

# 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



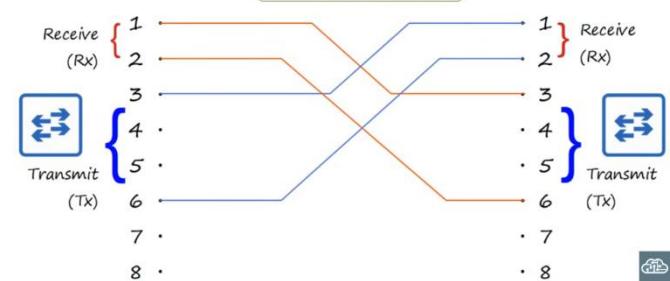
UTP Cables (10BASE-T, 100BASE-T)

Straight-through cable



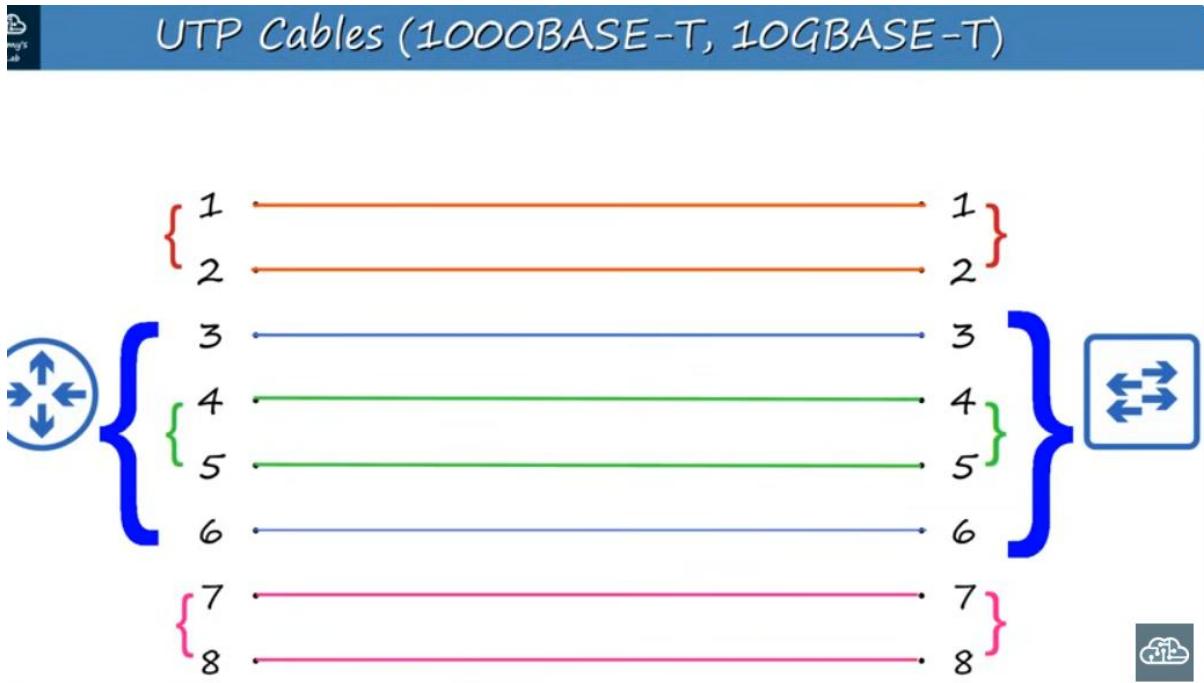
UTP Cables (10BASE-T, 100BASE-T)

Crossover cable



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

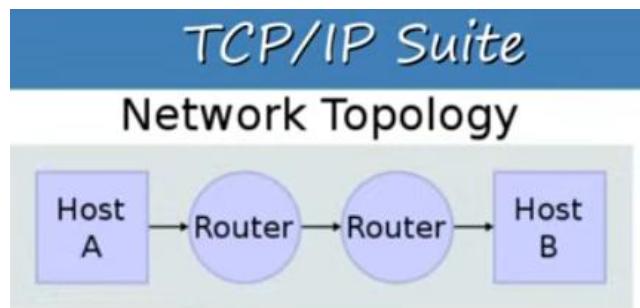
But nowadays Auto MDI-X 😊



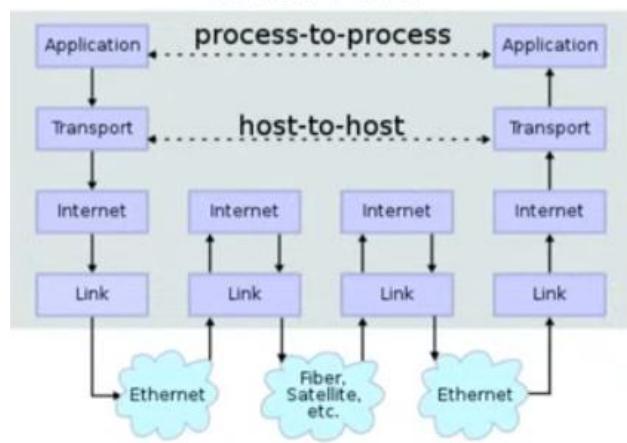
## Fiber-Optic Cable Standards

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



### Data Flow



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:

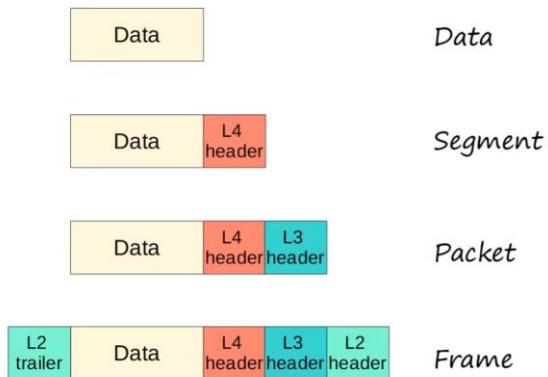
```
R1(config)#banner motd # message #
```

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

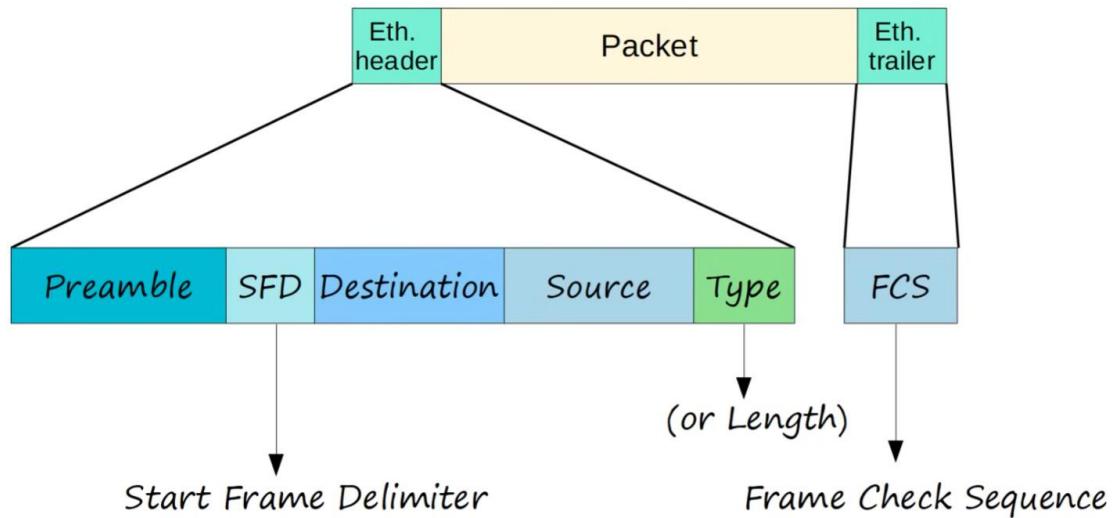
5. /

#### 6. Ethernet LAN Switching (part 1)

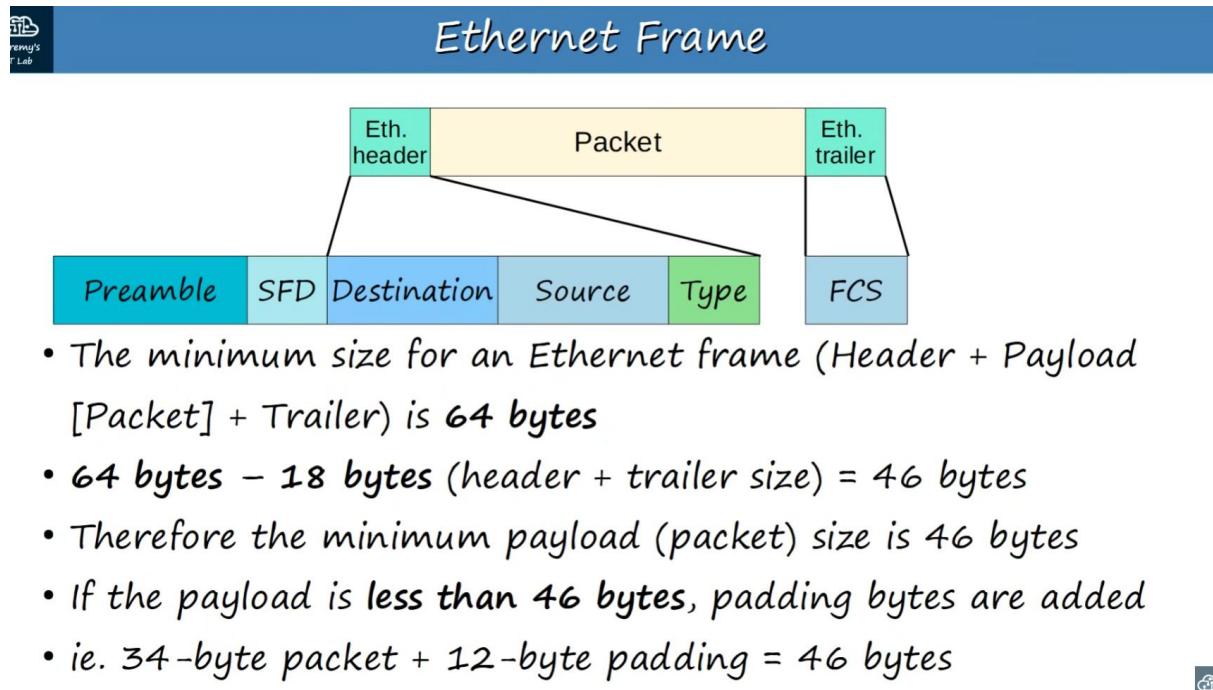
All PDUs:



Frame:



## 7. Ethernet LAN Switching (part 2)



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:	Internet Address	Physical Address
169.254.146.29 --- 0x9	169.254.255.255	ff-ff-ff-ff-ff-ff
	224.0.0.2	01-00-5e-00-00-02
	224.0.0.22	01-00-5e-00-00-16
	224.0.0.251	01-00-5e-00-00-fb
	224.0.0.252	01-00-5e-00-00-fc
	239.255.255.250	01-00-5e-7f-ff-fa
	255.255.255.255	ff-ff-ff-ff-ff-ff
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)

Have a look at MAC address table on a switch:

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)



Class	First octet	First octet numeric range
A	0xxxxxxx	0-127 <span style="color:red">126</span>
B	10xxxxxx	128-191
C	110xxxxx	192-223
Multicast addresses	D	1110xxxx
Reserved (experimental)	E	1111xxxx

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 it 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		DSCH		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																																													
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.

10.10.30.10 10.10.30.10 Summary, L1 10.10.10.1 Level 1, L2 10.10.10.1  
- TS-TS inter area \* - candidate default II - per-user stat

**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  $32 - \text{mask}$ .

E.g. /27 → 5 host bits

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

$n$  = 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	Prefix Length	Number of Subnets	Number of Hosts
/17	2	32766	/25	512	126
/18	4	16382	/26	1024	62
/19	8	8190	/27	2048	30
/20	16	4094	/28	4096	14
/21	32	2044	/29	8192	6
/22	64	1022	/30	16384	2
/23	128	510	/31	32768	0 (2)
/24	256	254	/32	65536	0 (1)

## 15. Subnetting (part 2)

/

## 16. Subnetting (part 3 -VLSM)

When VLSM you need to define this for every subnet:

Network address:

Broadcast address:

First usable address:

Last usable address:

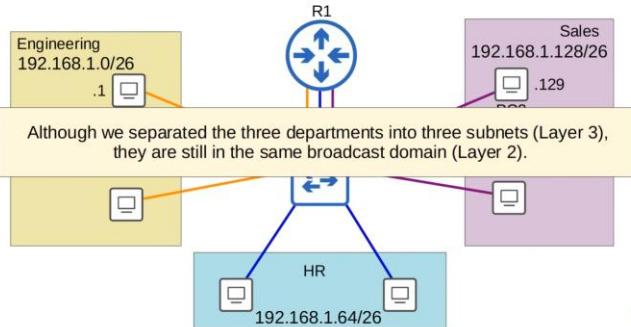
Total number of usable host addresses:

Show info of a specific interface:

R1#show ip interface g0/0

## 17. VLANs (part 1)

## What is a VLAN?



## Engineering's IT Lab

## 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

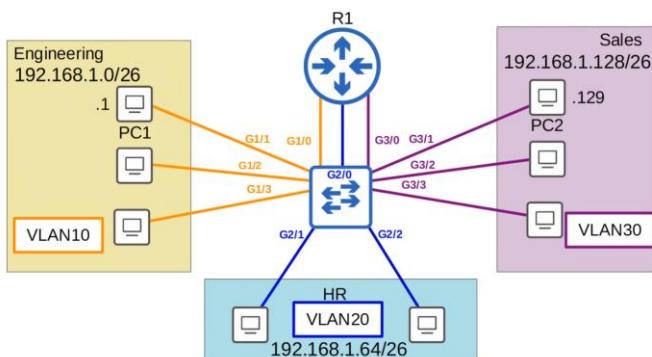
```
SW1(config)#interface range g1/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
SW1(config-if-range)#interface range g2/0 - 2
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport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

## VLAN Configuration



```
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 ?
```

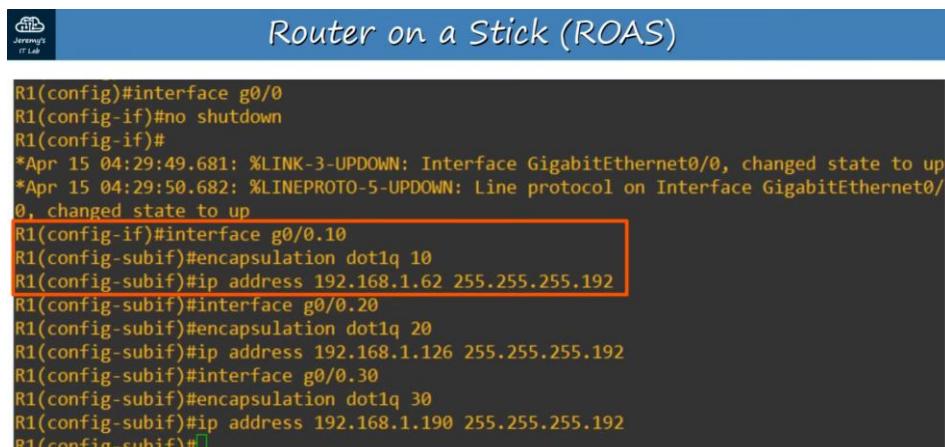
Set native vlan:

```
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)

```
R1(config)#int g0/0.10
R1(config-if)#encapsulation dot1q 10
R1(config-subif)#ip address 192.168.1.62 255.255.255.192
```

The subinterface number **does not** have to match the VLAN number.  
However it is **highly recommended** that they do match, to make it easier to understand.

\*Apr 15 04:29:49.681: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up  
\*Apr 15 04:29:50.682: %LINK-3-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up



## 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)#[REDACTED]
```

*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)#[REDACTED]
```

-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)



### Spanning Tree Protocol

- 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

Caveat; tie breakers are NEIGHBOR IDs!

Look at the STP:

**SW1#show spanning-tree**

**SW1#show spanning-tree vlan 1**

**SW1#show spanning-tree detail**

**SW1#show spanning-tree summary**

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

Enable portfast (only on ports connected to end hosts!):

**SW1(config-if)#spanning-tree portfast**

→ As a safety, this does not work on trunk ports, only on access ports. Because trunk ports might be connected to other switch.

You can also enable portfast with the following command:

**SW1(config)# spanning-tree portfast default**

This enables portfast on all access ports (not trunk ports).

```
| SW1(config)#interface g0/2  
| SW1(config-if)#spanning-tree bpduguard enable  
| SW1(config-if)#
```

You can also enable BPDU Guard with the following command:

```
SW1(config)# spanning-tree portfast bpduguard default
```

This enables BPDU Guard on all Portfast-enabled interfaces.

If an interface with BPDU Guard enabled receives a BPDU from another switch, the interface will be shut down to prevent a loop from forming.

## Configure the Primary Root Bridge

```
| SW3(config)#spanning-tree vlan 1 root primary  
| SW3(config)#do show spanning-tree  
  
VLAN0001  
  Spanning tree enabled protocol ieee  
  Root ID    Priority    24577  
            Address     cccc.cccc.cccc  
            This bridge is the root  
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
  
  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)  
            Address     cccc.cccc.cccc  
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
            Aging Time  15 sec
```

The **spanning-tree vlan vLan-number root primary** command sets the STP priority to 24576. If another switch already has a priority lower than 24576, it sets this switch's priority to 4096 less than the other switch's priority.

Remember; the root bridge is the switch acting as the root bridge itself! You configure this in the global configuration mode of the switch (not on an interface).



## Configure the Secondary Root Bridge

```
SW2(config)#spanning-tree vlan 1 root secondary
SW2(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
              Address     cccc.cccc.cccc
              Cost         4
              Port        1 (GigabitEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    28673 (priority 28672 sys-id-ext 1)
              Address     bbbb.bbbb.bbbb
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time   300 sec
```

The **spanning-tree vlan *vlan-number* root secondary** command sets the STP priority to 28672.

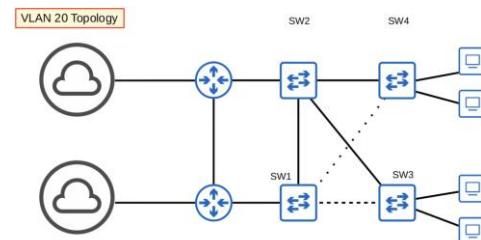
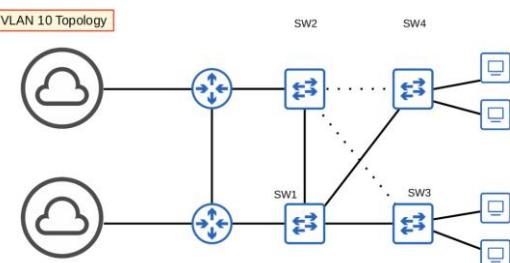
Will take over when primary root bridge fails.

You can manually choose the port cost (which adds to total cost) and port priority (which is a tie breaker in selecting the port) of each interface.

```
SW1(config-if)#spanning-tree {vlan 1} cost 100
SW1(config-if)#spanning-tree {vlan 1} port-priority 240
```

*Example of primary and secondary root over different VLANs:*

```
SW1(config)# spanning-tree vlan 10 root primary
SW1(config)# spanning-tree vlan 20 root secondary
SW2(config)# spanning-tree vlan 20 root primary
SW2(config)# spanning-tree vlan 10 root secondary
```



## 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

Enable RSTP via:

**SW1(config)#spanning-tree mode rapid-pvst**

• All switches running Rapid STP send their own BPDUs every hello time (2 seconds).

• Switches 'age' the BPDU information much more quickly. In classic STP, a switch waits 10 hello intervals (20 seconds). In rapid STP, a switch considers a neighbor lost if it misses 3 BPDUs (6 seconds). It will then 'flush' all MAC addresses learned on that interface.

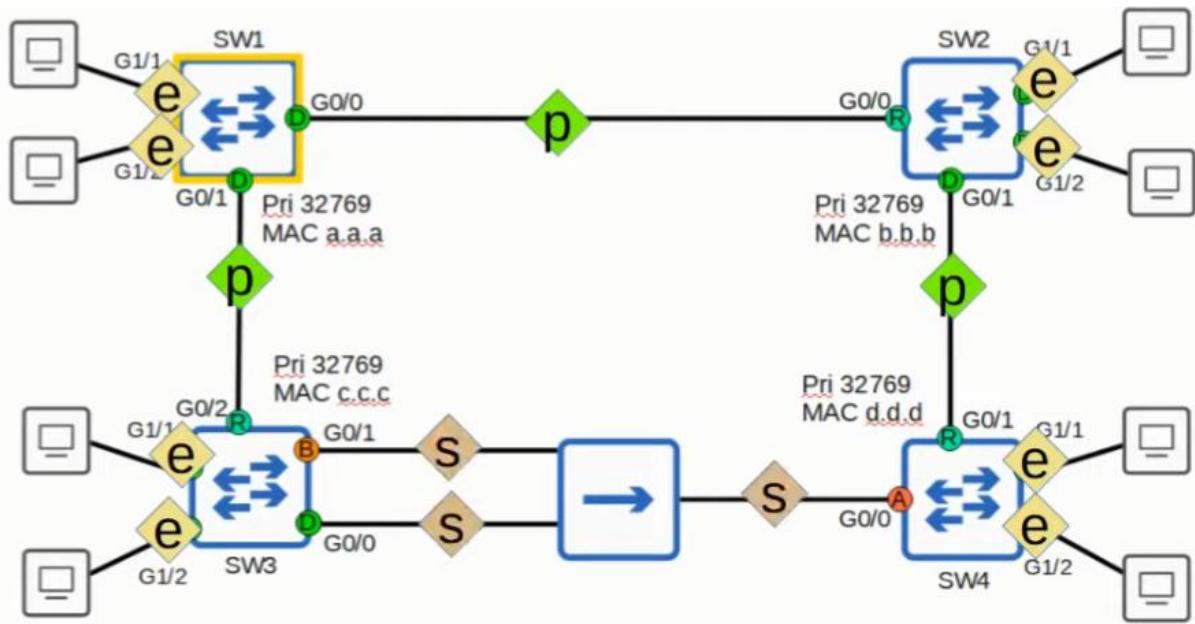
• RSTP distinguishes between three different 'link types'.

• **Edge**: a port that is connected to an end host. Moves directly to forwarding, without negotiation.

• **Point-to-point**: a direct connection between two switches.

• **Shared**: a connection to a hub. Must operate in half-duplex mode.

These link types are automatically detected, thus portfast is not necessary to configure in RSTP.



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.

! See end of video solution lab RSTP for explanation of different link types and how to see via CLI.

## 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. <b>Links of all speeds are equal.</b>
EIGRP	Metric based on bandwidth & delay (by default)	Complex formula that can take into account many values. By default, the bandwidth of the <b>slowest link in the route</b> 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 <b>not</b> 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'.

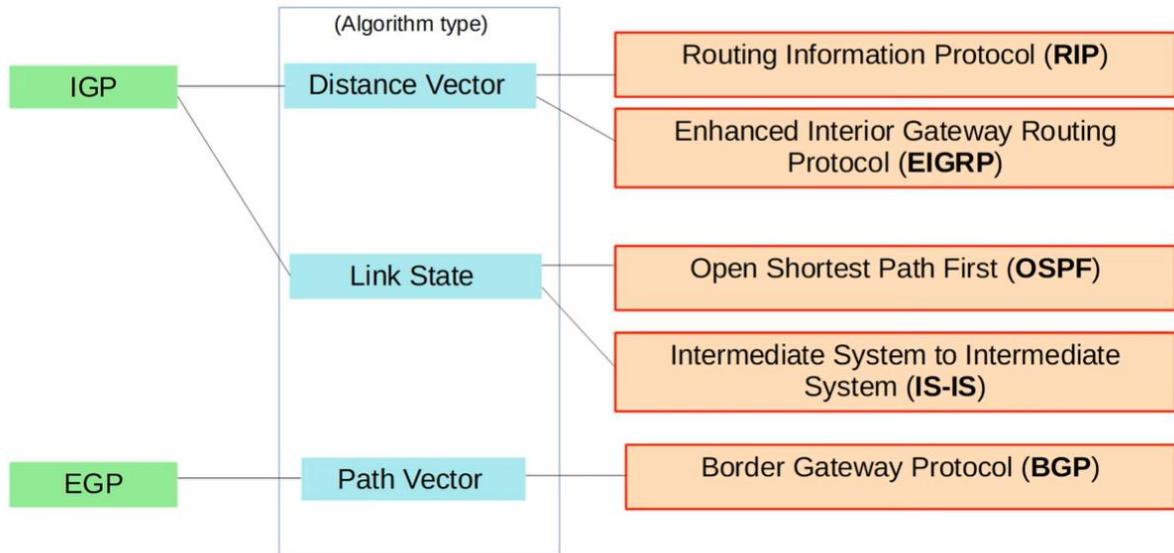
You cannot compare different protocol routes, because they use different metrics (appels en peren).

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

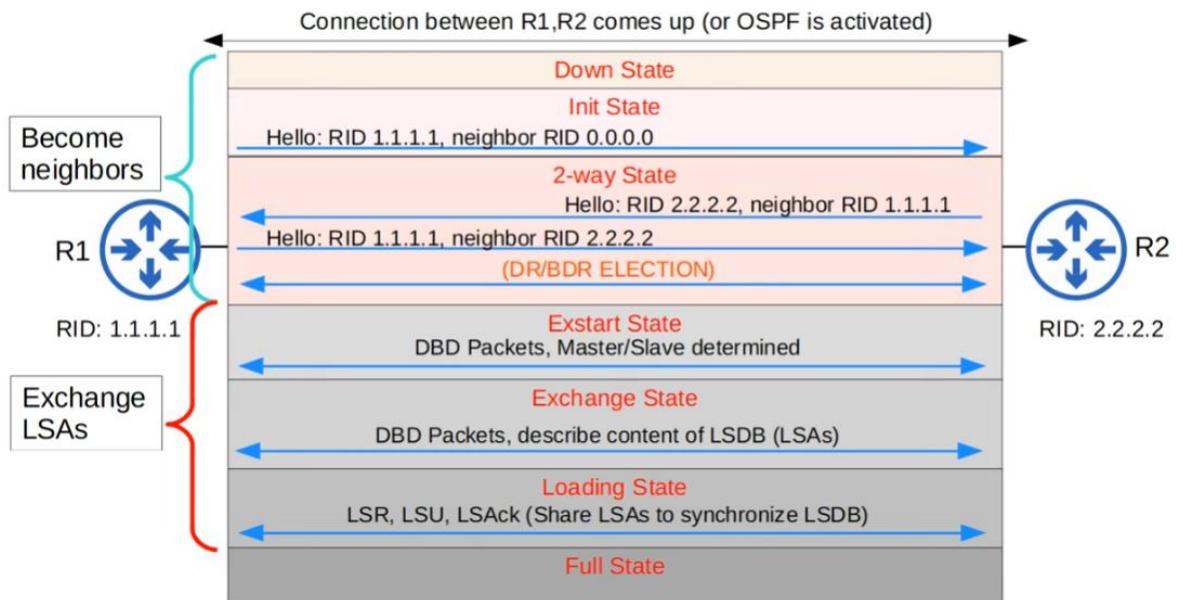
Route protocol/type	AD
IS-IS	115
RIP	120
EIGRP (external)	170
Internal BGP (iBGP)	200
Unusable route	255

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	<b>Hello</b>	Neighbor discovery and maintenance.
2	<b>Database Description (DBD)</b>	Summary of the LSDB of the router. Used to check if the LSDB of each router is the same.
3	<b>Link-State Request (LSR)</b>	Requests specific LSAs from the neighbor.
4	<b>Link-State Update (LSU)</b>	Sends specific LSAs to the neighbor.
5	<b>Link-State Acknowledgement (LSAck)</b>	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

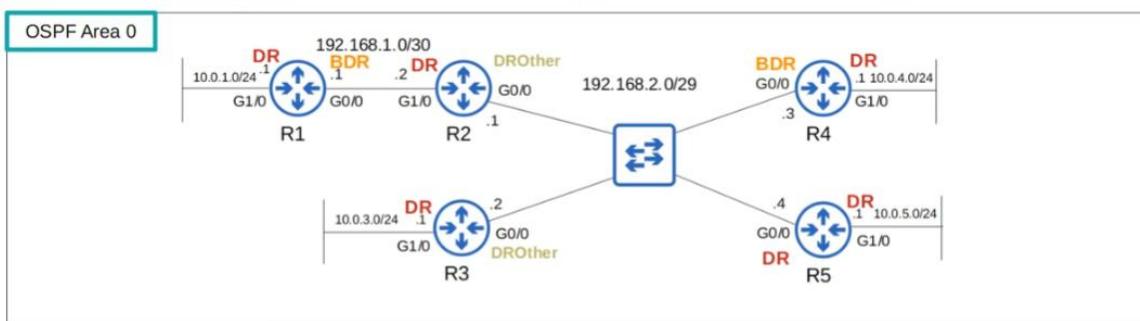
R1#(config)#router ospf 1

R1(config-router)#passive-interface default

Then select the active ones

R1(config-router)#no passive-interface g0/0

## 27. OSPF (part 3)



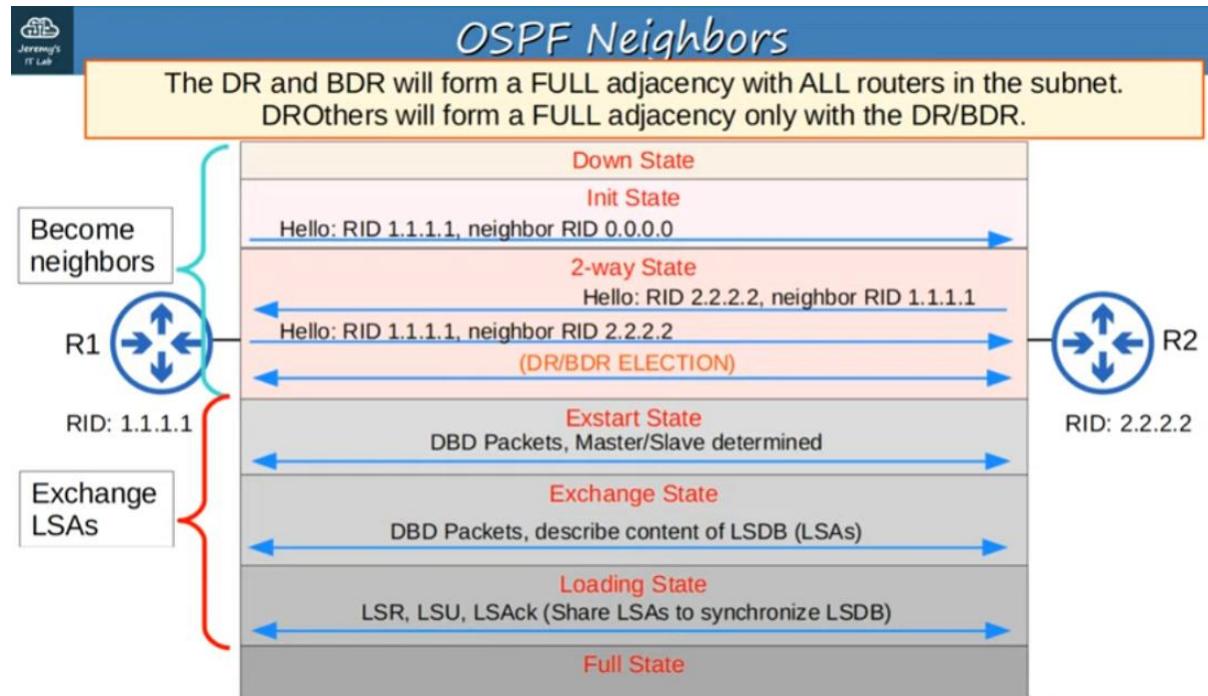
- The DR/BDR election order of priority:
  - Highest **OSPF interface priority**
  - Highest OSPF Router ID
- 'First place' becomes the DR for the subnet, 'second place' becomes the BDR
- The default OSPF interface priority is 1 on all interfaces

## Set OSPF priority

R1(config)#int g0/0

R1(config-if)#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
```

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!

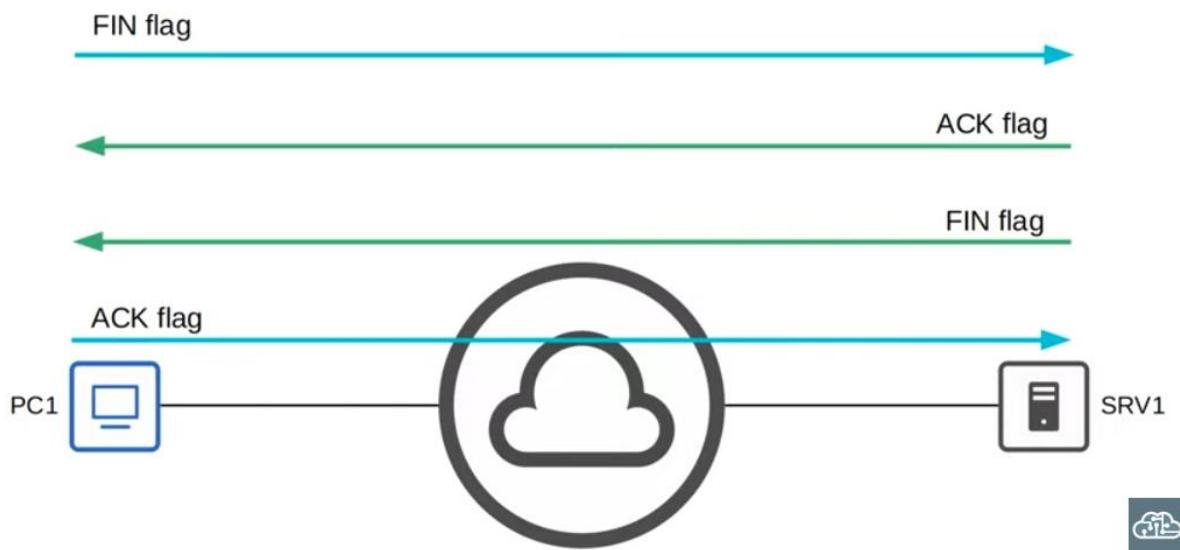
## 29. TCP & UDP

### Establishing Connections: Three-Way Handshake

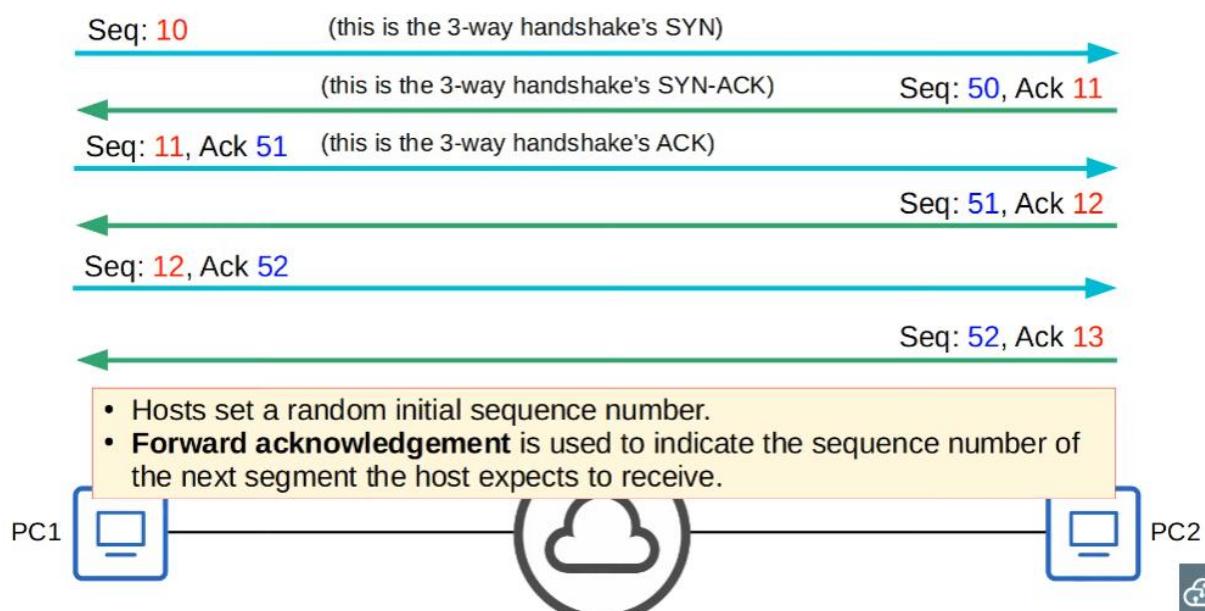




## Terminating Connections: Four-Way Handshake



## TCP: Sequencing / Acknowledgment



## 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:[ ]:[ ]: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.

1: Divide the MAC address in half

**1234 5678 90AB** → **1234 56 | 78 90AB**

2: Insert FFFE in the middle

**1234 56FF FE78 90AB**

3: Invert the 7<sup>th</sup> bit

**1234 56FF FE78 90AB** → **1034 56FF FE78 90AB**



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

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

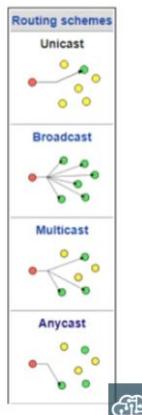
Scopes:

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!



## Neighbor Advertisement (NA)



- Source IP: R2 G0/0 IP
- Destination IP: R1 G0/0 IP
- Source MAC: R2 G0/0 MAC
- Destination MAC: R1 G0/0 MAC

```
> Frame 7: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:02:09:7c:00:08 (ca:02:09:7c:00:08), Dst: ca:01:09:6d:00:08 (ca:01:09:6d:00:08)
> Internet Protocol Version 6, Src: 2001:db8::78:9abc, Dst: 2001:db8::12:3456
> Internet Control Message Protocol v6
```

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?



- 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  
lt  
neq  
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	UDP
• FTP data (20)	• DHCP server (67)
• FTP control (21)	• DHCP client (68)
• SSH (22)	• TFTP (69)
• Telnet (23)	• SNMP agent (161)
• SMTP (25)	• SNMP manager (162)
• HTTP (80)	• Syslog (514)
• POP3 (110)	
• HTTPS (443)	
	<b>TCP &amp; UDP</b>
	• DNS (53)



## Extended ACL entry practice (2)

- 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
```

- 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
```

- 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:

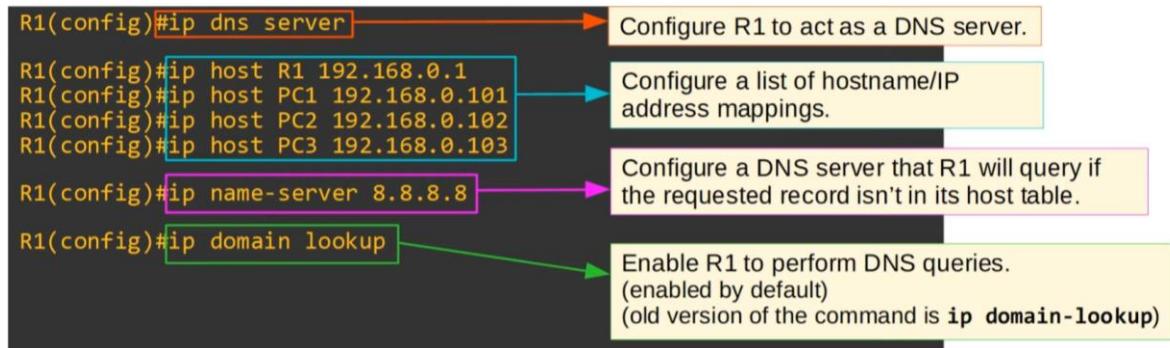
```
nslookup youtube.com
```

**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:

```
R1(config)#do ping youtube.com
Translating "youtube.com"
% Unrecognized host or address, or protocol not running.

R1(config)#ip name-server 8.8.8.8 → Configure R1 to use the specified DNS server.

R1(config)#ip domain lookup → Enable R1 to perform DNS queries. (default)

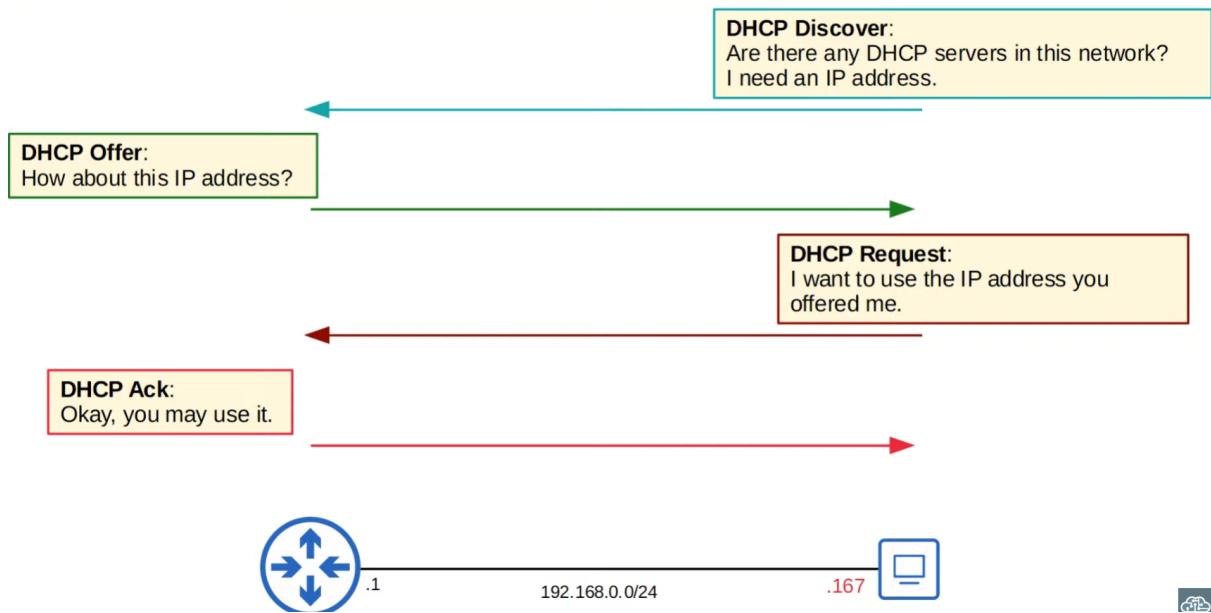
R1(config)#do ping youtube.com
Translating "youtube.com"...domain server (8.8.8.8) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.217.25.110, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/13 ms
```

Windows:  
C:\Users\user>ipconfig /all  
C:\Users\user>xsl lookup name  
C:\Users\user>ipconfig /displaydns  
C:\Users\user>ipconfig /flushdns  
C:\Users\user>ping ip-address -n number

Cisco IOS:  
R1(config)#ip dns server  
R1(config)#ip host hostname ip-address  
R1(config)#ip name-server ip-address  
R1(config)#ip domain lookup  
R1(config)#ip domain name domain-name  
R1#show hosts

## 36. DHCP

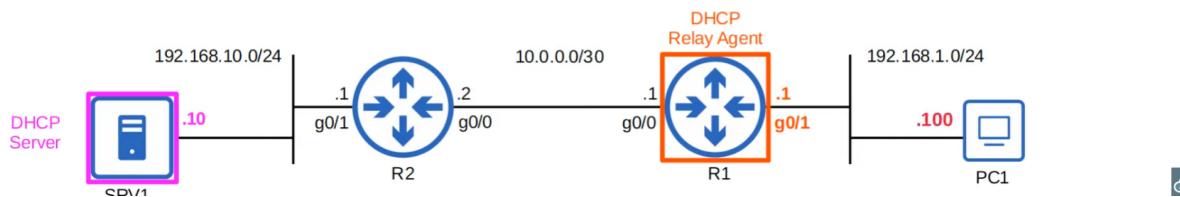
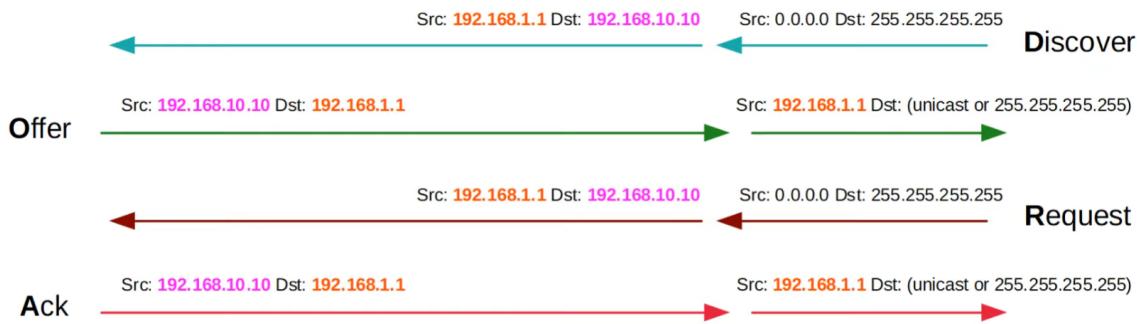


## DHCP D-O-R-A

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



## DHCP Relay



## DHCP Server Configuration in IOS

```
R1(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10
```

Specify a range of addresses that **won't** be given to DHCP clients.

```
R1(config)#ip dhcp pool LAB_POOL
```

Create a DHCP pool.

```
R1(dhcp-config)#network 192.168.1.0 ?  
/nn or A.B.C.D Network mask or prefix length  
<cr>
```

Specify the subnet of addresses to be assigned to clients (except the excluded addresses)

```
R1(dhcp-config)#network 192.168.1.0 /24
```

```
R1(dhcp-config)#dns-server 8.8.8.8
```

Specify the DNS server that DHCP clients should use.

```
R1(dhcp-config)#domain-name jeremysitlab.com
```

Specify the domain name of the network.  
(ie. PC1 = pc1.jeremysitlab.com)

```
R1(dhcp-config)#default-router 192.168.1.1
```

Specify the default gateway.

```
R1(dhcp-config)#lease 0 5 30
```

Specify the lease time.  
**lease days hours minutes** OR  
**lease infinite**

To see all clients with assigned DHCP:

R1#show ip dhcp binding

## DHCP Relay Agent Configuration in IOS

```
R1(config)#interface g0/1
R1(config-if)#ip helper-address 192.168.10.10
R1(config-if)#do show ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
    MTU is 1500 bytes
  Helper address is 192.168.10.10
[output omitted]
```

Configure the interface connected to the subnet of the client devices.

Configure the IP address of the DHCP server as the 'helper' address.



## Command Summary

```
C:\Users\user> ipconfig /release
C:\Users\user> ipconfig /renew
```

```
R1(config)# ip dhcp excluded-address Low-address high-address
R1(config)# ip dhcp pool pool-name
R1(dhcp-config)# network ip-address {/prefix-Length | subnet-mask}
R1(dhcp-config)# dns-server ip-address
R1(dhcp-config)# domain-name domain-name
R1(dhcp-config)# default-router ip-address
R1(dhcp-config)# lease {days hours minutes | infinite}
R1# show ip dhcp binding
```

DHCP server

```
R1(config-if)# ip helper-address ip-address
```

DHCP relay agent

```
R1(config-if)# ip address dhcp
```

DHCP client

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.

```
R1(config)#username jeremy secret ccnp
R1(config)#line console 0
R1(config-line)#login local
R1(config-line)#end
R1#exit
```

Tell the device to require a user to login using one of the configured usernames on the device.

Exec-timeout will log you out automatically after inactivity.

### Vlan:

```
SW1(config)#interface vlan1
SW1(config-if)#ip address 192.168.1.253 255.255.255.0
SW1(config-if)#no shutdown
SW1(config-if)#exit

SW1(config)#ip default-gateway 192.168.1.254
```

Configure the IP address on the SVI in the same way as on a multilayer switch.  
Enable the interface if necessary.

Configure the switch's default gateway.  
In this case, PC2 isn't in the same LAN as SW1. If SW1 doesn't have a default gateway, it can't communicate with PC2.

### Telnet:

```
SW1(config)#enable secret cna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1
SW1(config)#line vty 0 15
SW1(config-line)#login local
SW1(config-line)#exec-timeout 5 0
SW1(config-line)#transport input telnet
SW1(config-line)#access-class 1 in
```

If an enable password/secret isn't configured, you won't be able to access privileged exec mode when connecting via Telnet.

Configure an ACL to limit which devices can connect to the VTY lines.

Telnet/SSH access is configured on the VTY lines. There are 16 lines available, so up to 16 users can be connected at once. (VTY stands for Virtual Teletype)

Configure an ACL to limit which devices can connect to the VTY lines.

Apply the ACL to the VTY lines.

\*access-class applies an ACL to the VTY lines, ip access-group applies an ACL to an interface.

### SSH:

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

```
SW1(config)#ip domain name jeremysitlab.com
SW1(config)#crypto key generate rsa
The name for the keys will be: SW1.jeremysitlab.com
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

SW1(config)#
*Feb 21 04:22:35.778: %SSH-5-ENABLED: SSH 1.99 has been enabled

SW1(config)#do show ip ssh
SSH Enabled - version 1.99
```

The **FQDN** of the device is used to name the RSA keys.  
FQDN = Fully Qualified Domain Name (host name + domain name)

Generate the RSA keys.  
**crypto key generate rsa modulus Length** is an alternate method.  
\*length must be 768 bits or greater for SSHv2

```

SW1(config)#enable secret ccna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1

SW1(config)#ip ssh version 2          (optional, but recommended) Restrict SSH to version 2 only.

SW1(config)#line vty 0 15             Configure all VTY lines, just like Telnet.

SW1(config-line)#login local         Enable local user authentication.
*you cannot use login for SSH, only login local.

SW1(config-line)#exec-timeout 5 0     (optional, but recommended) Configure the exec timeout.

SW1(config-line)#transport input ssh  Best practice is to limit VTY line connections to SSH only.

SW1(config-line)#access-class 1 in    (optional, but recommended) Apply the ACL to restrict VTY line connections.

```

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.

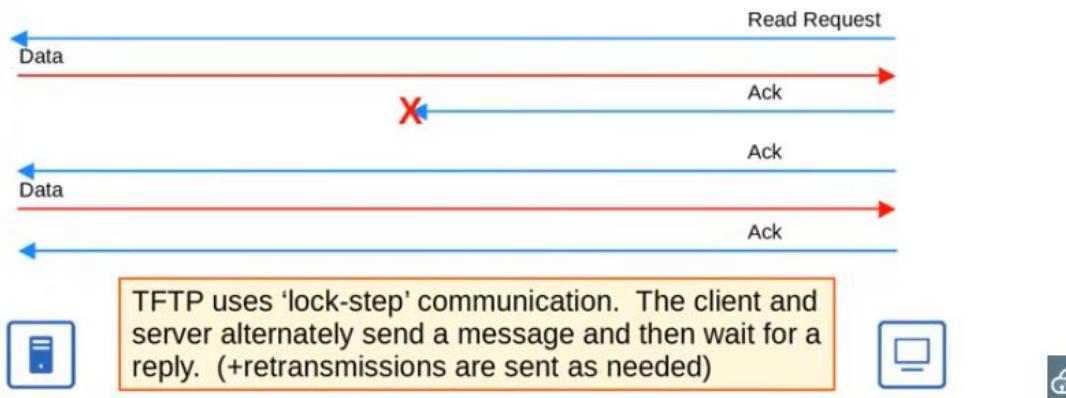
R2(config)#ip domain name jeremysitlab.com
R2(config)#crypto key generate rsa
The name for the keys will be: R2.jeremysitlab.com
[output omitted]

```

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 resend its previous message.

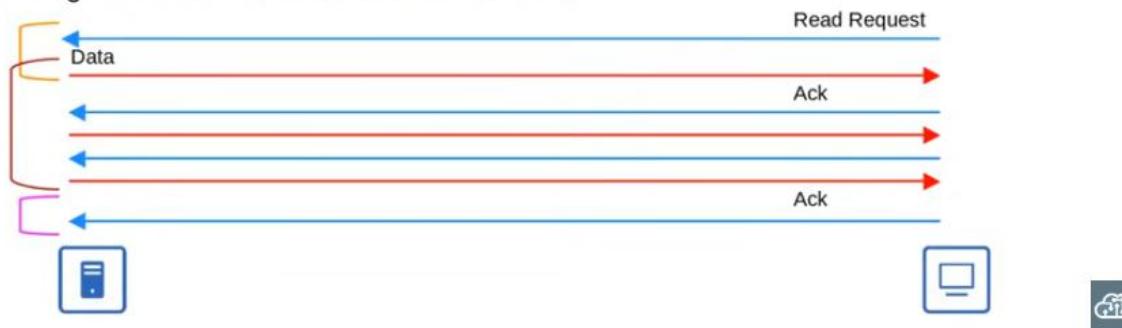


TFTP file transfers have three phases:

1: **Connection**: TFTP client sends a request to the server, and the server responds back, initializing the connection.

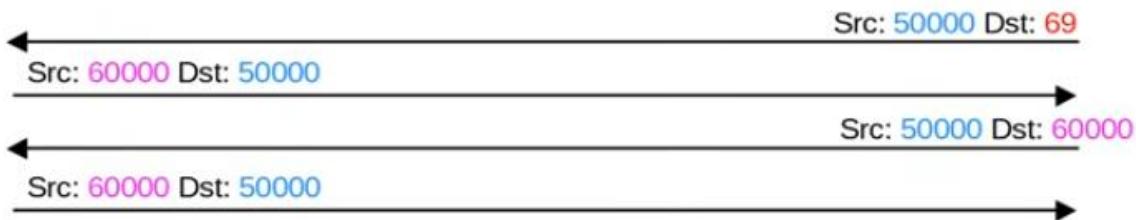
2: **Data Transfer**: The client and server exchange TFTP messages. One sends data and the other sends acknowledgments.

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



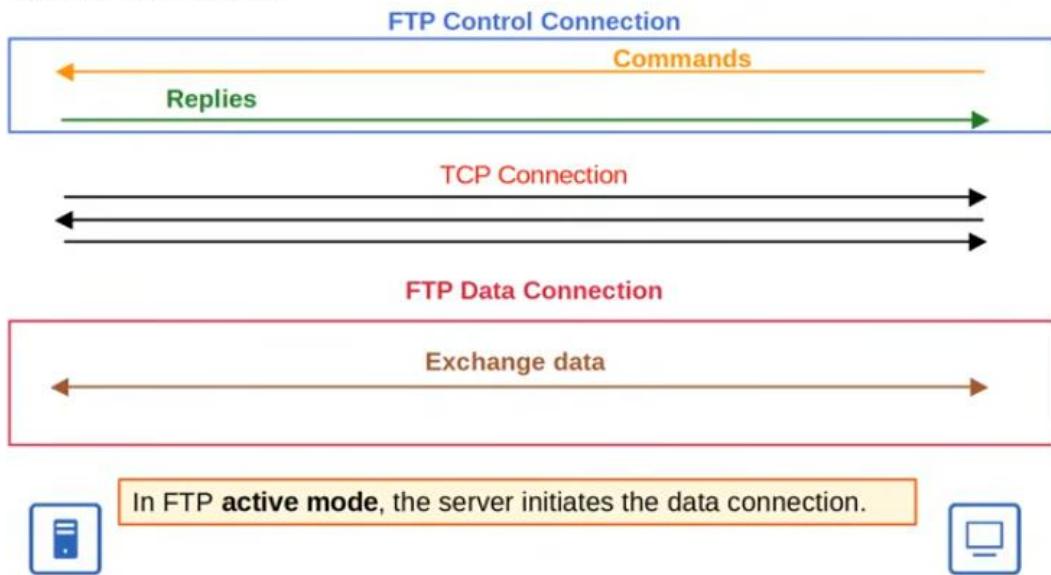
First initialisation; server listens to UDP port 69. Then, random port is selected by server to identify the next data transfers.

Client uses random port from start.



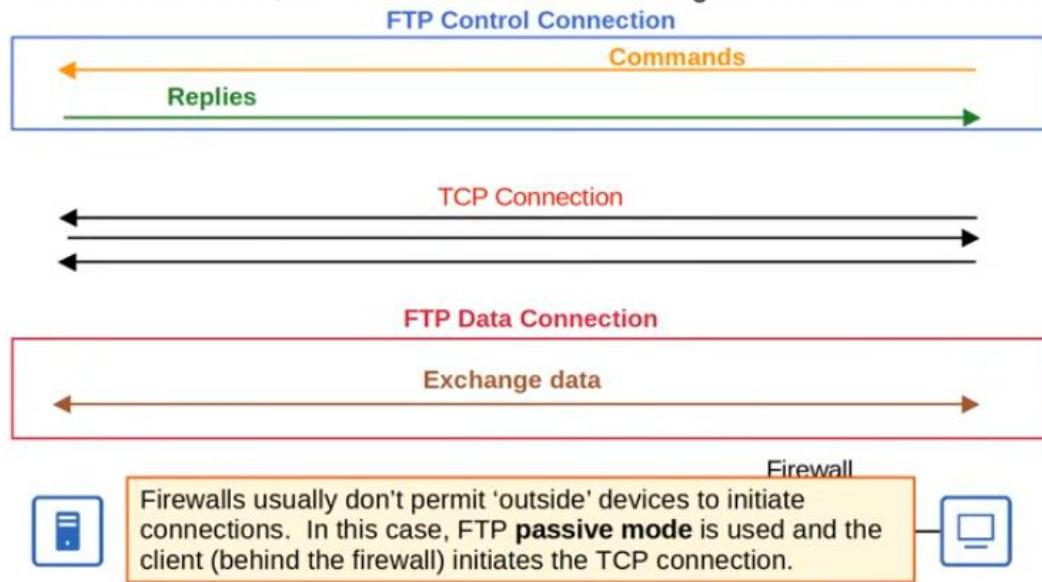
\*This is beyond the scope of the CCNA, but is an interesting part of TFTP's operation.

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



se = server

In FTP **passive mode**, the client initiates the data connection. This is often necessary when 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

- Uses TCP (20 for data, 21 for control) for connection-based communication
- Clients can use FTP commands to perform various actions, not just copy files
- Username/PW authentication

- More complex

## TFTP

- Uses UDP (69) for connectionless communication (although a basic form of 'connection' is used within the protocol itself)
- Clients can only copy files to or from the server
- No authentication
- Simpler



Look at the files on network device:

R1#show file systems

Current IOS:

R1#show version

### Copying Files (TFTP)

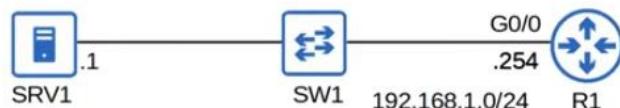
```
R1#copy tftp: flash:  
Address or name of remote host []? 192.168.1.1  
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin  
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?  
Accessing tftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin....  
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from  
192.168.1.1: !!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
[OK - 33591768 bytes]  
33591768 bytes copied in 4.01 secs (879550 bytes/sec)
```

copy source destination

Enter the TFTP server IP.

Enter the file name on the server

Enter the name you want to save it as on flash (hit enter to accept the default)



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



## Copying Files (FTP)

```
R1(config)#ip ftp username cisco  
R1(config)#ip ftp password cisco  
R1(config)#exit
```

```
R1#copy ftp: flash:  
Address or name of remote host []? 192.168.1.1  
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin  
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?
```

```
Accessing ftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin...  
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from  
192.168.1.1: !!!!!!! [output omitted]
```

Configure the FTP username/password that the device will use when connecting to an FTP server.

Same username and password needed on the FTP server!



## Upgrading Cisco IOS

```
R1#show flash  
  
System flash directory:  
File Length Name/status  
3 33591768 c2900-universalk9-mz.SPA.151-4.M4.bin  
4 33591768 c2900-universalk9-mz.SPA.155-3.M4a.bin  
2 28282 sigdef-category.xml  
1 227537 sigdef-default.xml  
[67439355 bytes used, 188304645 available, 255744000 total]  
249856K bytes of processor board System flash (Read/Write)
```

**boot system filepath**  
\*If you don't use this command, the router will use the first IOS file it finds in flash

```
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#boot system flash:c2900-universalk9-mz.SPA.155-3.M4a.bin  
R1(config)#exit  
R1#write memory  
Building configuration...  
[OK]  
R1#reload  
Proceed with reload? [confirm]
```



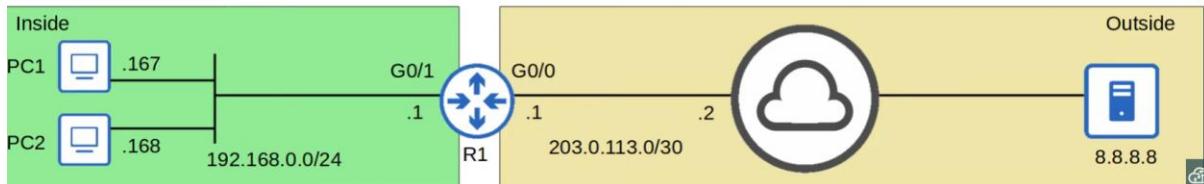
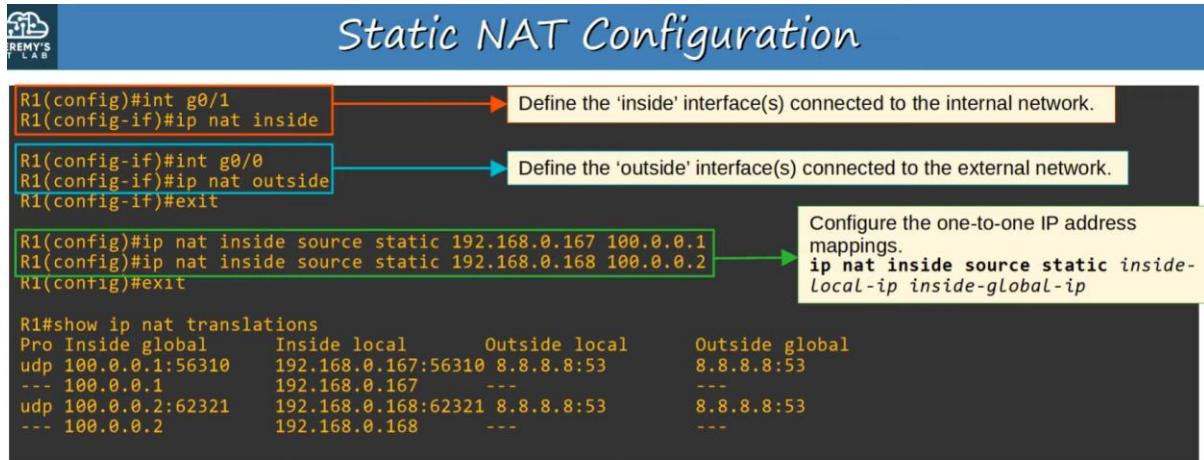
## Command Review

```
R1# show file systems  
R1# show version  
R1# show flash  
R1# copy source destination  
R1(config)# boot system filepath  
R1(config)# ip ftp username username  
R1(config)# ip ftp password password
```

## 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

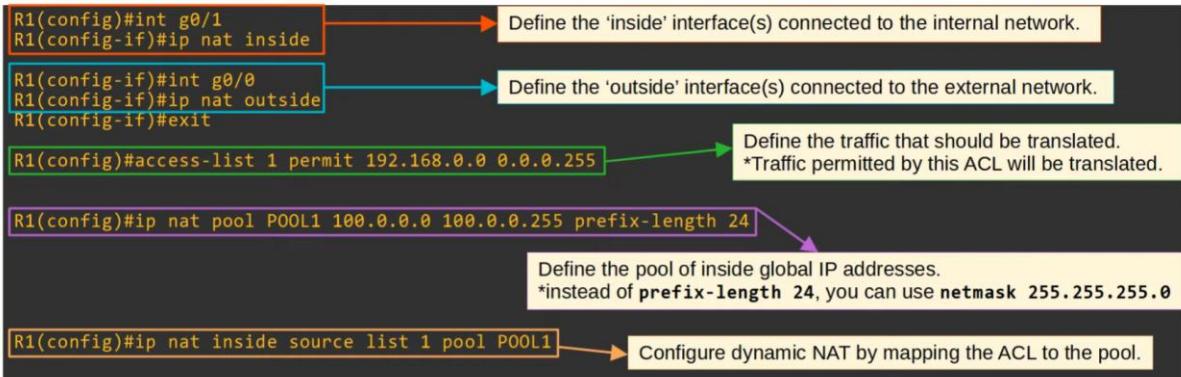
**Local/Global** = Perspective

```
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)



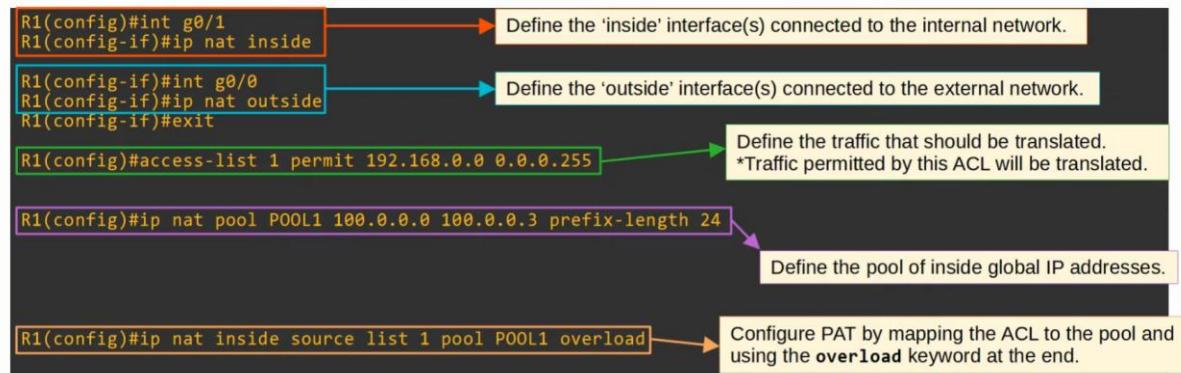
## Dynamic NAT Configuration



Configure PAT via pool OR via interface.



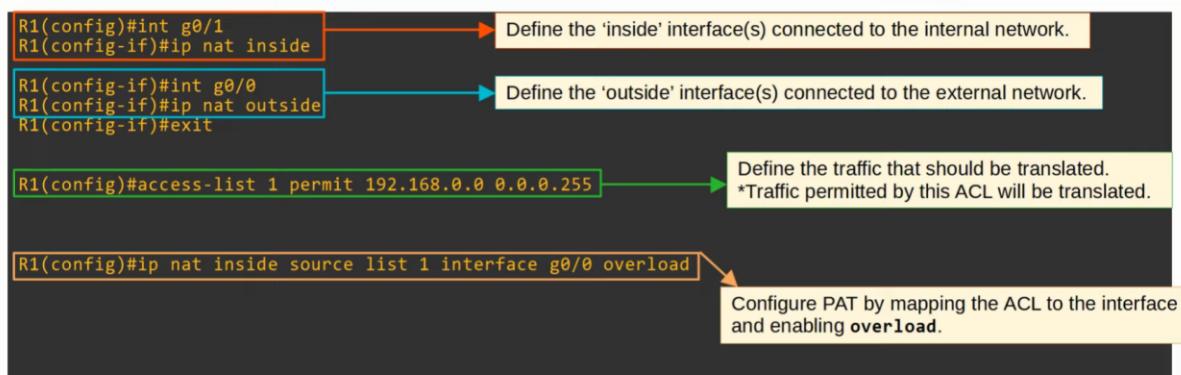
## PAT Configuration (pool)



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



## PAT Configuration (interface)



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



```
R1(config)# ip nat pool pool-name start-ip end-ip prefix-length prefix-length
R1(config)# ip nat pool pool-name start-ip end-ip netmask subnet-mask
R1(config)# ip nat inside source list access-list pool pool-name
R1(config)# ip nat inside source list access-list pool pool-name overload
R1(config)# ip nat inside source list access-list interface interface overload
```

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