Computer Networks (CN)

key terms (chronologically cfr. the study GPS)

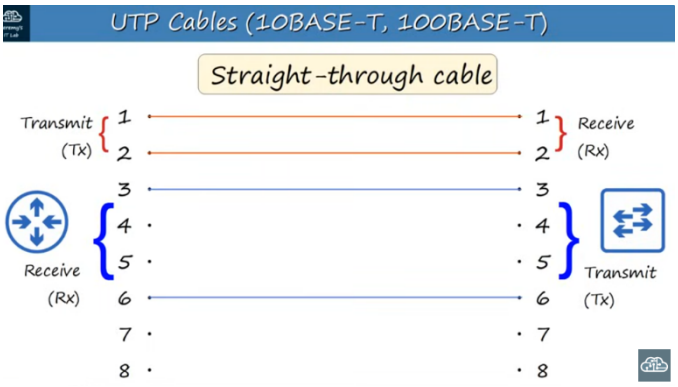
1. Network devices

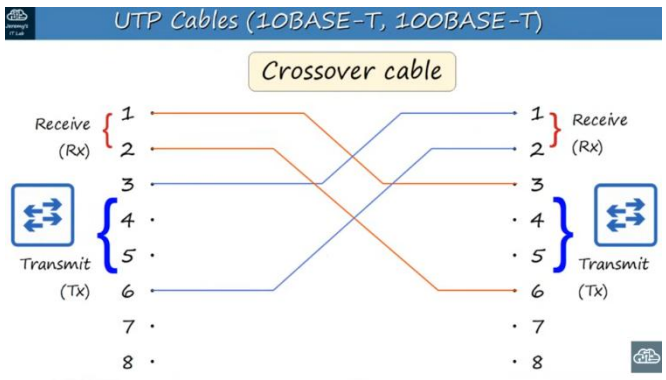
Display computer network info via command prompt:
ipconfig /all

2. Interfaces and cables

Ethernet Standards (copper)

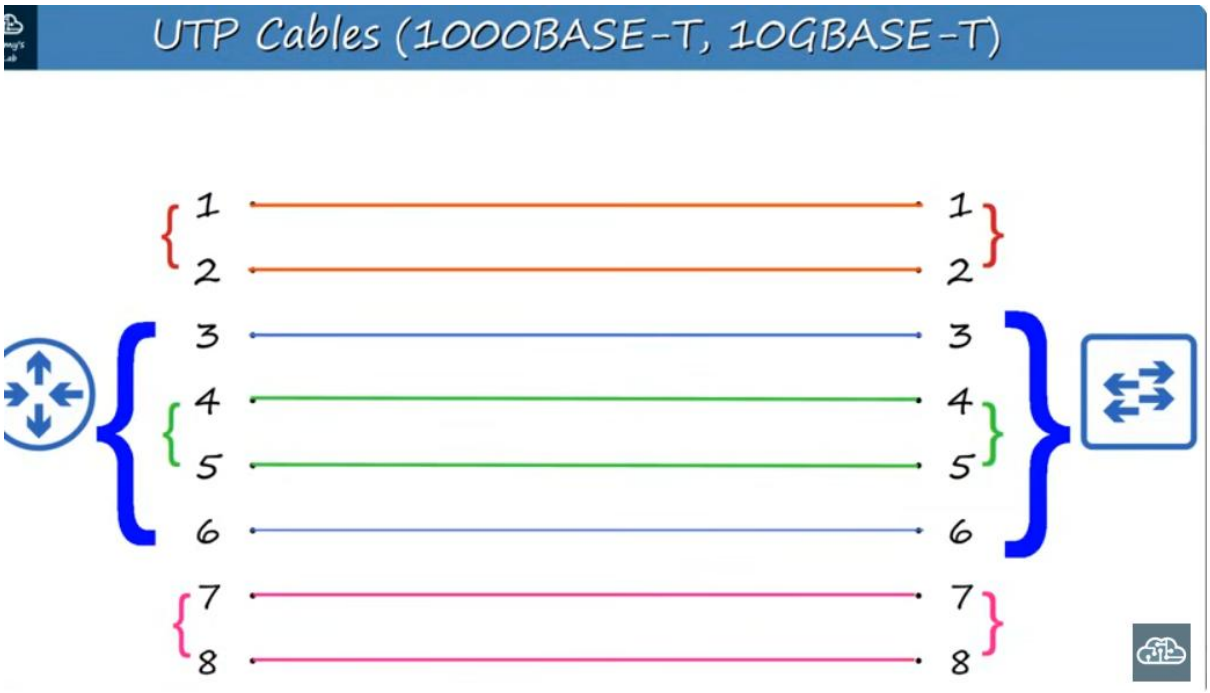
Speed	Common Name	IEEE Standard	Informal Name	Maximum Length
10 Mbps	Ethernet	802.3i	10BASE-T	100 m
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100 m
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100 m
10 Gbps	10 Gig Ethernet	802.3an	10GBASE-T	100 m





Device Type	Transmit (Tx) Pins	Receive (Rx) Pins
Router	1 and 2	3 and 6
Firewall	1 and 2	3 and 6
PC	1 and 2	3 and 6
Switch	3 and 6	1 and 2

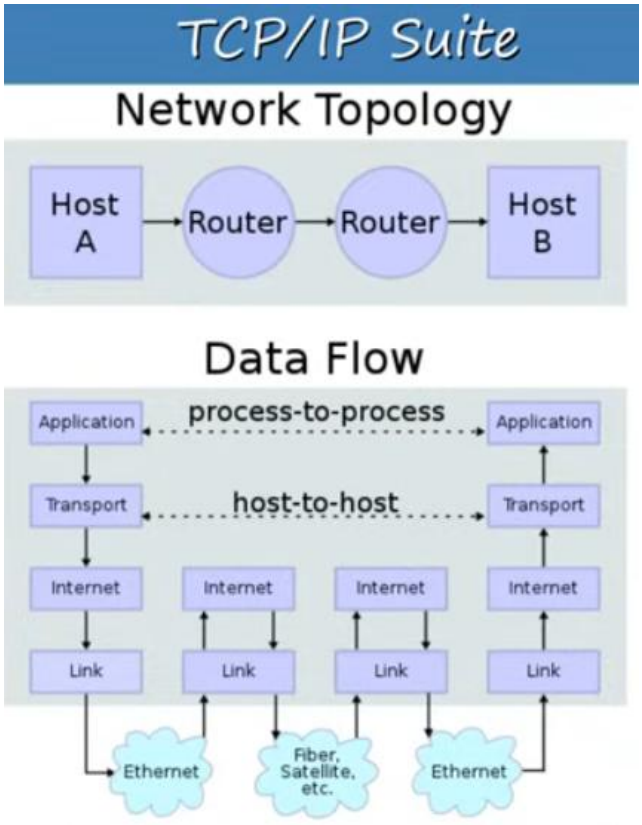
But nowadays Auto MDI-X 😊



Fiber-Optic Cable Standards

Informal Name	IEEE Standard	Speed	Cable Type	Maximum Length
1000BASE-LX	802.3z	1 Gbps	Multimode or Single-Mode	550 m (MM) 5 km (SM)
10GBASE-SR	802.3ae	10 Gbps	Multimode	400 m
10GBASE-LR	802.3ae	10 Gbps	Single-Mode	10 km
10GBASE-ER	802.3ae	10 Gbps	Single-Mode	30 km

3. OSI Model and TCP/IP Suite



Routers do not know about the upper layers! They just send lower layer segments through. The end hosts however do → same layer interaction! The transport layer has not been changed through complete process, as if sent directly → host-to-host.

4. Intro to the CLI

Router>enable User EXEC mode
Router#configure terminal Priviliged EXEC mode
Router(config)#exit Global configuration mode
Router#disable
Router>exit
… CLI closed …

Save running-config after restart:
Router#copy running-config startup-config

Setup password:
Router(config)#enable secret password

- ➔ this will secure privileged exec mode!
- ➔ to also enable this on console or vty access: use the **login** command in config. terminal

(To simple secure console login with **enable password** password command in consol line 0.)

Set hostname:
Router(config)#hostname R1

Set banner:
R1(config)#banner motd # message **#**

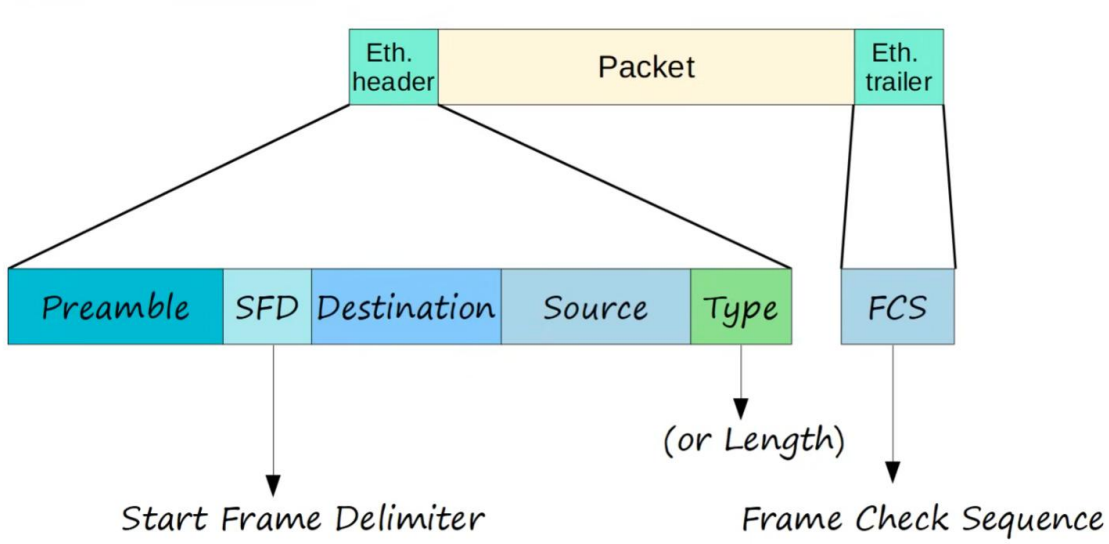
User ctrl. + shift + 6 to interrupt an active process on Cisco device.

5. /

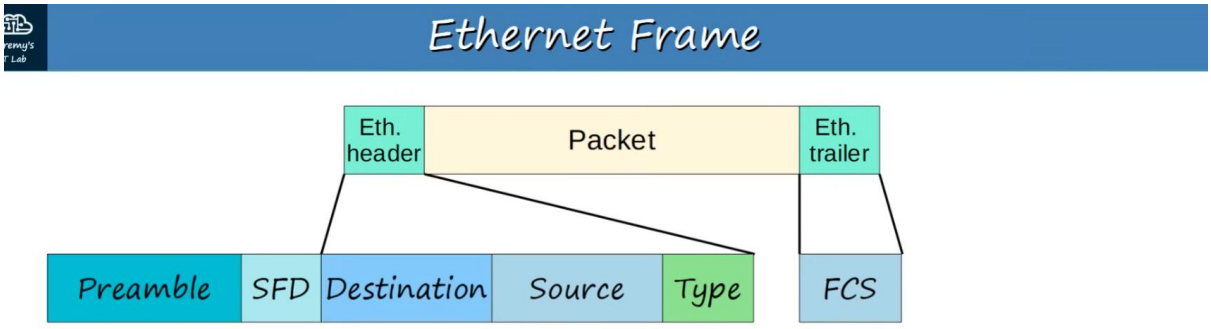
6. Ethernet LAN Switching (part 1)

All PDUs:

Frame:



7. Ethernet LAN Switching (part 2)



- The minimum size for an Ethernet frame (Header + Payload [Packet] + Trailer) is **64 bytes**
- **64 bytes – 18 bytes** (header + trailer size) = 46 bytes
- Therefore the minimum payload (packet) size is 46 bytes
- If the payload is **less than 46 bytes**, padding bytes are added
- ie. 34-byte packet + 12-byte padding = 46 bytes

Padding bytes are all zeros.

ARP

- ARP stands for 'Address Resolution Protocol'
- ARP is used to discover the Layer 2 address (MAC address) of a known Layer 3 address (IP address)
- Consists of two messages:

  ARP Request

  ARP Reply

- **ARP Request** is **broadcast** = sent to all hosts on the network
- **ARP Reply** is **unicast** = sent only to one host (the host that sent the request)

ARP Table

```
C:\Users\user>arp -a

Interface: 169.254.146.29 --- 0x9
  Internet Address      Physical Address      Type
  169.254.255.255       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

Interface: 192.168.0.167 --- 0xd
  Internet Address      Physical Address      Type
  192.168.0.1           98-da-c4-dd-a8-e4     dynamic
  192.168.0.255         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
```

- Use `arp -a` to view the ARP table (Windows, macOS, Linux)
- `Internet Address` = IP address (Layer 3 address)
- `Physical Address` = MAC address (Layer 2 address)
- `Type static` = default entry
- `Type dynamic` = learned via ARP

Ping

- A network utility that is used to test reachability
- Measures round-trip time
- Uses two messages:

  ICMP Echo Request

  ICMP Echo Reply

- Command to use ping: **ping** *(ip-address)*

<u>Have a look at MAC address table on a switch:</u>

SW1#show mac address-table

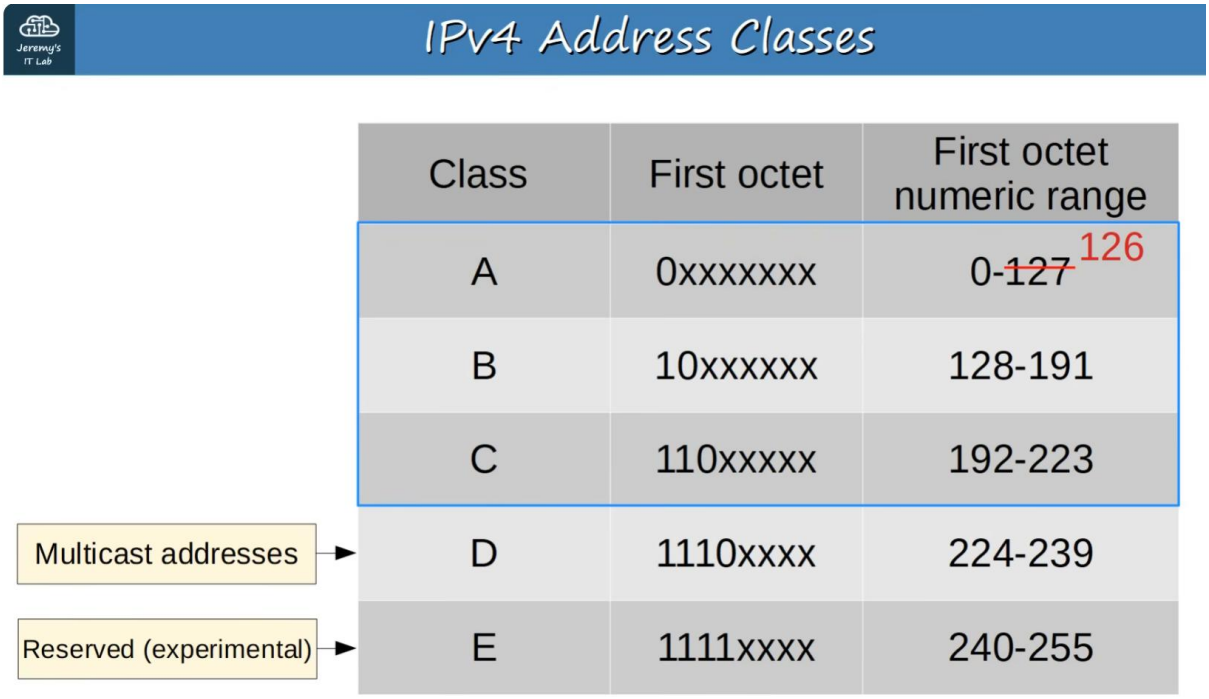
→ after 5 minutes of not being used, a MAC address dynamically learned will erase itself

SW1#clear mac address-table dynamic

→ clear all

SW1#clear mac address-table dynamic {address MAC address **| interface** g0/0**}**
→ clear specific address or interface

8. IPv4 Addressing (part 1)



127 range (127.0.0.0 → 127.255.255.255) is reserved for loopback addresses which test own ping.

A: /8, B: /16, C: /24 thus class A is for large companies with lots of end hosts, while C is for small companies!

If dest. broadcast IP address is used, dest. MAC address will be FFFF.FFFF.FFFF !

9. IPv4 Addressing (part 2)

View all IP addresses on device interfaces:
R1#show ip interface brief

Default status of interface is administratively down. Enable them with the **no shutdown** command.
This is layer 1.
The protocol line is layer 2.

Configure an IP address on an interface:
R1(config-if)#ip address 10.255.255.254 255.0.0.0
R1(config)if)#no shutdown

More details about interface:
R1#show interfaces g0/0

Enter description on an IF:
R1(config-if)#description ## message **##**

R1#show interfaces description
→ have a look at all IF descriptions

10. Switch Interfaces

Also same command to overview interfaces on a switch:
SW1#show ip interface brief

Get more details:
SW1#show interface status
➔ Speed, duplex type, status, description

Adjust interface settings:
SW1(config-if)#speed ?
SW1(config-if)#duplex ?
SW1(config-if)#description ## message **##**

Select multiple interfaces:
SW1(config)#interface range g0/0-5

Router interfaces have the `shutdown` command applied by default
=will be in the `administratively down/down` state by default

Switch interfaces do NOT have the '`shutdown`' command applied by default
=will be in the `up/up` state if connected to another device
OR
in the `down/down` state if not connected to another device

- ***Half duplex***: *The device cannot send and receive data at the same time. If it is receiving a frame, it must wait before sending a frame.*

- ***Full duplex***: *The device* ***can*** *send and receive data at the same time. It does not have to wait.*

Speed/Duplex Autonegotiation

- *Interfaces that can run at different speeds (10/100 or 10/100/1000) have default settings of* **`speed auto`** *and* **`duplex auto`**.

- *Interfaces 'advertise' their capabilities to the neighboring device, and they negotiate the best* **`speed`** *and* **`duplex`** *settings they are both capable of.*

Speed/Duplex Autonegotiation

- What if autonegotiation is disabled on the device connected to the switch?

- **SPEED**: The switch will try to sense the speed that the other device is operating at.
  If if fails to sense the speed, it will use the slowest supported speed (ie. 10 Mbps on a 10/100/1000 interface)

- **DUPLEX**: If the speed is 10 or 100 Mbps, the switch will use half duplex.
  If the speed is 1000 Mbps or greater, use full duplex.

Thus if switch senses 100 Mbps, it will also use half duplex. If that PC has full duplex? MISMATCH with collision… USE AUTO NEGOTIATION ON ALL DEVICES!

11. IPv4 Header

IPv4 Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

12. Static Routing

Show routing table:
R1#show ip route

Connected route = the network the interface is connected to.

Local route = the actual IP address on the interface (with a /32 mask)

Configure static route:
R1(config)#ip route destination-address mask [next-hop | exit-interface]
➔ next-hop is another IP address

Remove IP route from routing table:
R1(config)#no ip route destination-address mask

13. The Life of a Packet

/

14. Subnetting (part 1)

Amount of usable addresses given a certain mask. Number of host bits is 32 – mask.
E.g. /27 → 5 host bits

$$2^n - 2 = \text{usable addresses}$$

$$n = \text{number of host bits}$$

Cheat sheet:

Subnets/Hosts (Class C)

Prefix Length	Number of Subnets	Number of Hosts
/25	2	126
/26	4	62
/27	8	30
/28	16	14
/29	32	6
/30	64	2
/31	128	0 (2)
/32	256	0 (1)

Subnets/Hosts (Class B)

Prefix Length	Number of Subnets	Number of Hosts
/17	2	32766
/18	4	16382
/19	8	8190
/20	16	4094
/21	32	2044
/22	64	1022
/23	128	510
/24	256	254

Prefix Length	Number of Subnets	Number of Hosts
/25	512	126
/26	1024	62
/27	2048	30
/28	4096	14
/29	8192	6
/30	16384	2
/31	32768	0 (2)
/32	65536	0 (1)

15. Subnetting (part 2)

/

16. Subnetting (part 3 -VLSM)

When VLMS you need to define this for every subnet:

Network address:

Broadcast address:

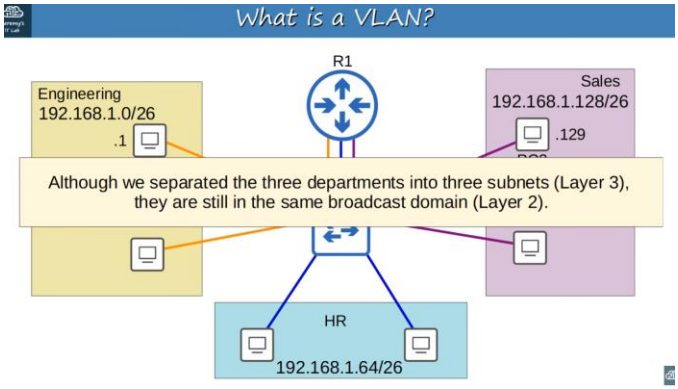
First usable address:

Last usable address:

Total number of usable host addresses:

Show info of a specific interface:
R1#show ip interface g0/0

17. VLANs (part 1)



What is a VLAN?

VLANs...

- are configured on switches on a **per-interface** basis.
- **logically** separate end hosts at Layer 2.

Switches do not forward traffic directly between hosts in different VLANs.

Show VLANs on a switch:

SW1#show vlan brief

→ by default all interfaces in vlan1

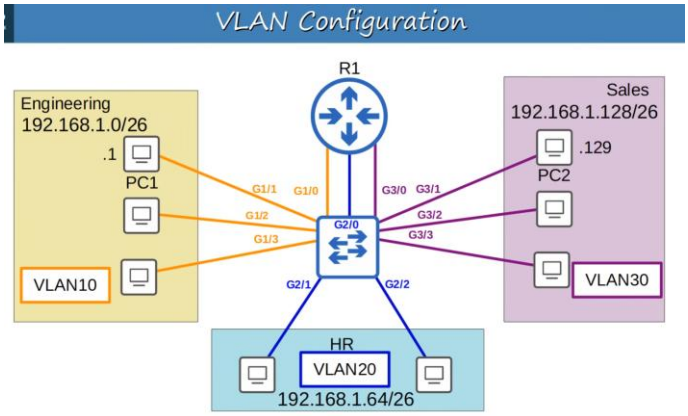
VLAN Configuration

```
SW1(config)#interface range g1/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
SW1(config-if-range)#interface range g2/0 - 2
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 20
% Access VLAN does not exist. Creating vlan 20
SW1(config-if-range)#interface range g3/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 30
% Access VLAN does not exist. Creating vlan 30
SW1(config-if-range)#
```

An access port is a switchport which belongs to a single VLAN, and usually connects to end hosts like PCs.

Switchports which carry multiple VLANs are called 'trunk ports'.

This example is NOT using trunk ports.




```
SW1(config)#vlan 10
SW1(config-vlan)#name ENGINEERING
SW1(config-vlan)#vlan 20
SW1(config-vlan)#name HR
SW1(config-vlan)#vlan 30
SW1(config-vlan)#name SALES
```

18. VLANs (part 2)

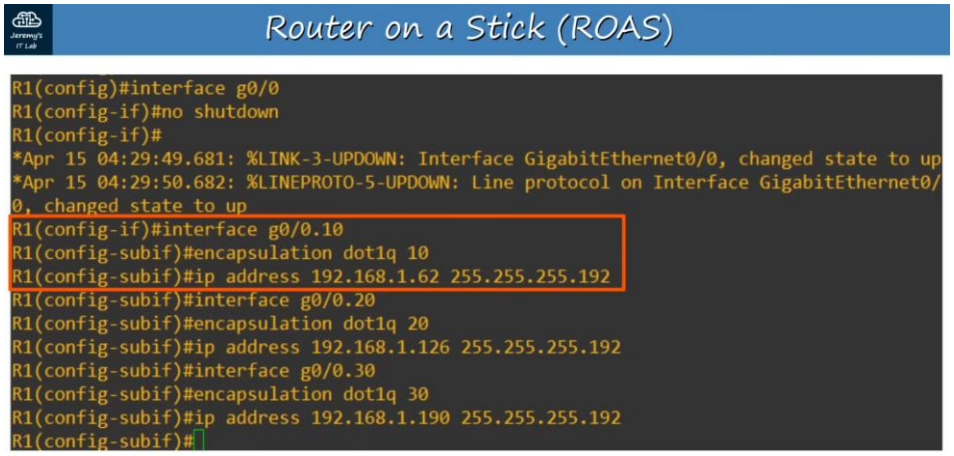
SW1(config-if-range)#switch mode access vs. **SW1(config-if)#switch mode trunk**

Set interface as a trunk:
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,30
SW1(config-if)#switchport trunk allowed vlan add 20
SW1(config-if)#switchport trunk allowed vlan ?

Set native vlan:
SW1(config-if)#switchport trunk native vlan number
→ make sure native vlans between connected switches match!

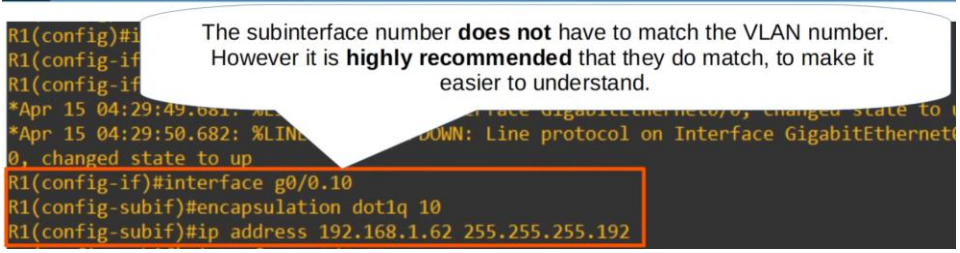
View trunk interfaces:
SW1#show interfaces trunk

Set router on a stick (router interface connected to switch interface with multiple vlans):



Number after dot1q is the vlan number which the subinterface connects to.

Router on a Stick (ROAS)



Router on a Stick (ROAS)

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES NVRAM  up                    up
GigabitEthernet0/0.10      192.168.1.62    YES manual up                    up
GigabitEthernet0/0.20      192.168.1.126   YES manual up                    up
GigabitEthernet0/0.30      192.168.1.190   YES manual up                    up
GigabitEthernet0/1         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/3         unassigned      YES NVRAM  administratively down down
```

Router on a Stick (ROAS)

- ROAS is used to route between multiple VLANs using a single interface on the router and switch.
- The switch interface is configured as a regular trunk.
- The router interface is configured using **subinterfaces**. You configure the VLAN tag and IP address on each subinterface.
- The router will behave as if frames arriving with a certain VLAN tag have arrived on the subinterface configured with that VLAN tag.
- The router will tag frames sent out of each subinterface with the VLAN tag configured on the subinterface.

19. VLANs (part 3)

Native VLAN on a router (ROAS)

- There are **2 methods** of configuring the native VLAN on a router:

  -Use the command **encapsulation dot1q** *vlan-id* **native** on the router subinterface.

```
R1(config)#int g0/0.10
R1(config-subif)#encapsulation dot1q 10 native
R1(config-subif)#
```

Option 2:

```
R1(config)#no interface g0/0.10
R1(config)#interface g0/0
R1(config-if)#ip address 192.168.1.62 255.255.255.192
R1(config-if)#
```

-Configure the IP address for the native VLAN on the router's physical interface (the **encapsulation dot1q** *vlan-id* command is not necessary)

Option 2:

```
!
interface GigabitEthernet0/0
 ip address 192.168.1.62 255.255.255.192
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/0.20
 encapsulation dot1Q 20
 ip address 192.168.1.126 255.255.255.192
!
interface GigabitEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.1.190 255.255.255.192
!
```

The g0/0.10 is gone, but used as the port with native vlan. The other subinterfaces for the other (not native) vlans still needed to add the dot1q tag in the Ethernet header.

20.STP (part 1)

1) One switch is elected as the root bridge. All ports on the root bridge are **designated ports** (forwarding state). Root bridge selection:
   1: Lowest bridge ID

2) Each remaining switch will select ONE of its interfaces to be its **root port** (forwarding state). Ports across from the root port are always **designated** ports.
   Root port selection:
   1: Lowest root cost
   2: Lowest neighbor bridge ID
   3: Lowest neighbor port ID

3) Each remaining collision domain will select ONE interface to be a **designated port** (forwarding state). The other port in the collision domain will be **non-designated** (blocking)
   Designated port selection:
   1: Interface on switch with lowest root cost
   2: Interface on switch with lowest bridge ID

21.STP (part 2)

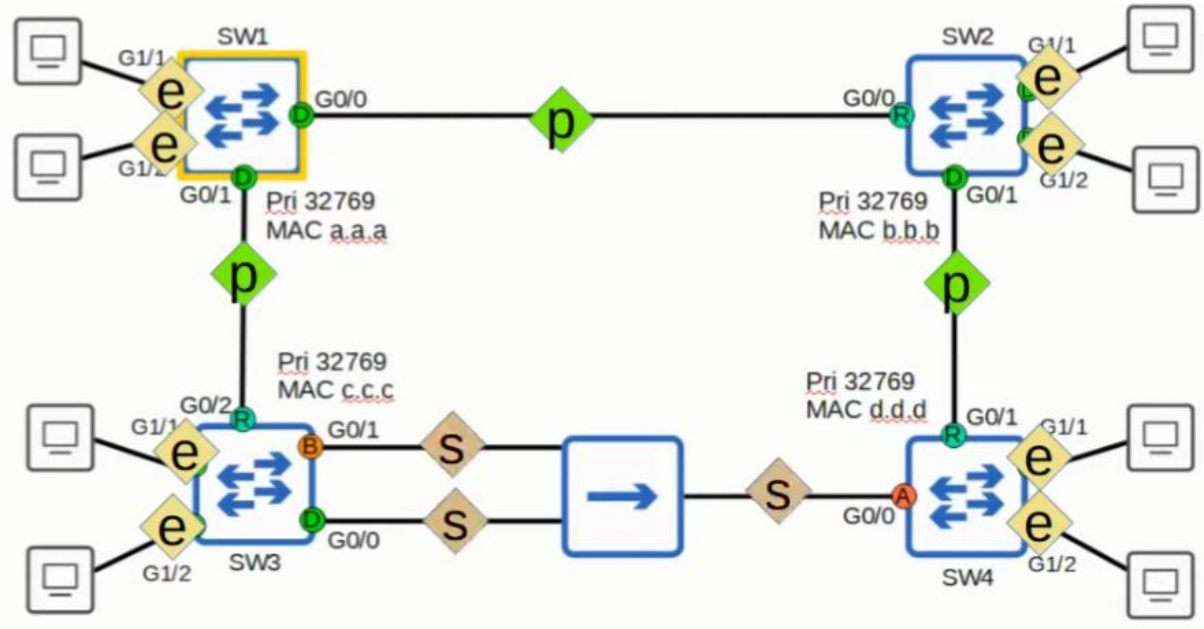
STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Blocking	NO/YES	NO	NO	Stable
Listening	YES/YES	NO	NO	Transitional
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable
Disabled	NO/NO	NO	NO	Stable

STP Timer	Purpose	Duration
Hello	How often the root bridge sends hello BPDUs	2sec
Forward delay	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
Max Age	How long an interface will wait after ceasing to receive Hello BDPUs to change the STP topology.	20sec (10* hello)

22.Rapid STP

Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	X	200
1 Tbps	X	20

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable



Summary of rapid STP with all interface states and RSTP link types.

Hubs are fully replaced by switches now, thus **backup port state** and **shared links** will *not* be seen in real life anymore.

23. EtherChannel

```
SW(config) port-channel load-balance mode
 #configures the EtherChannel load-balancing method on the switch

SW# show etherchannel load-balance
 #displays information about the load-balancing settings

SW(config-if)# channel-group number mode {desirable|auto|active|passive|on}
  #configures an interface to be part of an EtherChannel

SW# show etherchannel summary
  #displays a summary of EtherChannels on the switch

SW# show etherchannel port-channel
  #displays information about the virtual port-channel interfaces on the switch
```

24. Dynamic routing

IGP	Metric	Explanation
RIP	Hop count	Each router in the path counts as one 'hop'. The total metric is the total number of hops to the destination. **Links of all speeds are equal.**
EIGRP	Metric based on bandwidth & delay (by default)	Complex formula that can take into account many values. By default, the bandwidth of the **slowest link in the route** and the total delay of all links in the route are used.
OSPF	Cost	The cost of each link is calculated based on bandwidth. The total metric is the total cost of each link in the route.
IS-IS	Cost	The total metric is the total cost of each link in the route. The cost of each link is **not** automatically calculated by default. All links have a cost of 10 by default.

RIP very primitive as it both counts Gigabitethernet and slower links as 'one hop'.

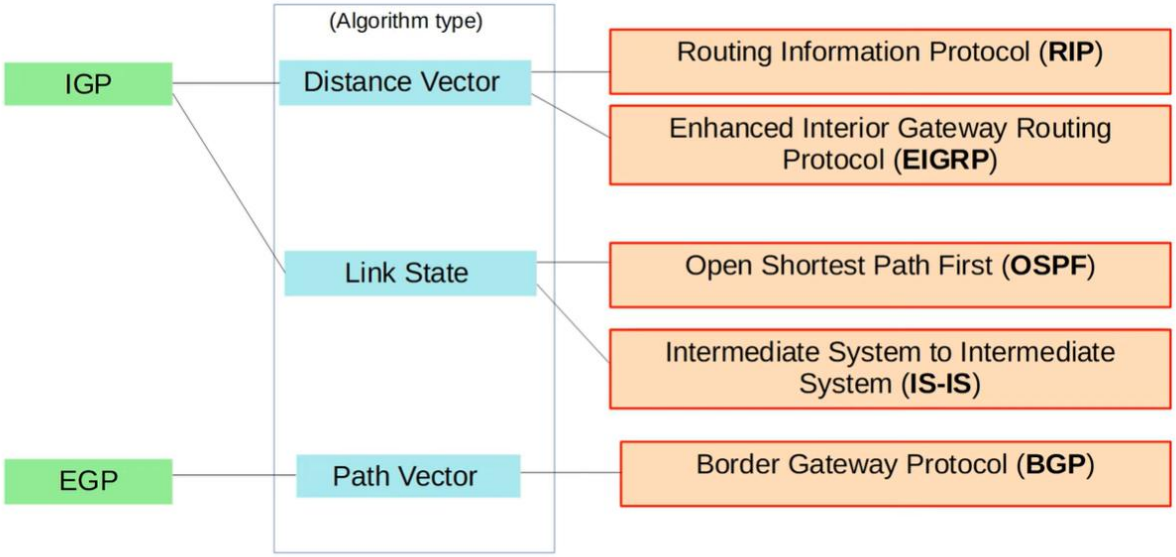
You cannot compare different protocol routes, because they use different metrics (appels en peren).

Route protocol/type	AD
Directly connected	0
Static	1
External BGP (eBGP)	20
EIGRP	90
IGRP	100
OSPF	110

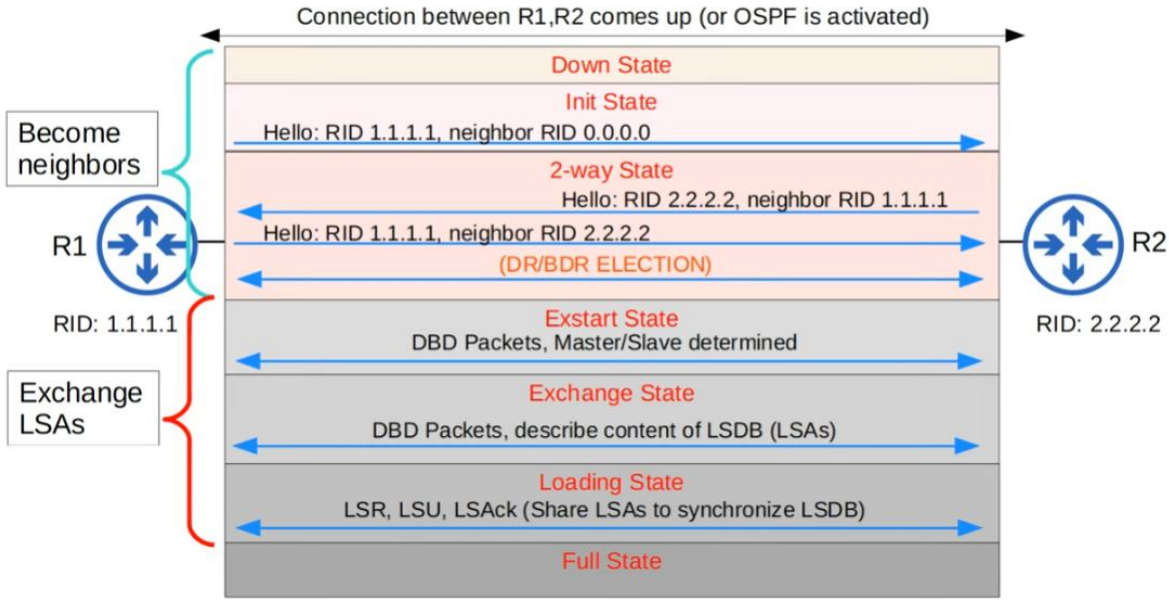
Route protocol/type	AD
IS-IS	115
RIP	120
EIGRP (external)	170
Internal BGP (iBGP)	200
Unusable route	255

Lower administrative distance are preferred over higher AD. Might differ across brands of network devices!

25. OSPF (part 1)



26. OSPF (part 2)



Type	Name	Purpose
1	**Hello**	Neighbor discovery and maintenance.
2	**Database Description (DBD)**	Summary of the LSDB of the router. Used to check if the LSDB of each router is the same.
3	**Link-State Request (LSR)**	Requests specific LSAs from the neighbor.
4	**Link-State Update (LSU)**	Sends specific LSAs to the neighbor.
5	**Link-State Acknowledgement (LSAck)**	Used to acknowledge that the router received a message.

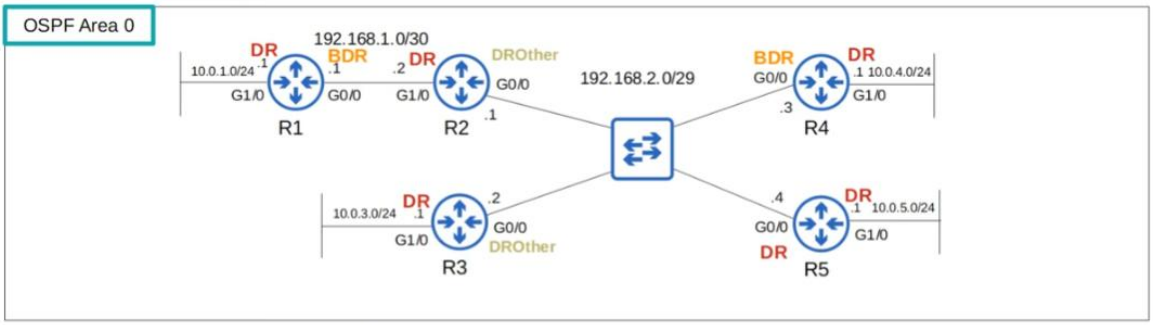
R1#show ip ospf neighbor
R1#show ip ospf interface g0/0

Configure IF as OSPF IF
R1(config)#int g0/0
R1(config-if)#ip ospf *process-id* **area** *area*

OR

Configure all IF as OSPF passive IF
R1#(config)#router ospf 1
R1(config-router)#passive-interface default
Then select the active ones
R1(config-router)#no passive-interface g0/0

27. OSPF (part 3)



- The DR/BDR election order of priority:
  1: Highest **OSPF interface priority**
  2: Highest OSPF Router ID

- 'First place' becomes the DR for the subnet, 'second place' becomes the BDR

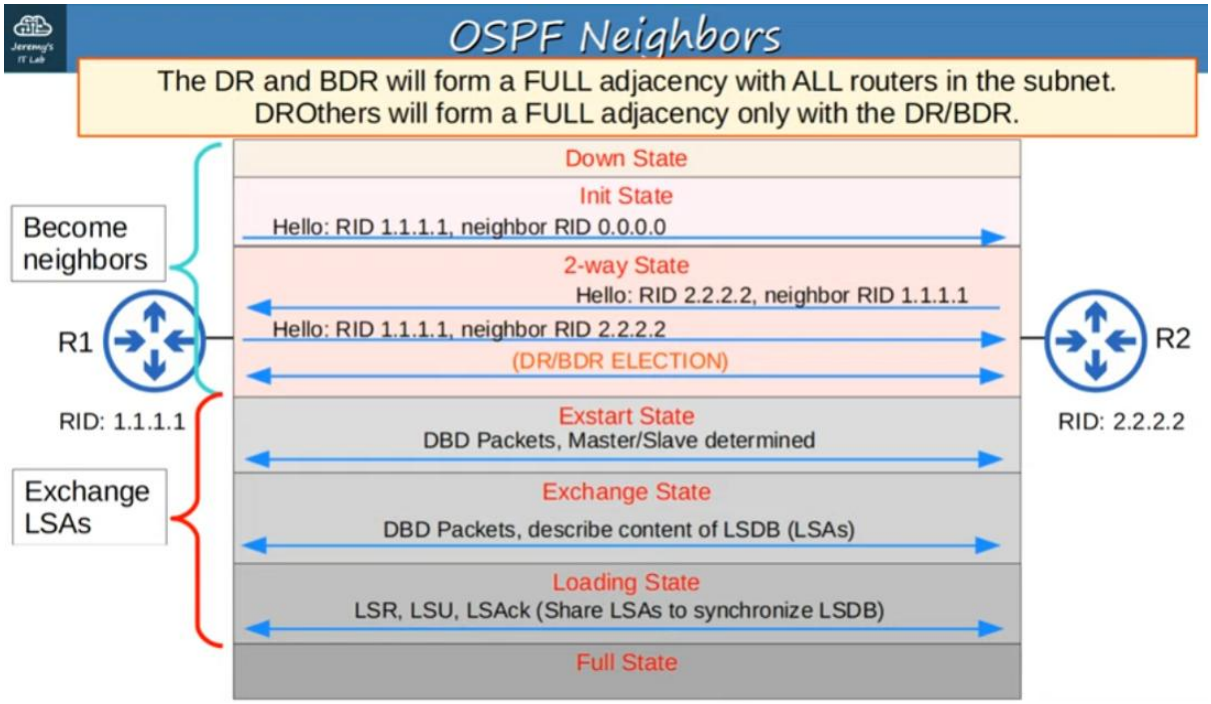
- The default OSPF interface priority is 1 on all interfaces

Set OSPF priority

R1(config)#int g0/0

R1(config-ig)#ip ospf priority 255

➔ Caveat, you need to reset to take effect (or do beforehand). Then first the BDR becomes DR. Thus actually two resets needed to go from DROther to BDR to DR.



R1#show ip ospf interface brief

R1#show ip ospf interface g0/0

R1#show ospf neighbor

OSPF Neighbor Requirements

1) Area number must match

2) Interfaces must be in the same subnet

3) OSPF process must not be **shutdown**

4) OSPF Router IDs must be unique

5) Hello and Dead timers must match

6) Authentication settings must match

7) IP MTU settings must match

8) OSPF Network Type must match.

Can become OSPF neighbors, but OSPF doesn't operate properly.

Get LSDB (which is the same on each router within a OSPF area)

R1#show ip ospf database

28. First Hop Redundancy Protocols

Comparing FHRPs

FHRP	Terminology	Multicast IP	Virtual MAC	Cisco proprietary?
HSRP	Active/Standby	v1: 224.0.0.2 v2: 224.0.0.102	v1: 0000.0c07.acXX v2: 0000.0c9f.fXXX	Yes
VRRP	Master/Backup	224.0.0.18	0000.5e00.01XX	No
GLBP	AVG / AVF	224.0.0.102	0007.b400.XXYY	Yes

eg. group 200 is 0xc8 in decimal

Configure HSRP:

R1(config)#interface g0/0

(R1(config-if)#standby version 2 **) if you want to use version 2**

R1(config-if)#standby ?

```
R1(config-if)#standby 1 ip 172.16.0.254
R1(config-if)#
R1(config-if)#standby 1 priority ?
  <0-255>  Priority value

R1(config-if)#standby 1 priority 200
R1(config-if)#
R1(config-if)#standby 1 preempt
R1(config-if)#
```

The **active router** is determined in this order:
1 – Highest priority (default 100)
2 – Highest IP address

Preempt causes the router to take the role of active router, even if another router already has the role.

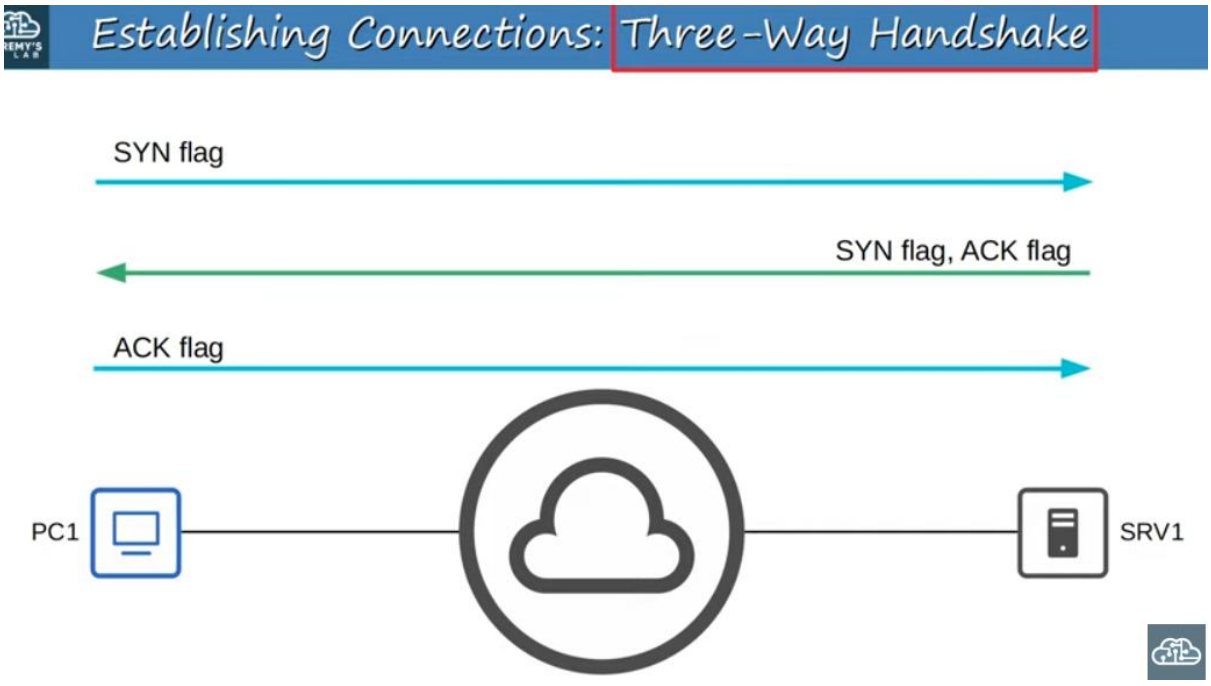
Only necessary on the router you want to become active

```
R2(config-if)#standby version 2
R2(config-if)#
R2(config-if)#standby 1 ip 172.16.0.254
R2(config-if)#
R2(config-if)#standby 1 priority 50
R2(config-if)#
R2(config-if)#standby 1 preempt
```

HSRP version 1 and version 2 are not compatible.
If R1 uses version 2, R2 must use version 2 also.

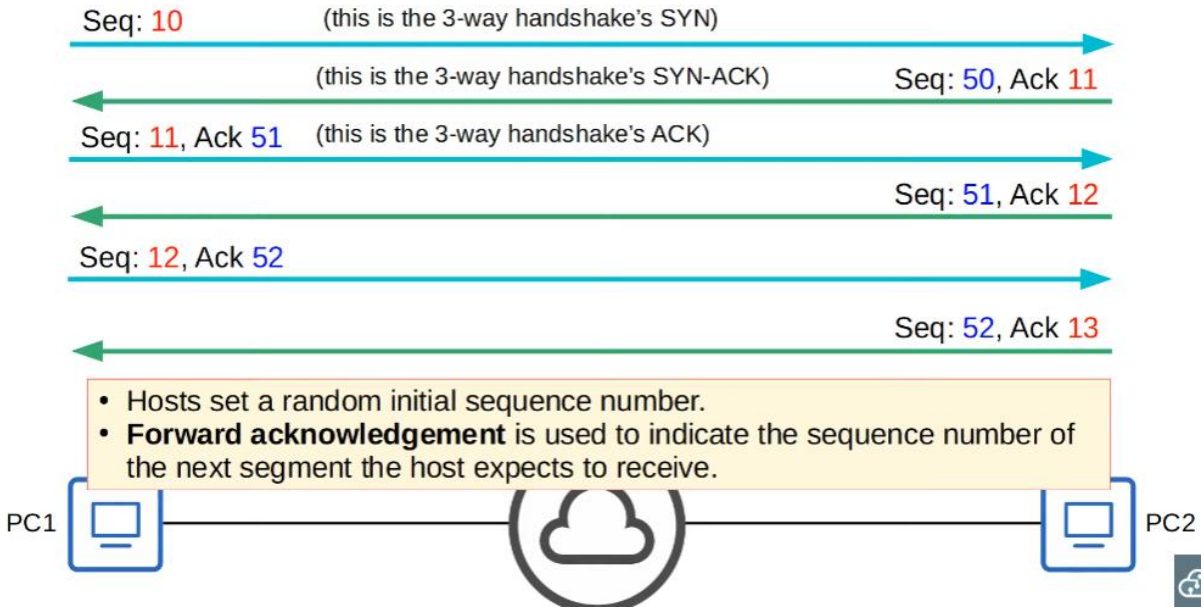
Last two red rectangles are unnecessary, because R1 higher priority anyways and pre-empt only needed on active router!

29. TCP & UDP



Terminating Connections: Four-Way Handshake

TCP: Sequencing / Acknowledgment



- Hosts set a random initial sequence number.
- **Forward acknowledgement** is used to indicate the sequence number of the next segment the host expects to receive.

Comparing TCP & UDP

TCP	UDP
Connection-oriented	Connectionless
Reliable	Unreliable
Sequencing	No sequencing
Flow control	No flow control
Use for downloads, file sharing, etc	Used for VoIP, live video, etc

BOTH use port numbers to allow session multiplexing and application layer protocols.

Port Numbers

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

30. IPv6 (part 1)

Hexadecimal

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9

Decimal	Binary	Hexadecimal
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

0b: binary. 0d: decimal. 0x: hexadecimal.

Shortening (abbreviating) IPv6 addresses

- Consecutive quartets of 0s can only be abbreviated once in an IPv6 address.

2001:0000:0000:0000:20A1:0000:0000:34BD

~~2001::20A1::34BD~~

How many quartets of 0 are here?

How many quartets of 0 are here?

2001::20A1:0:0:34BD

Finding the IPv6 prefix

Host Address	Prefix
FE80:0000:0000:0000:4c2c:e2ed:6a89:2a27/9	
2001:0DB8:0001:0B23:BA89:0020:0000:00C1/64	
2001:0DB8:0BAD:CAFE:1300:0689:9000:0CDF/71	
2001:0DB8:0000:FEED:0DAD:018F:6001:0DA3/62	
2001:0DB8:9BAD:BABE:0DE8:AB78:2301:0010/63	

Solution in video 27'20''.

Configure IPv6 address:

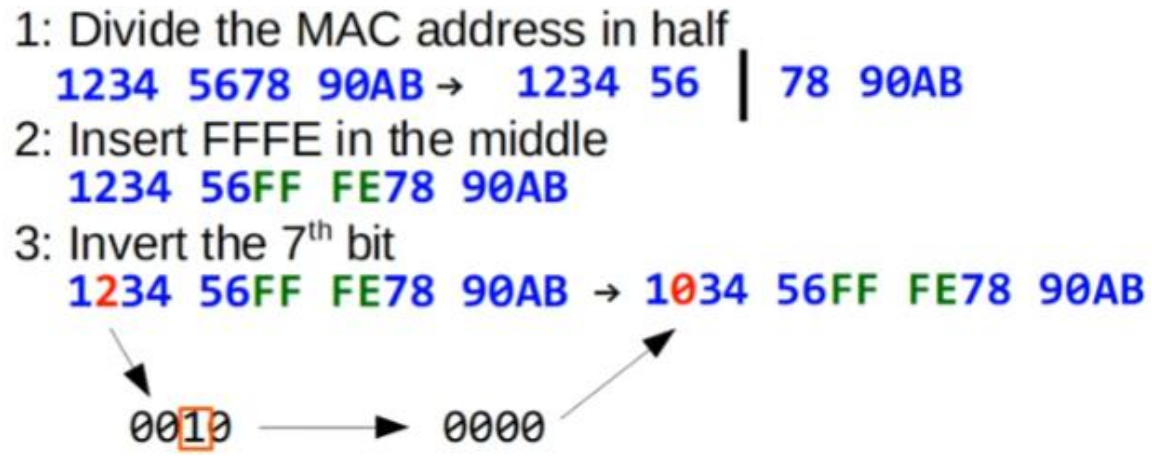
R1(config)#ipv6 unicast-routing (otherwise no IPv6 traffic can pass through!)

R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8:0:0::1/64
R1(config-if)#no shutdown

R1#show ipv6 interface brief

31. IPv6 (part 2)

EUI-64 identifier will be the host portion of the /64 IPv6 address.



R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8::/64 **eui-64**
R1(config-if)#no shutdown

Automatically generate link local address on enabled IPv6 interface:

R1(config-if)#ipv6 enable

➔ FE80::/10 with EUI-64 host portion

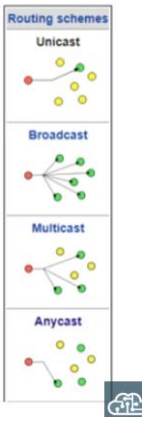
Multicast addresses

Purpose	IPv6 Address	IPv4 Address
All nodes/hosts (functions like broadcast)	FF02::1	224.0.0.1
All routers	FF02::2	224.0.0.2
All OSPF routers	FF02::5	224.0.0.5
All OSPF DRs/BDRs	FF02::6	224.0.0.6
All RIP routers	FF02::9	224.0.0.9
All EIGRP routers	FF02::A	224.0.0.10

Scopes:
FF01: node local
FF02: link local

R1#show ipv6 int g0/0

➔ **you can see which multicast scopes it joined!**



32. IPv6 (part 3)

Solicited-Node Multicast Address

- An IPv6 solicited-node multicast address is calculated from a unicast address.

`ff02:0000:0000:0000:0000:0001:ff` + `Last 6 hex digits of unicast address`

`2001:0db8:0000:0001:0f2a:4fff:fea3:00b1`

↓

`ff02::1:ffa3:b1`

`2001:0db8:0000:0001:0489:4eda:073a:12b8`

↓

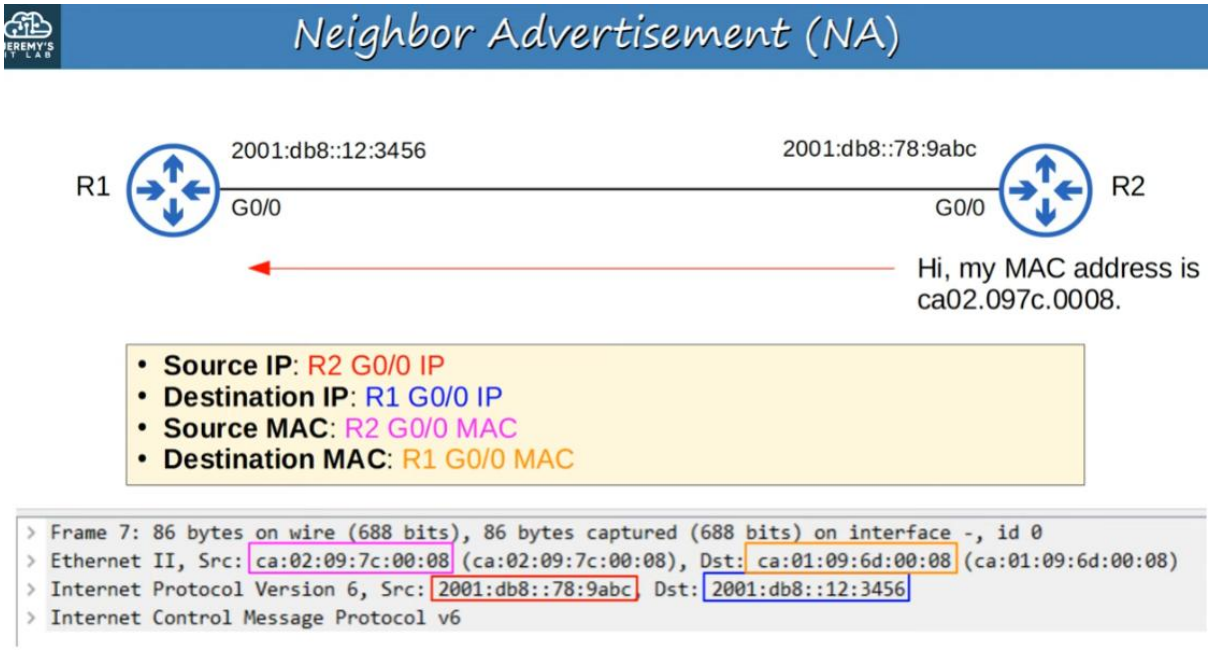
`ff02::1:ff3a:12b8`

Neighbor Solicitation (NS)

- **Source IP**: R1 G0/0 IP
- **Destination IP**: R2 solicited-node multicast address
- **Source MAC**: R1 G0/0 MAC
- **Destination MAC**: Multicast MAC based on R2's solicited-node address

```
> Frame 6: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:01:09:6d:00:08 (ca:01:09:6d:00:08), Dst: IPv6mcast_ff:78:9a:bc (33:33:ff:78:9a:bc)
> Internet Protocol Version 6, Src: 2001:db8::12:3456, Dst: ff02::1:ff78:9abc
> Internet Control Message Protocol v6
```

Know that NS is multicast, while the ARP request of IPv4 is broadcast!



Look at IPv6 neighbor table:
R1#show ipv6 neighbor

SLAAC:
R1#ipv6 address autoconfig

Routing table:
R1#show ipv6 route

Static routing:
R1(config)#ipv6 route destination/prefix-length {next-hop | exit-interface [next-hop]} [ad]
- directly attached via exit-interface (if IF is ethernet, not possible)
- recursive via next-hop
- fully specified via exit-interface AND next-hop (eg. mandatory for link-local)

Type of routes:

- Network /64
- Host /128
- Default ::/0
- Floating static

33. Standard ACLs

How ACLs work

- Configuring an ACL in global config mode will not make the ACL take effect.
- The ACL must be applied to an interface.
- ACLs are applied either inbound or outbound.
- ACLs are made up of one or more ACEs.
- When the router checks a packet against the ACL, it processes the ACEs in order, from top to bottom.
- If the packet matches one of the ACEs in the ACL, the router takes the action and stops processing the ACL. All entries below the matching entry will be ignored.

Implicit deny

- What will happen if a packet doesn't match any of the entries in an ACL?

ACL 2:
1: if source IP = 192.168.1.0/24, then permit
2: if source IP = 192.168.0.0/16, then deny
(3: if source IP = any, then deny)

- There is an 'implicit deny' at the end of all ACLs.
- The implicit deny tells the router to deny all traffic that doesn't match any of the configured entries in the ACL.

Configure standard numbered ACL:

R1(config)#access-list number {**deny** | **permit**} ip wildcard-mask

R1(config)# access-list 1 permit any

➔ **OR use R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255**

Add a note:

R1(config)#access-list 1 remark ## TEXT ##

R1(config)#do show access-lists

R1#show access-lists

Add ACL to an interface:

R1(config-if)#ip access-group number {**in** | **out**}

Named ACL:

Standard named ACLs are configured by entering 'standard named ACL config mode', and then configuring each entry within that config mode.

R1(config)# **ip access-list standard** *acl-name*
R1(config-std-nacl)# [*entry-number*] {**deny** | **permit**} *ip wildcard-mask*

```
R1(config)#ip access-list standard BLOCK_BOB

R1(config-std-nacl)#5 deny 1.1.1.1

R1(config-std-nacl)#10 permit any

R1(config-std-nacl)#remark ## CONFIGURED NOV 21 2020 ##

R1(config-std-nacl)#interface g0/0

R1(config-if)#ip access-group BLOCK_BOB in
```

34. Extended ACLs

In Day 34, you learned that numbered ACLs are configured in global config mode:

```
R1(config)# access-list 1 deny 192.168.1.1
R1(config)# access-list 1 permit any
```

You learned that named ACLs are configured with subcommands in a separate config mode:

```
R1(config)# ip access-list standard BLOCK_PC1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

However, in modern IOS you can also configure numbered ACLs in the exact same way as named ACLs:

```
R1(config)# ip access-list standard 1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

Delete ACE from ACL:

R1(config-std-nacl)#no sequence-number

When configuring/editing numbered ACLs from global config mode, you can't delete individual entries, you can only delete the entire ACL!

Thus use named ACL config mode, even with numbered ones.

Add ACE with specific sequence number:

R1(config-std-nacl)#sequence-number {**deny** | **permit**} …

Resequencing:

There is a *resequencing* function that helps edit ACLs.

The command is **ip access-list resequence** *acl-id starting-seq-num increment*

```
R1(config)#do show access-lists
Standard IP access list 1
    1 deny   192.168.1.1
    3 deny   192.168.3.1
    2 deny   192.168.2.1
    4 deny   192.168.4.1
    5 permit any
R1(config)#
R1(config)#ip access-list resequence 1 10 10
R1(config)#
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.3.1
    30 deny   192.168.2.1
    40 deny   192.168.4.1
    50 permit any
```

Extended ACL:

```
R1(config)# access-list number [permit | deny] protocol src-ip dest-ip

R1(config)# ip access-list extended {name | number}
R1(config-ext-nacl)# [seq-num] [permit | deny] protocol src-ip dest-ip
```

R1(config-ext-nacl)#deny ?

→ view all possible filter (protocol) options. Focus in videos on ip, tcp and udp.

Extended ACL entry practice (1)

1. Allow all traffic

```
R1(config-ext-nacl)#permit ip any any
```

2. Prevent 10.0.0.0/16 from sending UDP traffic to 192.168.1.1/32

```
R1(config-ext-nacl)#deny udp 10.0.0.0 0.0.255.255 host 192.168.1.1
```

3. Prevent 172.16.1.1/32 from pinging hosts in 192.168.0.0/24

```
R1(config-ext-nacl)#deny icmp host 172.16.1.1 192.168.0.0 0.0.0.255
```

Matching the TCP/UDP port numbers

- When matching TCP/UDP, you can optionally specify the source and/or destination port numbers to match.

```
R1(config-ext-nacl)#deny tcp src-ip  eq    src-port-num   dest-ip  eq    dst-port-num
                                     gt                             gt
                                     lt                             lt
                                     neq                            neq
                                     range                          range
```

- **eq 80** = equal to port 80
- **gt 80** = greater than 80 (81 and greater)
- **lt 80** = less than 80 (79 and less)
- **neq 80** = NOT 80
- **range 80 100** = from port 80 to port 100

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

Extended ACL entry practice (2)

1. Allow traffic from 10.0.0.0/16 to access the server at 2.2.2.2/32 using HTTPS.

```
R1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.255.255 2.2.2.2 0.0.0.0 eq 443
```

2. Prevent all hosts using source UDP port numbers from 20000 to 30000 from accessing the server at 3.3.3.3/32.

```
R1(config-ext-nacl)#deny udp any range 20000 30000 host 3.3.3.3
```

3. Allow hosts in 172.16.1.0/24 using a TCP source port greater than 9999 to access all TCP ports on server 4.4.4.4/32 except port 23.

```
R1(config-ext-nacl)#permit tcp 172.16.1.0 0.0.0.255 gt 9999 host 4.4.4.4 neq 23
```

Show active in and out ACL on an IF:
R1#show ip interface g0/0

35. DNS

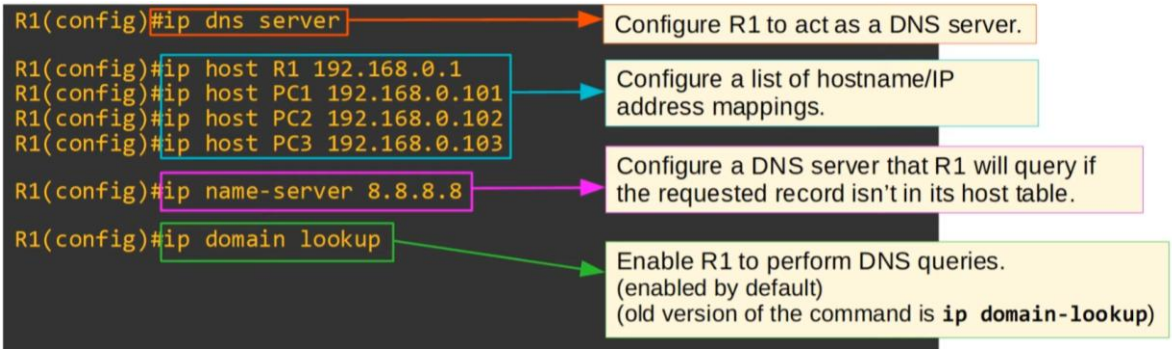
In command prompt on PC:
nslookup youtube.com

ping youtube.com (automatically looks up IP address of website first! No need for ns lookup first)

ipconfig /displaydns (show cached DNS mappings on device)

ipconfig /flushdns (remove DNS cache)

Use router as DNS server:

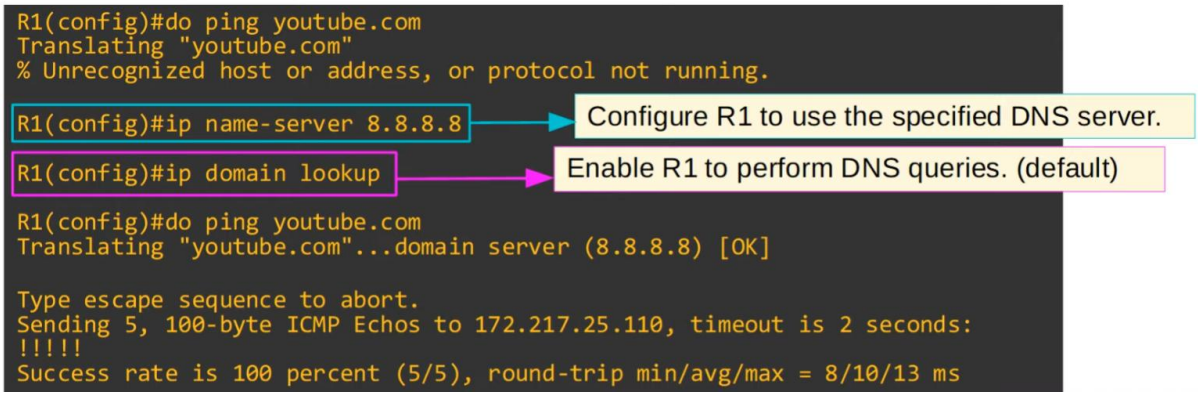


Now a host will query R1 first as DNS server instead of going on the internet, unless no match is found. Then R1 will go to name-server configured.

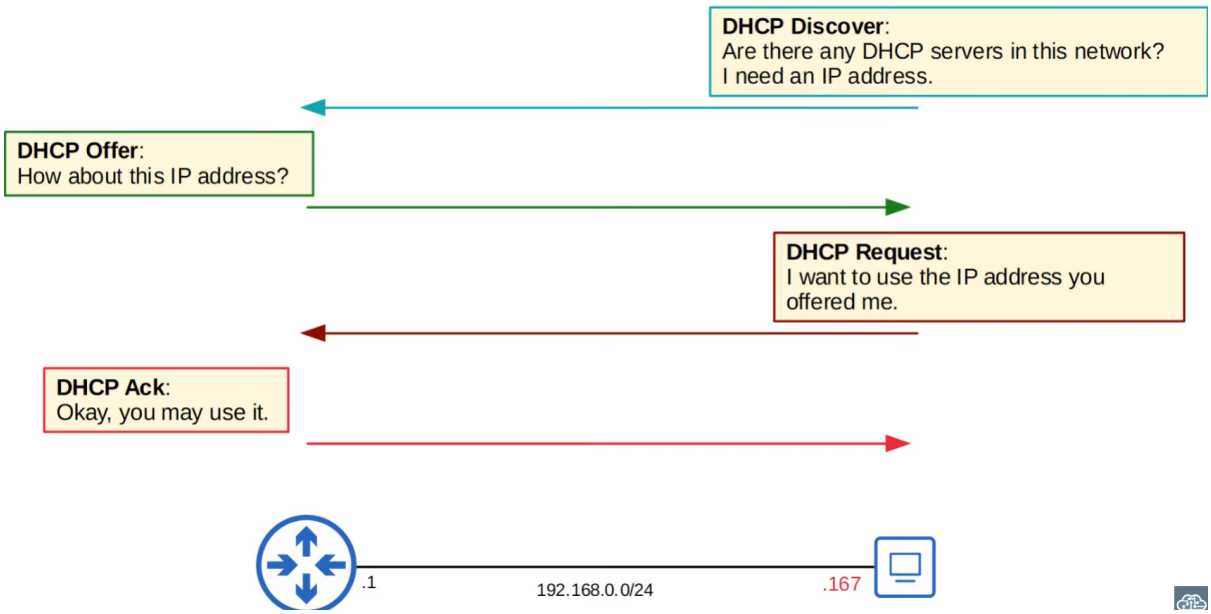
Learned mappings:

R1#show hosts

Use router as DNS client:

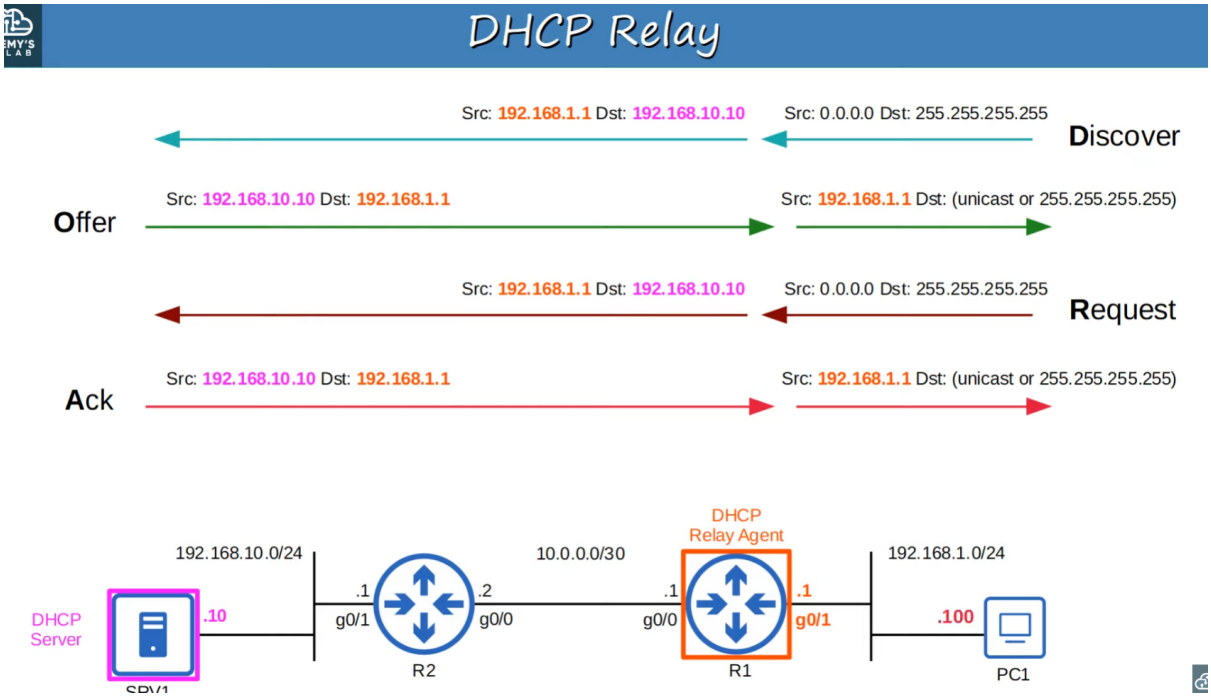


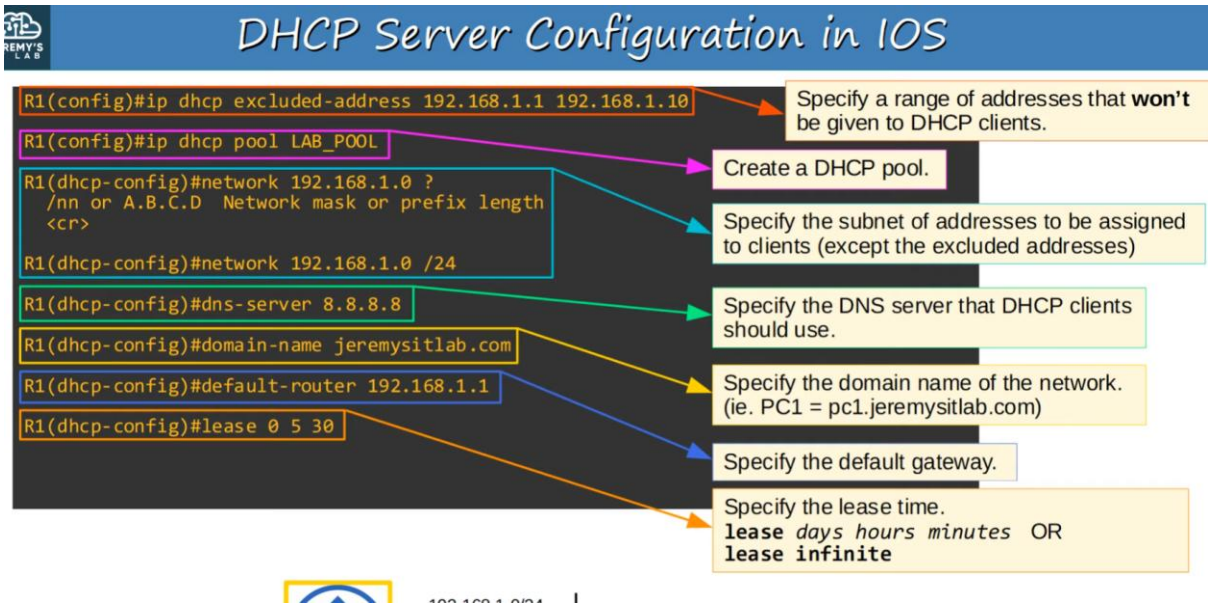
36. DHCP



DHCP D-O-R-A

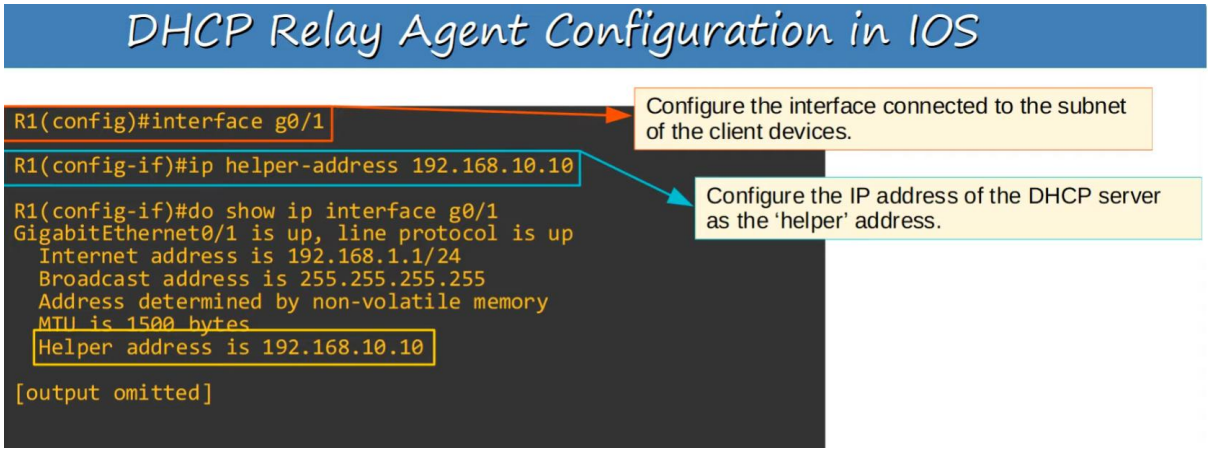
Discover	Client → Server	Broadcast
Offer	Server → Client	Broadcast or Unicast
Request	Client → Server	Broadcast
Ack	Server → Client	Broadcast or Unicast
Release	Client → Server	Unicast

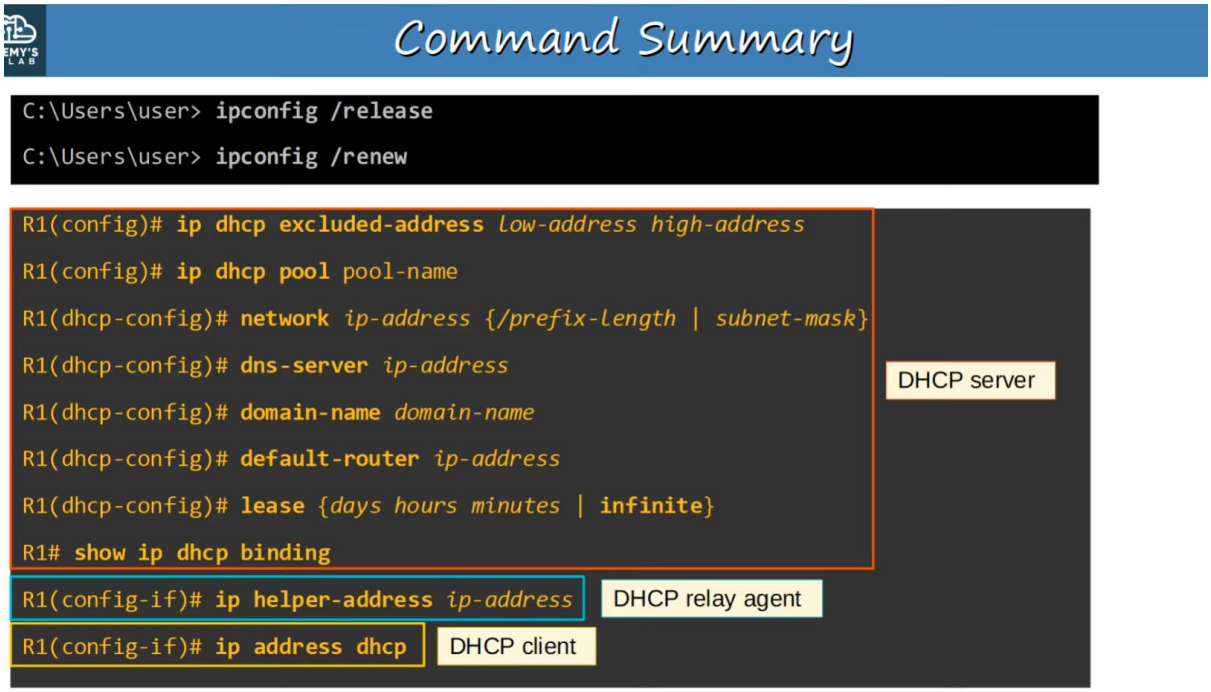




<u>To see all clients with assigned DHCP:</u>

R1#show ip dhcp binding





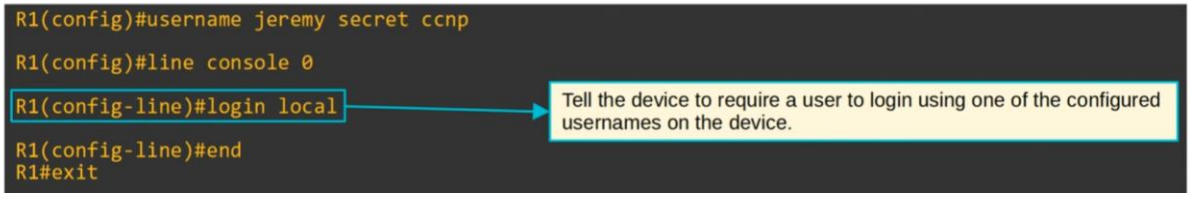
Last command tells router to learn its IP address via DHCP, just like an end host (eg. a PC) would.

37. SSH

Configure the (one and only) console line with a password before access:

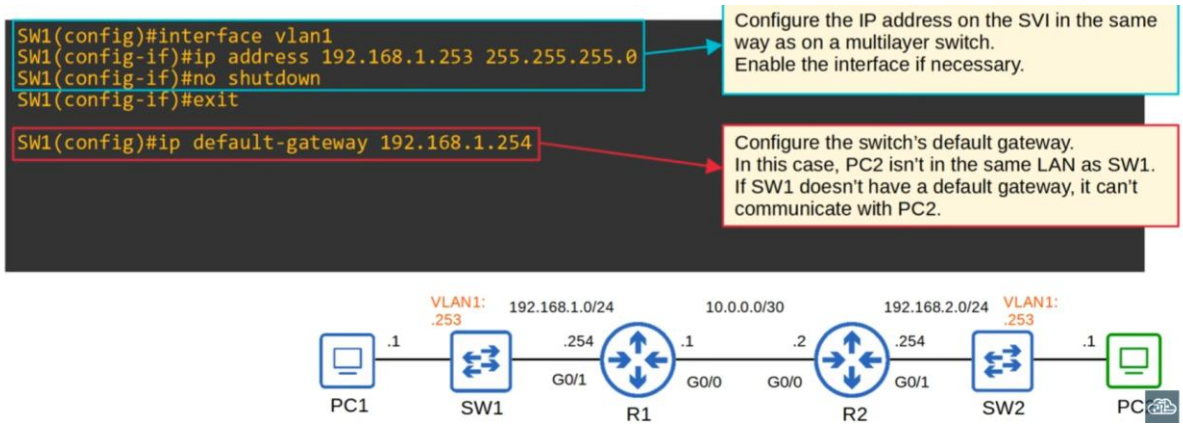
R1(config)#line console 0
R1(config-line)#password password
R1(config-line)#login
R1(config-line)#end

Use **login local** to require one of configured users with corresponding password to login.

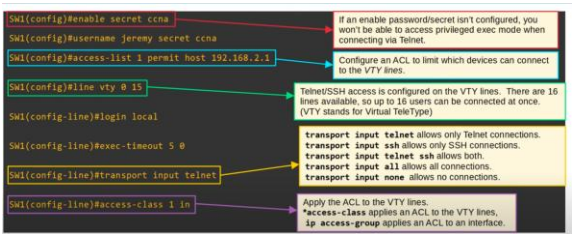


Exec-timeout will log you out automatically after inactivity.

Vlan:

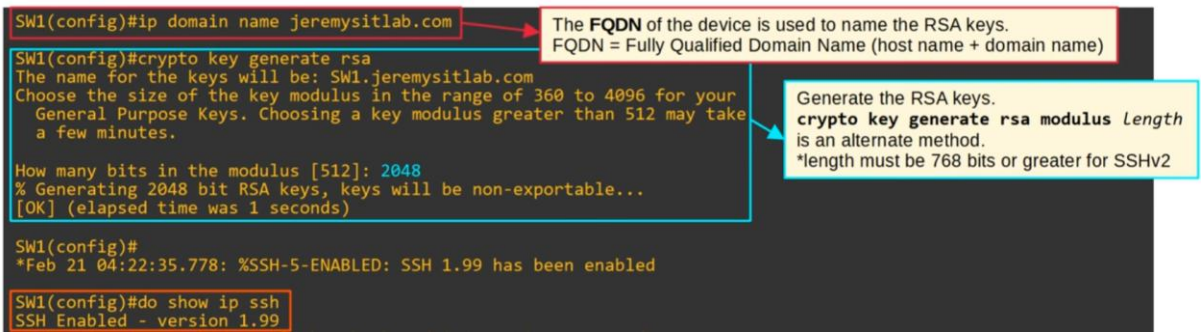


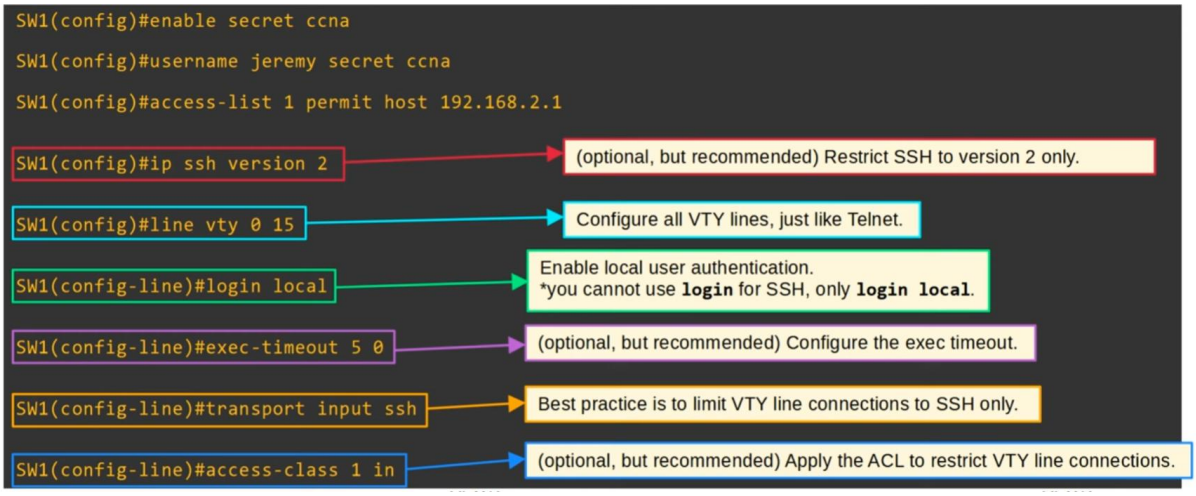
Telnet:



SSH:

First generate RSA keys! You will need custom device name and domain name first: FQDN.





Mandatory steps:

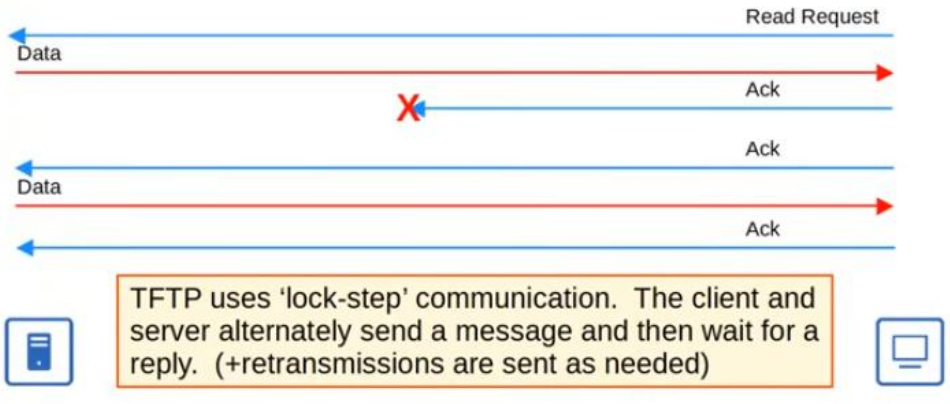
1) Configure host name
2) Configure DNS domain name
3) Generate RSA key pair
4) Configure enable PW, username/PW
5) Enable SSHv2 (only)
6) Configure VTY lines

```
Router(config)#crypto key generate rsa
% Please define a hostname other than Router.
Router(config)#hostname R2
R2(config)#crypto key generate rsa
% Please define a domain-name first.
R2(config)#ip domain name jeremysitlab.com
R2(config)#crypto key generate rsa
The name for the keys will be: R2.jeremysitlab.com
[output omitted]
```

Connect: **ssh -l** *username ip-address* OR **ssh** *username@ip-address*

38. FTP & TFTP

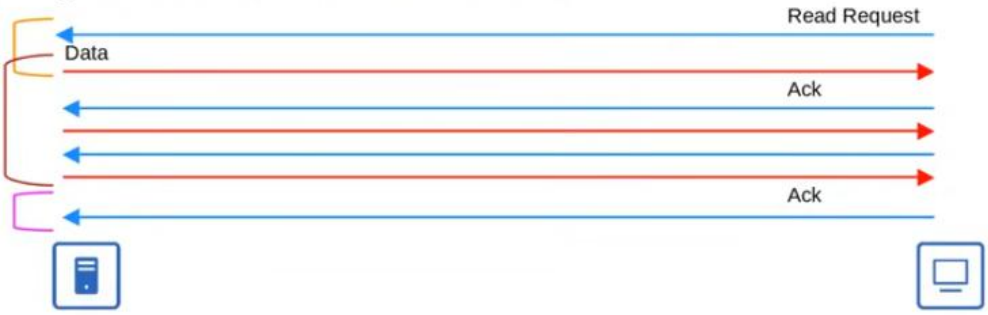
Timers are used, and if an expected message isn't received in time, the waiting device will re-send its previous message.



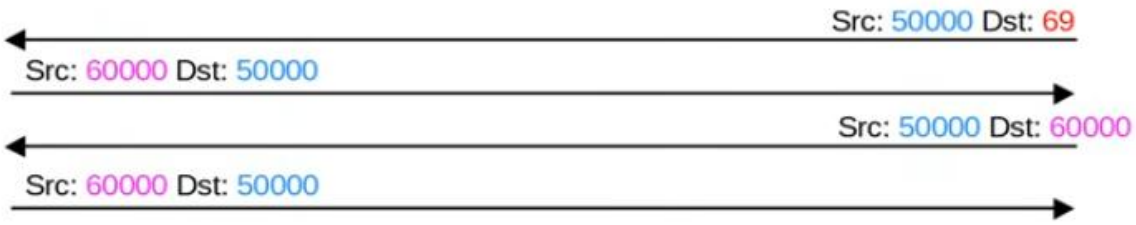
TFTP file transfers have three phases:
1: **Connection**: TFTP client sends a request to the server, and the server responds back, initializing the connection.

2: **Data Transfer**: The client and server exchange TFTP messages. One sends data and the other sends acknowledgments.

3: **Connection Termination**: After the last data message has been sent, a final acknowledgment is sent to terminate the connection.

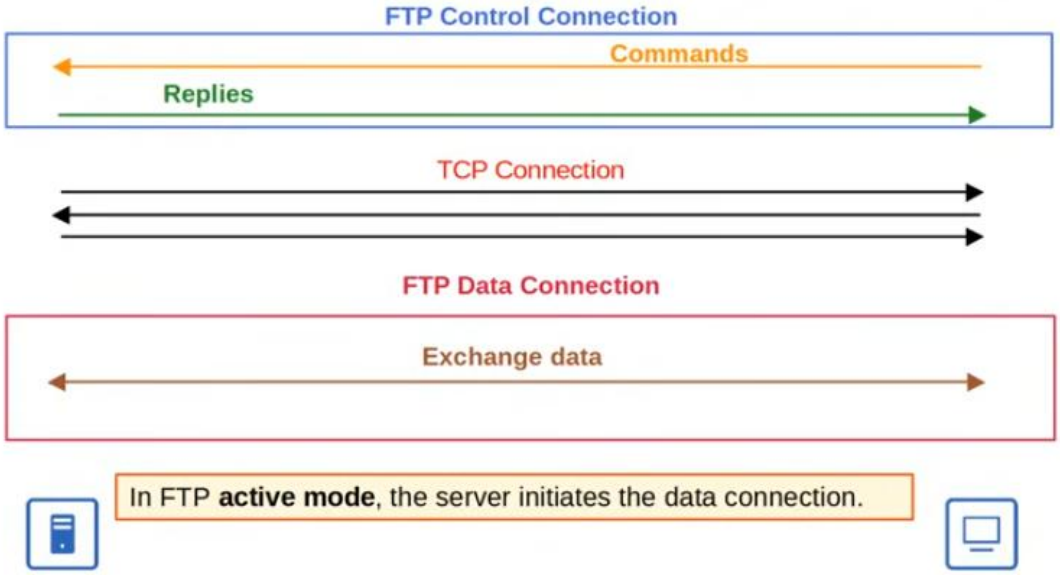


First initialisation; server listens to UDP port 69. Then, random port is selected by server to identify the next data transfers.
Client uses random port from start.



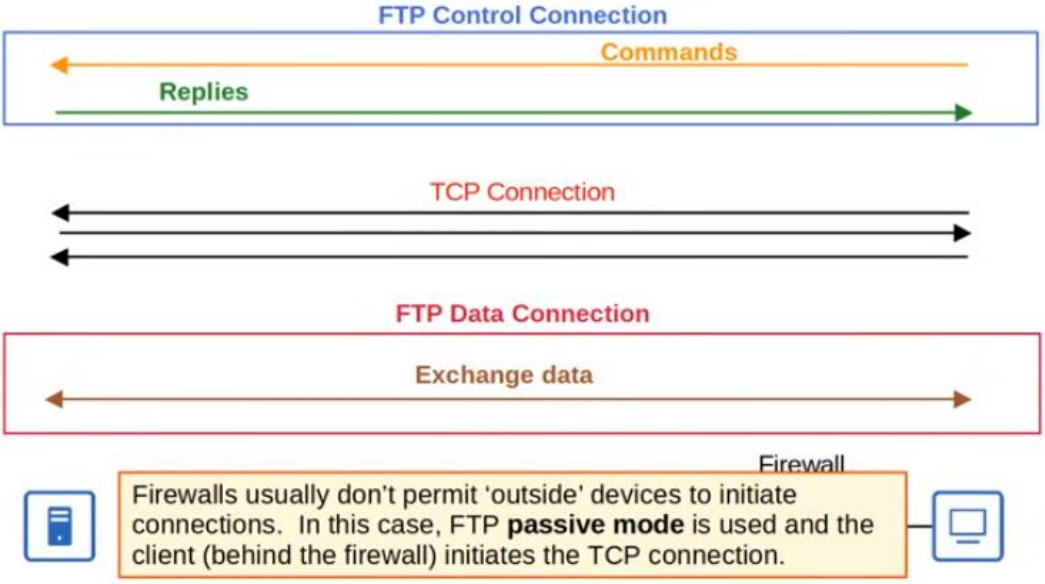
*This is beyond the scope of the CCNA, but is an interesting part of TFTP's operation.

The default method of establishing FTP data connections is **active mode**, in which the se initiates the TCP connection.



se = server

In FTP **passive mode**, the client initiates the data connection. This is often necessary wh the client is behind a firewall, which could block the incoming connection from the server.



wh = when

Look at the difference in TCP connection arrows! FTP data connection can either be active or passive initiated.

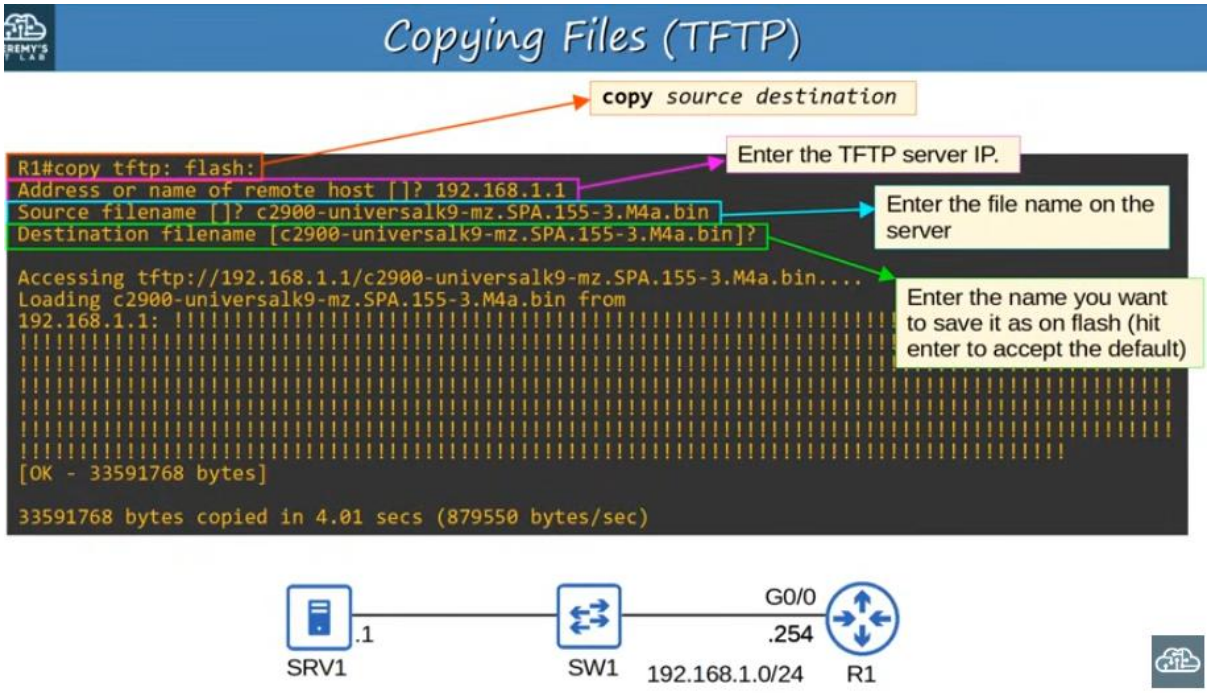
FTP	TFTP
-Uses TCP (20 for data, 21 for control) for connection-based communication	-Uses UDP (69) for connectionless communication (although a basic form of 'connection' is used within the protocol itself)
-Clients can use FTP commands to perform various actions, not just copy files	-Clients can only copy files to or from the server
-Username/PW authentication	-No authentication
-More complex	-Simpler

Look at the files on network device:

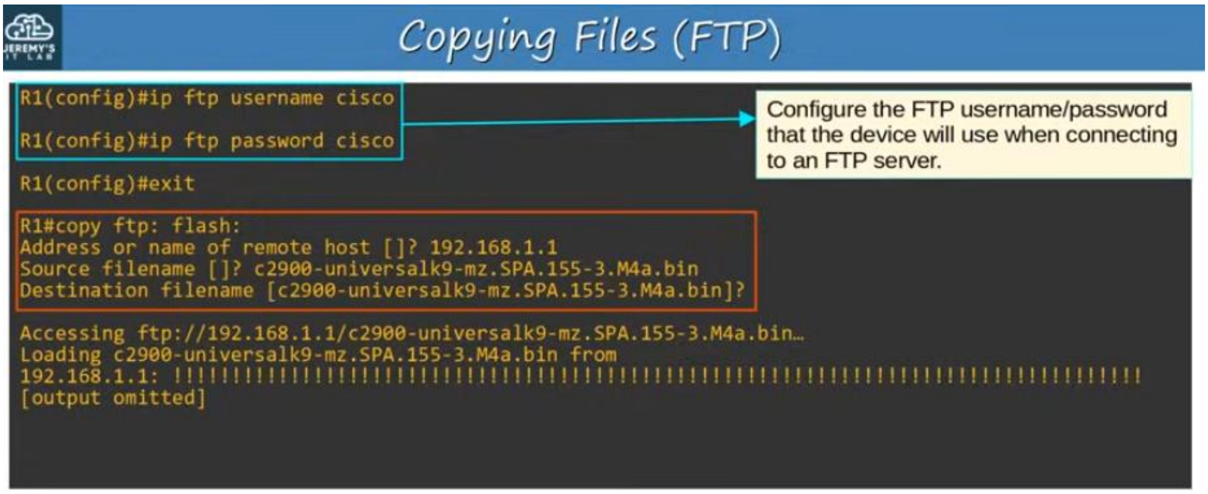
R1#show file systems

Current IOS:

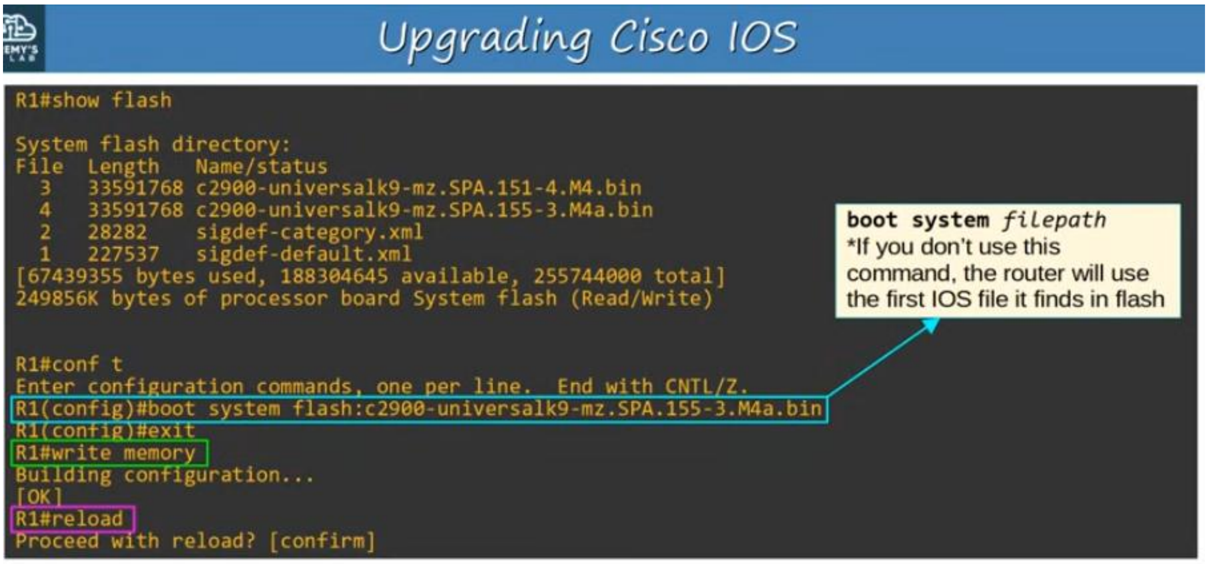
R1#show version



Remember: you need to know the name of the file *beforehand* if you use TFTP!

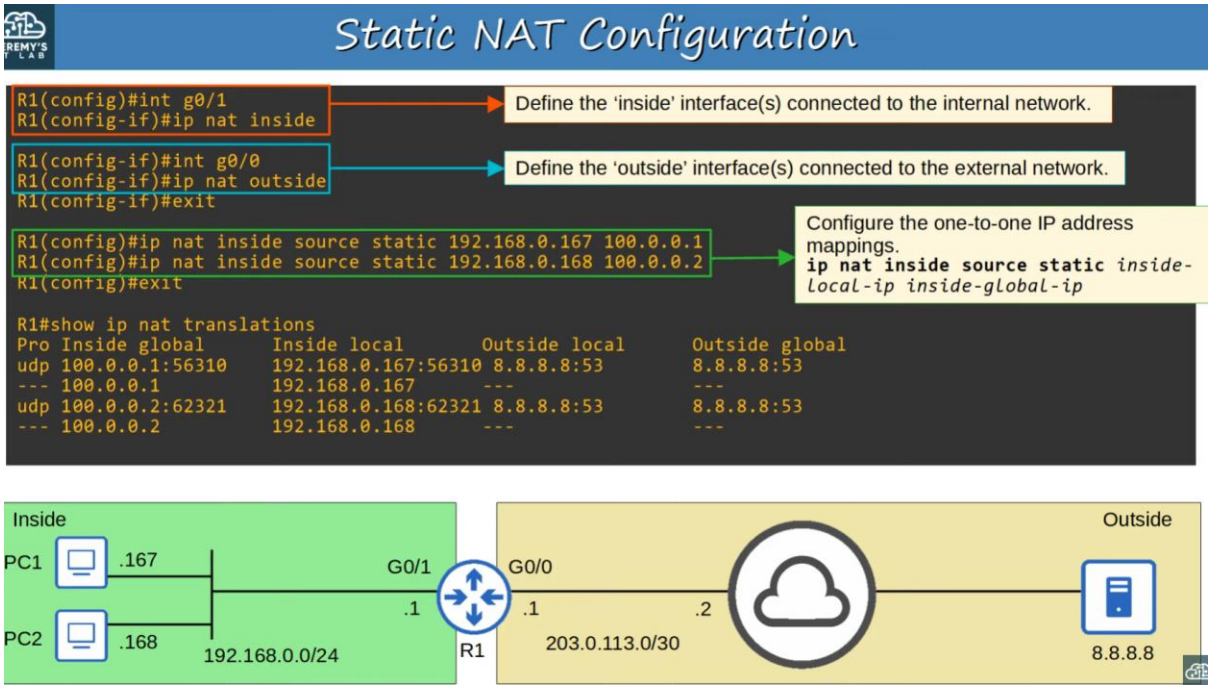


Same username and password needed on the FTP server!



39. NAT (part 1)

View NAT information:
R1#show ip nat translations
R1#show ip nat statistics



You need to own the public IP addresses as well!
Number after inside global IP address is the port number used in UDP. This remains the same after IP address translation.

→ **Inside Local** = The IP address of the *inside* host, from the perspective of the local network
*the IP address actually configured on the inside host, usually a private address

→ **Inside Global** = The IP address of the *inside* host, from the perspective of *outside* hosts
*the IP address of the inside host after NAT, usually a public address

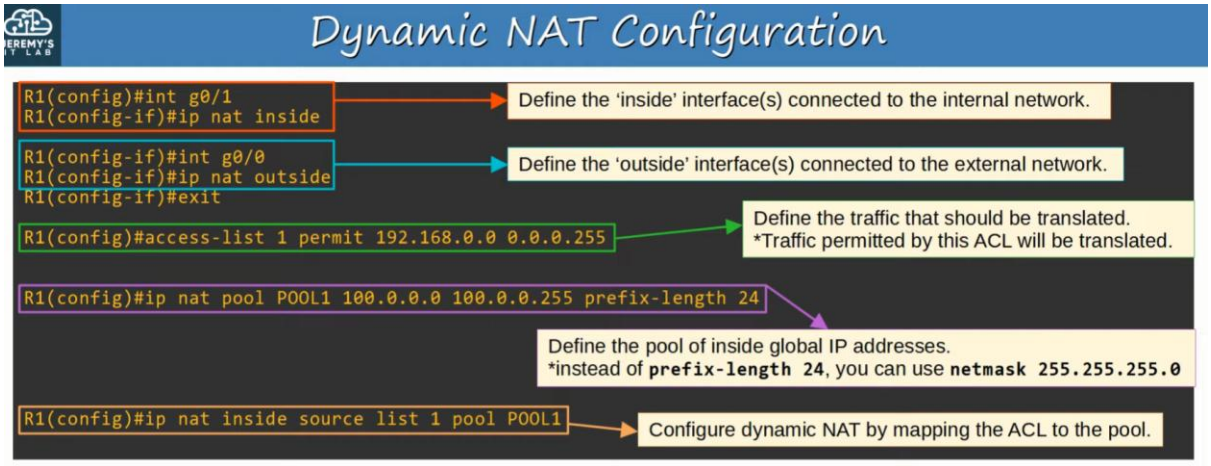
→ **Outside Local** = The IP address of the *outside* host, from the perspective of the local network

→ **Outside Global** = The IP address of the *outside* host, from the perspective of the outside network

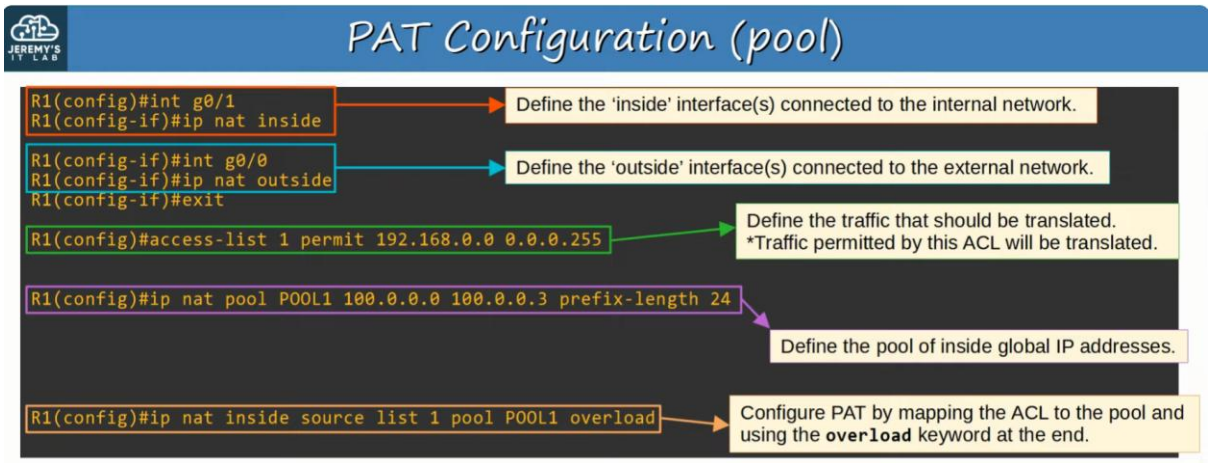
Inside/Outside = Location of the host
Local/Global = Perspective

```
R1(config-if)# ip nat inside
R1(config-if)# ip nat outside
R1(config)# ip nat inside source static inside-local-ip inside-global-ip
R1# show ip nat translations
R1# show ip nat statistics
R1# clear ip nat translation *
```

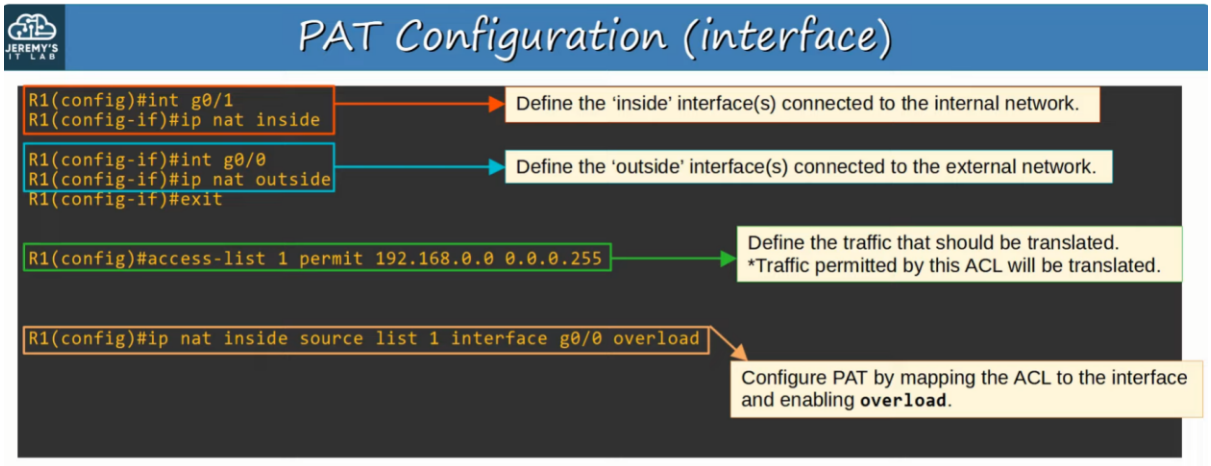
40.NAT (part 2)



Configure PAT via pool OR via interface.



Difference with dynamic NAT is the **word overload.**



This uses the interface of the router as inside global IP but only different port number for each host!

41. Network Management
42. Network Troubleshooting
43. Network Security Concepts