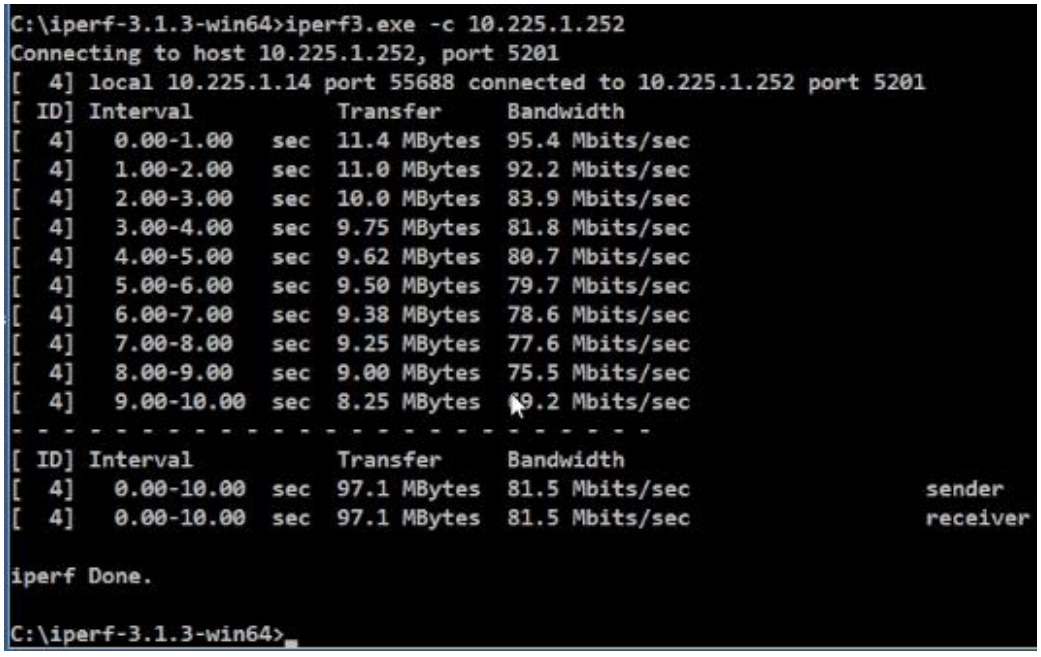
1. Speedtest sites - multiple locations
   - speedtest.net
   - speakeasy.net/speedtest
   - speedof.me
2. Watch speed and consistancy of speed
3. Internal testing - use iPerf (client / server)
4. Wired vs. wireless - never accept only a wireless test
5. Know your baseline usage and bandwidth capacity

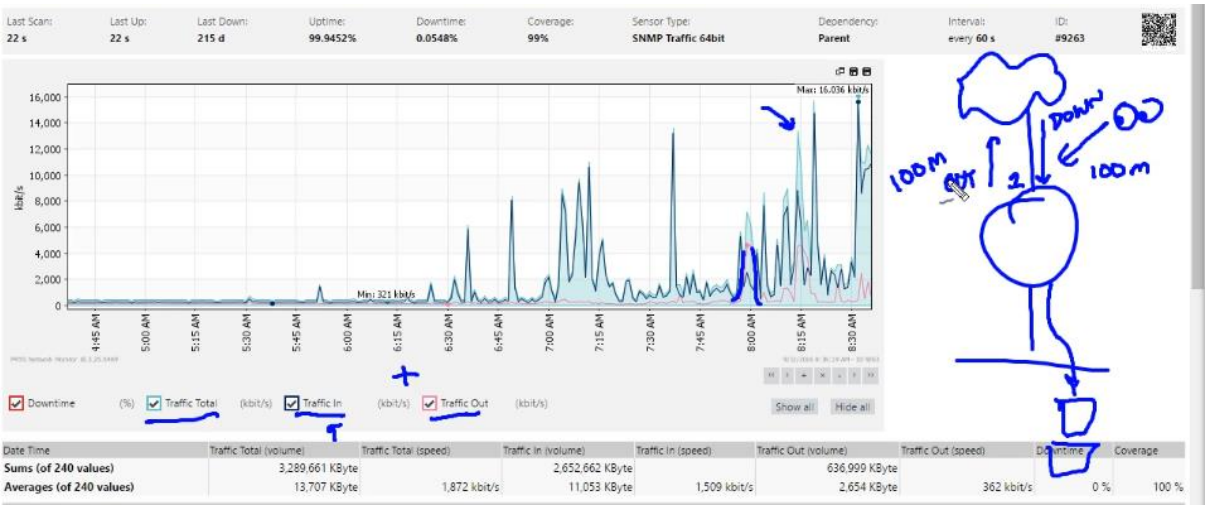






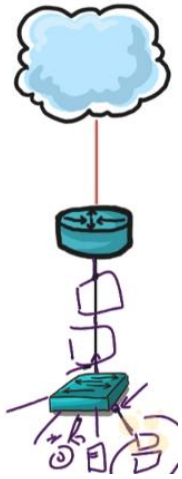
Server actively running and listening on 5201 port. Good test to check internal network speeds before even checking the external internet.





Key Interface Counters

- Trace the switchport path between source and destination
- Use the "show interface" command to review statistics
- Key "performance" counters:
  - duplex/speed
  - 5 minute output rates
  - input errors
  - CRC
  - collisions
  - late collisions
  - interface resets



```
monet>show interfaces serial 0
Serial 0 is up, line protocol is up  <-- Interface status line
  Hardware is MCI Serial
  Internet address is 131.108.156.98, subnet mask is 255.255.255.240
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 0:00:00, output 0:00:00, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 5762 drops, input queue 0/75, 301 drops
  Five minute input rate 9000 bits/sec, 16 packets/sec
  Five minute output rate 9000 bits/sec, 17 packets/sec
     5780806 packets input, 785841604 bytes, 0 no buffer
     Received 757 broadcasts, 0 runts, 0 giants
     146124 input errors, 87243 CRC, 58857 frame, 0 overrun, 0 ignored, 3 abort
     5298821 packets output, 765669598 bytes, 0 underruns
     0 output errors, 0 collisions, 2941 interface resets, 0 restarts
     2 carrier transitions
```

Output drops

Input drops

CRC errors

Abort errors

Input errors

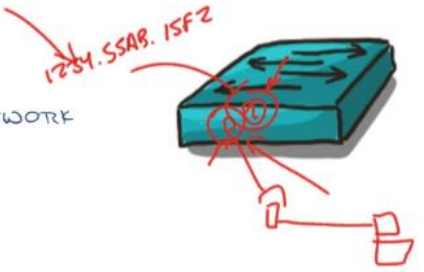
Framing errors

Interface resets

Carrier transitions

```
FastEthernet0/1 is up, line protocol is up
  Hardware is AmdFE, address is 0006.d7b2.c1e1 (bia 0006.d7b2.c1e1)
  Internet address is 10.1.3.2/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 1000 bits/sec, 3 packets/sec
  5 minute output rate 3000 bits/sec, 2 packets/sec
     980 packets input, 59426 bytes
     Received 6 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     1868 packets output, 174674 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The Purpose of Port Security



- We care about things plugging in to the network
- Two types of security:
  1. (everyone) Limit the number of MAC addresses on each port
  2. (security centric) Limit what MAC addresses can access the port

Static
Sticky

- Three response types: protect, restrict, and shutdown

1. Connect two devices to your Switch 1 (S1), port 1 and 2.
2. Assign the devices IP addresses on the same network and begin a continual ping between the devices.
3. View the MAC address table - verify the devices are appearing as expected.
4. Enable port security on the ports to only allow a single MAC address.
5. On one of the devices, use a tool like Ostinato and generate traffic from multiple source addresses (alternate: another hub/switch); verify port security works.
6. Re-enable the port; ensure pings continue to work.
7. On both ports, enable the sticky MAC address feature.
8. Move the device on port 1 to port 2 and vice versa. Verify port security is functioning as expected.
9. Re-enable the ports and remove port security.

```
S1#show port-security interface fastEthernet 0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : f8b1.56fe.9da9:1
Security Violation Count   : 1

S1#
```

```
S1#show interfaces fa0/1
FastEthernet0/1 is down, line protocol is down (err-disabled)
  Hardware is Fast Ethernet, address is 000c.854b.ee81 (bia 000c.854b.ee81)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed, media type is 10/100BaseTX
```

```
!
interface FastEthernet0/1
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky f8b1.56fe.9da9
!
interface FastEthernet0/2
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0011.43fe.5425
!
```

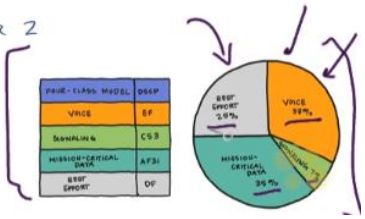
Understanding the Problem

- There is virtually no security at layer 2

- There is no segmentation at layer 2

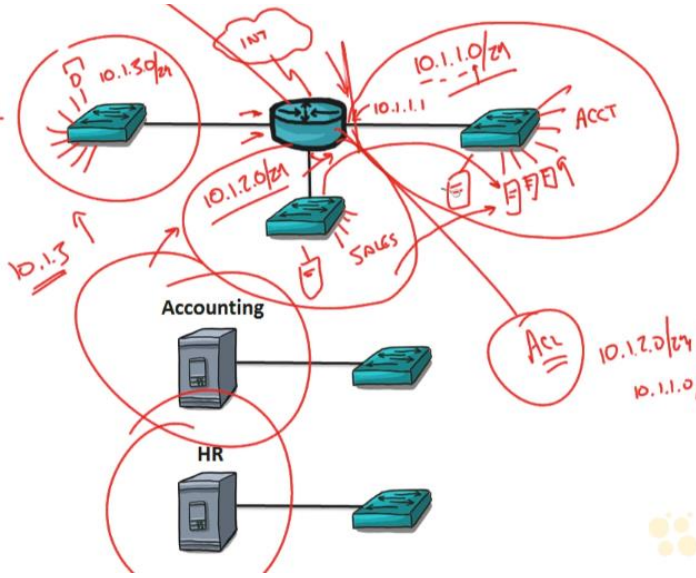
- There is no real way to differentiate devices at layer 2

Q65



Old School Solutions

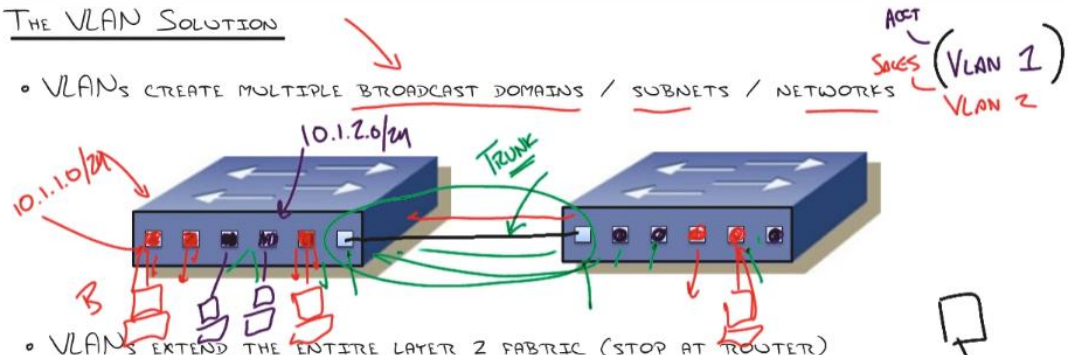
1. Routers everywhere



2. Servers on every network

The VLAN Solution

- VLANs create multiple broadcast domains / subnets / networks

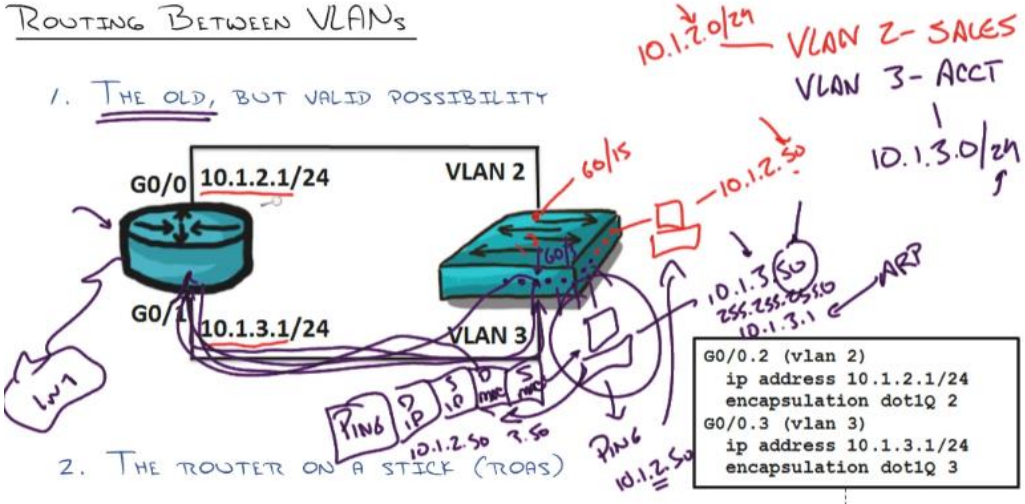


- VLANs extend the entire layer 2 fabric (stop at router)

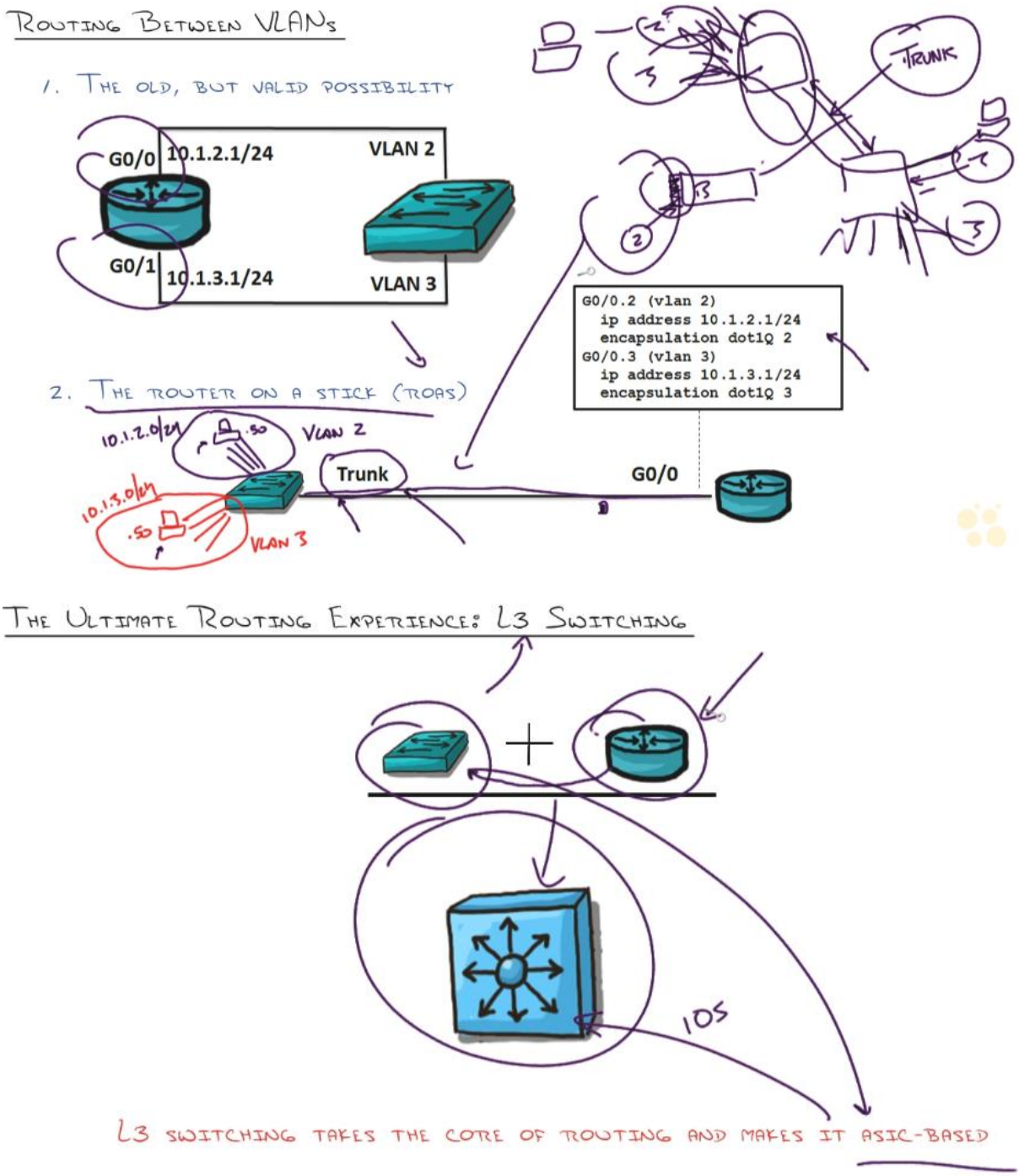
- VLANs segment and isolate traffic

Routing Between VLANs

1. The old, but valid possibility

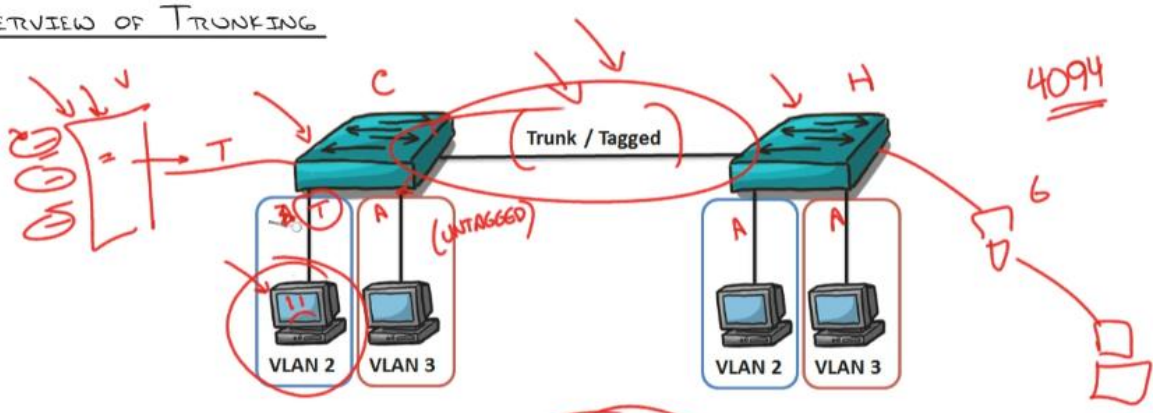


2. The router on a stick (ROAS)



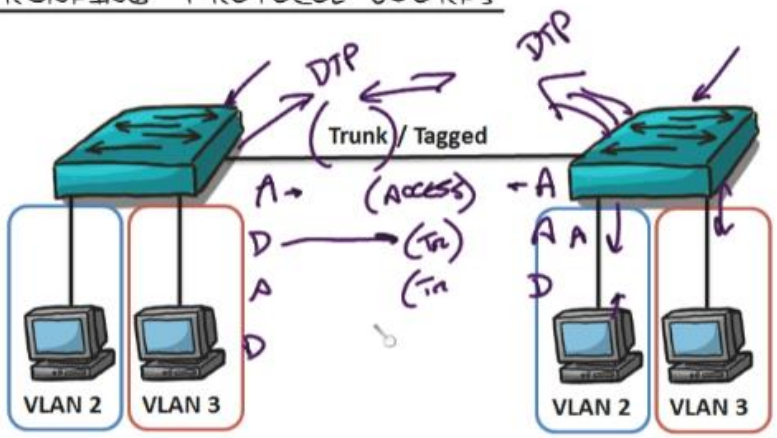
ASIC based devices are limited with features which is why the routers still exist.

Overview of Trunking



- Tagging done between switches with 802.1Q protocol
- Tag removed on exiting [access port]
- Trunk ports configured on each interface

How the Dynamic Trunking Protocol Works



- Enabling trunk ports:
  - manual configuration (trunk / nonegotiate)
  - dynamic configuration (dynamic auto / dynamic desirable)
- Dynamic configuration can be creepy; use nonegotiate

	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Limited Connectivity
Access	Access	Access	Limited Connectivity	Access

The Strange and Mysterious Native VLAN

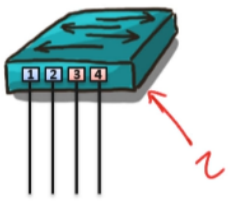
- Trunk ports should receive traffic with tags

- ...But what if it doesn't? NATIVE VLAN
- ...But how?
  - switch originated traffic
  - pass-through devices
  - virtualized servers

Native vlan should be configured the same on both sides

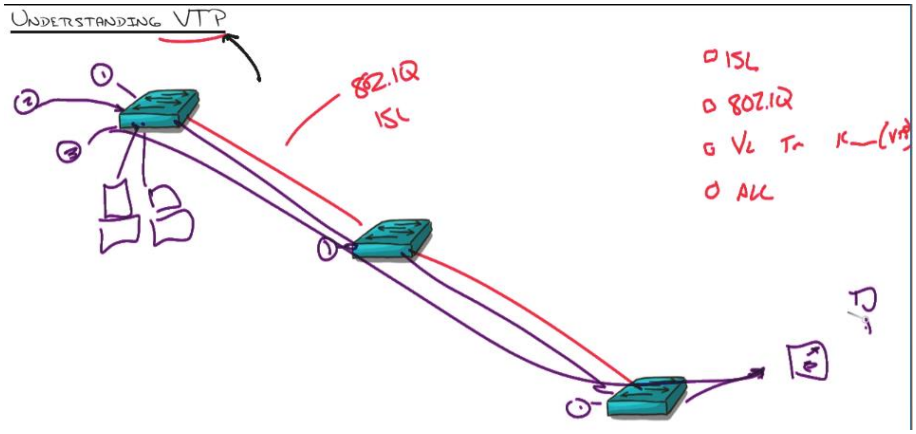
Switching VLANs:

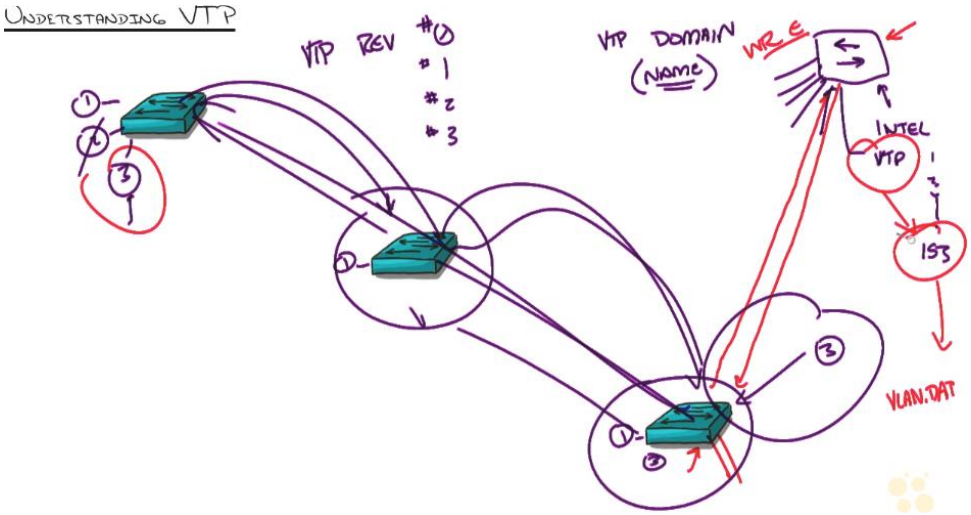
Configuring and Testing VLANs



VLAN 2 = 10.1.2.0/24
VLAN 3 = 10.1.3.0/24

1. Create and name VLAN 2 (accounting) and VLAN 3 (sales)
2. Assign designated ports to appropriate VLANs
3. Turn off STP on all ports so they come on right away (spanning-tree portfast)
4. Plug two computers into ports 1 and 2 - assign them IP addresses from the subnet
5. Start a continual ping between the two computers (command prompt -> ping <ip> -t)
6. Unplug one computer and move it to port 3. What happens to pings?
7. Unplug the other computer and move it to port 4. What happens now?
8. Unplug one computer and move it back to port 1 or 2. Then, with the pings still going...switch the port to VLAN 3.

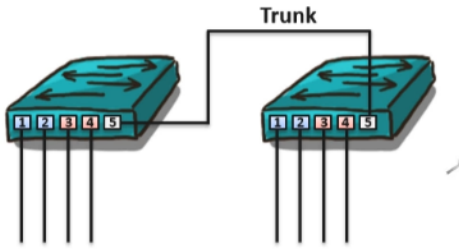





```
Switch#show vtp status
VTP Version                     : running VTP1 (VTP2 capable)
Configuration Revision          : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
Switch#
```

Switching VLANs:

Configuring Trunking and VTP



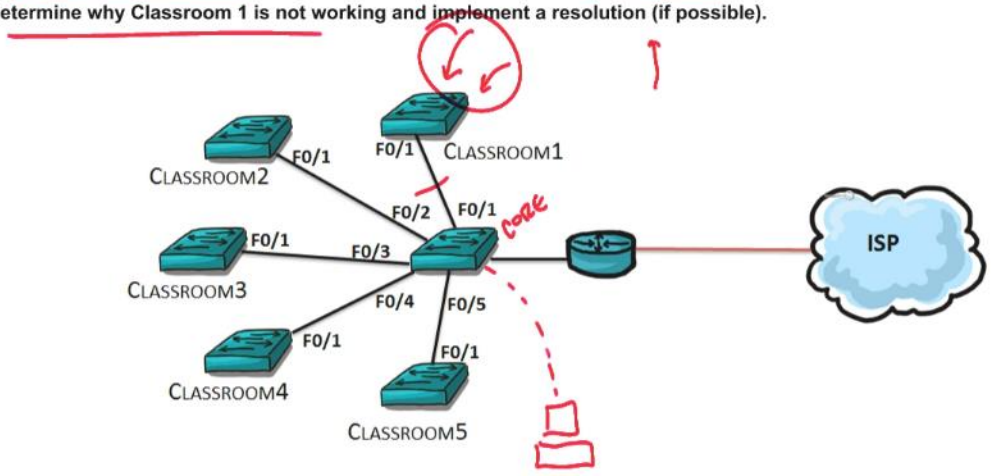
VLAN 2 = 10.1.2.0/24
VLAN 3 = 10.1.3.0/24

1. Use two switches - completely erase the configuration on both of them.
2. Access switch1 (left...S1) and configure VTP Mode: Server, Domain: CBTNuggets (set a VTP password if you like).
3. Add the VLANs shown and assign ports on S1.
4. Configure Port 5 as a trunk link, manual configuration (don't disable DTP), 802.1q encapsulation. Check the config on S2.
5. Plug the two switches together on port 5. What happens? What is the config of S2, port 5? What VLANs does it have?
6. Configure VTP on S2 and see if VLANs replicate.
7. Verify the Trunk link - what VLANs are permitted to cross?
8. Plug some PCs in to the switches and move them around similarly to the last lab and verify all is working as expected.

Scenario 1: Student Testing is not Working

Hudson Elementary is running state-mandated student tests in multiple classrooms this week. Each of the five testing classrooms has a 24-port switch feeding the 20 desktop computers in the room and is connected back to the core MDF switch. Classroom 1 is reporting connectivity failures, while the other four classrooms seem to be working fine.

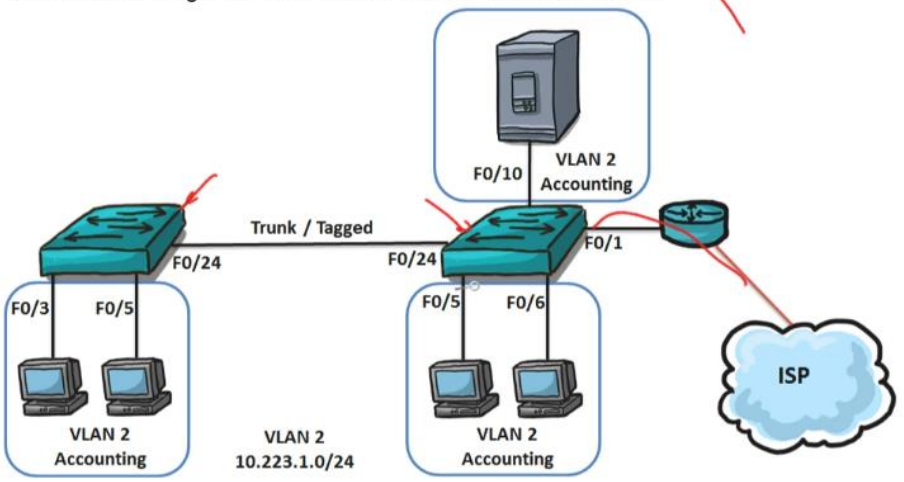
Determine why Classroom 1 is not working and implement a resolution (if possible).



Scenario 2: Accounting Spreadsheets are "Unbeatable"

The accounting group has recently grown and added a few more team members to the group. To accommodate the expansion, the junior on-site technician created a new IP subnet for the accounting VLAN that increased the capacity of the network as well as adding an additional expansion switch. After these changes, the accounting team has been very vocal in communicating that everything seems to be running slow, especially commonly accessed Excel spreadsheets.

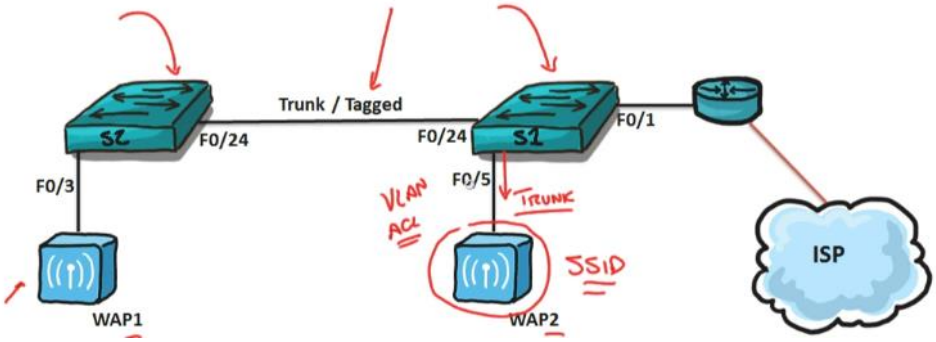
Use your ninja-level troubleshooting skills to determine if this is a network related issue.



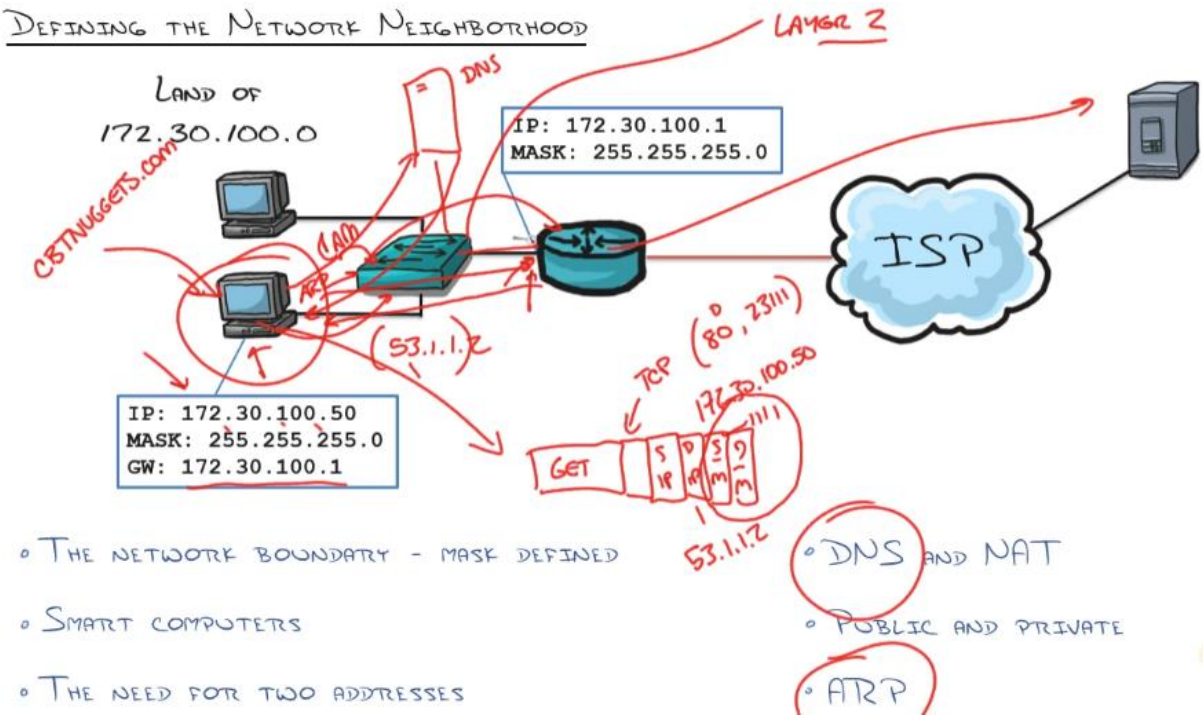
Scenario 3: Public WIFI is Unavailable

The Public wireless at the Mellow Flaky Flakes cereal company has recently gone down. Guests are complaining that they are now incurring higher cell phone bills and are threatening to sue Flaky Flakes for undue stress and increased data charges.

Save Flaky Flakes from the lawsuit. Determine why Public WIFI is unavailable and fix it.

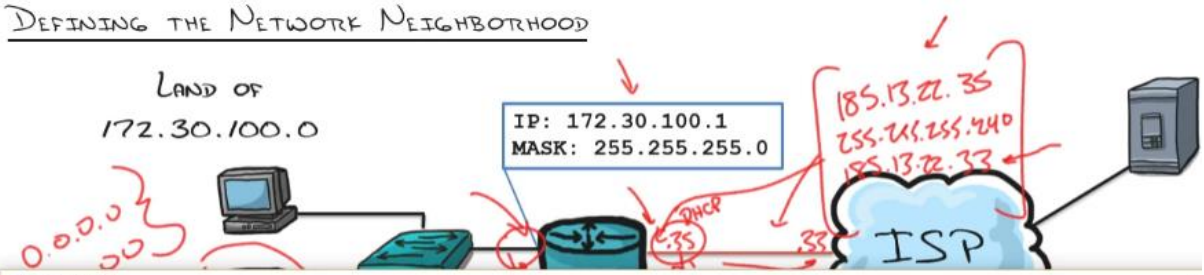


Defining the Network Neighborhood



- The network boundary - mask defined
- Smart computers
- The need for two addresses
- DNS and NAT
- Public and private
- ARP

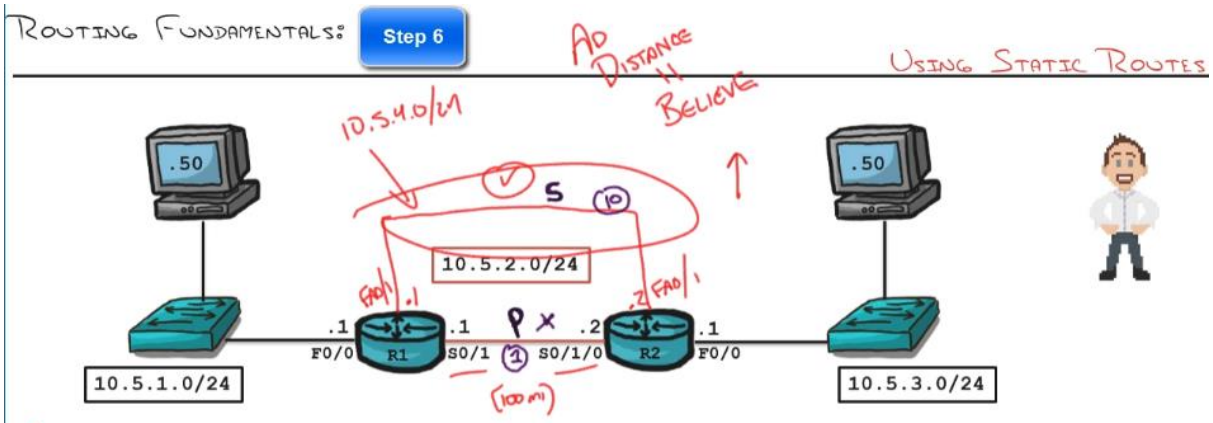
Defining the Network Neighborhood



COM4 - PuTTY

```
% Invalid input detected at '^' marker.

Router#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 185.13.22.33 to network 0.0.0.0

     185.13.0.0/28 is subnetted, 1 subnets
C       185.13.22.32 is directly connected, FastEthernet0/1
     172.30.0.0/24 is subnetted, 1 subnets
C       172.30.100.0 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 185.13.22.33
```

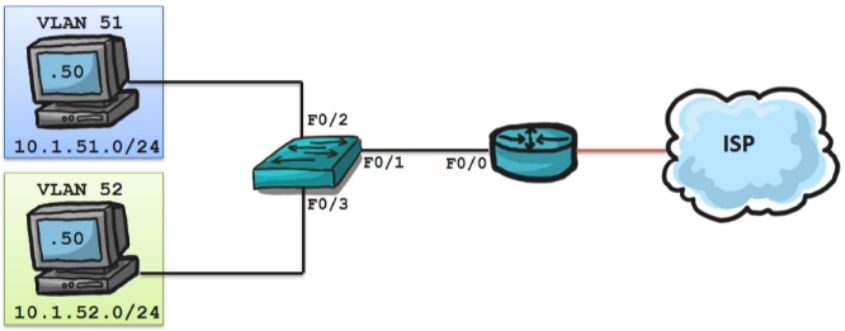
```
R1#show controllers s0/1
Interface Serial0/1
Hardware is PowerQUICC MPC860
DCE V.35, no clock
idb at 0x84C6BEF8, driver data structure at 0x84C73614
SCC Registers:
General [GSMR]=0x2:0x00000000, Protocol-specific [PSMR]=0x8
Events [SCCE]=0x0000, Mask [SCCM]=0x0000, Status [SCCS]=0x03
Transmit on Demand [TODR]=0x0, Data Sync [DSR]=0x7E7E
Interrupt Registers:
Config [CICR]=0x00367F80, Pending [CIPR]=0x04008800
Mask   [CIMR]=0x00200000, In-srv  [CISR]=0x00000000
Command register [CR]=0x540
Port A [PADIR]=0x0400, [PAPAR]=0xFFFF
       [PAODR]=0x0000, [PADAT]=0xE8FE
Port B [PBDIR]=0x03A0F, [PBPAR]=0x0100E
       [PBODR]=0x00000, [PBDAT]=0x3FF7D
Port C [PCDIR]=0x00C, [PCPAR]=0x000
       [PCSO]=0x0A0,  [PCDAT]=0xFF2, [PCINT]=0x00F
Receive Ring
        rmd(680127F0): status 9000 length 60C address 75DCFA4
        rmd(680127F8): status 9000 length 60C address 75D5424
        rmd(68012800): status 9000 length 60C address 75D3A24
        rmd(68012808): status 9000 length 60C address 75D40A4
 --More--
```



1. Configure R1 and R2 with the IP addresses shown in the diagram
2. Verify the routing table on both routers - what networks do they know how to reach?
3. Ping directly between the two routers. Are you successful?
4. Ping various devices on the LANs attached to the routers. What works, what doesn't?
5. Implement static routing on the routers shown so that full connectivity is achieved.
6. BONUS: Add a secondary connection between the routers. Set up floating static routing so that redundancy is achieved.

Routing Fundamentals:

Routing Between VLANs

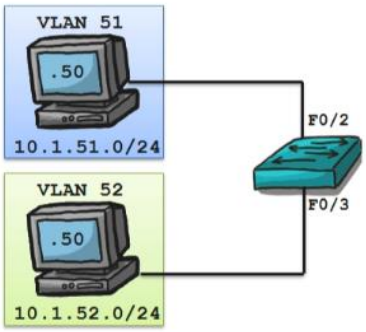


1. Connect devices as shown.
2. Create VLAN 51 (ENG) and VLAN 52 (MGMT) on the switch and assign ports as shown.
3. Configure a router-on-a-stick design. The router should be the first IP address on each subnet.

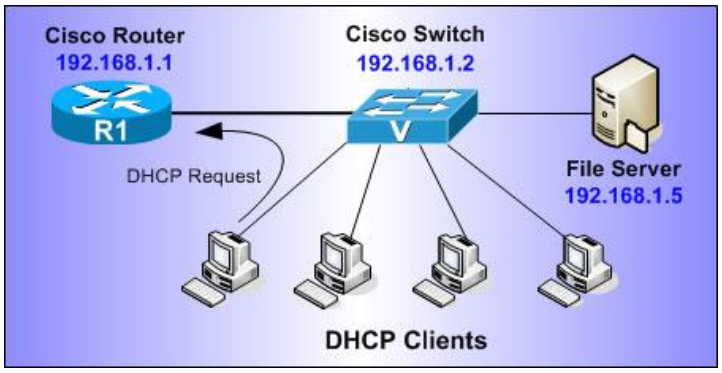
   NOTE: The trunk port between the router and switch should ONLY allow VLAN 51 and 52 (VLAN 1 native)
4. The computer in VLAN 51 should be able to ping the computer in VLAN 52 (and vice versa).

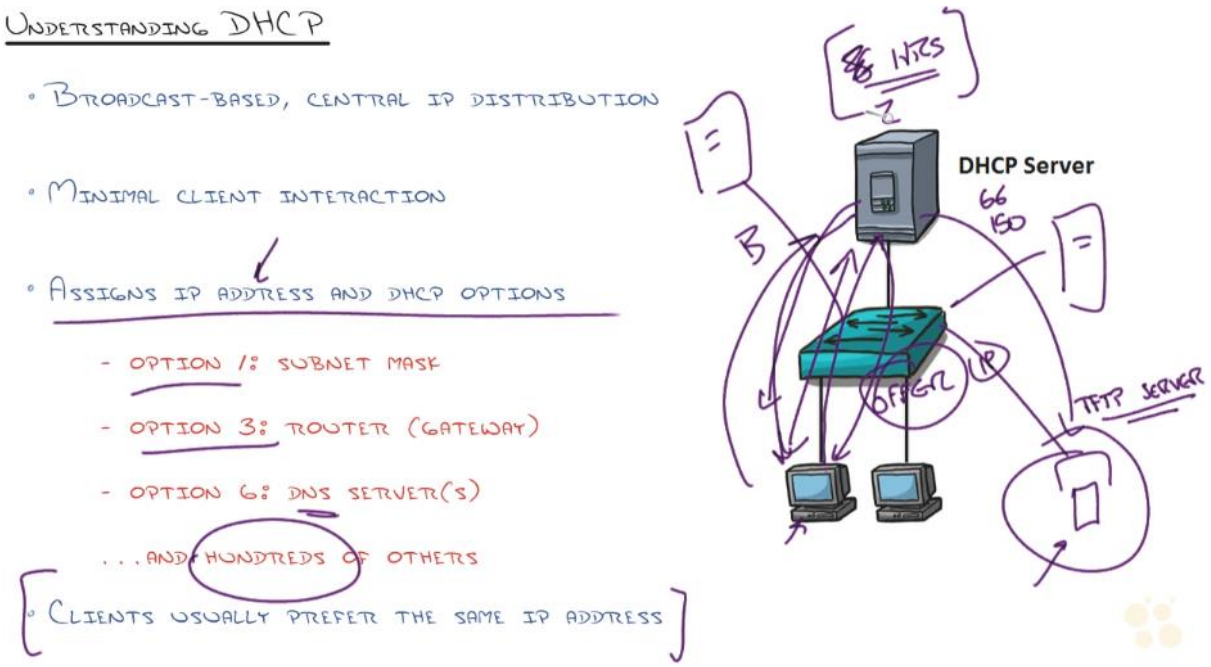
Routing Fundamentals:

Layer 3 Switching

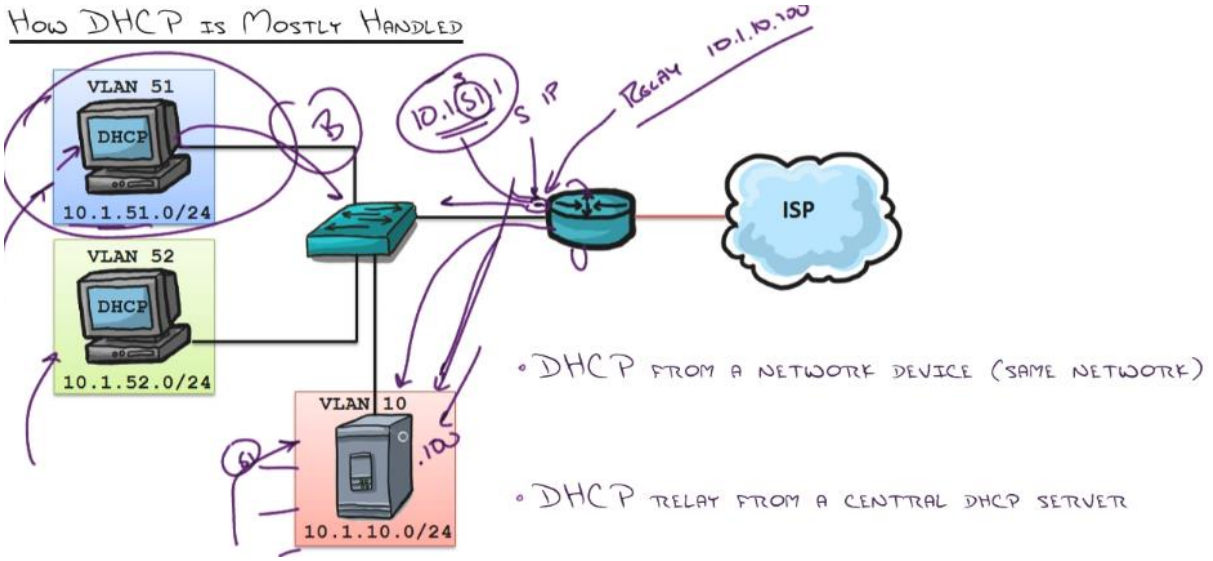


1. Connect devices as shown.
2. Create VLAN 51 (ENG) and VLAN 52 (MGMT) on the switch and assign ports as shown.
3. Configure a Layer 3 Switch design. The SVIs should have the first IP address on each subnet.
4. The computer in VLAN 51 should be able to ping the computer in VLAN 52 (and vice versa).






```
NEO#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.1.51.20          01f8.b156.fe9d.a9       Mar 02 2002 12:34 AM    Automatic
NEO#
```



Difference between Option 150 and Option 66:

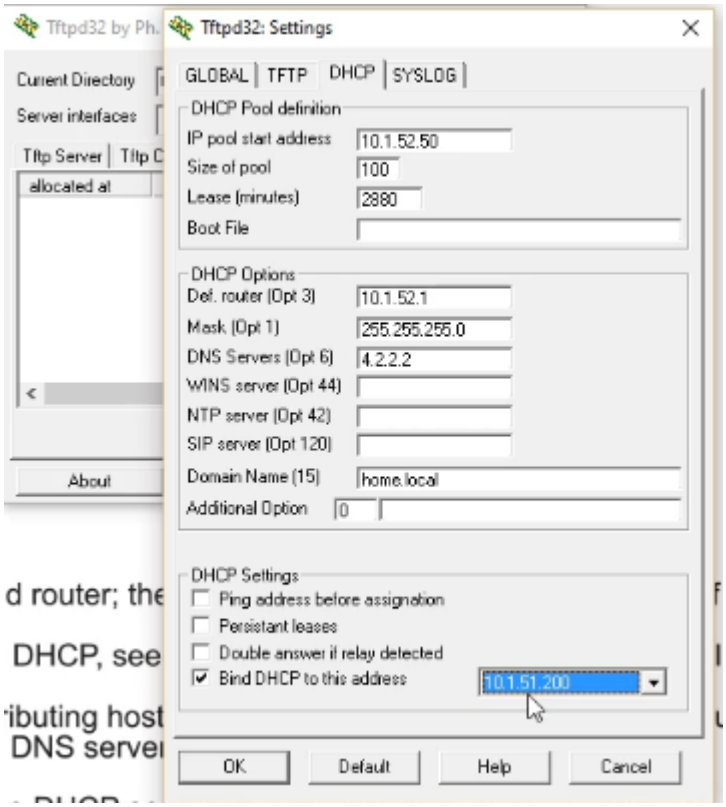
•DHCP option 150 supports a list of TFTP servers (Multiple Server IPs)

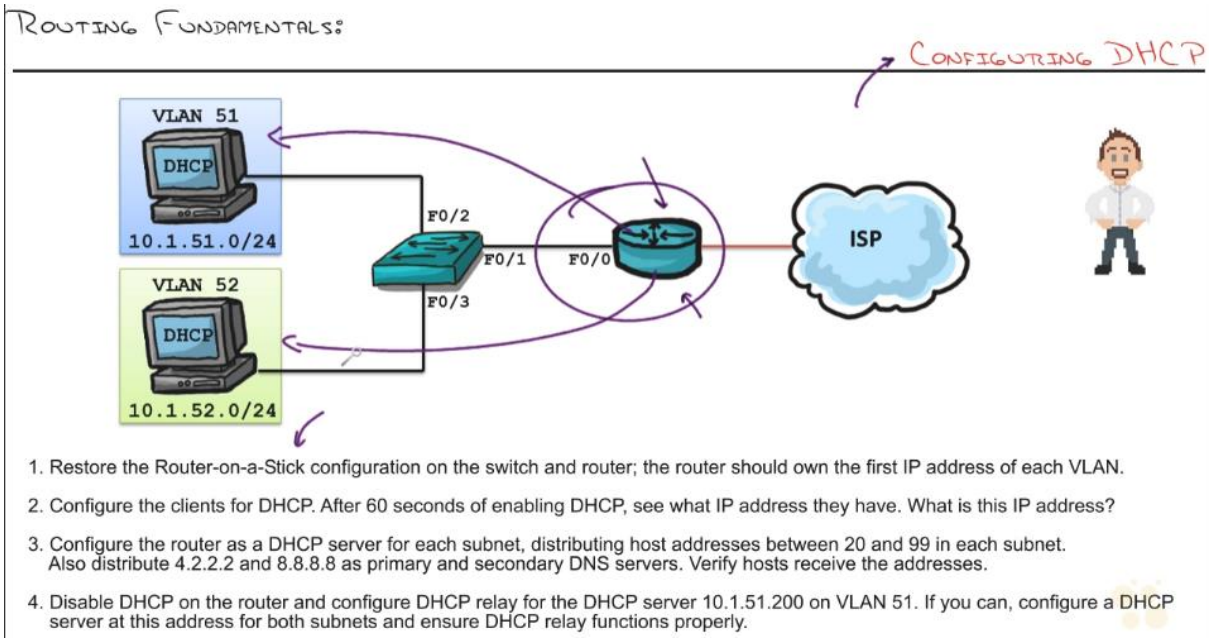
•DHCP option 66 only supports the IP address or the hostname of a single TFTP server.

DHCP Option 150 is Cisco proprietary. Option 66 is an IEEE standard.

For Cisco phones IP addresses can be assigned manually or by using DHCP. Devices also require access to a TFTP server that contains device configuration name files (.cnf file format), which enables the device to communicate with Cisco Call Manager.

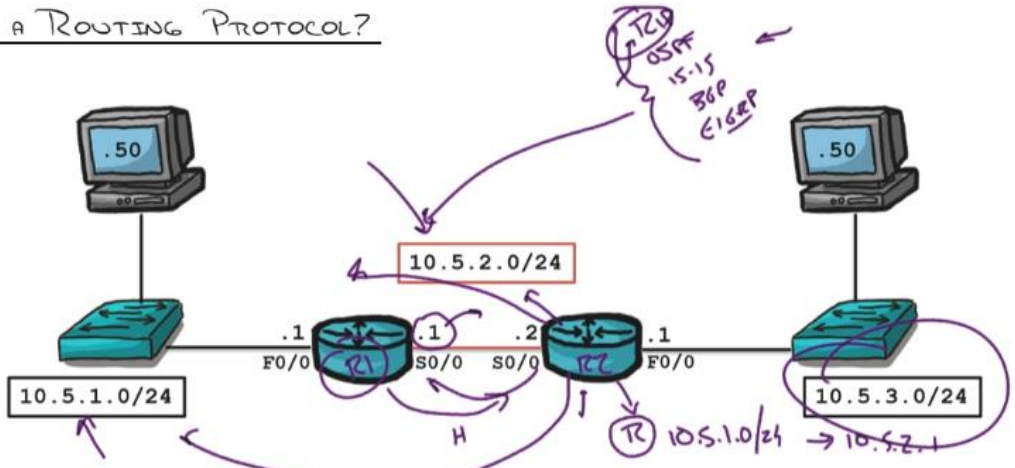
Cisco IP Phones download their configuration from a TFTP server. When a Cisco IP Phone starts, if it does not have both the IP address and TFTP server IP address pre-configured, it sends a request with option 150 to the DHCP server to obtain this information.
DHCP Option 150 is Cisco proprietary. The IEEE standard that matches with this requirement is Option 66. Like option 150, option 66 is used to specify the Name of the TFTP server.





1. Restore the Router-on-a-Stick configuration on the switch and router; the router should own the first IP address of each VLAN.
2. Configure the clients for DHCP. After 60 seconds of enabling DHCP, see what IP address they have. What is this IP address?
3. Configure the router as a DHCP server for each subnet, distributing host addresses between 20 and 99 in each subnet. Also distribute 4.2.2.2 and 8.8.8.8 as primary and secondary DNS servers. Verify hosts receive the addresses.
4. Disable DHCP on the router and configure DHCP relay for the DHCP server 10.1.51.200 on VLAN 51. If you can, configure a DHCP server at this address for both subnets and ensure DHCP relay functions properly.

What is a Routing Protocol?

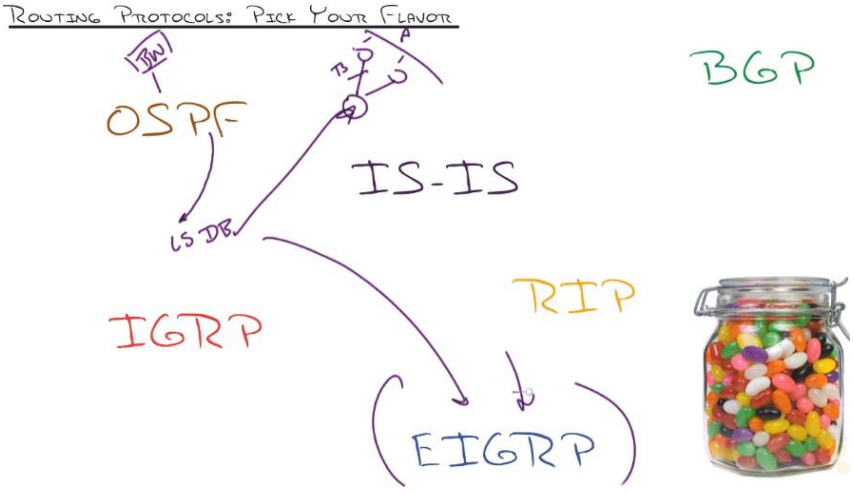


- A router knows about ONE type of network by default
- Routing protocol: educate a router without your involvement
- Routing protocols are dynamic, forming neighbors, detecting failures

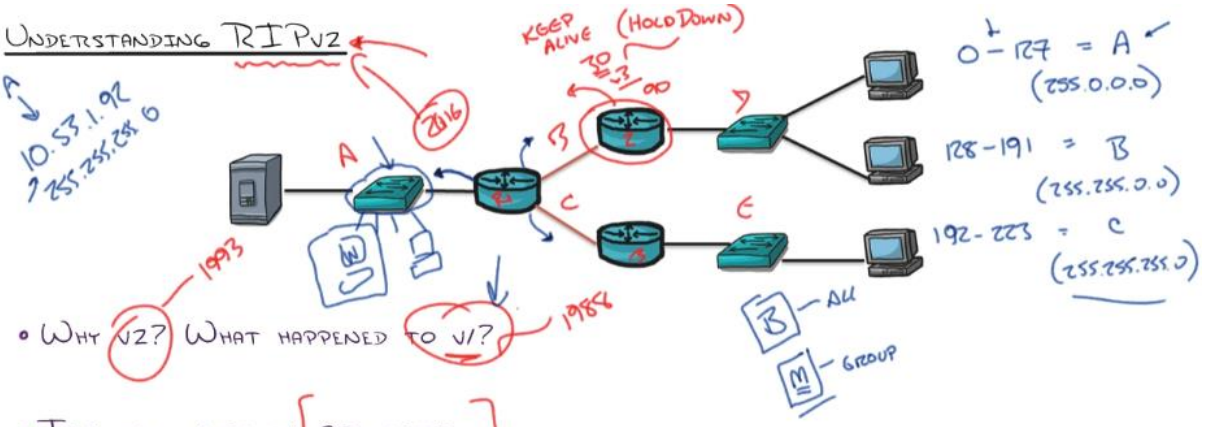
Distance Vector vs. Link State

- DV
  - only knows what the neighbor tells it
  - memory / processor efficient
  - loop prevention mechanisms needed
- LS
  - maintains a map of the network system
  - resource consuming
  - maintains loop free by nature

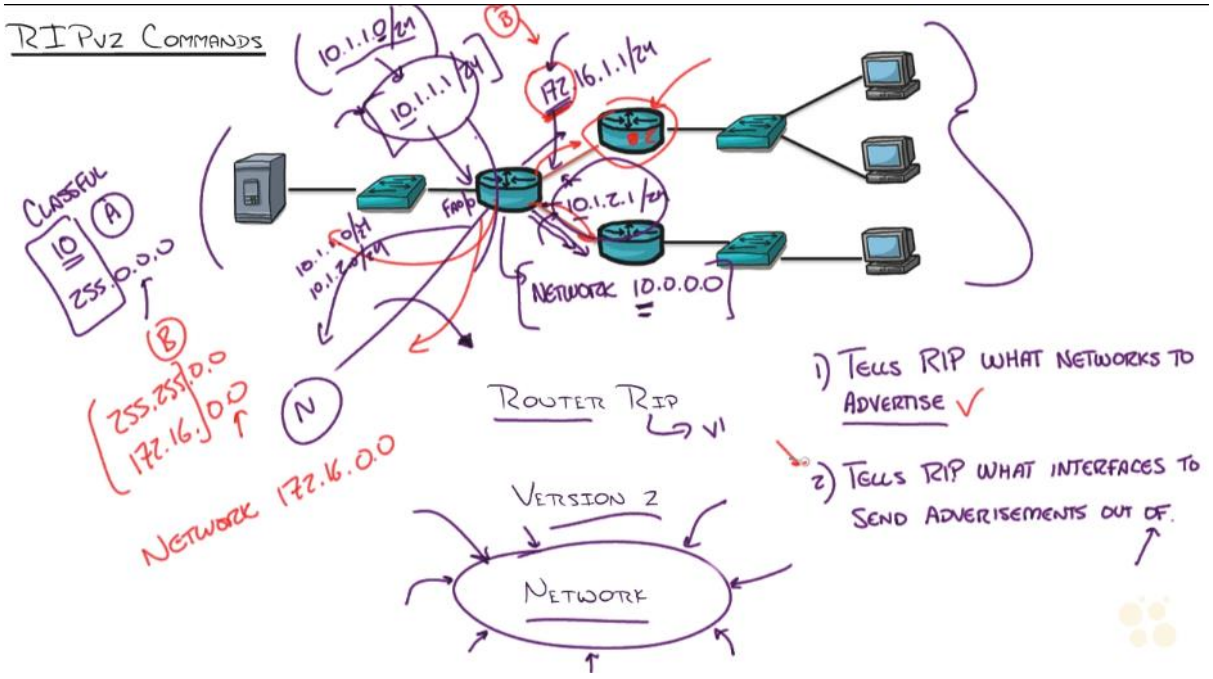
Routing Protocols: Pick Your Flavor



Understanding RIPv2



- Why v2? What happened to v1?
- Tell your life in 30 seconds
- It's time to be a little classless...
- Where does this protocol fit? ~~Any~~ NO where. Some

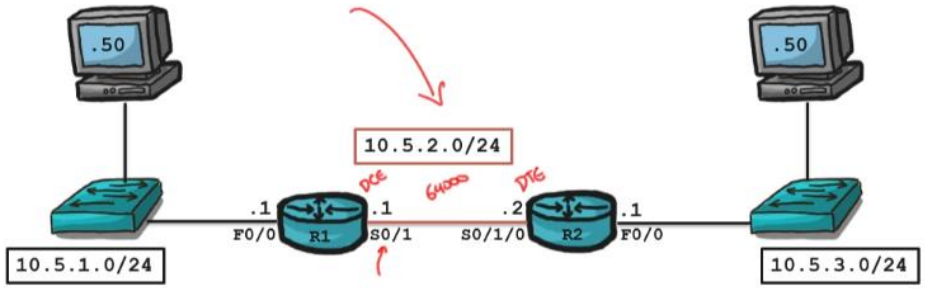



```
R2#show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 0 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/0       2     2
    Serial0/1/0           2     2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.5.2.1              120      00:00:04
  Distance: (default is 120)
```

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.5.3.0 is directly connected, FastEthernet0/0
C       10.5.2.0 is directly connected, Serial0/1/0
R       10.5.1.0 [120/1] via 10.5.2.1, 00:00:15, Serial0/1/0
R2#
```

Routing Protocols:

Configuring RIPv2



1. Assign IP addresses and connect devices as show. Ensure there are no static routes on R1 and R2.

2. Test pings between the devices on the two networks. What results do you see?

3. Configure RIPv2 to populate the routing table on both routers. Verify this is successful.

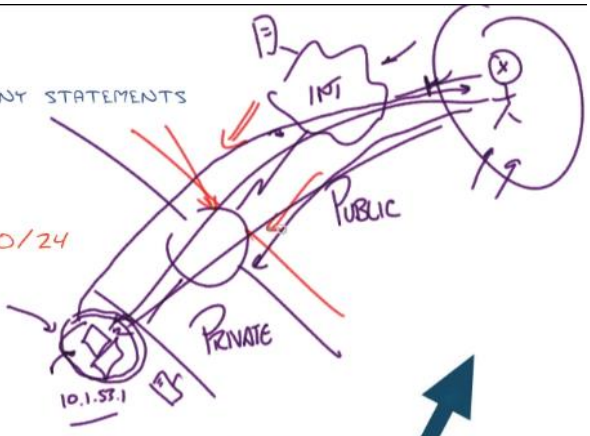
4. Ping between the two devices again. What results do you see?

5. BONUS: Add a secondary interface/network between R1 and R2. Add it to the RIP routing process; how does it affect the routing table?

Access-Lists... They're Everywhere!!!

What they are: A list of permit and deny statements

- permit 192.168.2.50
- deny 192.168.1.0/24
- permit tcp port 80 for 200.1.1.1
- permit all tcp traffic for 210.0.1.0/24



What they can be used for:

- access control
- NAT
- quality of service
- demand dial routing
- policy routing
- route filtering
- making french toast

Using ACLs for Security

List is read from top to bottom; stops at first match

Invisible implicit deny at the bottom

ACL is applied to an interface inbound or outbound

```
R1#show access-list
Standard IP access list 1
    10 permit 192.168.1.0, wildcard bits 0.0.0.255
    20 deny   172.16.0.0, wildcard bits 0.0.255.255
R1#
```

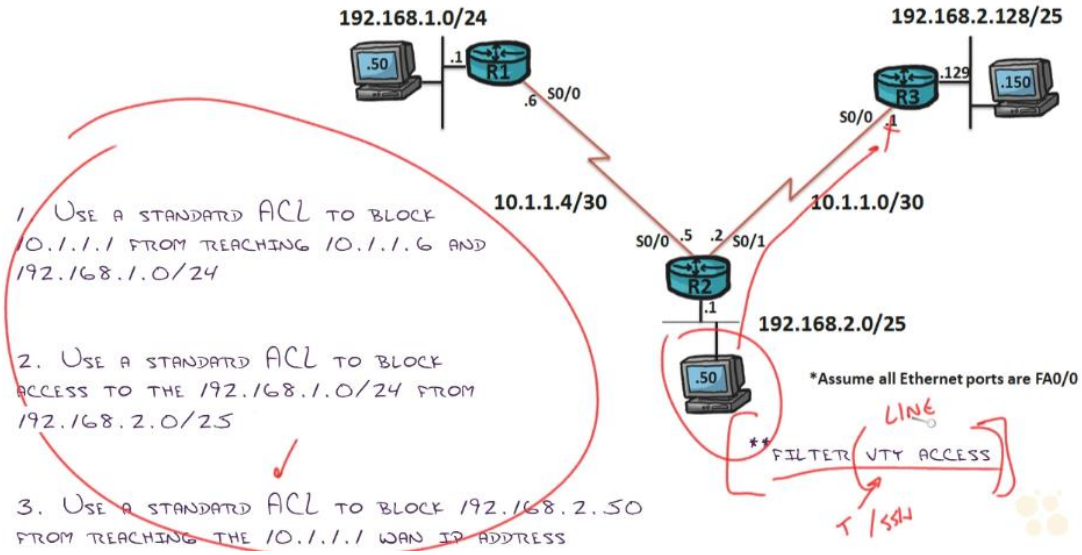
Standard and Extended Access Lists

- Standard
  - Matches based on source address
  - Lower processor utilization
  - Affect depends on application

  } ICND1

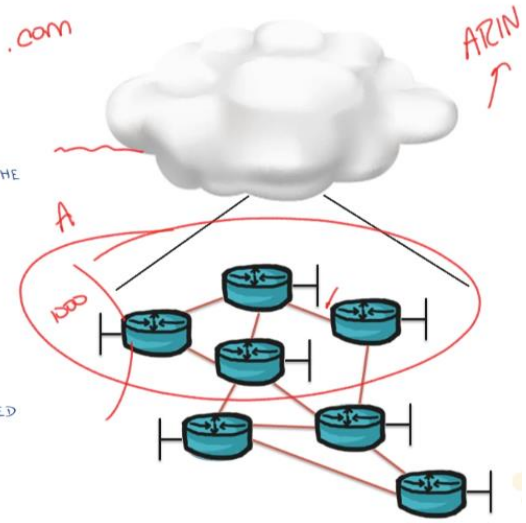
- Extended
  - Matches based on source/destination address, protocol, source/destination port number
  - Higher processor utilization
  - Syntax takes some time to learn
- Reflexive (established)
  - Allows return traffic for internal requests

ACL Configuration Scenario

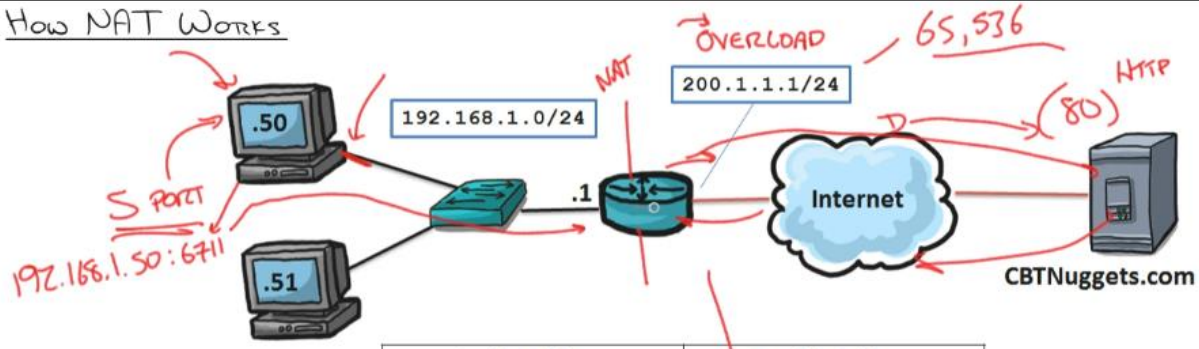


What is NAT?

- Internet = Big Network
- Too many devices compared to the number of total IP addresses
- IP management necessary
- Public and private divide needed
- Welcome to NAT!



How NAT Works

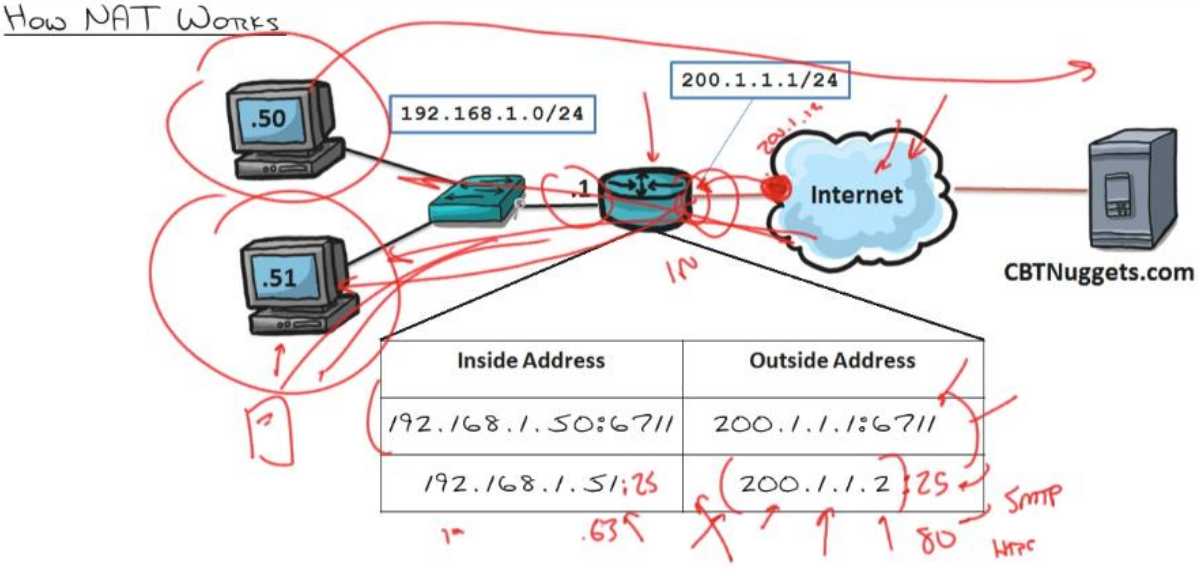


Inside Address	Outside Address
192.168.1.50:6711	200.1.1.1:6711
192.168.1.51:1536	200.1.1.1:1536

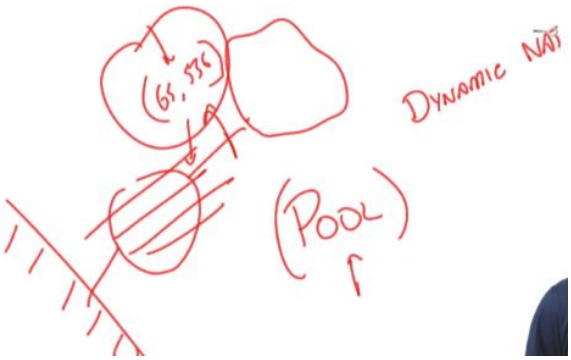
SOCKET

This form of NAT is commonly called PAT

How NAT Works

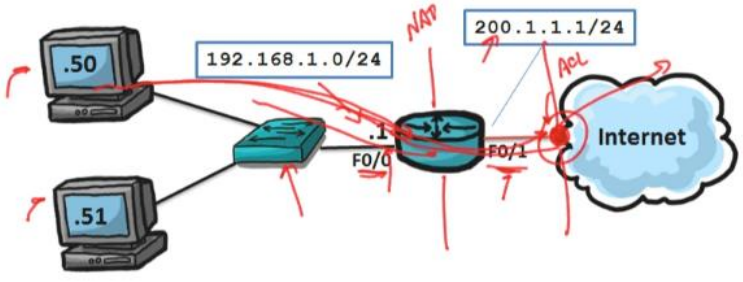


STATIC NAT CAN TRANSLATE FULL IP ADDRESSES OR SPECIFIC PORTS



Network Address Translation:

CONFIGURING NAT OVERLOA



RFC1918

1) ACL TO ID ADDRESSES TO BE TRANSLATED

2) ID INSIDE AND OUTSIDE INTERFACES

3) DEFINE NAT OPERATIONS

1. Assign IP addresses and connect devices as shown. The computers can be configured via DHCP or statically.

2. Configure a default route to the Internet - the ISP router is 200.1.1.32. Test connectivity to the Internet from the router using ping.

3. Test connectivity from the computers to the Internet. Are they working? Why or why not?

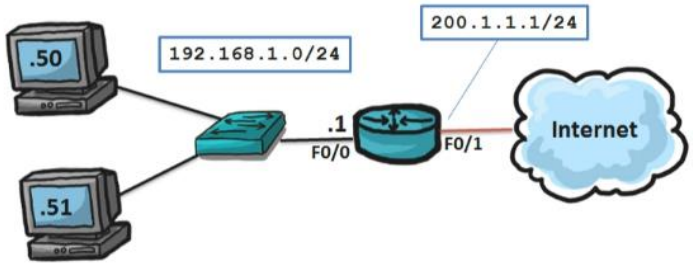
4. Configure NAT Overload to use the F0/1 interface IP address. Verify NAT is functioning using relevant show commands.

5. BONUS: Convert the NAT Configuration to an overloaded NAT Pool using the range 200.1.1.2 - 200.1.1.5.

```
Neo#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
udp 200.1.1.1:137      192.168.1.50:137   23.99.3.152:137    23.99.3.152:137
udp 200.1.1.1:137      192.168.1.50:137   64.4.23.171:137    64.4.23.171:137
udp 200.1.1.1:137      192.168.1.50:137   111.221.77.147:137 111.221.77.147:137
udp 200.1.1.1:137      192.168.1.50:137   157.55.56.171:137  157.55.56.171:137
udp 200.1.1.1:137      192.168.1.50:137   157.55.235.153:137 157.55.235.153:137
udp 200.1.1.1:35977    192.168.1.50:35977 65.55.223.29:443   65.55.223.29:443
udp 200.1.1.1:35977    192.168.1.50:35977 71.70.146.148:33786 71.70.146.148:33786
udp 200.1.1.1:35977    192.168.1.50:35977 73.4.148.119:58010 73.4.148.119:58010
udp 200.1.1.1:35977    192.168.1.50:35977 95.215.78.190:9241 95.215.78.190:9241
udp 200.1.1.1:35977    192.168.1.50:35977 154.68.21.234:2137 154.68.21.234:2137
udp 200.1.1.1:35977    192.168.1.50:35977 157.55.130.162:443 157.55.130.162:443
udp 200.1.1.1:35977    192.168.1.50:35977 181.95.222.63:51478 181.95.222.63:51478
udp 200.1.1.1:35977    192.168.1.50:35977 189.121.88.179:29912 189.121.88.179:29912
udp 200.1.1.1:49334    192.168.1.50:49334 8.8.8.8:53         8.8.8.8:53
udp 200.1.1.1:49344    192.168.1.50:49344 4.2.2.2:53         4.2.2.2:53
udp 200.1.1.1:49344    192.168.1.50:49344 8.8.8.8:53         8.8.8.8:53
udp 200.1.1.1:49616    192.168.1.50:49616 4.2.2.2:53         4.2.2.2:53
udp 200.1.1.1:49616    192.168.1.50:49616 8.8.8.8:53         8.8.8.8:53
udp 200.1.1.1:49630    192.168.1.50:49630 4.2.2.2:53         4.2.2.2:53
 --More--
```

Network Address Translation:

Configuring Static NAT



1. Assign IP addresses and connect devices as shown. The computers can be configured via DHCP or statically.
2. Configure static NAT to translate all traffic from 192.168.1.50 to the IP address 200.1.1.10. Verify this is working correctly.
3. Configure static NAT to translate all web traffic (HTTP / HTTPS) to the new web server at 192.168.1.51. This traffic will come in on the IP address 200.1.1.11. Only translate these specific services.
4. Verify all NAT configurations are implemented correctly.

Rationale for IPv6

- Yes, Virginia... there IS an IP address shortage!
- Current IP addresses poorly allocated
- New network devices on the rise
- NAT (our current solution) is now seen as a hinderence to innovation
- Potential future features: IPsec everywhere, mobility, simpler header

USA

IPv6 Addressing

- Address size moved from 32-bit (IPv4) to 128-bit (IPv6)
- Provides

  340,282,366,920,938,463,463,374,607,431,770,000,000

  . . . addresses

- To make addresses more manageable, divided into 8 groups of 4 hex characters

  2001:0050:0000:0000:0000:0AB4:1E2B:98AA

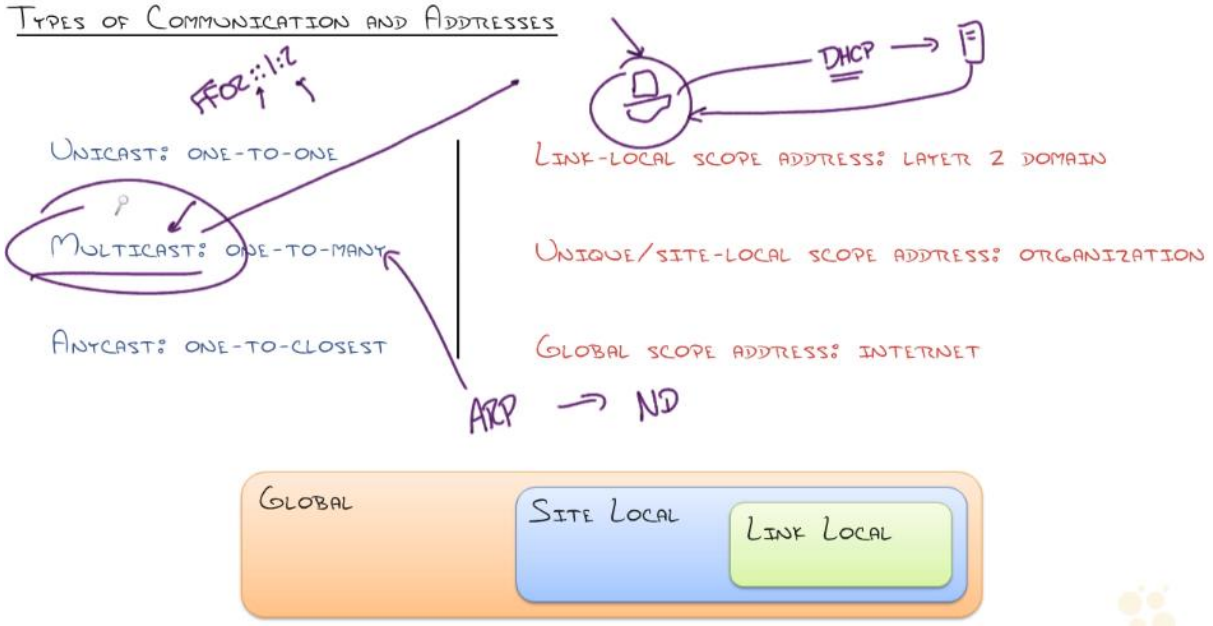
- Rule 1: eliminate groups of consecutive zeros

  2001:0050::0AB4:1E2B:98AA

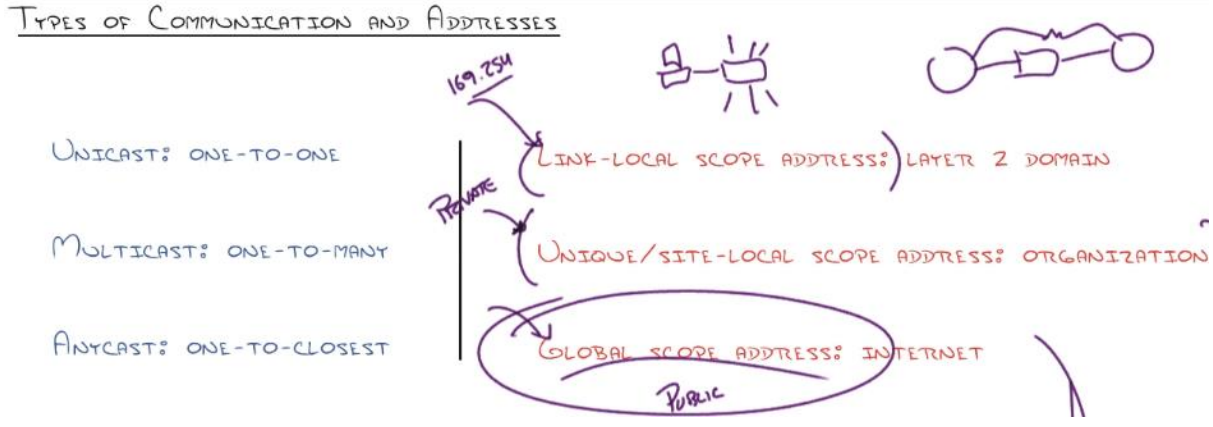
- Rule 2: drop leading zeros

  2001:50::AB4:1E2B:98AA

Types of Communication and Addresses



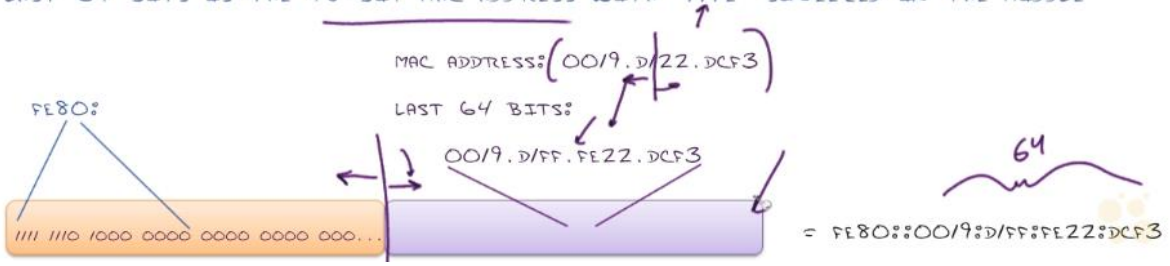
Types of Communication and Addresses

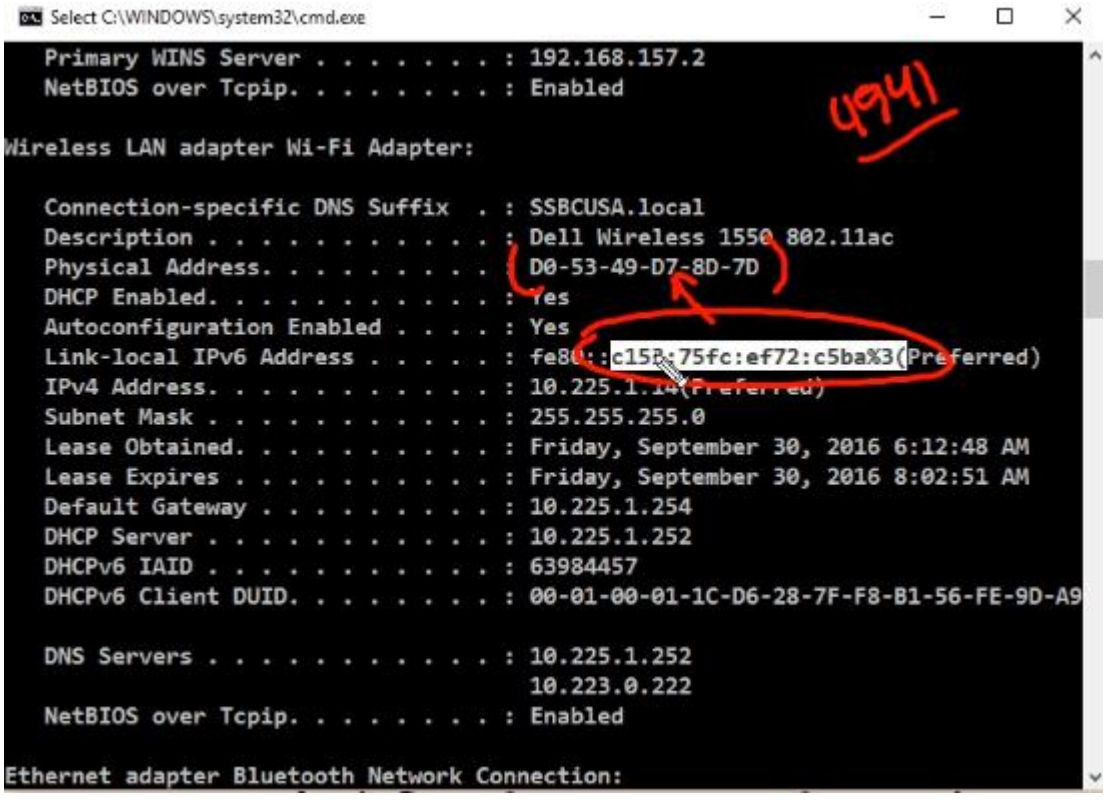


Link Local Addresses

- Assigned automatically as an IPv6 host comes online
- Similar to the 169.254.x.x addresses of IPv4
- Always begin with "FE80" (first 10 bits: 1111 1110 10) followed by 54 bits of zeros
- Last 64 bits is the 48-bit MAC address with "FFFE" squeezed in the middle

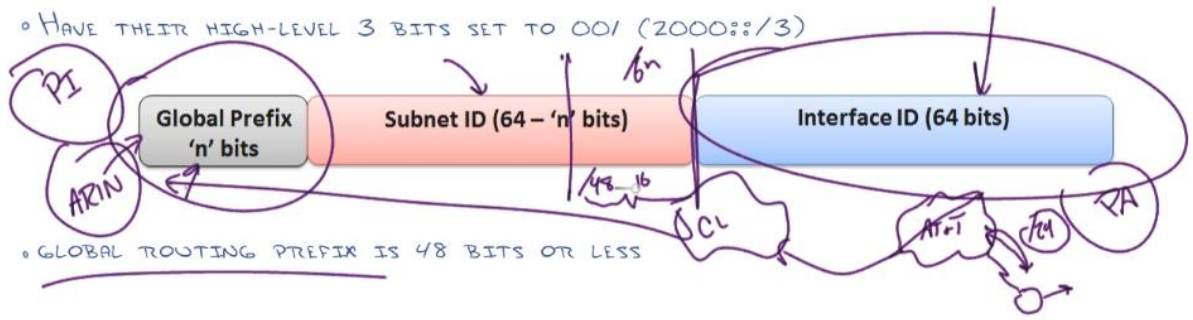
[FE80::] /64 / EUI-64





Global Addresses

- Have their high-level 3 bits set to 001 (2000::/3)

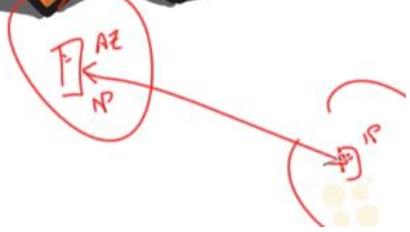


- Global routing prefix is 48 bits or less
- Subnet ID is comprised of bits are left over after global routing prefix
- The primary addresses comprising the IPv6 internet are from 2001::/16

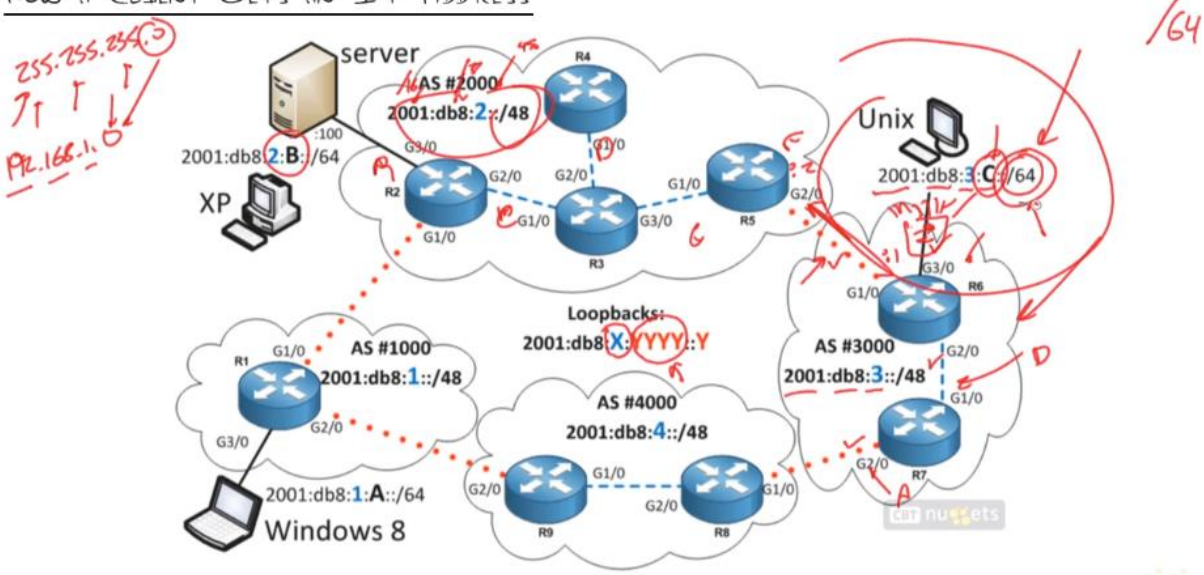
  - /32 subnets assigned to providers
  - /48 /56 /64 subnets assigned to customers

The Kinds of Addresses a Client Could Have

- Global — 2xxx::/3 — 001
- Link local — EUI-64 — FFFE — (H) — FE80::/(64)
- Unique local
- Multicast — 224.0.0.5 — OSPF
- Anycast — one to closest



How a Client Gets an IP Address



Sneak Peek of IPv6 at CBTNuggets

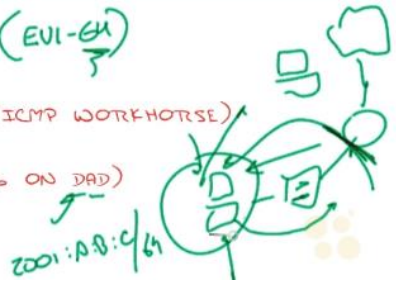
List of Methods

- Static Addressing: What it always has been
- DHCPv6: What it always has been (multicast-powered)

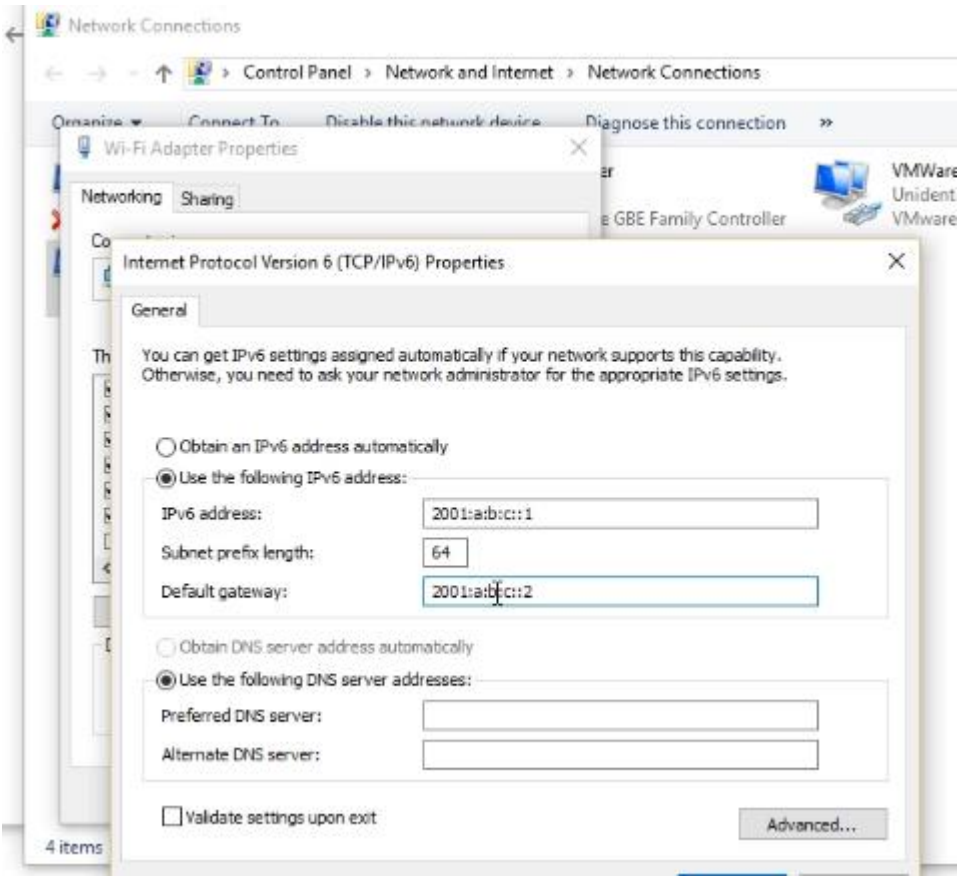
IPv6 multicast address	Description
FF02::1	The **all-nodes** address used to reach all nodes on the same link.
FF02::2	The **all-routers** address used to reach all routers on the same link.
FF02::5	The **all-OSPF routers** address used to reach all OSPF routers on the same link.
FF02::6	The OSPF **designated** routers address used to reach all OSPF designated routers on the same link.
FF02::9	The **all-RIP** routers address used to reach all RIP routers on the same link.
FF02::A	The **all-EIGRP** routers address used to reach all EIGRP routers on the same link.
FF02::1:FFxx:xxxx	The **solicited-node** address used in the address resolution process to resolve the IPv6 address of a link-local node to its link-layer address. The last 24 bits (xx:xxxx) of the solicited-node address are the last 24 bits (6 hex characters) of an IPv6 unicast address.

SLACC: Stateless Address Auto Configuration

- Dynamically asks router to discover network (ICMP workhorse)
- Generates host portion of the address (relying on DAD)

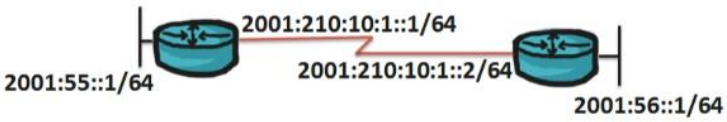


IPv6 multicast address	Description
FF02::1	The **all-nodes** address used to reach all nodes on the same link.
FF02::2	The **all-routers** address used to reach all routers on the same link.
FF02::5	The **all-OSPF routers** address used to reach all OSPF routers on the same link.
FF02::6	The OSPF **designated** routers address used to reach all OSPF designated routers on the same link.
FF02::9	The **all-RIP** routers address used to reach all RIP routers on the same link.
FF02::A	The **all-EIGRP** routers address used to reach all EIGRP routers on the same link.
FF02::1:FF*xx:xxxx*	The **solicited-node** address used in the address resolution process to resolve the IPv6 address of a link-local node to its link-layer address. The last 24 bits (*XX:XXXX*) of the solicited-node address are the last 24 bits (6 hex characters) of an IPv6 unicast address.



IPv6:

Interface Configuration and Static Routes



1. Assign IP addresses and connect devices as shown.

2. Test by pinging between the two devices.

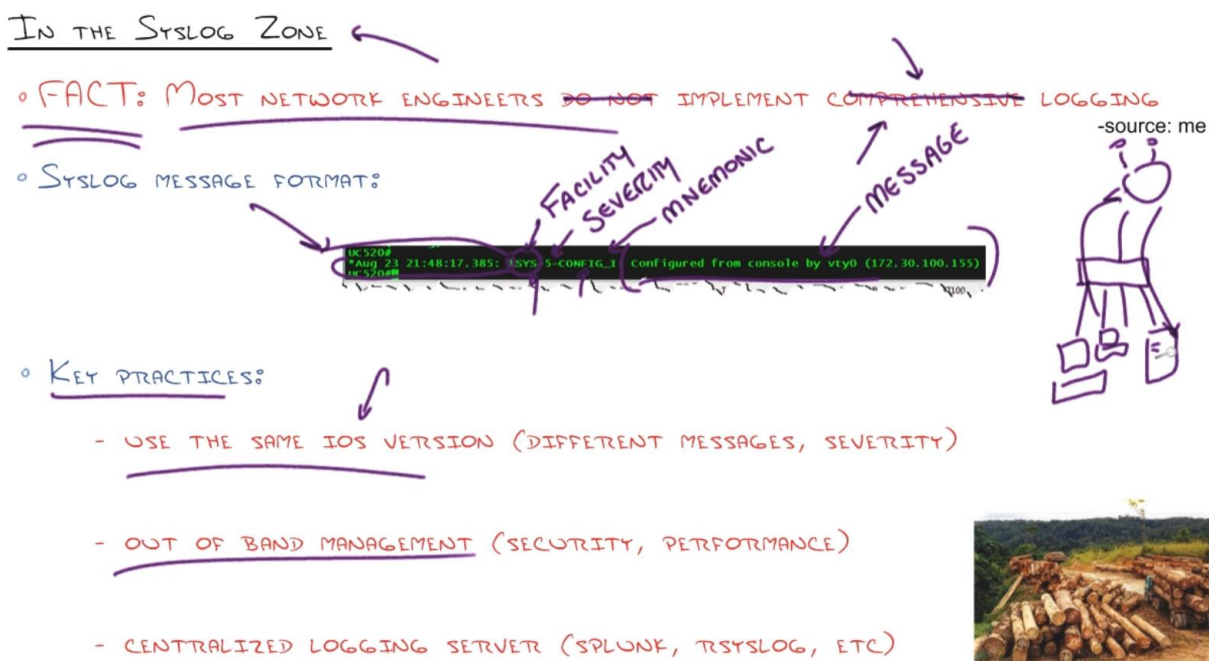
3. Configure a static route allowing the two LAN connections to reach each other.

4. Test using ping sourced from each LAN interface of the routers.

```
R1#show ipv6 interface brief
FastEthernet0/0            [administratively down/down]
FastEthernet0/1            [administratively down/down]
Serial1/0                  [up/up]
    FE80::C001:28FF:FE9D:0
    2001:210:10:1::1
Serial1/1                  [administratively down/down]
Serial1/2                  [administratively down/down]
Serial1/3                  [administratively down/down]
R1#
```

```
R1#show ipv6 interface
Serial1/0 is up, line protocol is down
  IPv6 is enabled, link-local address is FE80::C001:28FF:FE9D:0 [TEN]
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:210:10:1::1, subnet is 2001:210:10:1::/64 [TEN]
  Joined group address(es):
    FF02::1
    FF02::2
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
R1#
```

```
R1#show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
C   2001:55::/64 [0/0]
     via ::, Loopback0
L   2001:55::1/128 [0/0]
     via ::, Loopback0
S   2001:56::/64 [1/0]
     via 2001:210:10:1::2
C   2001:210:10:1::/64 [0/0]
     via ::, Serial1/0
L   2001:210:10:1::1/128 [0/0]
     via ::, Serial1/0
L   FF00::/8 [0/0]
     via ::, Null0
```




```
S1#
00:32:19: %SYS-5-CONFIG_I: Configured from console by console
S1#
```

SYS: Facility

Severity: 5

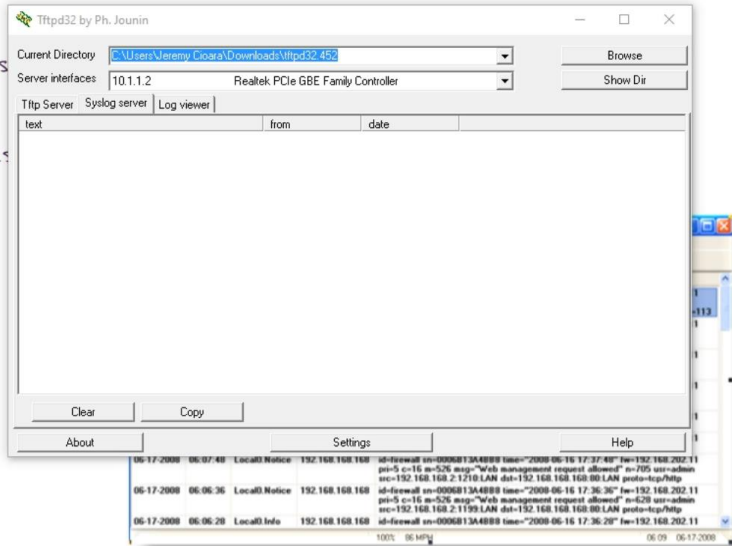
CONFIG_I: Mnemonic

Message:

```
S1(config)#logging console ?
  <0-7>          Logging severity level
  alerts         Immediate action needed           (severity=1)
  critical       Critical conditions               (severity=2)
  debugging      Debugging messages                (severity=7)
  emergencies    System is unusable                (severity=0)
  errors         Error conditions                  (severity=3)
  guaranteed     Guarantee console messages
  informational  Informational messages            (severity=6)
  notifications  Normal but significant conditions (severity=5)
  warnings       Warning conditions                (severity=4)
  xml            Enable logging in XML
  <cr>
```

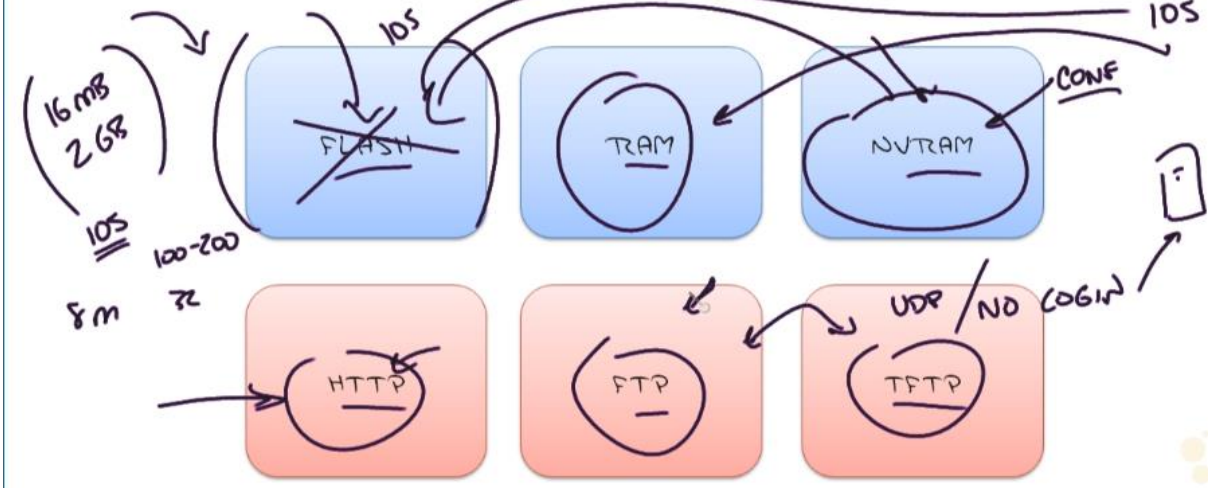
Capturing Syslog

- Syslog captures key status mess
- Each device can store syslog mes
- Syslog uses UDP port 514

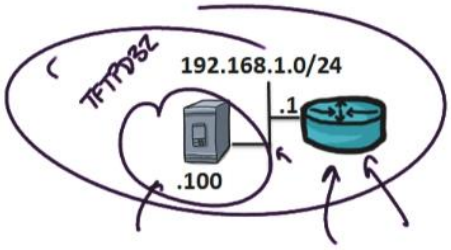


Understanding File Storage Locations

- Cisco devices have multiple storage locations where they can access files
- Copy command allows you to perform file management between locations



IOS / CONFIG BACKUP AND RESTORE

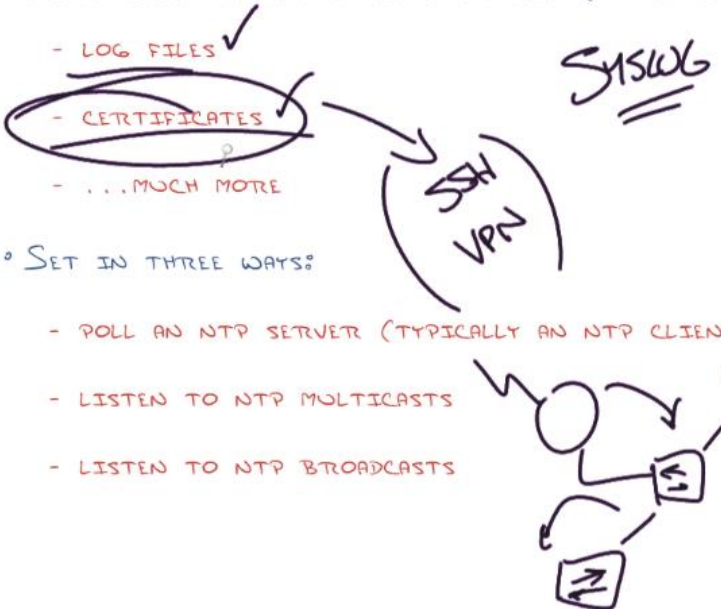


1. Access the device shown in the picture.
2. Backup the IOS to the TFTP server shown.
3. Backup the running configuration to the TFTP server shown.
4. Restore both the IOS and the running configuration to the router in the proper way.

```
R1#show flash

System flash directory:
File  Length    Name/status
  1   29715256  c2600-adventerprisek9-mz.124-13b.bin
  2   1642      BackupCFG [deleted]
  3   1642      cbtrouter-confg
  4   899       2016different-cfg
[29719700 bytes used, 3310440 available, 33030140 total]
32768K bytes of processor board System flash (Read/Write)
```

THE NETWORK TIME PROTOCOL (NTP)

- KEEPS CISCO CLOCKS IN SYNCHRONIZED (NO CLOCK DRIFT); VALUABLE FOR:
  - LOG FILES
  - CERTIFICATES
  - ...MUCH MORE
- SET IN THREE WAYS:
  - POLL AN NTP SERVER (TYPICALLY AN NTP CLIENT)
  - LISTEN TO NTP MULTICASTS
  - LISTEN TO NTP BROADCASTS

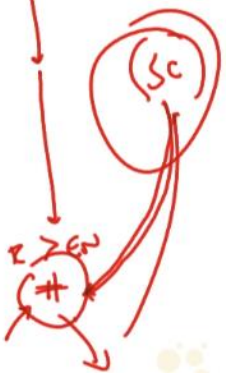


PENGUIN STRATUM

```
R1#show ntp associations

     address         ref clock      st  when  poll reach  delay  offset    disp
 ~64.6.144.6       0.0.0.0          16     -    64     0    0.0    0.00  16000.
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
R1#show ntp associations

     address         ref clock      st  when  poll reach  delay  offset    disp
*~64.6.144.6       128.252.19.1      2     0    64   377   122.1  -24.03    21.9
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

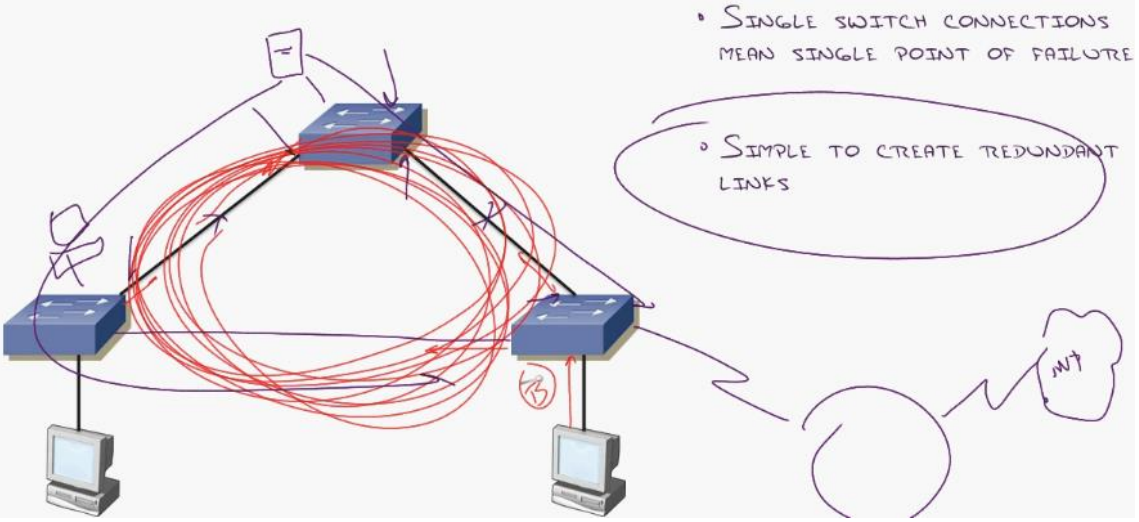
How a Cisco Device Boots

1. Check the configuration register
   - 2/00: rommon
   - 2/01: rxboot
   - 2/02: boot normally
   - 2/42: ignore nvram
2. Check for "boot system" commands in the startup config
3. Look for the first IOS image in flash
4. Broadcast for a TFTP server

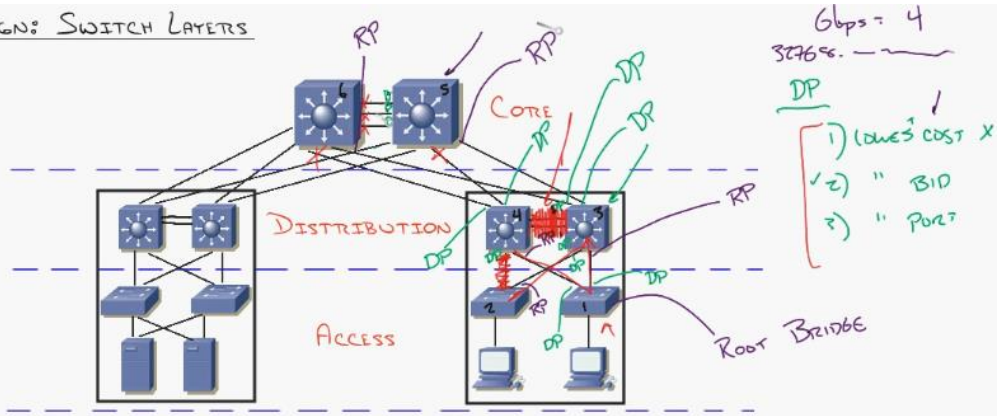


Fact: Redundancy is Good!

- Single switch connections mean single point of failure
- Simple to create redundant links



An Ideal Design: Switch Layers

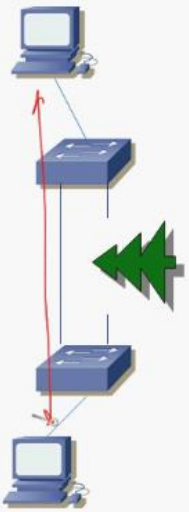


Gbps = 4
32768.
DP
1) (lowest cost
2) " BID
3) " Port

- Layered approach allows for easy, manageable growth
- EtherChannel can provide more bandwidth on key links
- Redundant connections eliminate a single point failure

Redundancy Chaos!!

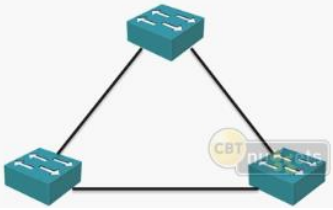
STP



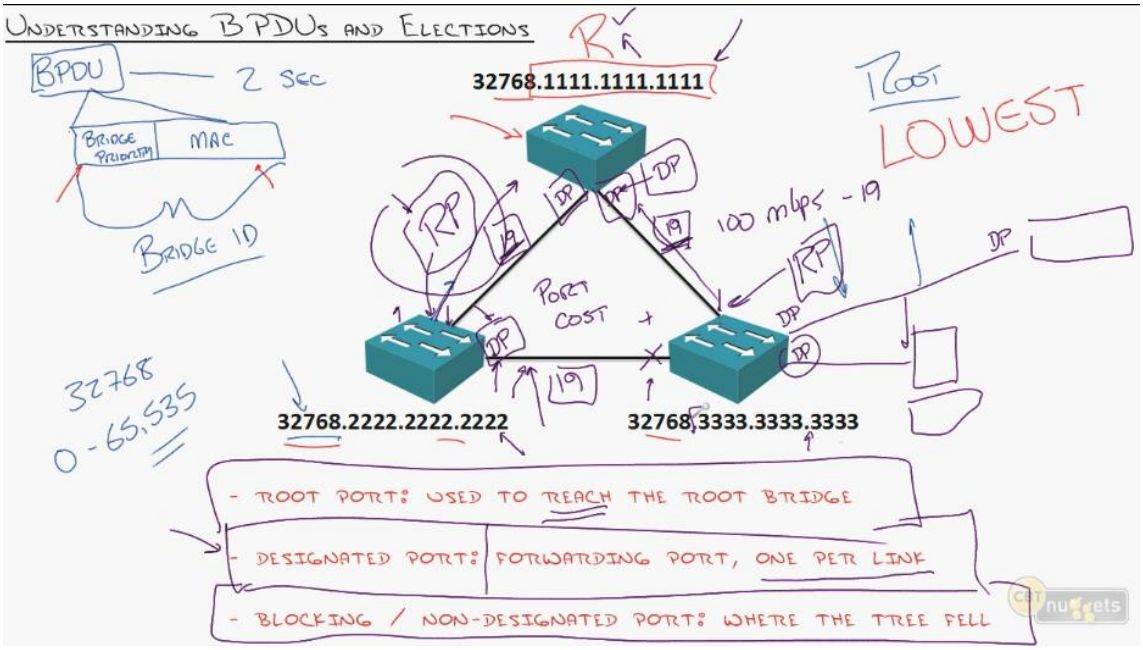
- Switches forward broadcast packets out all ports by design
- Redundant connections are necessary in business networks
- The place of spanning tree: drop trees on redundant links (until they are needed)

Key Facts About STP

- Original STP (802.1D) was created to prevent loops
- Switches send "probes" into the network called Bridge Protocol Data Units (BPDU) to discover loops
- The BPDU probes also help elect the core switch of the network, called the Root Bridge
- The simplistic view of STP: all switches find the best way to reach the root bridge then block all redundant links



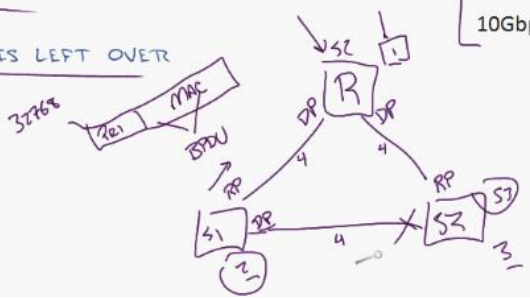
Understanding BPDUs and Elections



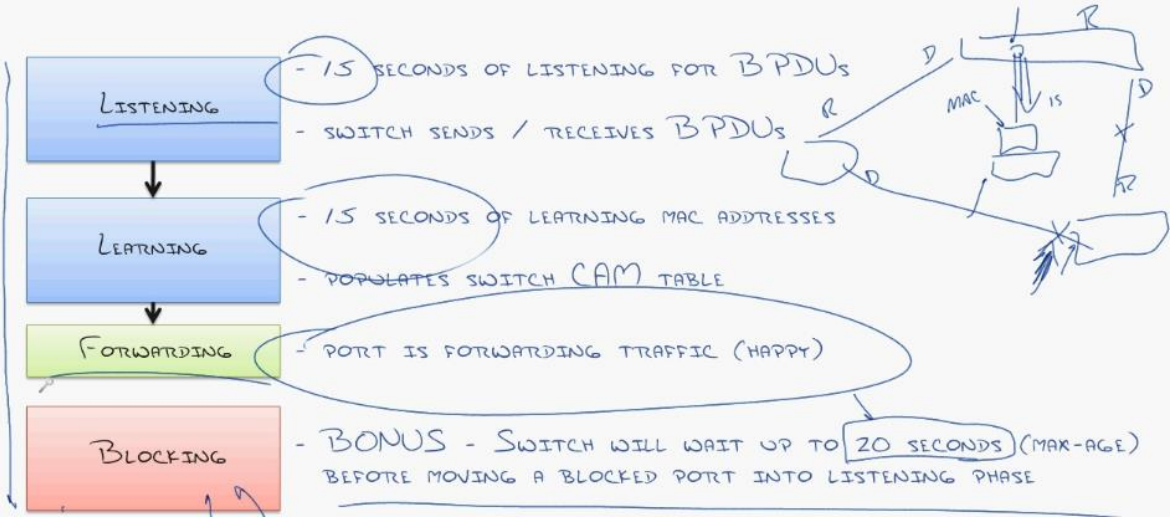
How STP Finds the Best Path

1. Elect the root
2. Find the best path to the root
   - lowest cost
   - lowest bridge ID
   - lowest port number
2.5 En 2 dp per link
3. Block whatever is left over

Link Bandwidth	Cost
10Mbps	100
100Mbps	19
1Gbps	4
10Gbps	2



STP was created a long time ago...



Problems and Solutions

Problems with PCs: Modern PCs can boot faster than the 30 seconds

Solution: Portfast

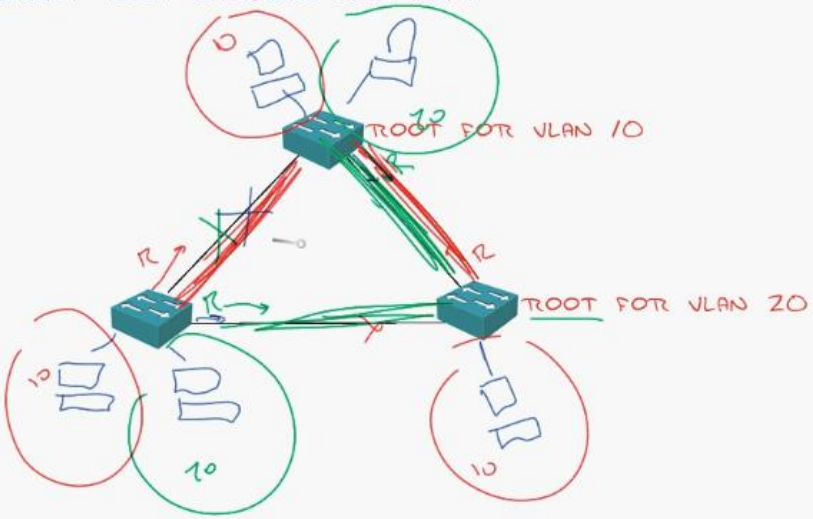
```
Switch(config-if)# spanning-tree portfast

%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface  when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/6 but will only
have effect when the interface is in a non-trunking mode.
```

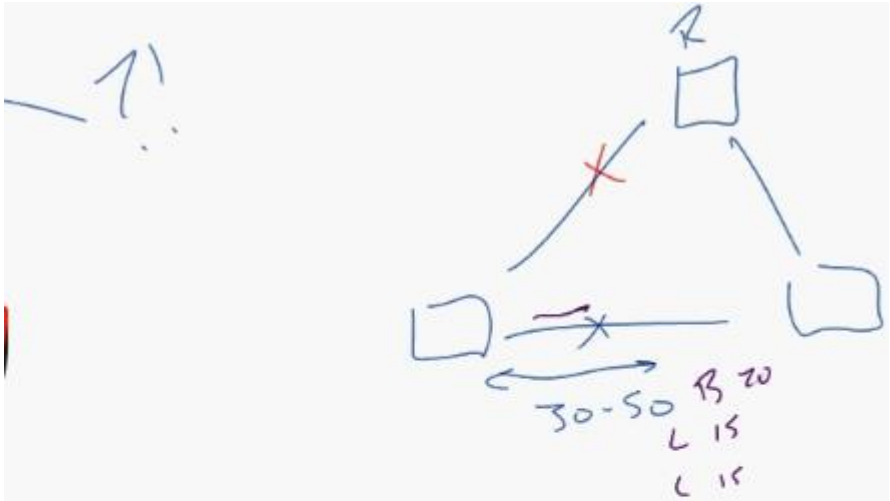
Problems with uplink ports: 50 seconds of down time causes big problems

Solution: Rapid Spanning-Tree

Initial STP Enhancement: PVST+

- Runs an instance of STP per-VLAN
- Allows different root bridges per VLAN





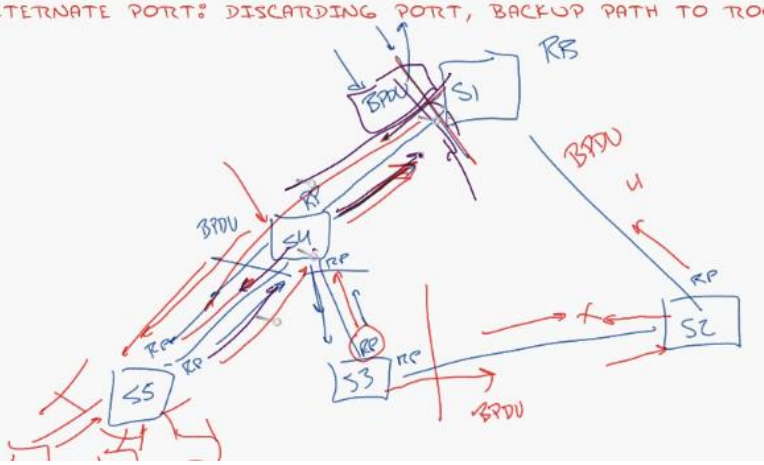
Rapid STP:

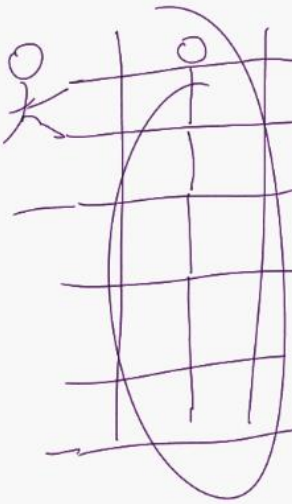
STP 802.1D

- 802.1w
- ~~PROACTIVE~~ SYSTEM
- REDEFINED PORT ROLES
- MANY STP SIMILARITIES

How RSTP Improves Performance

- More logical port types
  - root port: used to reach the root bridge
  - designated port: forwarding port, one per link
  - alternate port: discarding port, backup path to root

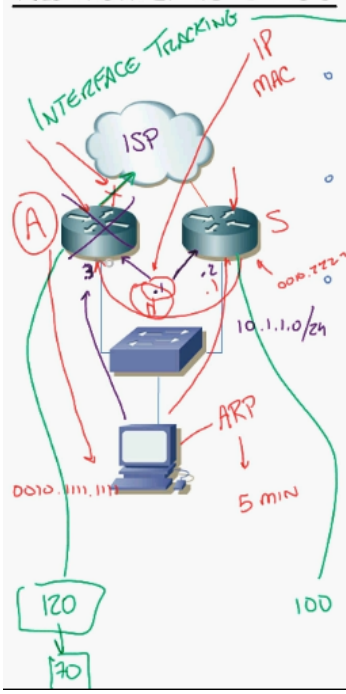




Redundancy is Good!

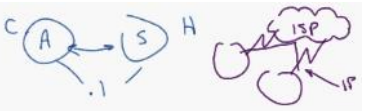
50/0 – 50 priority

- HSRP
- VRRP
- GLBP



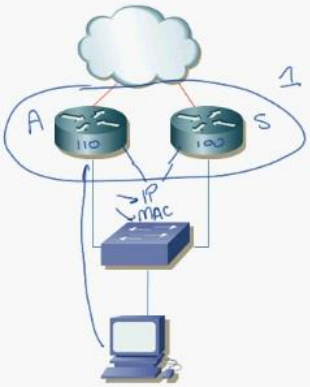
- "Two is One, One is None"
- Each client can only have one default gateway
- Questions:
  - How does a router know when to take over?
  - How does a client switch?
  - What about ARP cache issues?
  - What if only the WAN link fails?

Flavors of Redundancy



- Cisco Hot-Standby Router Protocol (HSRP):
  - Created by Cisco, for Cisco in 1994
  - Hello - 3 seconds; Hold - 10 seconds
- Virtual Router Redundancy Protocol (VRRP)
  - Created by the IETF in 1999
  - Hello - 1 second; Hold - 3 seconds
- Gateway Load Balancing Protocol (GLBP)
  - Created by Cisco, for Cisco in 2005
  - Identical to HSRP, but allows active-active load-balancing

A Focus on HSRP

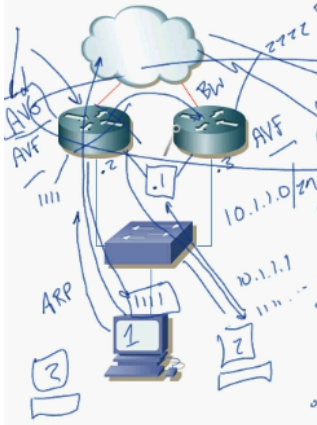


- Gateways organized into standby groups
- One gateway active, others in standby state
- Phantom (virtual) router IP and MAC address generated
- Hello messages sent once every 3 seconds; dead after 10 sec

A Focus on GLBP

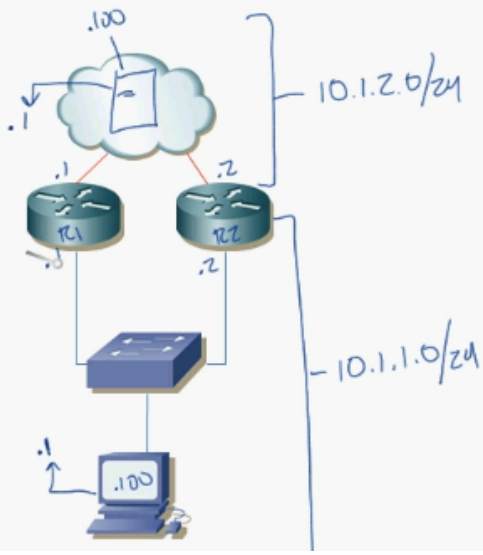
AVG – Resp to ARP requests

MAC1: 1111:1111:1111
MAC2: 1111:1111:2222



- One gateway assigned active virtual gateway (AVG) role
- Additional gateways act as active virtual forwarders (AVF)
- Phantom (virtual) router IP and MAC addresses generated
- AVG responds to ARP requests, load balances MAC replies
- Hello messages sent once every 3 seconds; dead after 10 sec

HSRP Base Configuration



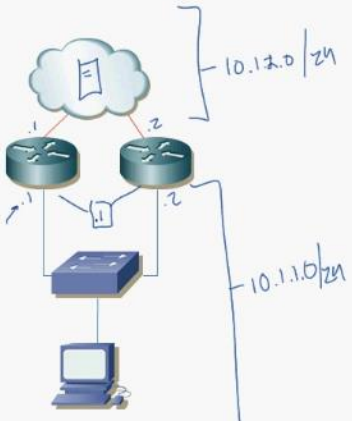
STEP 1: CREATE STANDBY GROUP

STEP 2: REASSIGN IP ADDRESSES

STEP 3: VERIFY

STEP 4: OPTIMIZE AND TUNE

VRRP Base Configuration

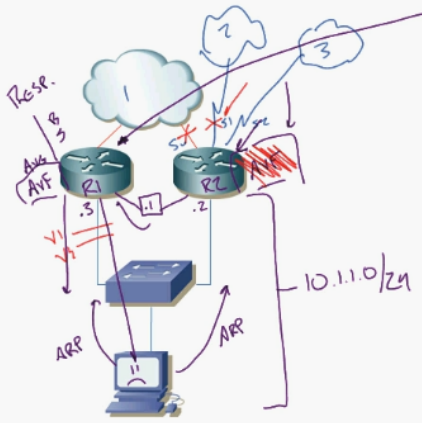


STEP 1: CREATE VRRP GROUP

STEP 2: VERIFY

STEP 3: OPTIMIZE AND TUNE

GLBP Base Configuration



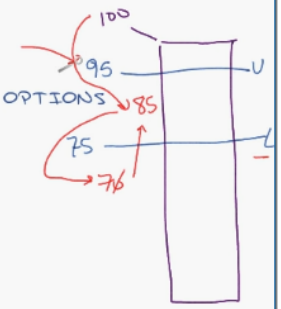
HSRP - STANDBY
VRRP
GLBP

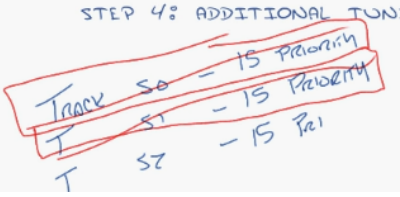
STEP 1: CREATE GLBP GROUP

STEP 2: VERIFY

STEP 3: CONFIGURE WEIGHTING OPTIONS

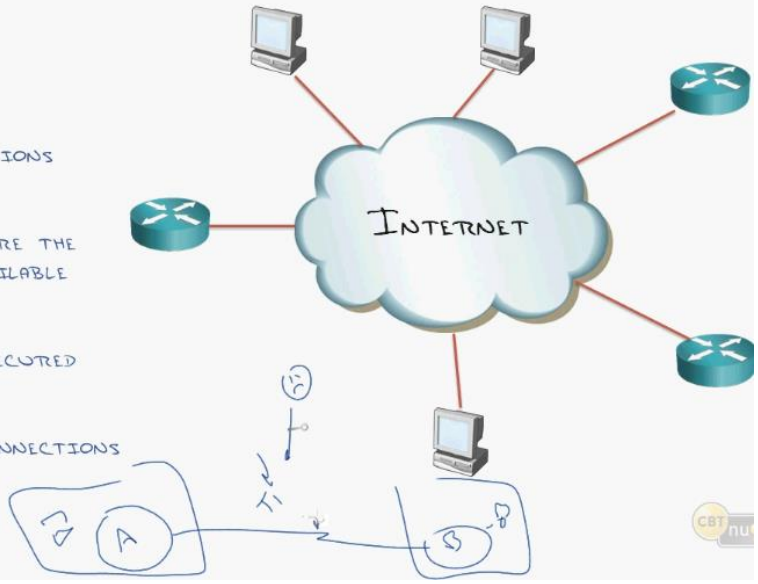
STEP 4: ADDITIONAL TUNING





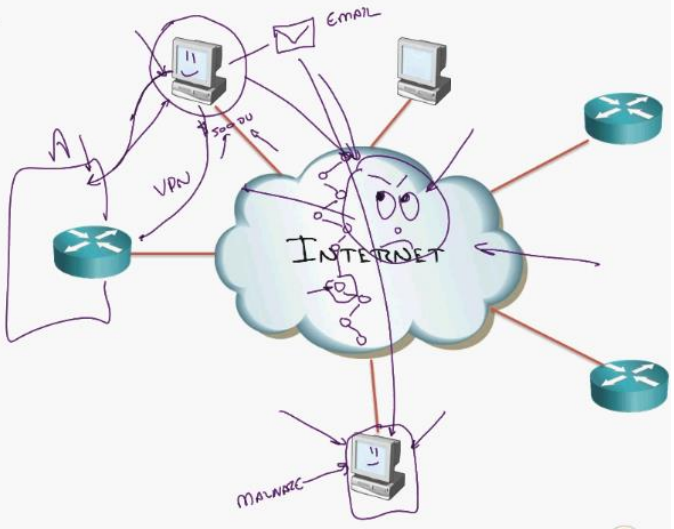
The Benefits of VPNs

- VPNs Defined
- Cheaper Connections
- Available anywhere the Internet is available
- Encrypted and secured
- Many-to-many connections

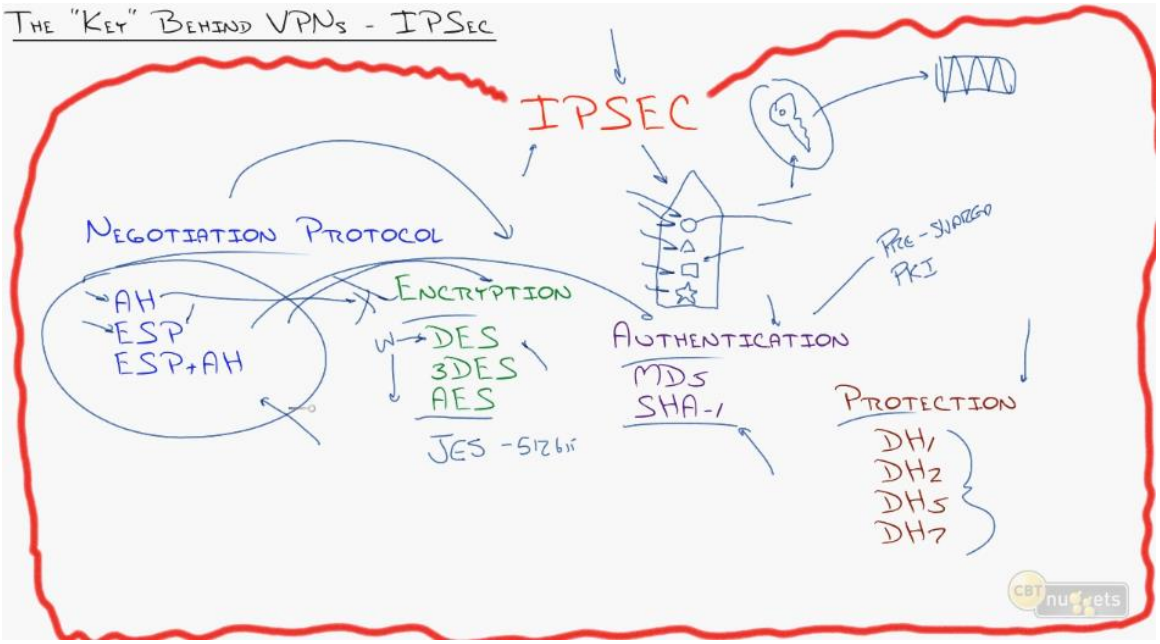


VPNs Must Provide...

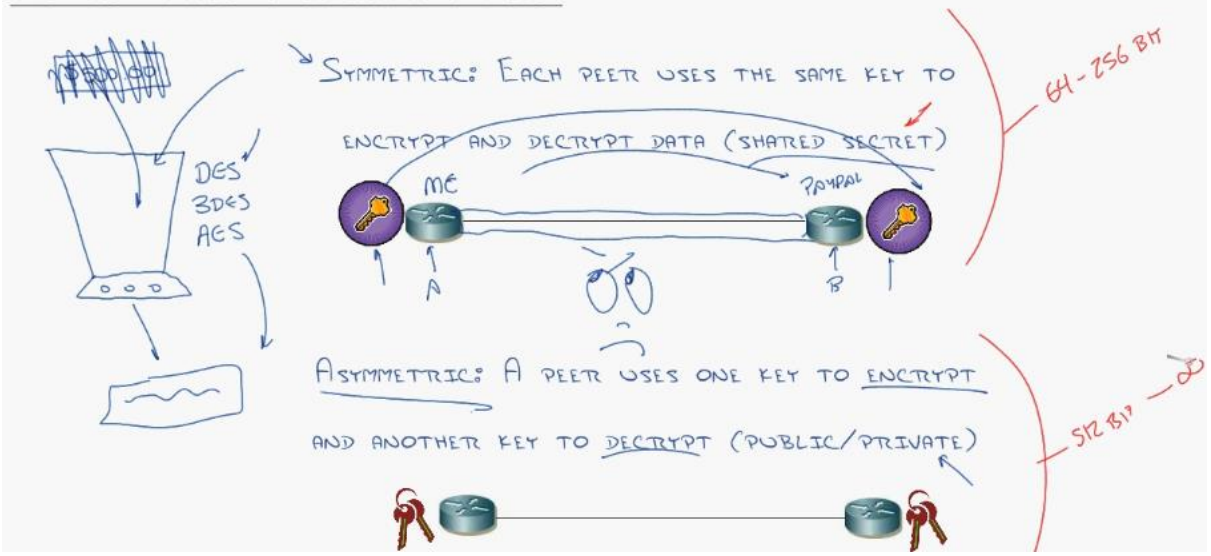
- Authentication
- Data Integrity
- Confidentiality
- Anti-Replay



The "Key" Behind VPNs - IPSec

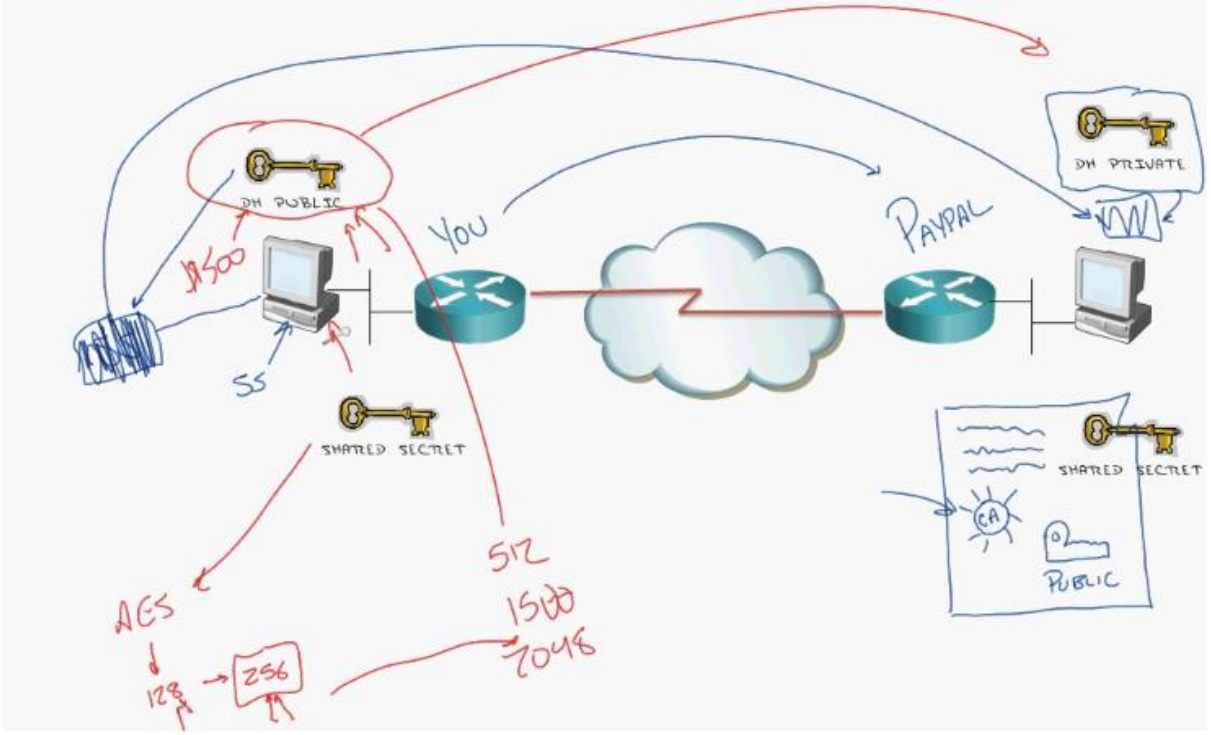


The Two Types of Encryption Keys



Security Over a Public Network?

...can it be true?!?



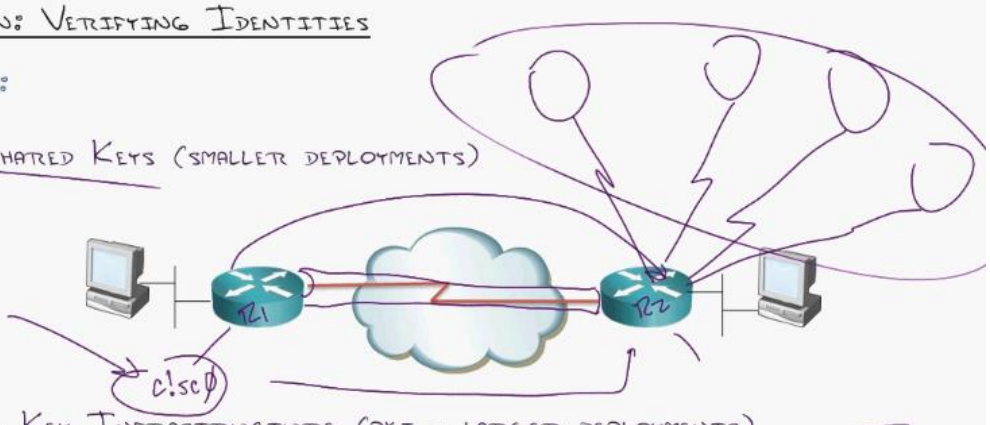
Modern Day Encryption Algorithms

- DES - Created by IBM, 56-bit key (s)
- 3DES - Uses three DES keys on each block of data to create 168-bit keys (s)
- AES - Newer, more efficient algorithm, 128-, 192-, and 256-bit key (s)
- RSA - Used for "misc" encryption, 512-, 768-, 1024-bit...or larger (a)
- DH - Used commonly on VPN connections to allow secure transfer of shared secret keys (and helps generate shared secret keys). 768-, 1024-, 1536-bit...or larger (a)

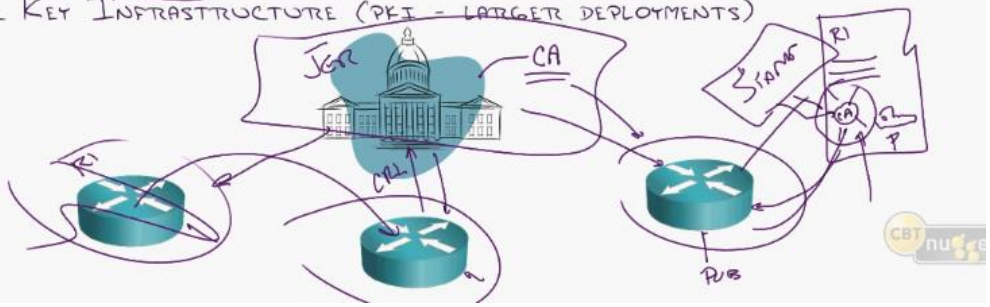
Authentication: Verifying Identities

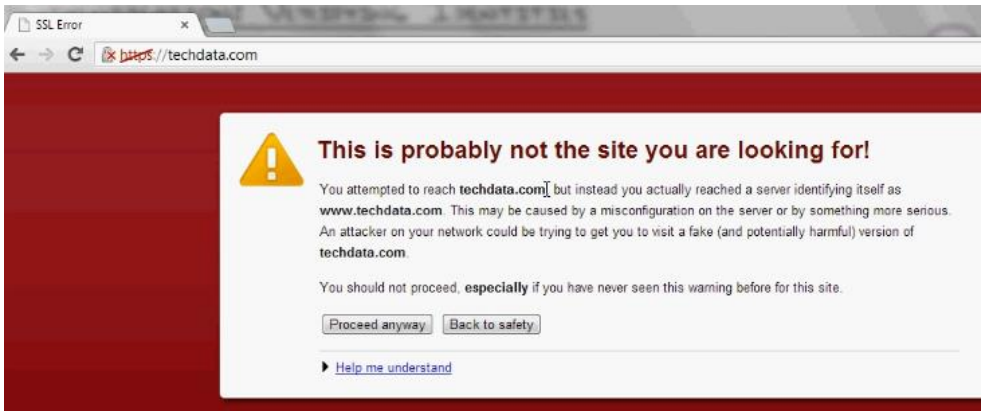
Two methods:

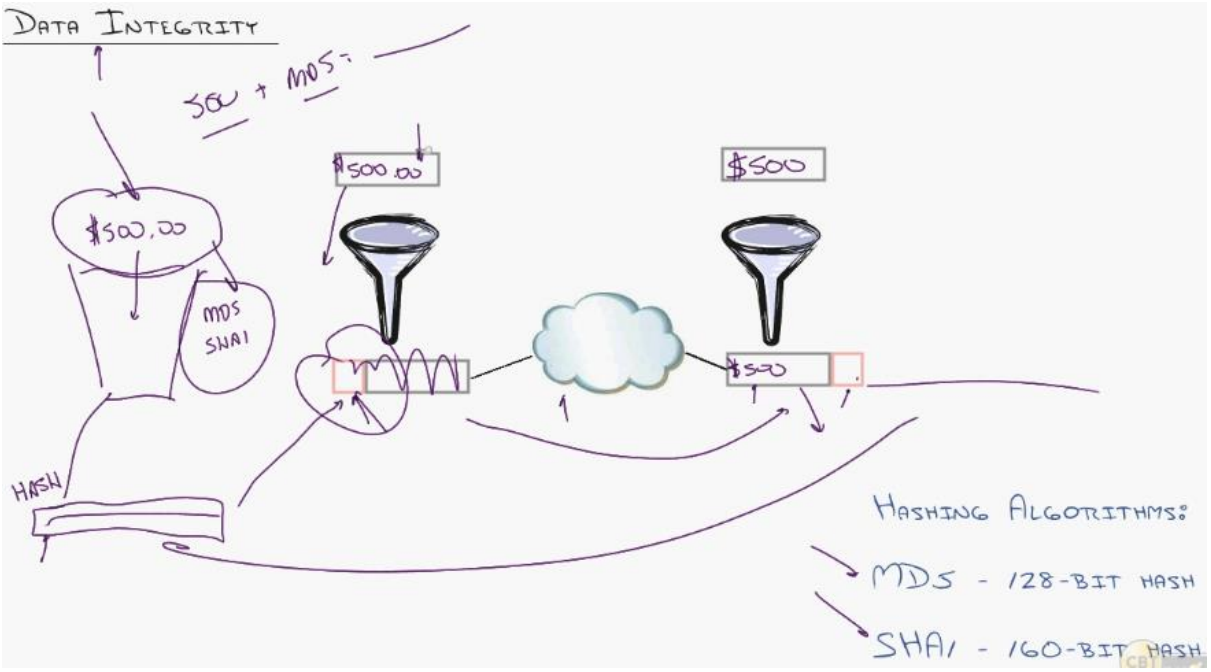
- Pre-Shared Keys (smaller deployments)

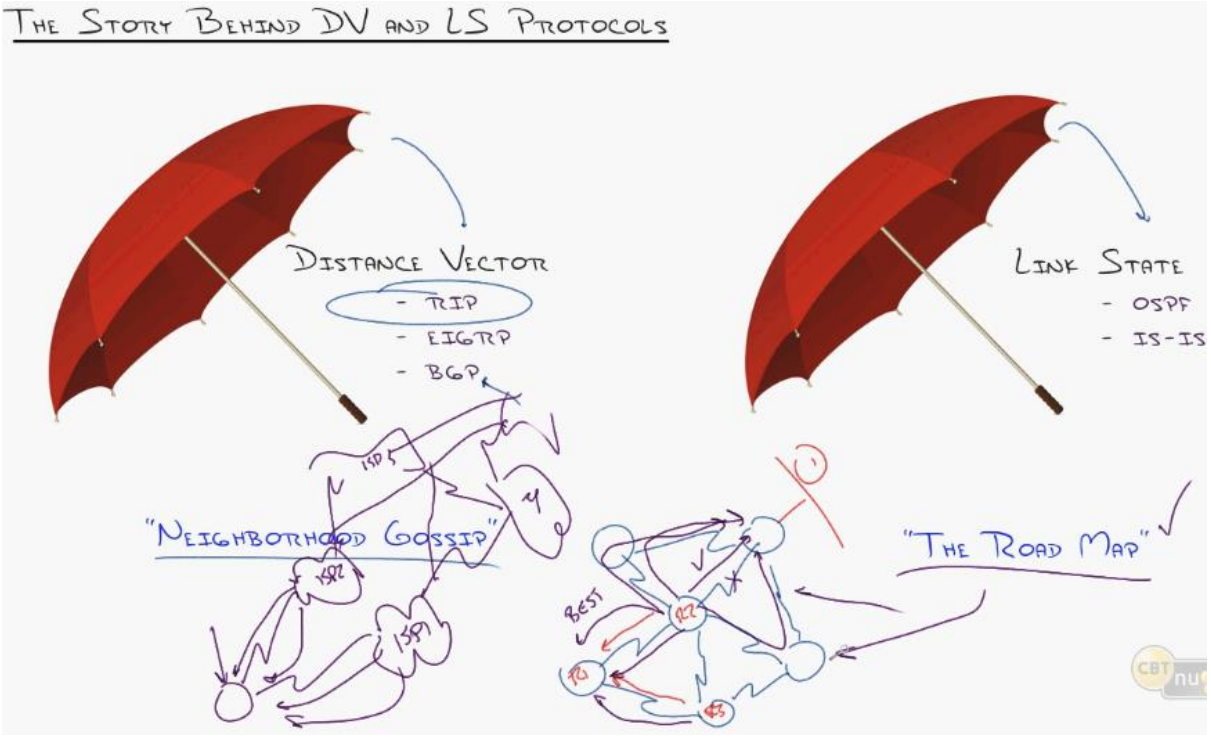


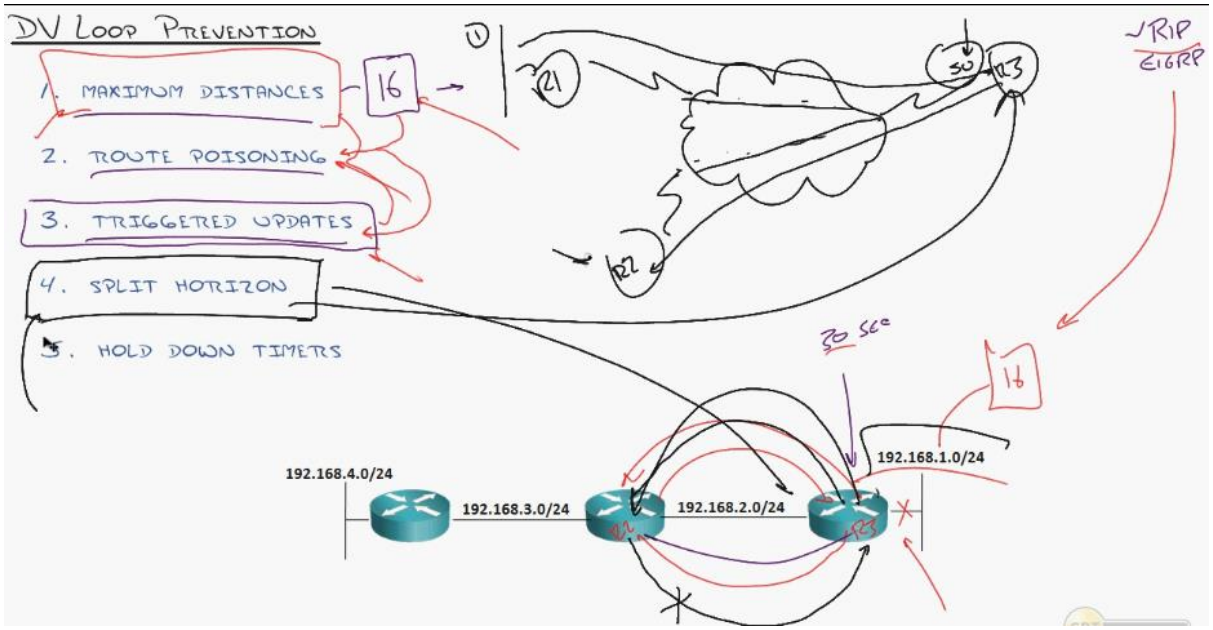
- Public Key Infrastructure (PKI - larger deployments)

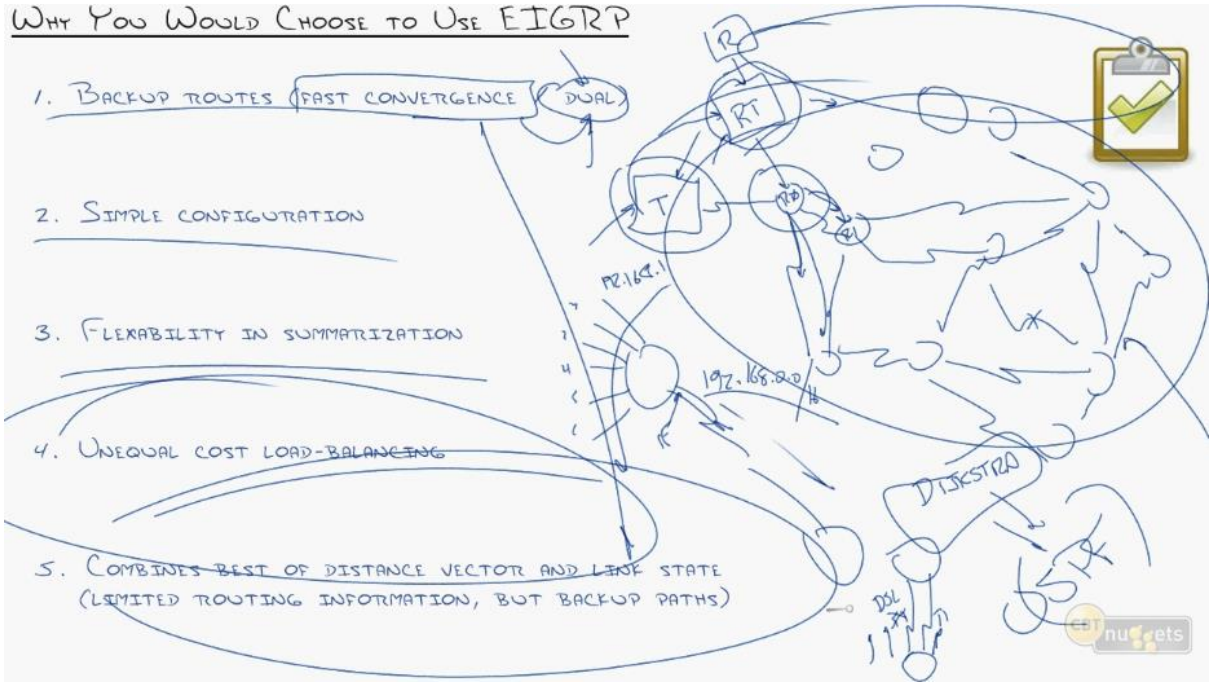





DATA INTEGRITY
500 + MD5:
$500.00
$500,00
MD5
SHA1
$500
HASH
HASHING ALGORITHMS:
MD5 - 128-BIT HASH
SHA1 - 160-BIT HASH


THE STORY BEHIND DV AND LS PROTOCOLS
DISTANCE VECTOR
- RIP
- EIGRP
- BGP
LINK STATE
- OSPF
- IS-IS
"NEIGHBORHOOD GOSSIP"
"THE ROAD MAP"
BEST

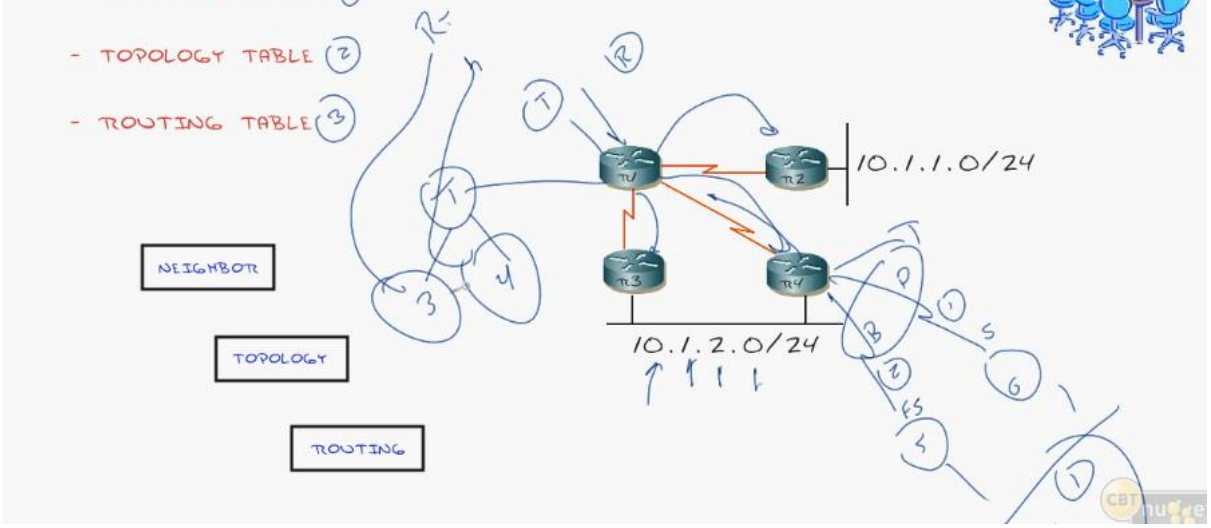

DV Loop Prevention
1. Maximum Distances
16
2. Route Poisoning
3. Triggered Updates
4. Split Horizon
5. Hold Down Timers
RIP
EIGRP
30 sec
16
192.168.4.0/24
192.168.3.0/24
192.168.2.0/24
192.168.1.0/24


Why You Would Choose to Use EIGRP
1. Backup routes
Fast convergence
DUAL
2. Simple configuration
3. Flexability in summarization
4. Unequal cost load-balancing
5. Combines best of distance vector and link state
(limited routing information, but backup paths)
Dijkstra
DSL

EIGRP Tables and Terminology

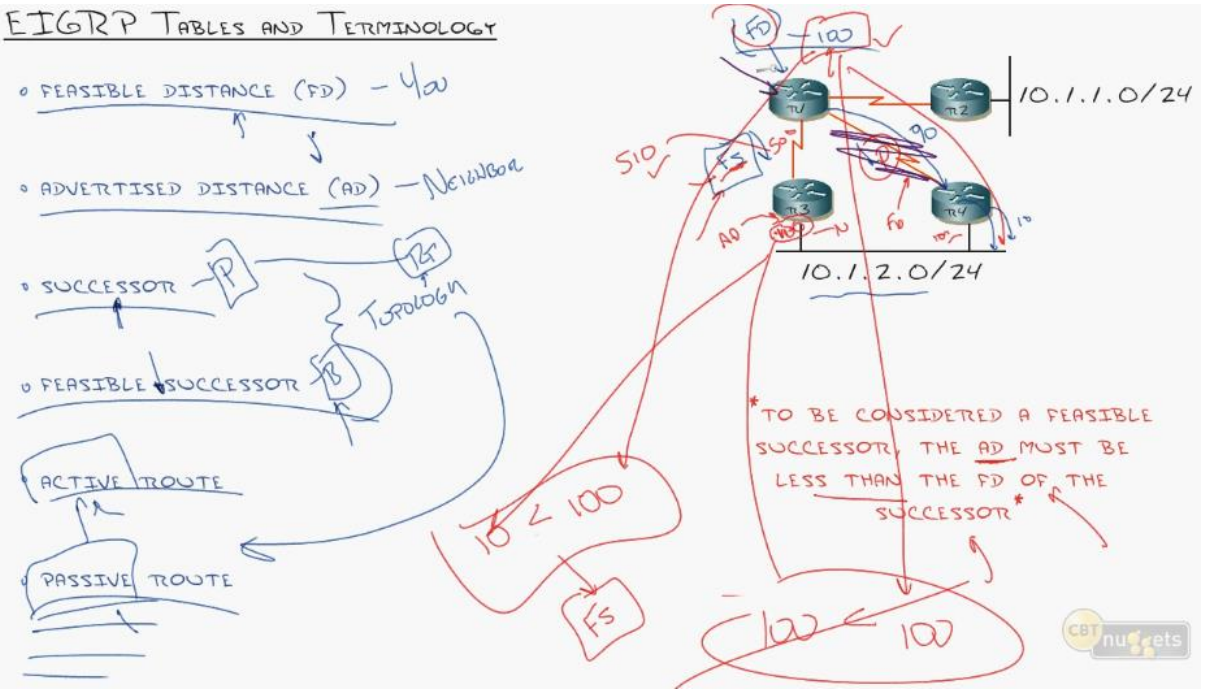
A router running EIGRP maintains three tables:

- Neighbor table (1)
- Topology table (2)
- Routing table (3)

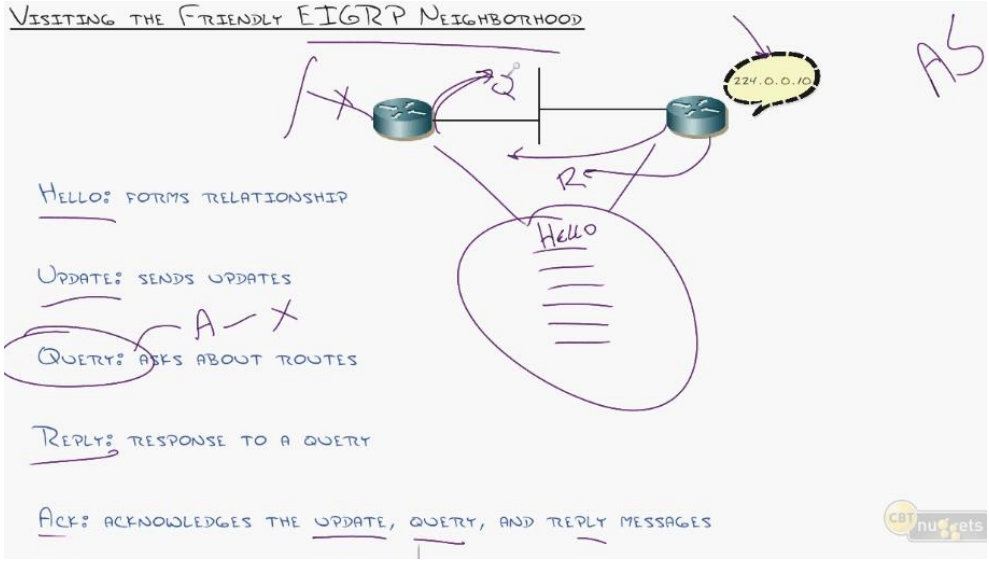


EIGRP Tables and Terminology

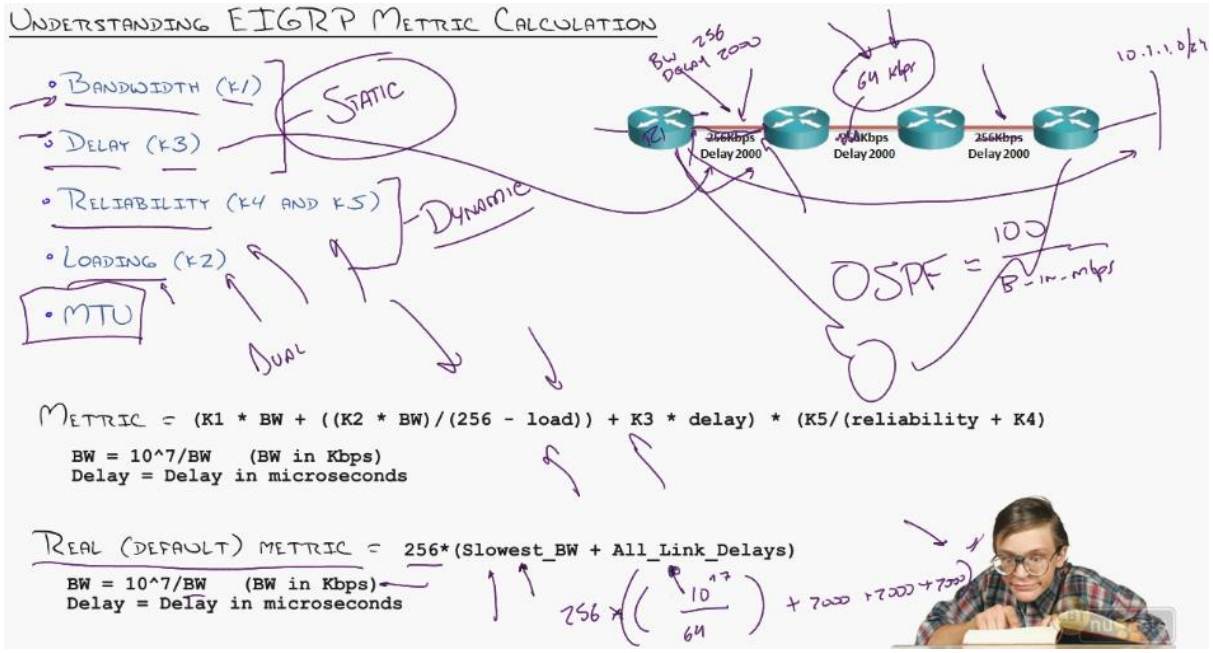
- Feasible Distance (FD) – Low
- Advertised Distance (AD) – Neighbor
- Successor
- Feasible Successor
- Active Route
- Passive Route



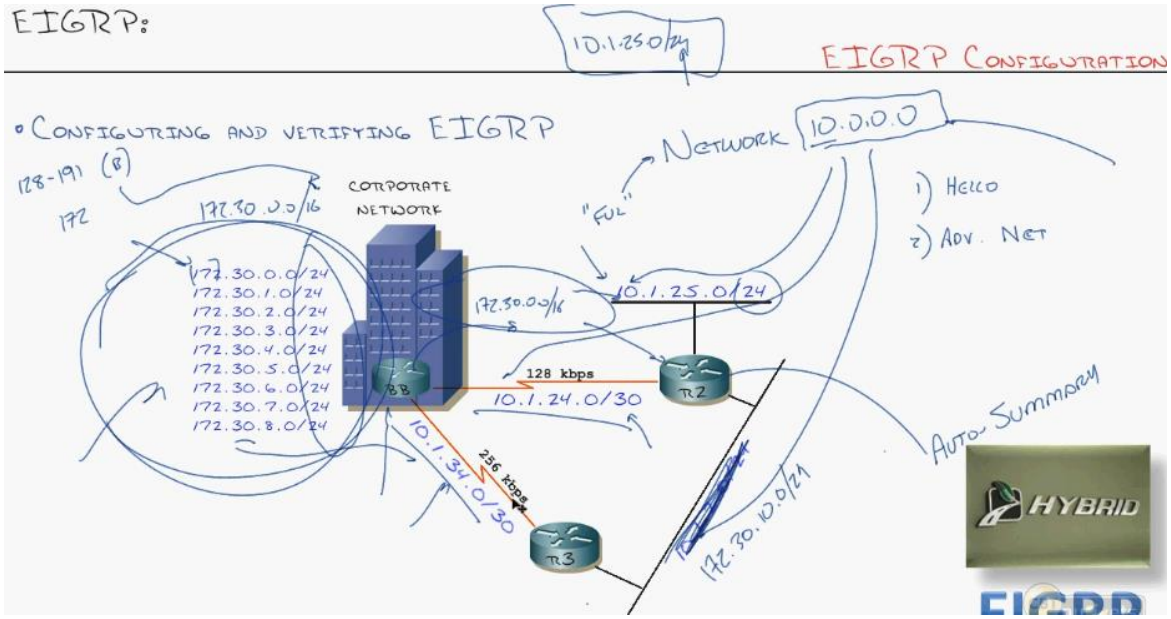
Visiting the Friendly EIGRP Neighborhood



Understanding EIGRP Metric Calculation

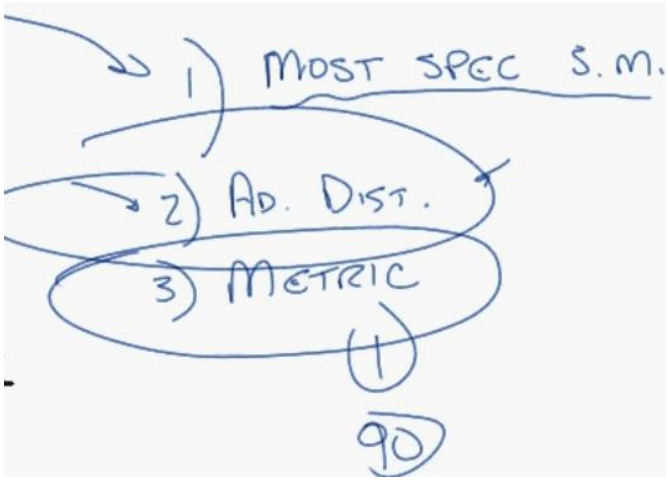


EIGRP:

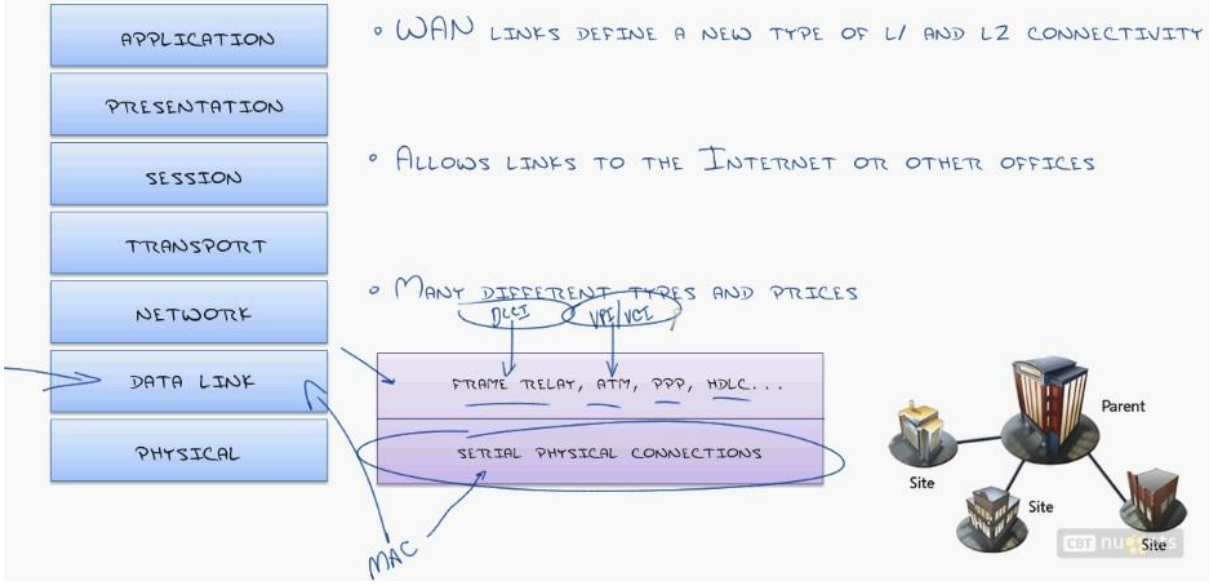


```
R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.2.2                Fa0/0             14 00:38:13 1022   5000  0  5
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.2.0/24 is directly connected, FastEthernet0/0
D       10.1.25.0/24 [90/409600] via 10.1.2.2, 00:38:53, FastEthernet0/0
D       10.1.24.0/30 [90/2195456] via 10.1.2.2, 00:38:53, FastEthernet0/0
C       10.1.34.0/30 is directly connected, Serial0/0
```

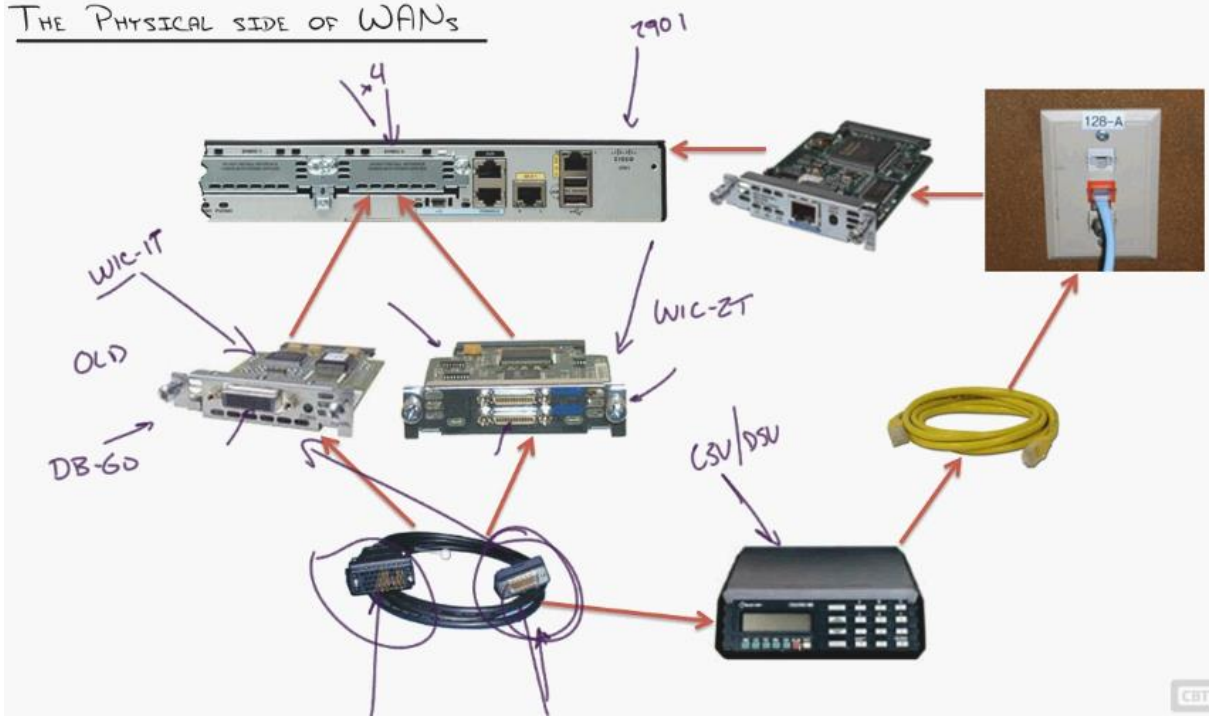
```
BB#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.30.8.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.2.0/24, 2 successors, FD is 2195456
        via 10.1.24.2 (2195456/281600), Serial0/1
        via 10.1.34.2 (2195456/281600), Serial0/0
P 10.1.25.0/24, 1 successors, FD is 2297856
        via 10.1.24.2 (2297856/128256), Serial0/1
        via 10.1.34.2 (2323456/409600), Serial0/0
P 10.1.24.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/1
P 10.1.34.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0
```

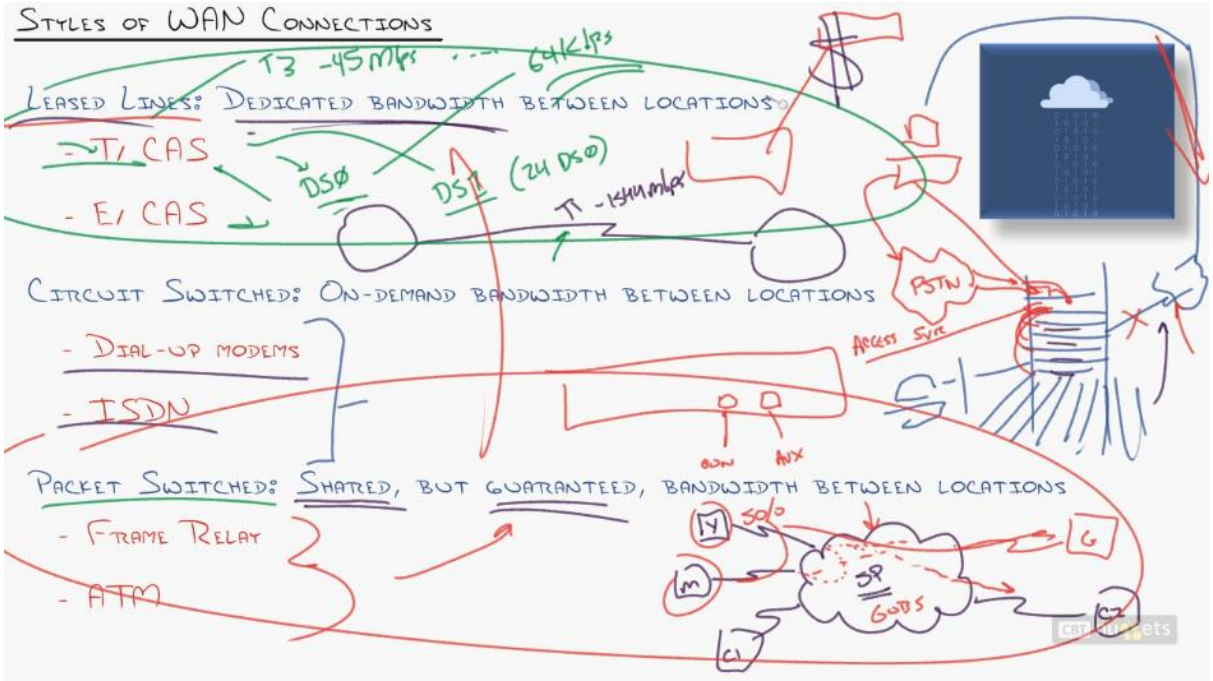


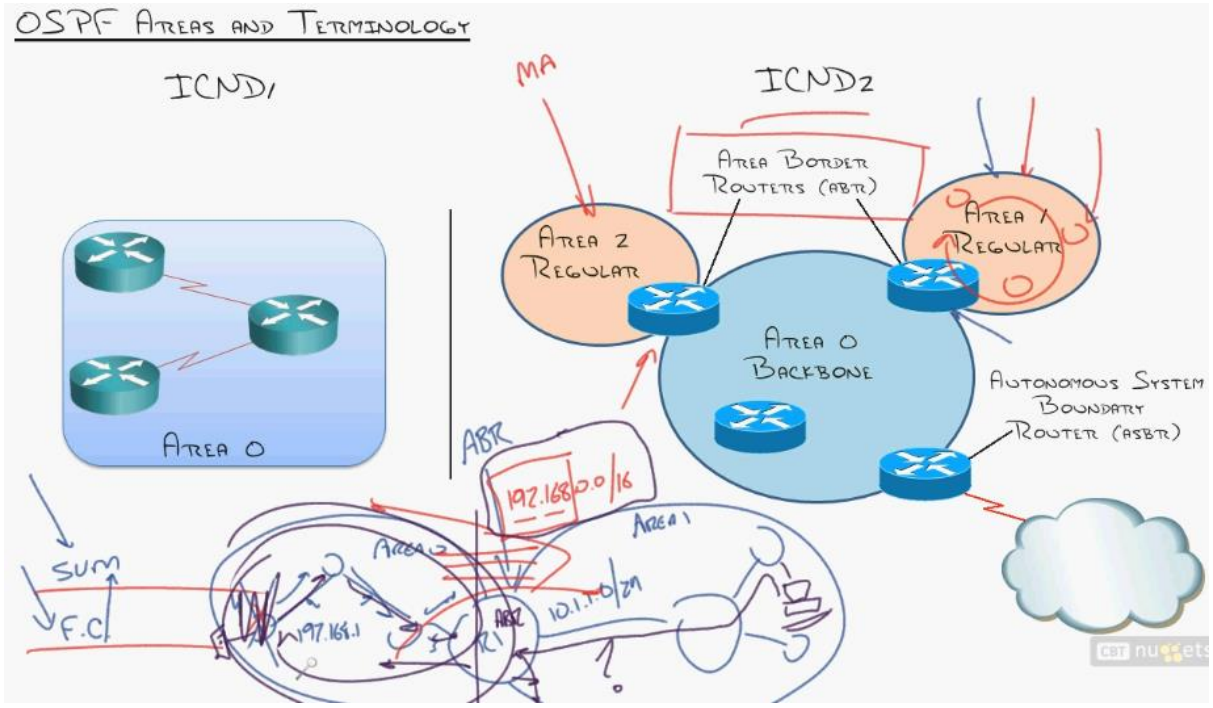
Where WAN Connections Fit



The Physical side of WANs

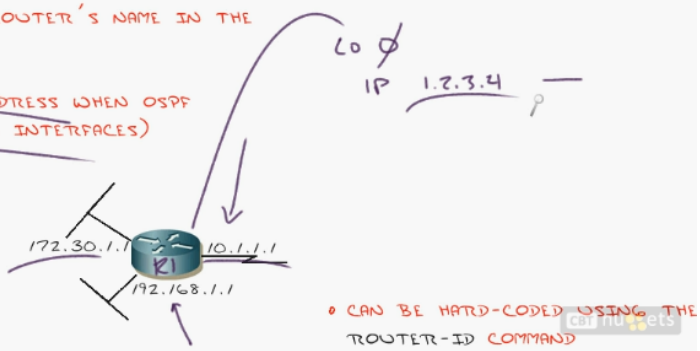



STYLES OF WAN CONNECTIONS
T3 - 45 Mbps
64Kbps
LEASED LINES: DEDICATED BANDWIDTH BETWEEN LOCATIONS
- T1 CAS
- E1 CAS
DS0
DS1 (24 DS0)
T1 - 1544Mbps
CIRCUIT SWITCHED: ON-DEMAND BANDWIDTH BETWEEN LOCATIONS
- DIAL-UP MODEMS
- ISDN
PSTN
ACCESS SVR
AUX
PACKET SWITCHED: SHARED, BUT GUARANTEED, BANDWIDTH BETWEEN LOCATIONS
- FRAME RELAY
- ATM
SP
GODS


OSPF AREAS AND TERMINOLOGY
ICND1
MA
ICND2
AREA BORDER ROUTERS (ABR)
AREA 2 REGULAR
AREA 1 REGULAR
AREA 0 BACKBONE
AUTONOMOUS SYSTEM BOUNDARY ROUTER (ASBR)
AREA 0
ABR
192.168.0.0/16
AREA 1
AREA 2
SUM
F.C.
192.168.1
10.1.1.0/24

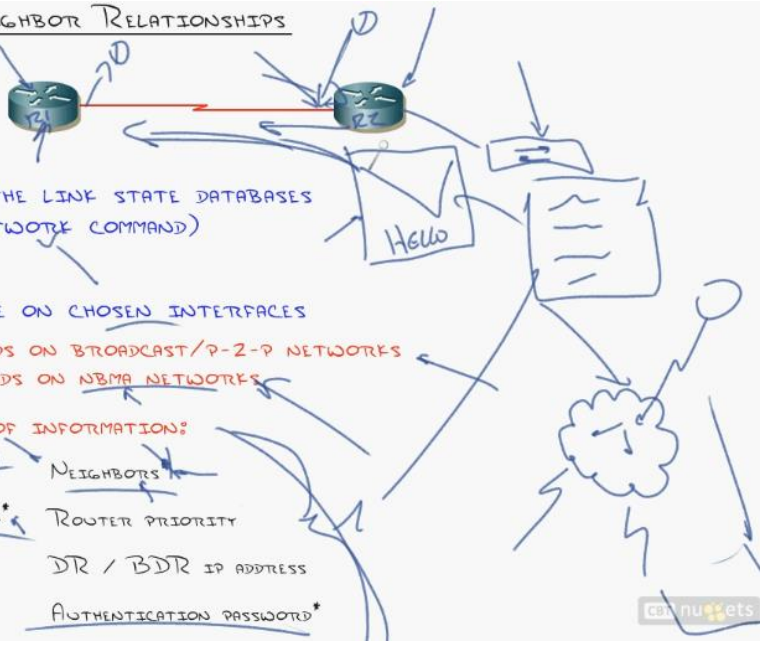
Understanding OSPF Neighbor Relationships

1. Determine your own Router ID
   - The router ID is simply the router's name in the OSPF process
   - Highest active interface IP address when OSPF starts (loopbacks beat physical interfaces)



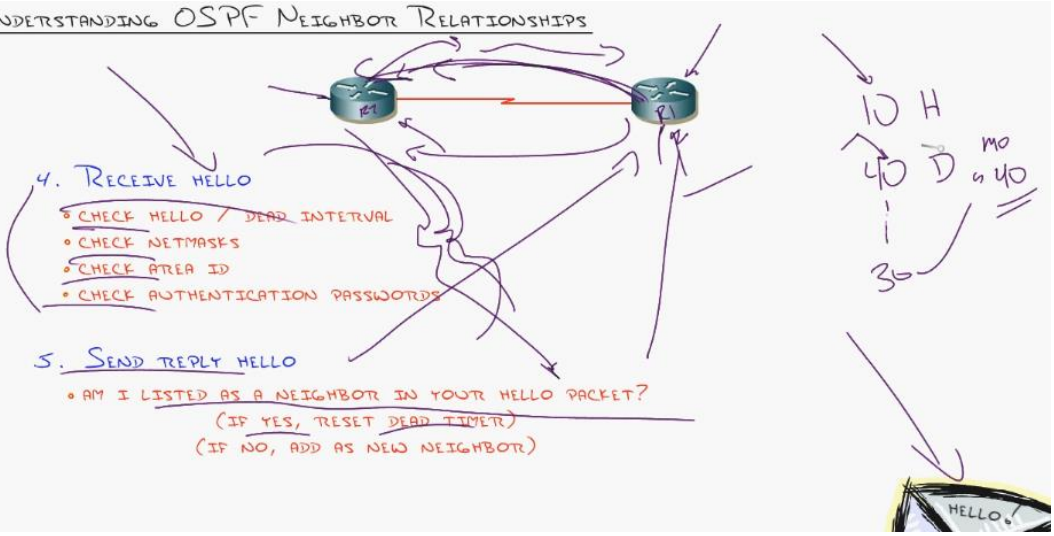
   - Can be hard-coded using the router-id command

Understanding OSPF Neighbor Relationships



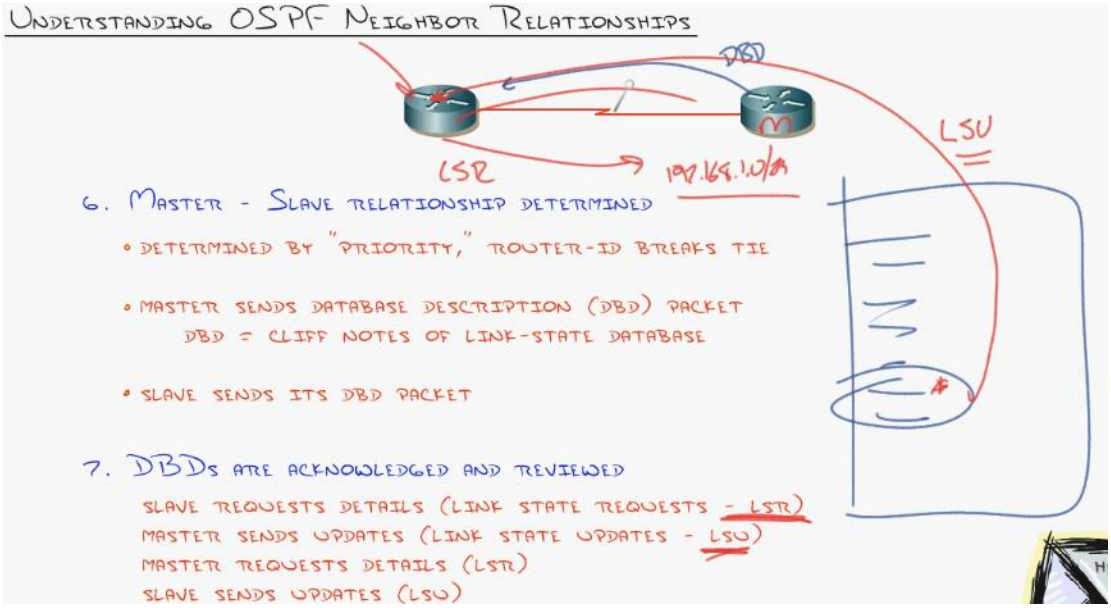
2. Add interfaces to the link state databases (dictated by the network command)
3. Send a hello message on chosen interfaces
   - Once every 10 seconds on broadcast/p-2-p networks
   - Once every 30 seconds on NBMA networks
   - Contains all sorts of information:
     - Router ID
     - Neighbors*
     - Hello and dead timers*
     - Router priority
     - Network mask*
     - DR / BDR IP address
     - Area ID*
     - Authentication password*

Understanding OSPF Neighbor Relationships

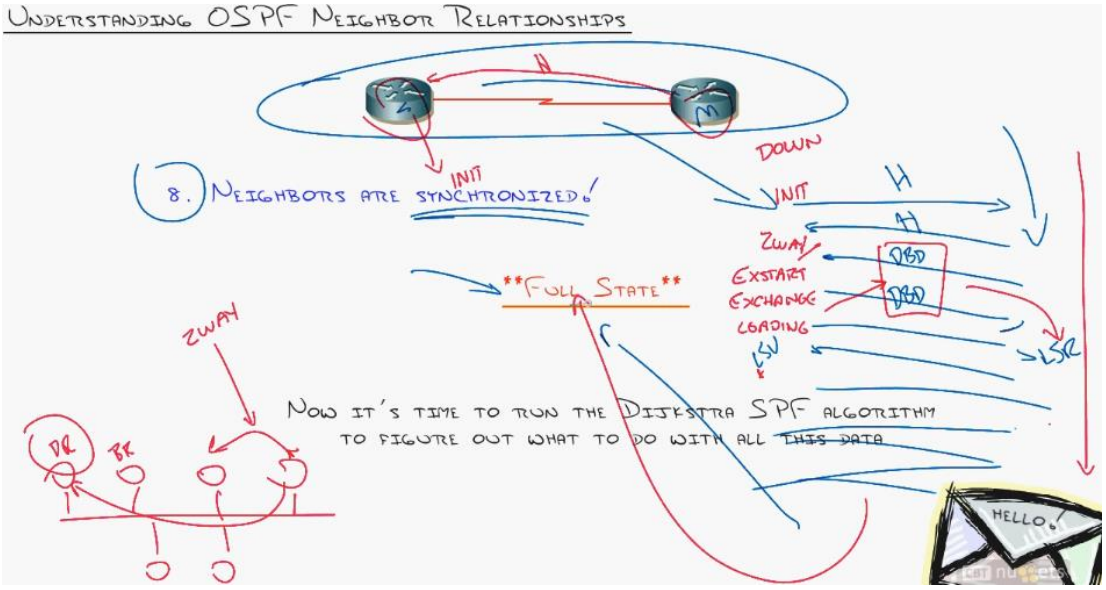


4. Receive Hello
   - Check Hello / Dead interval
   - Check netmasks
   - Check area ID
   - Check authentication passwords
5. Send reply Hello
   - Am I listed as a neighbor in your Hello packet?
     (If yes, reset dead timer)
     (If no, add as new neighbor)

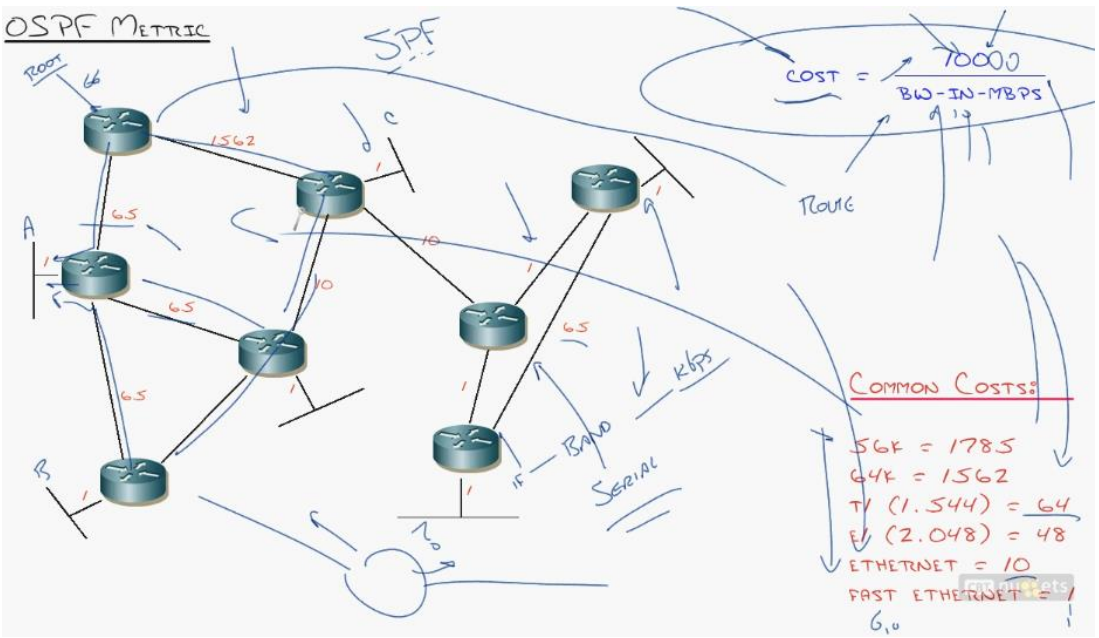
Understanding OSPF Neighbor Relationships



Understanding OSPF Neighbor Relationships



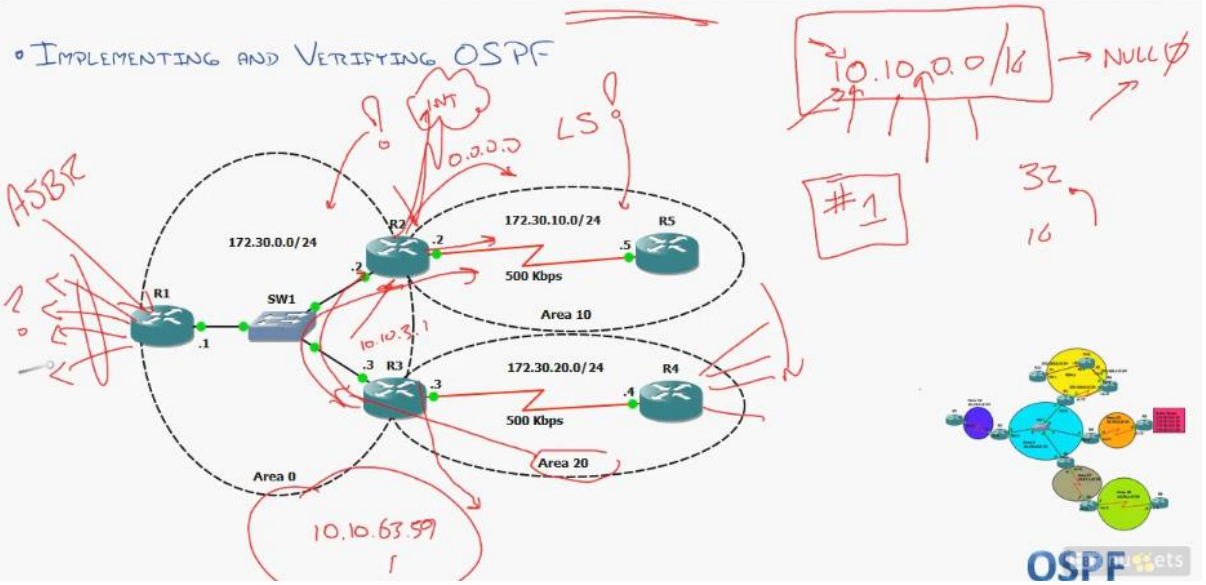
OSPF Metric



OSPF:

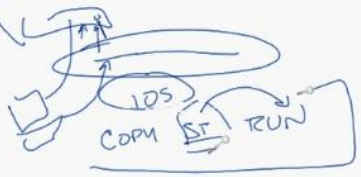
OSPF Multi-Area Configuration and Verificatio

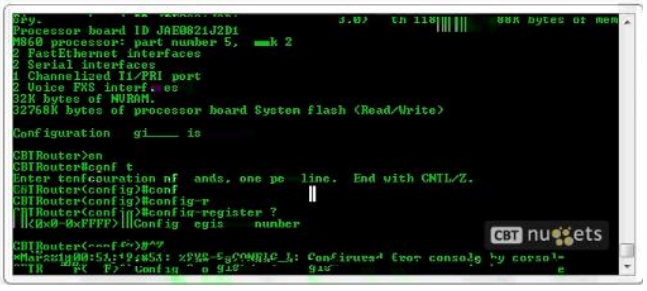
- Implementing and Verifying OSPF



How a Cisco Device Boots

1. Check the configuration register
   - 2100: ROMMON
   - 2101: RXBOOT
   - 2102: Boot normally
   - 2142: Ignore NVRAM
2. Check for "boot system" commands in the startup config
3. Look for the first IOS image in flash
4. Broadcast for a TFTP server





IOS License Management

- Before IOS version 15 (12.x versions and prior), all-inclusive IOS features
  - IP BASE
  - ADV. SECURITY
  - ADV. IP SERVICES
  - IP VOICE
  - ENT. BASE
  - ENT. SERVICES
- Post IOS version 15, all features in image; activated with license key
  - IP BASE
  - DATA (MPLS, ATM, MULTI-PROTOCOL)
  - UNIFIED COMMUNICATION (VOIP)
  - SECURITY (FIREWALL, VPN, ENCRYPTION)

Verifying License Information

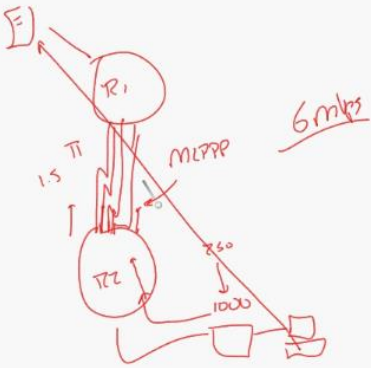
(show version output)

```
-------------------------------------------------
Device#   PID                  SN
-------------------------------------------------
*0        CISCO2951            FHH1211P037


Technology Package License Information for Module:'c2951'


-----------------------------------------------------------------
Technology    Technology-package          Technology-package
              Current       Type          Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      None          ipbasek9
security      disable       None          disable
uc            uc            Evaluation    uc
data          None          None          None

Configuration register is 0x0
```

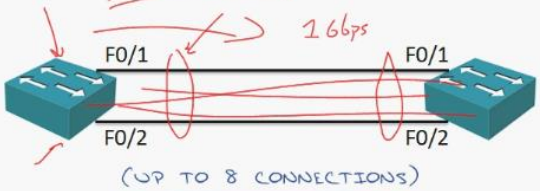
Licenses installed with "license install" command

Bundling Instead of Blocking

- How would STP handle the above situation?



- Common EtherChannel options:



(up to 8 connections)

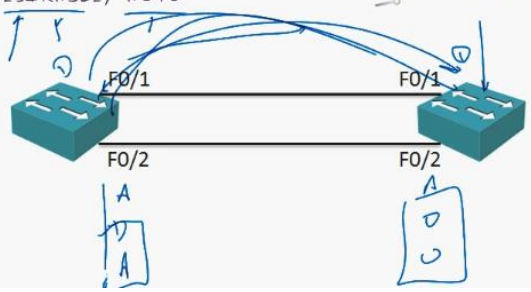
EtherChannel Protocols

PAgP

- Cisco proprietary
- Modes: On, Desirable, Auto

LACP

- Industry standard
- Modes: On, Active, Passive



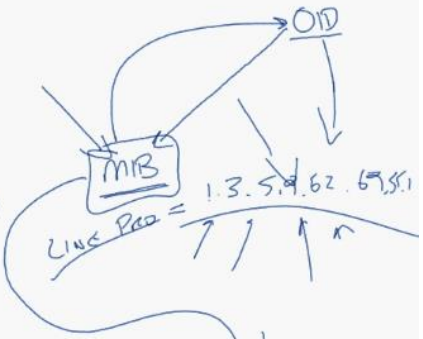
EtherChannel Configuration

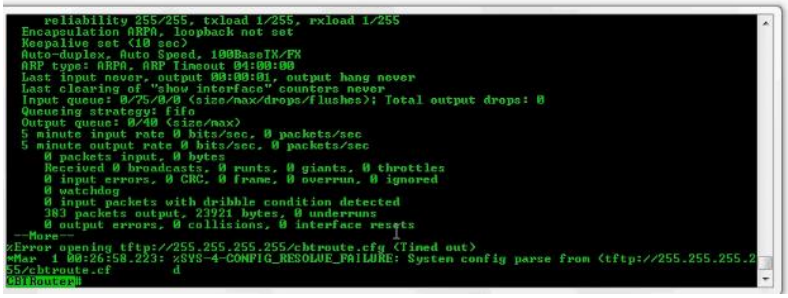
- Base interfaces must have identical configuration (speed, duplex, mode, VLANs)
- Use the channel-group command to create the EtherChannel
- All configuration done on the virtual port-channel interface after bundling
- Best verification: show etherchannel summary

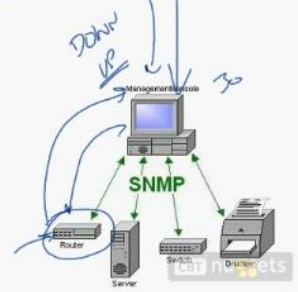
```
Port-channel10         unassigned      YES unset  down                    down
CBTSV1#show eth
CBTSV1#show etherchannel summ
CBTSV1#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

10     Po10(SD)         PAgP      Fa0/1(I)    Fa0/2(I)

CBTSV1#
```

Understanding SNMP

- SNMP "simply" grabs statistics from devices
- Monitoring applications manipulate and massage data
- Three SNMP versions

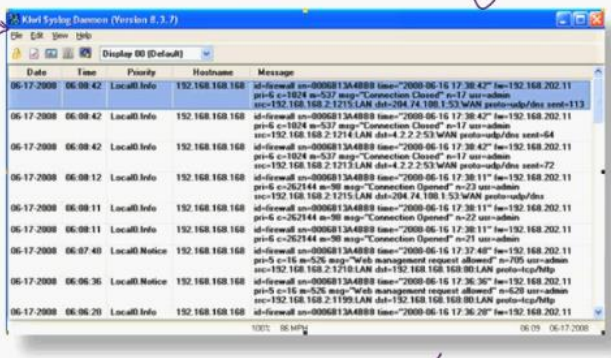






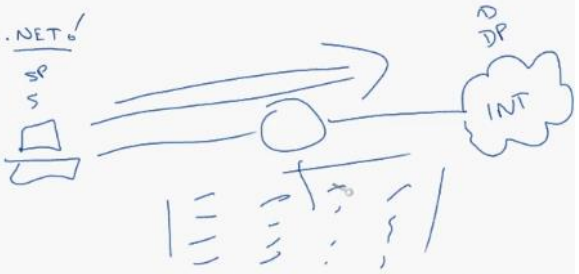
Capturing Syslog

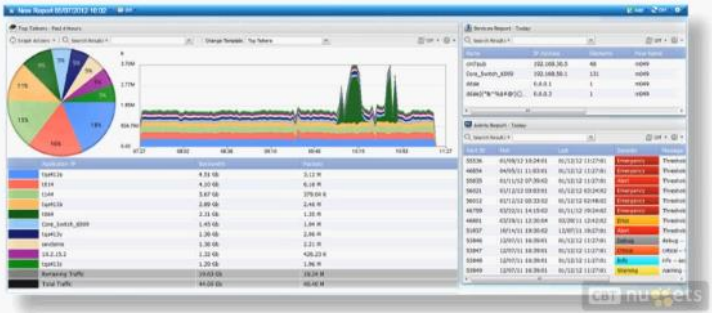
- Syslog captures key status messages from Cisco devices
- Each device can store syslog messages locally or on a remote server
- Syslog uses UDP port 514



The Power of Netflow

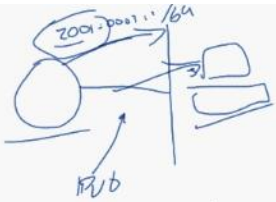
- Netflow - Tracks the flows...on the...net!
- Traffic Accounting:
  - per IP address
  - per protocol
  - per port



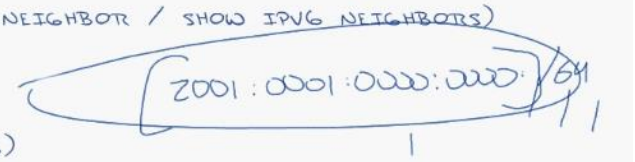


IPv6 General Troubleshooting

1. Test end-to-end connectivity (ping, tracert)



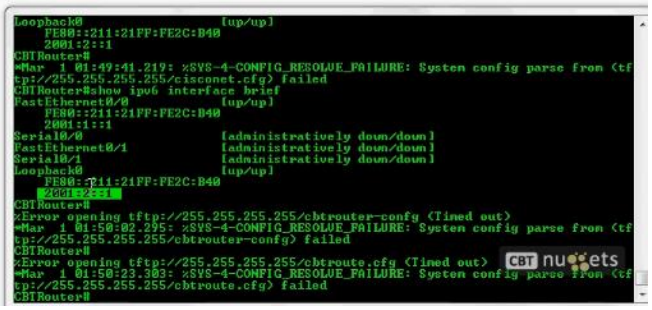
2. Verify L3 to L2 mapping (netsh neighbor / show ipv6 neighbors)



3. Verify default gateway (ipconfig)

4. Verify DNS (nslookup, hosts)

5. Verify ACLs




```
CBTRouter(config-if)#do show ipv6 int
% Ambiguous command:  "do show ipv6 int"
CBTRouter(config-if)#do show ipv6 int fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::211:21FF:FE2C:B40
  Global unicast address(es):
    2001:1::1, subnet is 2001:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF2C:B40
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
CBTRouter(config-if)#
```

EIGRP FOR IPV6

- ALREADY SUPPORTED PROTOCOL "MODULES" - PERFECT FOR IPV6!
- HELLO PACKETS - FF02::A LINK-LOCAL MULTICAST
- METRIC: THE SAME (BW, DELAY, RELIABILITY, LOAD, MTU)
- ENGINE: THE SAME (DUAL)
- LOAD BALANCING: THE SAME (UNEQUAL PATH SUPPORT)
- ENABLED PER INTERFACE INSTEAD OF USING THE NETWORK COMMAND

```
R3#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
D   2001:11::/64 [90/2684416]
     via FE80::C205:EFF:FE64:0, Serial0/0
D   2001:12::/64 [90/2172416]
     via FE80::C205:EFF:FE64:0, Serial0/0
C   2001:23::/64 [0/0]
     via ::, Serial0/0
L   2001:23::3/128 [0/0]
     via ::, Serial0/0
C   2001:33::/64 [0/0]
     via ::, FastEthernet0/0
L   2001:33::3/128 [0/0]
     via ::, FastEthernet0/0
L   FF00::/8 [0/0]
     via ::, Null0
R3#
```

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "eigrp 100"
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Interfaces:
    FastEthernet0/0
    Serial0/0
  Redistribution:
    None
  Maximum path: 16
  Distance: internal 90 external 170
```