# Navigate Between IOS Modes

Move between **User EXEC Mode** and **Privileged EXEC Mode**

```
switch> enable
switch# disable
switch>
```

Move between **Privileged EXEC Mode** and **Global Configuration Mode**

```
switch# configure terminal
switch(config)# exit
switch#
```

# Navigate Between IOS Modes

Move from **any sub-configuration mode** to **Privileged EXEC mode**

```
switch(config-line)# end
switch#
```

```
switch(config-line)# Ctrl+Z
switch#
```

From one sub-configuration mode to another can be moved directly

```
switch(config-line)# interface vlan 1
switch(config-if)#
```

# Navigate Between IOS Modes

**`exit`** can be used to move from any mode to its parent mode.

```
switch(config-line)# exit
switch(config)#
```

```
switch(config-if)# exit
switch(config)#
```

```
switch(config)# exit
switch#
```

```
switch# exit
switch con0 is now available
press RETURN to get started.
```

# IOS - Commands and Keywords Shortening

- Commands and keywords can be shortened to the minimum number of characters that identify a unique selection
  - The **configure** command can be shortened to **conf** because configure is the only command that begins with **conf**.
    - An even shorter version of **con** will not work because more than one command begins with **con**
  - The **terminal** keyword can be shortened to **t** because terminal is the only keyword available in command configure.

```
switch# conf t
switch(config)#
```

```
switch# con t
% Ambiguous command: "con t"
```

# IOS - Hotkeys

| | |
|---|---|
| ? | List available commands |
| Tab | Autocomplete & Check if the current command viable |
| Ctrl + Z | Return to Privileged EXEC Mode |
| Ctrl + Shift + 6 | Cancel Cisco IOS Process |
| Up Arrow / Down Arrow | Allows user to scroll through former commands. |

```
Switch>hello
Translating "hello"...domain server (255.255.255.255) % Name lookup aborted
Switch>
```

# Save the Running Configuration File

- There are two system files that store the device configuration:
  - **startup-config**
    - Stored in Non-volatile Random Access Memory (NVRAM)
    - Contains all of the commands that will be used by the device upon reboot
    - NVRAM does not lose its contents when the device is powered off or restarted
  - **running-config**
    - Stored in Random Access Memory (RAM)
    - Modifying a running configuration affects the operation of a Cisco device immediately
    - It loses all of its content when the device is powered off or restarted
- To save changes made to the running configuration to the startup configuration file use the **copy running-config startup-config** privileged EXEC mode command

```
switch# copy running-config startup-config
```

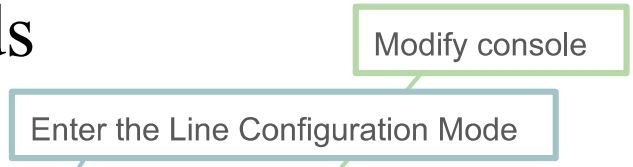
# Configure Hostnames

Edit hostname

```
switch# configure terminal
switch(config)# hostname SW-EC1f
SW-EC1f(config)#
```

- Remember to save in Privileged EXEC Mode after configuring.
- Add **no** in front of any command will remove that command.
  - e.g. switch(config)# **no hostname** *SW-EC1f*

# Configure Passwords




Console Access password

```
SW-EC1f(config)# line console 0
SW-EC1f(config-line)# password passwd1
SW-EC1f(config-line)# login
SW-EC1f(config-line)# exit
SW-EC1f(config)#
```

Privileged EXEC Access password

```
SW-EC1f(config)# enable password passwd2
```

# Encrypt Passwords

- The startup-config and running-config files display **password** in plaintext
  - This is a security threat since anyone can see the passwords used if they have access to these files.
- To encrypt passwords, use the **service password-encryption** global config command
  - The command applies weak encryption (Vigenère cipher) to all unencrypted passwords
  - This encryption applies only to passwords in the configuration file, not to passwords as they are sent over the network
  - The purpose of this command is to keep unauthorized individuals from viewing passwords in the configuration file.
  - **It cannot provide any security at all**

```
SW-EC1f(config)# service password-encryption
```

# Configure Users and their Secrets

- Create a user and set its secret
  - Secret is same with password
  - But it stores the result of MD5 hash function instead

```
SW-EC1f(config)# username user secret passwd3
SW-EC1f(config)# line console 0
SW-EC1f(config-line)# login local
SW-EC1f(config-line)# exit
SW-EC1f(config)#
```

```
SW-EC1f(config)# enable secret passwd4
```

# Banner Messages

Although requiring passwords is one way to keep unauthorized personnel out of a network, it is vital to provide a method for declaring that only authorized personnel should attempt to gain entry into the device

```
SW-EC1f(config)# banner motd !
helloworld hellloflvksdflsdk!
SW-EC1f(config)#
```

The Delimiter can be any character, being used to mark the start and end of message.

# Verify and Monitor Solution

- The Cisco IOS **show** commands are some of the most useful troubleshooting and verification tools included the Cisco IOS
  - Taking advantage of a large variety of options and sub-options, the show command can be used to narrow down and display information about practically any specific aspect of IOS

```
SW-EC1f# show running-config
SW-EC1f# show statup-config
SW-EC1f# show interfaces
SW-EC1f# show arp
SW-EC1f# show version
```

# Command Output Redirection

- The output of a **show** command can be filtered or redirected to a file.
- Redirection is available using a **pipe (|)** character after **any show command**, combined with the following keywords:

| | |
|---|---|
| **`begin`** | Begins unfiltered output of the show command with the first line that contains the regular expression. |
| **`exclude`** | Displays output lines that do not contain the regular expression. |
| **`include`** | Displays output lines that contain the regular expression. |
| **`section`** | Filter a section of output |

# Command Output Redirection - Examples

```
Switch# show running-config | section username
username ccna secret 5 $1$mERr$Bok4KDfVutXOJolNq009M/
Switch# show running-config | section line
line con 0
  login local
line vty 0 4
  login
line vty 5 15
  login
end
Switch#
```

# Cisco Discovery Protocol (CDP)

- A Cisco proprietary protocol
- Collect directly connected neighbor device information
  - hardware, software, device name, …
- Show all neighbors

```
Sw-Lab1# show cdp neighbors
```

- Show details of one neighbor

```
Sw-Lab1# show cdp entry Device-ID
```

# Cisco Discovery Protocol (CDP)

- Disable CDP globally

```
Sw-Lab1(config)# no cdp run
```

- Disable CDP for an interface

```
Sw-Lab1(config)# interface fastEthernet 0/1
Sw-Lab1(config-if)# no cdp enable
```

# Link Layer Discovery Protocol (LLDP)

- IEEE standard protocol
- Similar to CDP
- Not enabled by default

```
Sw-Lab1(config)# lldp run
```

or

```
Sw-Lab1(config)# interface fastEthernet 0/1
Sw-Lab1(config-if)# lldp receive
Sw-Lab1(config-if)# lldp transmit
```

# Link Layer Discovery Protocol (LLDP)

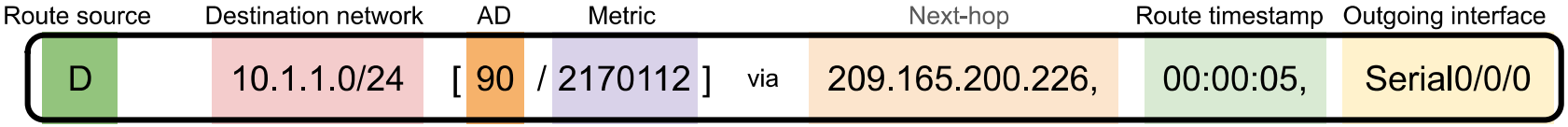
- Show lldp neighbors

```
Sw-Lab1# show lldp neighbors
```

# Command Review

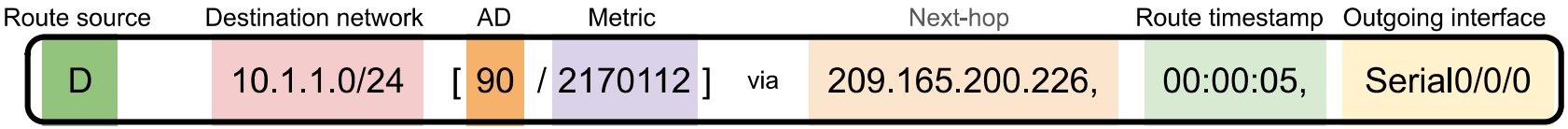
```
switch> enable
switch# conf term
switch(config)# hostname SW-EC1f
SW-EC1f(config)# enable secret passwd1
SW-EC1f(config)# username user secret passwd2
SW-EC1f(config)# line console 0
SW-EC1f(config-line)# login local
SW-EC1f(config-line)# exit
SW-EC1f(config)# banner motd !
helloworld hellloflvksdflsdk!
SW-EC1f(config)# exit
SW-EC1f# cop runn start
```

# Entries in the Routing Table




- **Route Source**: how the route was learned
- **Destination Network**: the destination of the packets
- **Administrative Distance (AD)**
  - The trustworthiness of the route source
  - The lower value, the more preferred route source
- **Metric**
  - The value assigned to reach the remote network.
  - The lower value, the more preferred route
- **Next-hop**: where the router should send to
- **Route Timestamp**: after the route was learned
- **Outgoing Interface**: the exit interface to forward packets

# AD v.s. Metric




- **Administrative Distance (AD)**
  - the trustworthiness of the route source
  - the lower value, the more preferred route source
- **Metric**
  - the value assigned to reach the remote network.
  - the lower value, the more preferred route

# Administrative Distance

| **Route Source** | **Administrative Distance** |
|---|---|
| Connected | 0 |
| Static | 1 |
| EIGRP summary route | 5 |
| External BGP | 20 |
| Internal EIGRP | 90 |
| IGRP | 100 |
| OSPF | 110 |
| RIP | 120 |
| External EIGRP | 170 |
| Internal BGP | 200 |

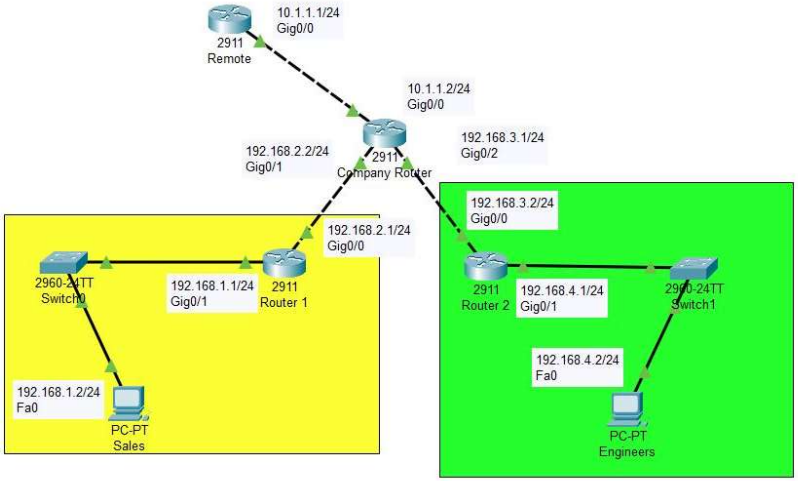
# Static Route: IP Route Command

- Configure static route

```
Router(config) # ip route network-address subnet-mask { ip-address |
exit-intf}
```

| Parameter | Description |
|---|---|
| network-address | ● Destination network address of the remote network to be added to the routing table |
| subnet-mask | ● Subnet mask of the remote network to be added to the routing table<br>● The subnet mask can be modified to summarize a group of networks |
| ip-address | ● Referred to as the next-hop router's IP address<br>● Creates a recursive lookup |
| exit-intf | ● Use the outgoing interface to forward packets<br>● Also referred to as a directly attached static route |

# Static Route Configuration




For **Router1** & **Router2**

**Method #1 (Router1):** exit-interface

**Try it !**

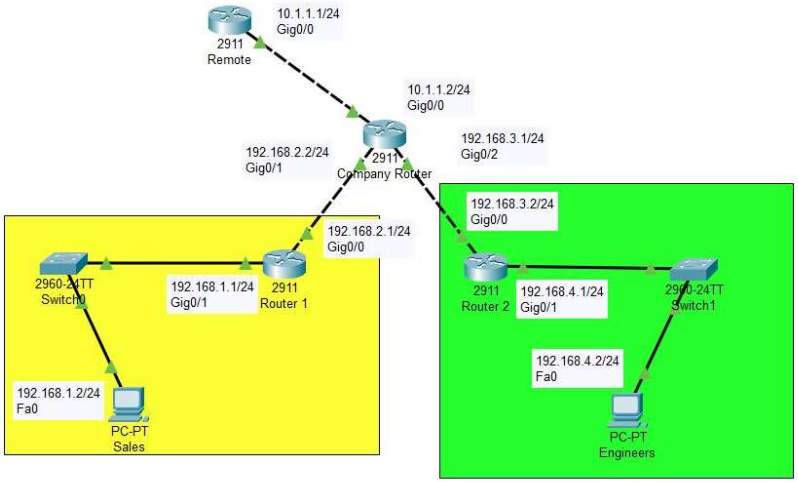
```
R1(config)# ip route 192.168.4.0 255.255.255.0 GigabitEthernet 0/0
```

**Method #2 (Router2):** IP-address

**Try it !**

```
R2(config)# ip route 192.168.1.0 255.255.255.0 192.168.3.1
```

# Static Route Configuration




For **CompanyRouter**

**Method #1 (Engineers):** exit-interface

**Try it !**

```
CompanyR(config)# ip route 192.168.4.0 255.255.255.0 GigabitEthernet
0/2
```

**Method #2 (Sales):** IP-address

**Try it !**

```
CompanyR(config)# ip route 192.168.1.0 255.255.255.0 192.168.2.1
```

# Show IP Route on Router

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP , M - mobile, B -
BGP
D - EIGRP, EX - EIGRP external,  O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
```

# Routing Table for **exit-intf** (Router1)

- **Method #1 exit-intf:** routing table only needs to search <u>once</u>

```
R1# show ip route

……

Gateway of last resort is not set
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.1.0/24 is directly connected, GigabitEthernet0/1
L 192.168.1.1/32 is directly connected, GigabitEthernet0/1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.2.0/24 is directly connected, GigabitEthernet0/0
L 192.168.2.1/32 is directly connected, GigabitEthernet0/0
S 192.168.4.0/24 is directly connected, GigabitEthernet0/0
```

# Routing Table for **IP-Address** (Router2)

- **Method #2 IP-address:** routing table needs to search twice

```
R2# show ip route
```

Try it !

```
……

Gateway of last resort is not set
S 192.168.1.0/24 [1/0] via 192.168.3.1
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.3.0/24 is directly connected, GigabitEthernet0/0
L 192.168.3.2/32 is directly connected, GigabitEthernet0/0
     192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.4.0/24 is directly connected, GigabitEthernet0/1
L 192.168.4.1/32 is directly connected, GigabitEthernet0/1
```

# **Default** Static Route

- Configure default static route

```
Router(config) # ip route 0.0.0.0 0.0.0.0 {ip-address | exit-intf}
```

| Parameter | Description |
|---|---|
| `0.0.0.0 0.0.0.0` | • Matches any network address |
| `ip-address` | • next-hop router's IP address<br>• commonly creates a recursive lookup |
| `exit-intf` | • use the outgoing interface to forward packets<br>• also referred to as a directly attached static route |

# Default Route on Company Router

- Configure default route on Company Router

```
CompanyR(config)# ip route 0.0.0.0 0.0.0.0 gigabitEthernet 0/0
```

Try it !

- Show ip route of Company Router

```
...
192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.3.0/24 is directly connected, GigabitEthernet0/2
L 192.168.3.1/32 is directly connected, GigabitEthernet0/2
S 192.168.4.0/24 [1/0] via 192.168.3.2
S* 0.0.0.0/0 is directly connected, GigabitEthernet0/0
```

# Default Route on Router 1 and Router 2

- Configure default route on Route 1

```
R1(config)# ip route 0.0.0.0 0.0.0.0 gigabitEthernet 0/0
```


- Show ip route of Route 1

```
...
C 192.168.2.0/24 is directly connected, GigabitEthernet0/0
L 192.168.2.1/32 is directly connected, GigabitEthernet0/0
S 192.168.4.0/24 is directly connected, GigabitEthernet0/0
S* 0.0.0.0/0 is directly connected, GigabitEthernet0/0
```

# Dynamic Routing: Router RIP Configuration Mode

- Enable RIP

```
Router(config) # router rip

Router(config-router) #
```

# Advertise Networks

- Start RIP routing

```
Router(config-router) # network {subnet}
```

- RIPv1 is a classful routing protocol for IPv4.
- Classful routing protocol: In contrary to classless, a protocol that does not support ip except A, B, C classes
- Therefore, if a subnet address is entered, the IOS automatically converts it to the classful network address.
    - For example, entering the network 192.168.1.32 command would automatically be converted to network 192.168.1.0 on the running configuration file.
    - No error message, but IOS corrects the input and enters the classful network address.

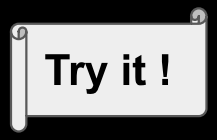
# Network Class

- A: 0.0.0.0 ~ 127.0.0.0 (subnet /8)
- B: 128.0.0.0 ~ 191.255.0.0 (subnet /16)
- C: 192.0.0.0 ~ 223.255.255.0 (subnet /24)
- D: 224.0.0.0 ~ 239.255.255.255 (multicast)
- E: 240.0.0.0 ~ 255.255.255.255 (reserved)

# Dynamic Routing: RIP

```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0
```


```
R2(config)# router rip
R2(config-router)# network 192.168.3.0
R2(config-router)# network 192.168.4.0
```



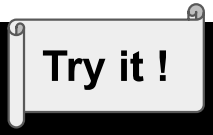

```
CompanyR(config)# router rip
CompanyR(config-router)# network 192.168.2.0
CompanyR(config-router)# network 192.168.3.0
```

# Dynamic Routing: RIP

```
R2# show ip route
……
R    192.168.1.0/24 [120/2] via 192.168.3.1, 00:00:09, GigabitEthernet0/0

R    192.168.2.0/24 [120/1] via 192.168.3.1, 00:00:09, GigabitEthernet0/0

192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.3.0/24 is directly connected, GigabitEthernet0/0
L 192.168.3.2/32 is directly connected, GigabitEthernet0/0
192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.4.0/24 is directly connected, GigabitEthernet0/1
L 192.168.4.1/32 is directly connected, GigabitEthernet0/1
```



# Propagate a Default Route

- Default static route be advertised to all other routers in the RIP routing domain
- Tell others in the same RIP routing domain where to connect Internet

```
Router(config)# ip route 0.0.0.0 0.0.0.0 {ip-address | exit-intf}
Router(config)# router rip
Router(config-router)# default-information originate
```

# Propagate a Default Route

- Configure default route on Company Router


```
CompanyR(config)# ip route 0.0.0.0 0.0.0.0 gigabitEthernet 0/0
CompanyR(config)# router rip
CompanyR(config-router)# default-information originate
```

# Propagate a Default Route

- Show ip route

```
CompanyR# show ip route
...

     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.3.0/24 is directly connected, GigabitEthernet0/1
L 192.168.3.1/32 is directly connected, GigabitEthernet0/1
S* 0.0.0.0/0 is directly connected, GigabitEthernet0/2
```

# Propagate a Default Route

- Show ip route

```
R1# show ip route
...
R 192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:23, GigabitEthernet0/0
R 192.168.4.0/24 [120/2] via 192.168.2.2, 00:00:23, GigabitEthernet0/0
R* 0.0.0.0/0 [120/1] via 192.168.2.2, 00:00:05, GigabitEthernet0/0
```

# Enable and RIPv2

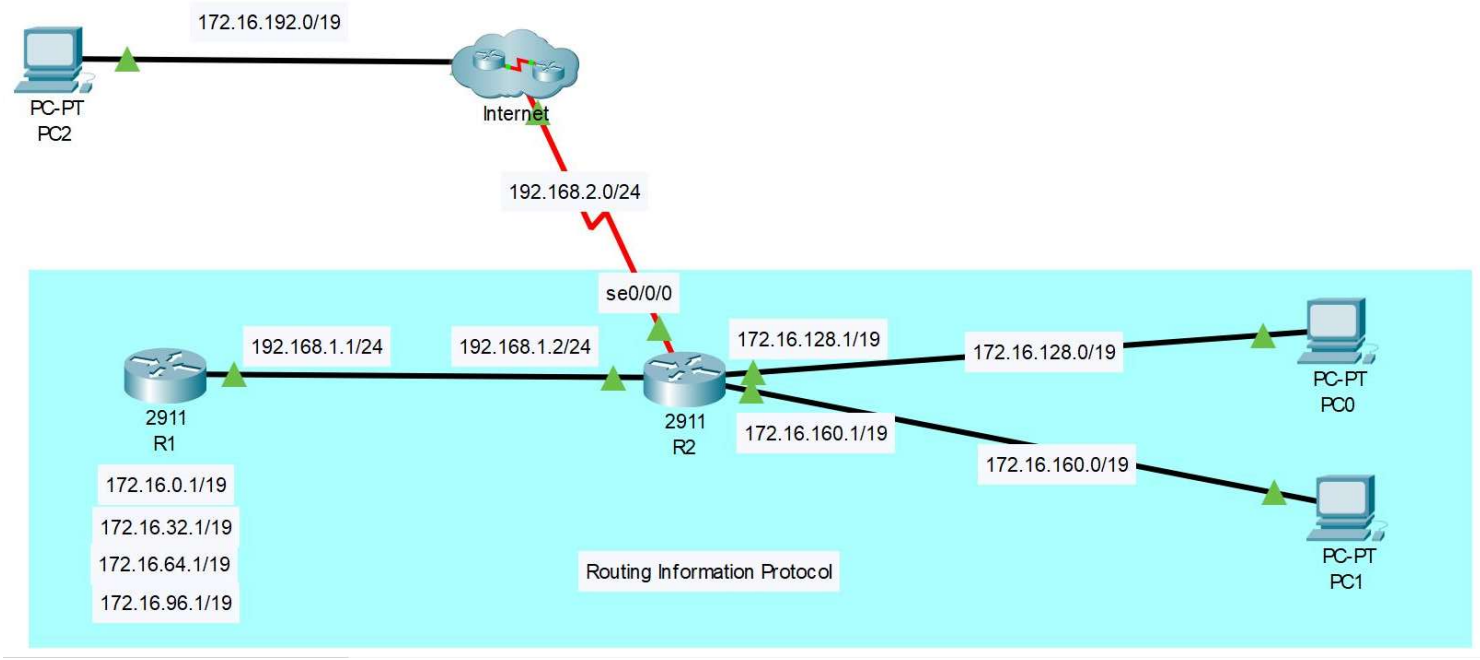
- Enable RIPv2
  - Make RIP a **classless** routing protocol

```
Router(config) # router rip

Router(config-router) # version 2
```

# Auto Summarization

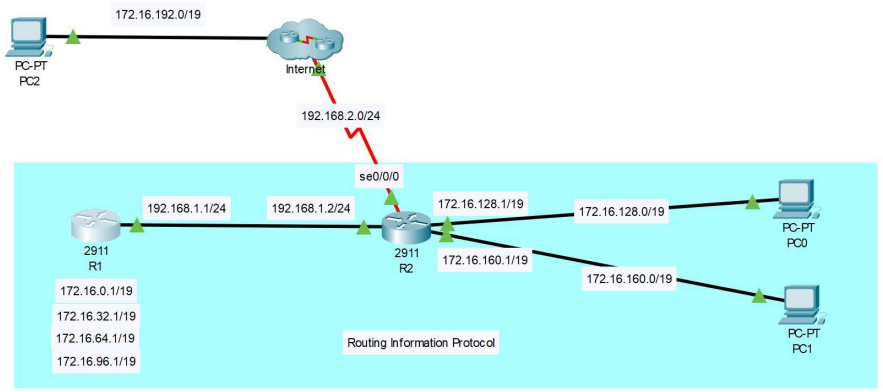
- Auto summarization is a feature which allows RIP to summarize its routes to their classful networks automatically.



# Auto Summarization

- Default enable

```
R1# show ip route
...
172.16.0.0/16 is variably subnetted, 9 subnets, 3 masks
R 172.16.0.0/16 [120/1] via 192.168.1.2, 00:00:21, GigabitEthernet0/0
C 172.16.0.0/19 is directly connected, Loopback0
L 172.16.0.1/32 is directly connected, Loopback0
```



# Disable Auto Summarization

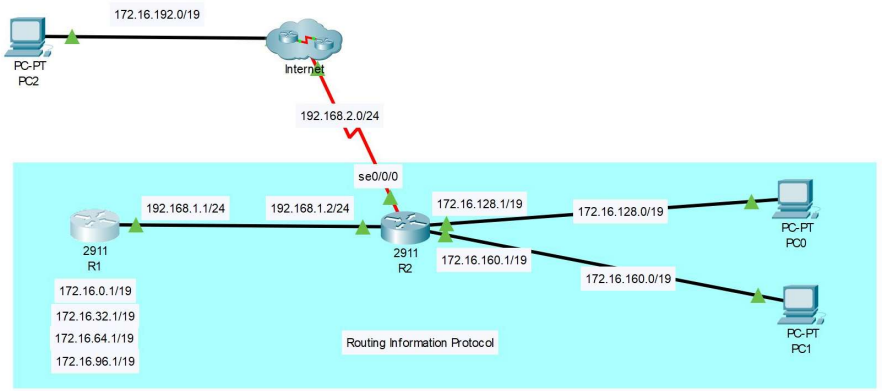
- Disable auto summarization on R2

```
R2(config)# router rip
R2(config-router)# no auto-summary
```

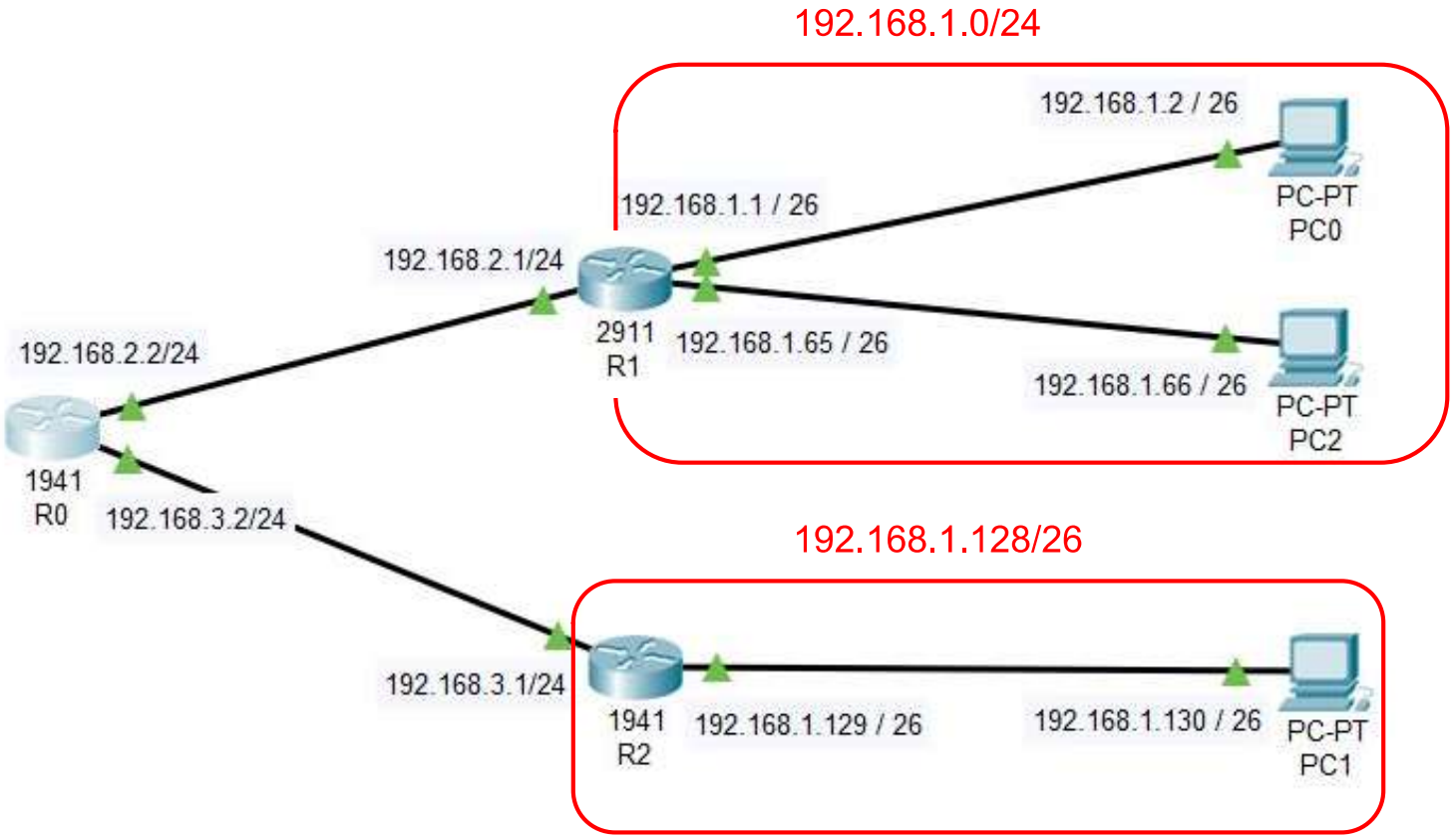

- This command has no effect when using RIPv1
- When automatic summarization has been disabled, RIPv2 no longer summarizes networks to their classful address at boundary routers

# Disable Auto Summarization

```
R1# show ip route
...
C 172.16.96.0/19 is directly connected, Loopback3
L 172.16.96.1/32 is directly connected, Loopback3
R 172.16.128.0/19 [120/1] via 192.168.1.2, 00:00:13, GigabitEthernet0/0
R 172.16.160.0/19 [120/1] via 192.168.1.2, 00:00:13, GigabitEthernet0/0
```



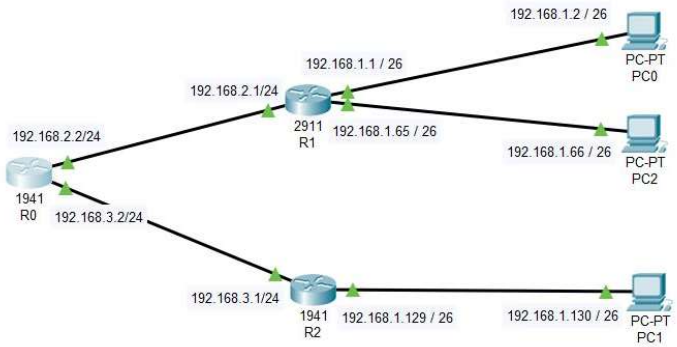
# Longest Prefix Matching



# Longest Prefix Matching

```
R0# show ip route
……
Gateway of last resort is not set

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
S       192.168.1.0/24 [1/0] via 192.168.2.1
S       192.168.1.128/26 [1/0] via 192.168.3.1
……
```



# Supplement: Timer

- Update Timer
  - How long to send route update packet
- Invalid Timer
  - How long to mark a route invalid since last update
- Hold-Down Timer
  - How long hold-down time be
  - A route enters hold-down state when it's **unreachable**
    - No update
    - Will be advertised as unreachable
- Flush Timer
  - How long to remove a route since last update
  - Starting at the same time as invalid timer

# Solution: Passive Interfaces

- Configure passive interface

Method #1: Passive all and *no* specific

```
Router(config)# router rip
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface Gigabit 0/1
```

Method #2: Passive specific

```
Router(config-router)# passive-interface GigabitEthernet 0/0
```

- The command stops routing updates out the specified interface

# Types of STP

[1] Cisco IOS use PVST+ as default mode

| Protocol | Standard | Resources Needed | Convergence | Tree Calculation |
|---|---|---|---|---|
| STP | 802.1D | Low | Slow | All VLANs |
| PVST+ [1] | Cisco | High | Slow | Per VLAN |
| RSTP | 802.1w | Medium | Fast | All VLANs |
| Rapid PVST+ | Cisco | Very high | Fast | Per VLAN |
| MSTP | 802.1s | Medium or high | Fast | Per Instance |

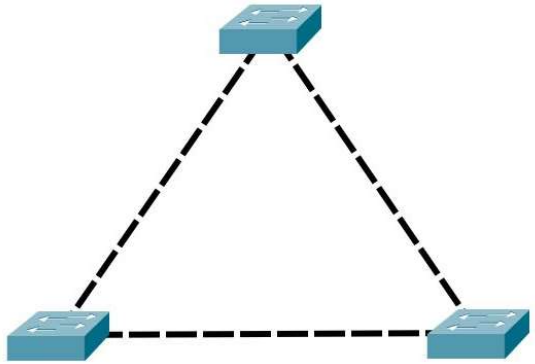
```
(config)# spanning-tree mode mode
```

# BPDU

- Bridge Protocol Data Units (BPDUs) frames contains
  - switch ID, MAC address, switch port priority, switch port cost, etc.
- Send out as multicast address at `01:80:c2:00:00:00`
- Types of BPDU
  - Configuration BPDU (CBPDU)
  - Topology Change Notification (TCN)
  - Topology Change Acknowledgement (TCA)
  -

# STP Process

- Elect a root bridge
- Place root interface to forwarding state
- Choose root port for each non-root bridge
- Choose designated port for each remaining link
- Put all other ports into blocking state

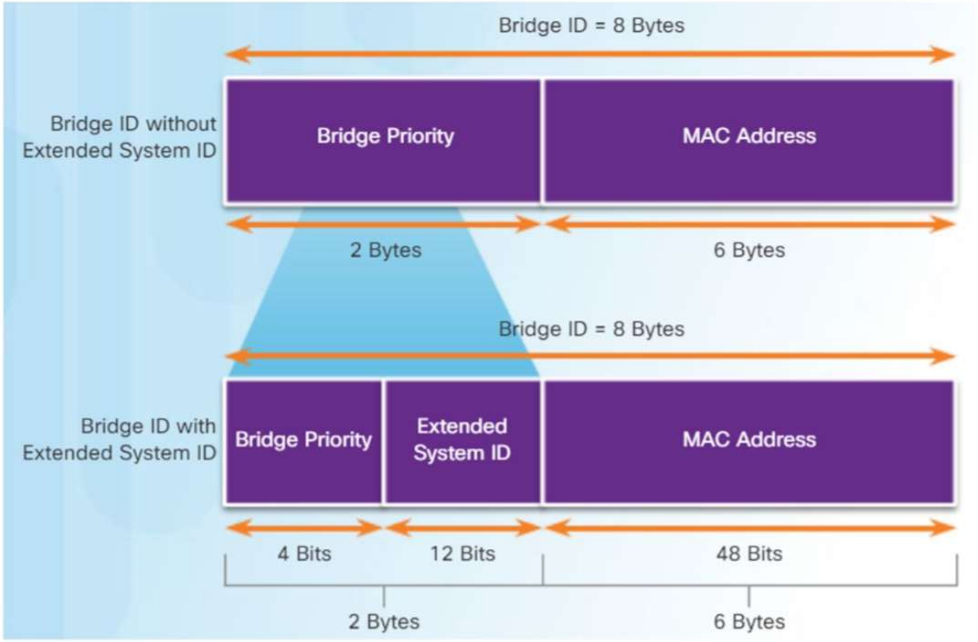


# Port Roles

- Root port
  - Ports on non-root switches with the best cost path to root bridge. These ports forward data to the root bridge.
- Designated port
  - Ports on root and designated switches. All ports on the root bridge will be designated.
- Blocked port
  - All other ports to bridges or switches are in a blocked state. Access ports going to workstations or PCs are not affected.

# Bridge Priority

- Bridge priority only allows to be in multiple of **4096**



# Bridge Priority

- Two ways to change bridge priority

```
(config)# spanning-tree vlan vlan-id root [ primary | secondary ]
(config)# spanning-tree vlan vlan-id priority value
```

```
switch# show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID Priority       32769
          Address        0025.b4c1.b400
          This bridge is the root
          Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID    Priority       32769  (priority 32768 sys-id-ext 1)
          Address        0025.b4c1.b400
          Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
          Aging Time     300 sec
```

priority + sys-id-ext

# STP Cost

- Configure interface cost

```
(config)# interface interface
(config-if)# spanning-tree vlan value cost value
```

- Show spanning tree cost

```
switch# show spanning-tree vlan 1
…
Interface  Role    Sts    Cost     Prio.Nbr     Type
---------  -----   -----  -----    --------     --------
Fa0/1       Root   FWD    19       128.8        P2p
Fa0/2       Desg   FWD    40       128.2        P2p
...
```

| Link Speed | Cost |
|---|---|
| 10 Gb/s | 2 |
| 1 Gb/s | 4 |
| 100 Mb/s | 19 |
| 10 Mb/s | 100 |

# Root port selection

1. Lowest cost
2. Lowest neighbor bridge ID
3. Lowest neighbor port priority
4. Lowest neighbor physical port number

# Designated port selection

1. Lowest accumulated cost to the root bridge
2. Lowest bridge ID

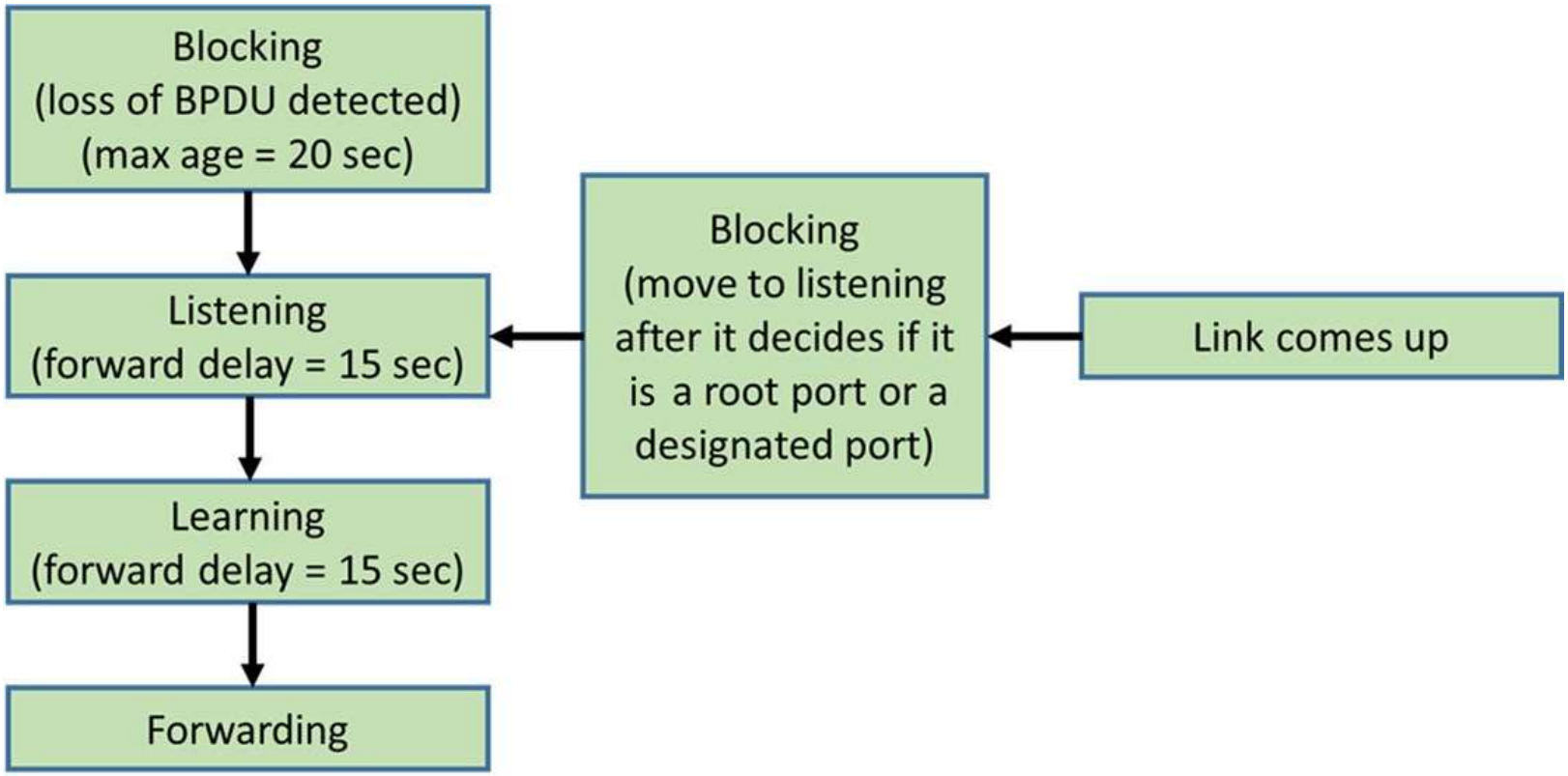
# Port States

| STATUS | Disabled | Blocking | Listening | Learning | Forwarding |
|---|---|---|---|---|---|
| Receive BPDU | X | O | O | O | O |
| Send BPDU | X | X | O | O | O |
| Learn MAC | X | X | X | O | O |
| Forwarding | X | X | X | X | O |
| Duration | Until no shutdown | Until topology changed | Forward Delay (default 15s) | Forward Delay (default 15s) | Until shutdown or not root/designated port |

# Topology Change

- Send TCN (topology change notification) on its root port
- Upstream bridge responds an TCA (topology change acknowledgement)
- Root bridge send BPDU with TC bit set
- Lower their MAC table address aging time (from 300s to 15s)

# Port States


Blocking
(loss of BPDU detected)
(max age = 20 sec)
Listening
(forward delay = 15 sec)
Learning
(forward delay = 15 sec)
Forwarding
Blocking
(move to listening
after it decides if it
is a root port or a
designated port)
Link comes up

# STP Convergence

- 50 seconds for going through all state

# STP Timer

- Hello Timer (default 2 sec)
  - The frequency of sending BPDU
- Max Age Timer (default 20 sec)
  - The maximum length of time a switch saves BPDU information
- Forward Delay Timer (default 15 sec)
  - The time spent on the listening and learning states

# STP Timer

- The diameter is the maximum number of switches that data must cross to connect any two switches
- Diameter configuration (deprecated on Packet Tracer)

```
(config)# spanning-tree vlan vlan-id root primary diameter size
```

| Diameter | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| Hello Time | 2 | 2 | 2 | 2 | 2 | 2 |
| Max Age | 10 | 12 | 14 | 16 | 18 | 20 |
| Forward Delay | 7 | 9 | 10 | 12 | 13 | 15 |

# STP Enhancements

- PortFast
- BPDU Guard
- BPDU Filter
- Root Guard

# PortFast

- Allow a port to enter from blocking to forwarding state immediately, bypassing the listening and learning states
- You will see warning as below after configuring PortFast

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

- Need to wait for convergence to communicate with a port on which the PortFast feature is disabled (normally a port connected to another switch).
- Never enable PortFast on a Trunk port!

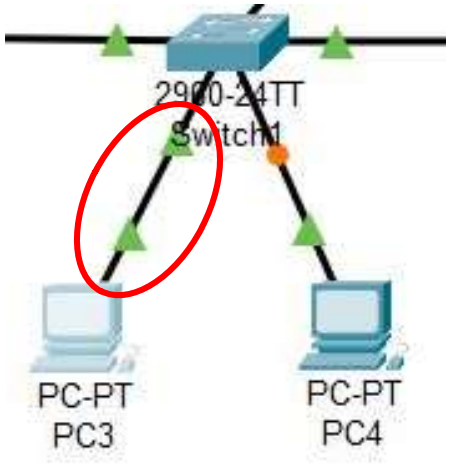
# PortFast Configuration

- Configure PortFast on a switch port

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

- Enable PortFast on all nontrunking interfaces

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




with Portfast without Portfast

# BPDU Guard

- If BPDU guard is enabled, it puts the port in an **err-disabled** state when receiving a BPDU
  - This will effectively shut down the port
- The BPDU guard feature provides a secure response to invalid configurations because you must manually put the interface back into service
  - **shutdown** the interface and **no shutdown** it back to recover
- Turn on BPDU Guard

```
(config)# interface interface
(config-if)# spanning-tree bpduguard enable
```

# BPDU Filter

- Similar to BPDU Guard, but just “filter” it
- Two different configuration styles, with different behaviors
  - Configure it globally
  - Configure it on the specified port

# BPDU Filter

- When configured globally
  - Affect all operational PortFast ports
  - If BPDUs are seen on the port
    1. The port loses its PortFast status
    2. BPDU filtering is disabled
    3. STP sends and receives BPDUs as any other STP port
  - Upon startup, the port transmits ten BPDUs. If this port receives any BPDUs during that time, PortFast and BPDU filtering are disabled
- When configured on an individual port
  - Ignore all BPDUs received
  - Send no BPDUs

# BPDU Filter Configuration

- Enable BPDU filtering globally (deprecated on Packet Tracer)

```
(config)# spanning-tree portfast bpdufilter default
```

- Enable BPDU filtering on a specific switch port (deprecated on Packet Tracer)

```
(config)# interface interface
(config-if)# spanning-tree bpdufilter enable
```

- To verify the configuration

```
# show spanning-tree summary totals
# show spanning-tree interface interface detail
```

# Root Guard

- Root guard is configured on a per-port basis
- If there is a superior BPDU received on the port, root guard does not take the BPDU into account and so puts the port into **root inconsistent state**
- Root guard configuration

```
(config)# interface interface
(config-if)# spanning-tree guard root
```

```
switch# show spanning-tree inconsistentports
Name                 Interface             Inconsistency
-------------------- --------------------- ------------------
VLAN0001             FastEthernet0/4       Root Inconsistent

Number of inconsistent ports (segments) in the system : 1
```

# VLAN Tagging

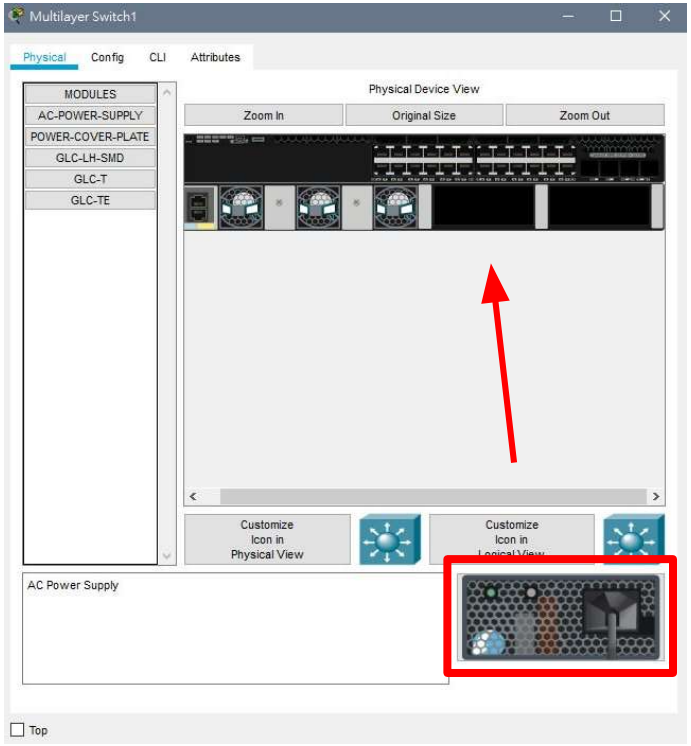
- Review: Tagging of access/trunk interface
- Access:
  - expect receive untagged frame
    - drop tagged frame when received
  - remove frame’s tag before sending out
- Trunk
  - expect receive tagged frame.
    - if frame is untagged, consider it is inside native VLAN
  - add frame’s tag before sending out

# Switched Virtual Interface (SVI)

- Switch does not have a physical interface to which an IP address can be assigned.
  - IP is configured on a virtual interface called switched virtual interface (SVI).
- Switched Virtual Interface (SVI)
  - A single SVI can only be mapped to a VLAN.
  - A SVI cannot be activated unless that VLAN associated with at least one active physical port.
  - SVI provides the Layer 3 processing for packets from all active physical ports associated with the VLAN.
    - routing packet from/to other SVI
    - no need physical router for inter-VLAN routing

# 建立實驗環境 (2/4)

- 新增一個機器時，可能需要手動拖曳電源模組



# 實驗所需指令 - VLAN

- 建立 VLAN

```
switch# configure terminal
switch(config)# vlan vlan-num ──► 可用的 vlan-num 參考後方投影片
switch(config-vlan)# name vlan-name (optional)
```

- 將 Port 指派給 VLAN

```
(config)# interface type interface_number
(config-if)# switchport mode access ──► access link
(config-if)# switchport access vlan vlan-num
```

# 實驗所需指令 - VLAN

- Cisco IOS 可用的 VLAN 範圍

| VLANs | Range | Usage |
|---|---|---|
| 0, 4095 | Reserved | For system use only. You cannot see or use these VLANs. |
| 1 | Normal | Cisco default. You can use this VLAN but you cannot delete it.<br>Cisco will use this VLAN to send Control Plane Traffic<br>(like CDP, BPDU) |
| 2-1001 | Normal | For Ethernet VLANs; you can create, use, and delete these VLANs. |
| 1002-1005 | Normal | You cannot delete VLANs 1002-1005. (Cisco defaults for FDDI and Token Ring) |
| 1006-4094 | Extended | For Ethernet VLANs only. |

# 實驗所需指令 - 情境示範

```
switch(config)# vlan 10
switch(config-vlan)# name department-A
...
switch(config)# vlan 20
switch(config-vlan)# name department-B
...
switch(config)# interface fastEthernet 0/1
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 10
...
switch(config)# interface fastEthernet 0/4
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 20
...
```

# 實驗所需指令 - Trunk

- 設定 Trunk link

```
(config)# interface type interface_number
(config-if)# switchport trunk encapsulation {isl|dot1q|negotiate}
! 預設為 negotiate，會視另一端協商使用 ISL 或是 802.1Q
! 有些較新的 Switch 不支援 ISL 會沒有這個指令，或是只有 dot1q 的選項
(config-if)# switchport trunk allowed vlan {vlan-list|except vlan-list|all}
(config-if)# switchport mode trunk ——→ trunk link
```

- 修改已經設定完的 Trunk link

```
(config-if)# switchport trunk allowed vlan {add|remove} vlan-list
```

# 實驗所需指令 - 情境示範

```
switch(config)# interface gigabitEthernet 0/1
switch(config-if)# switchport trunk allowed vlan 10,20
switch(config-if)# switchport mode trunk
! switch 2960 上沒有 encapsulation 的指令
...
L3switch(config)# interface gigabitEthernet 1/0/1
L3switch(config-if)# switchport trunk encapsulation dot1q
L3switch(config-if)# switchport trunk allowed vlan 10,20
L3switch(config-if)# switchport mode trunk
! L3 switch 3650 上有 encapsulation 的指令, 但只有 dot1q 的選項
...
```

# 實驗所需指令 - Inter-VLAN Routing

- 在 Router/L3 Switch 建立 VLAN

```
(config)# vlan vlan-num
(config-if)# name vlan-name (optional)
```

- 設定 VLAN Interface（= SVI），並設定 VLAN Gateway 的 IP

```
(config)# interface vlan vlan-num
(config-if)# ip address ip netmask
```

- 在 L3 Switch 上啟用 Routing

```
(config)# ip routing
```

# 實驗所需指令 - 情境示範

```
L3switch(config)# vlan 10
...
L3switch(config)# vlan 20
...
L3switch(config)# interface vlan 10
L3switch(config-if)# ip address 10.1.10.254 255.255.255.0
...
L3switch(config)# interface vlan 20
L3switch(config-if)# ip address 10.1.20.254 255.255.255.0
...
L3switch(config)# ip routing
```

- 最後再將 PC 上的 Gateway 設為那個 VLAN 下對應的 Gateway

# 實驗所需指令 - Native VLAN

- 設定 Native VLAN，Cisco Native VLAN 預設為 VLAN 1

```
(config)# interface type interface_number
(config-if)# switchport trunk encapsulation dot1q
(config-if)# switchport trunk native vlan vlan-num
```

# 實驗所需指令 - 情境示範

```
switch(config)# interface fastEthernet 0/5
switch(config-if)# switchport trunk native vlan 10
switch(config-if)# switchport mode trunk
```

- 就算 Hub 不支援 VLAN，現在部門 A2 與部門 A 可以直接透過 L2 溝通了

# 實驗所需指令(補充) - Dynamic trunking

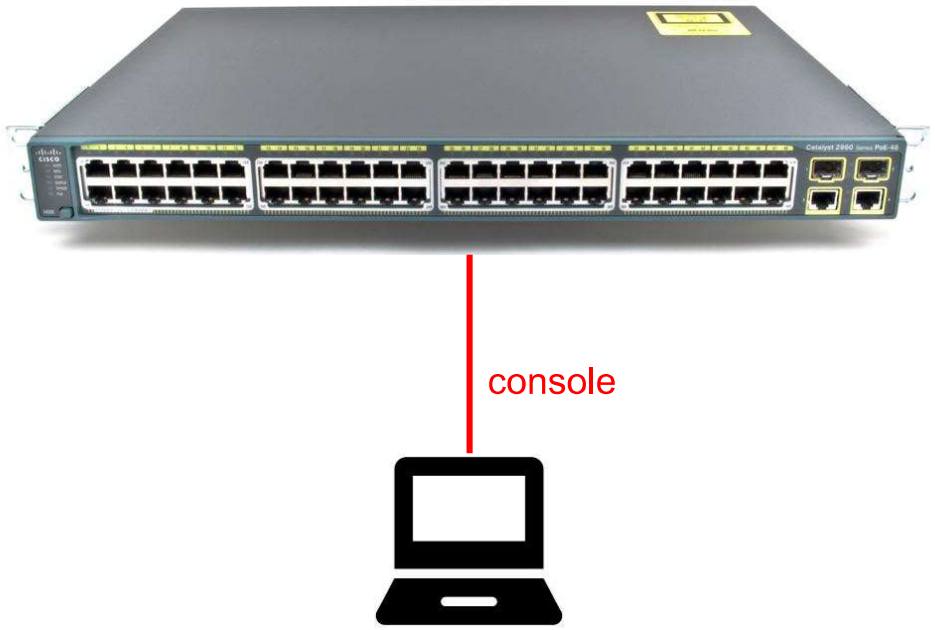
```
(config)# ...前略
(config-if)# switchport mode dynamic {desirable|auto}
```

- 除了前面提到的 trunk mode 與 access mode，還有 dynamic mode 可以動態協商是要使用 access 或是 trunk
  - dynamic desirable，port 會**主動**嘗試將鏈路轉為 trunking mode
  - 預設為 dynamic auto，**被動**協商，僅在另一端交換器主動要求時才會轉為trunking mode
- 有興趣可以自己查關鍵字："cisco dtp"、"dynamic trunking protocol"

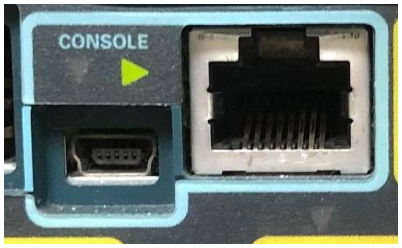
# 附錄: 檢驗指令

- **show running-config**
- **show interfaces** [**status** | **trunk**]
- **show vlan id** *vlan-id*
- **show vlan** [**brief**]
- **show interface** *type member/module/name* **switchport**
- **show ip route**

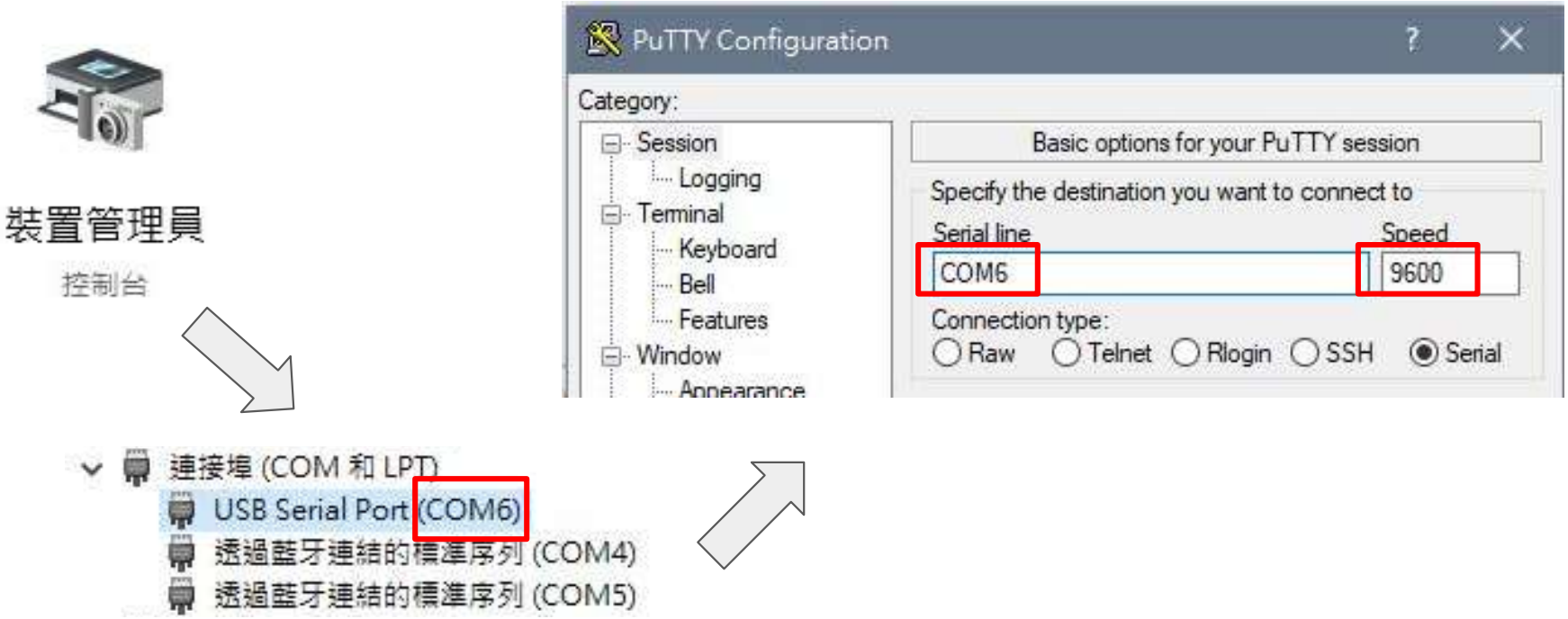
# Try the Real Switch



# Try the Real Switch - Console

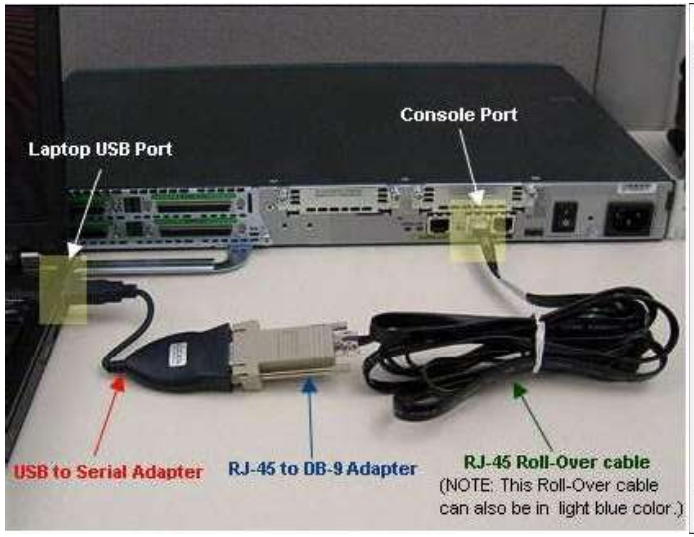


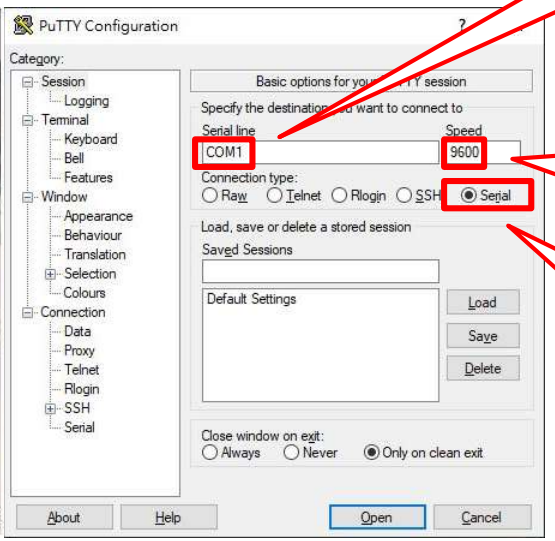
# Try the Real Switch - Console



# How to Access a Switch - Console

- Directly connect the ethernet cable to the console port on the switch.
  - No need to be able to access internet.
  - Filled in the transmission speed (bit rate) according to the brand and model.
    - e.g. 300, 1200, 2400, 9600, 115200, 19200 bit/s



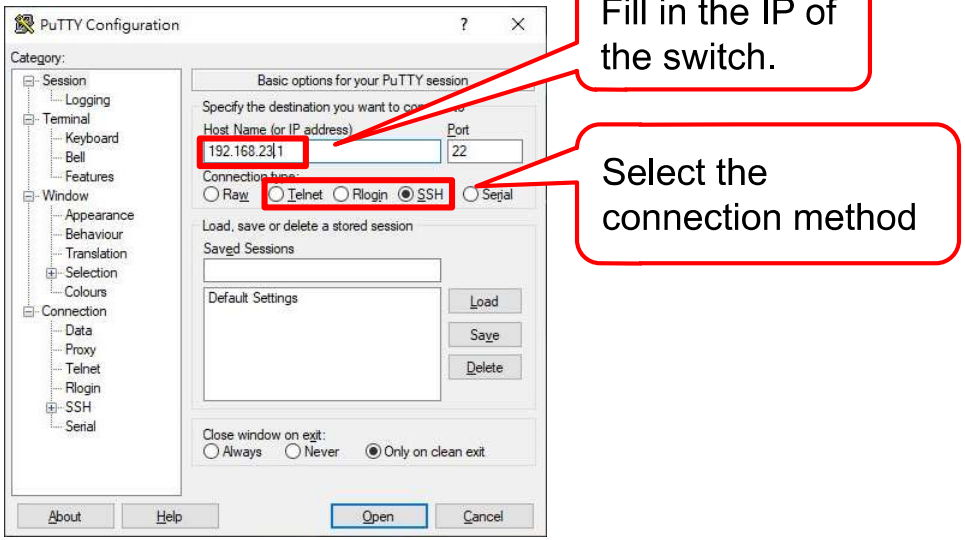

Select the connected interface

Fill in the transmission speed

Select Connection Type ‘Serial’

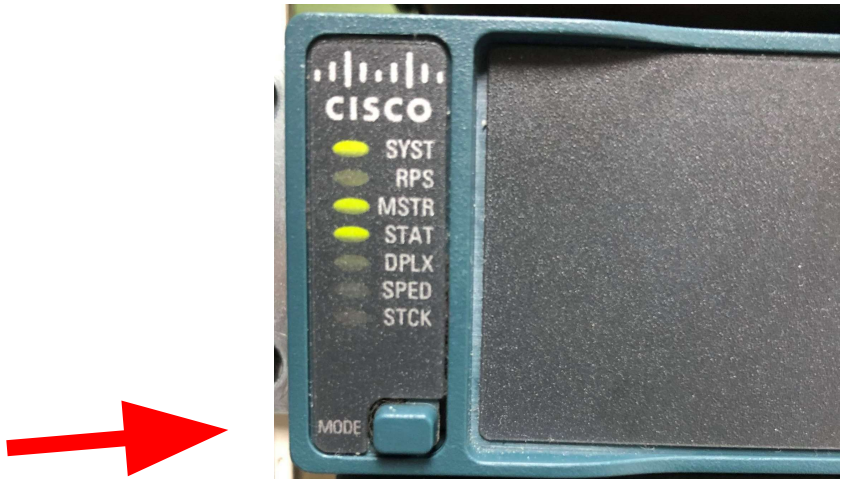
# How to Access a Switch - VTY (Virtual Teletype Terminal)

- Connect to the administration interface by network connection.
  - The basic networking must be configured.
  - e.g. ssh, telnet



# Try the Real Switch - Reset

- Unplug the power cord
- Plug in the power cord and immediately hold down the“MODE”button for a while



# Try the Real Switch - Reset

- Mount flash file system

```
The system has been interrupted ...
...

switch: flash_init
```

- Show files

```
switch: dir flash:

Directory of flash:/
   13  drwx         192   Mar 01 1993 22:30:48  c2960-lanbase-mz.122-25.FX
   11  -rwx        5825   Mar 01 1993 22:31:59  config.text
   18  -rwx         720   Mar 01 1993 02:21:30  vlan.dat
```

# Try the Real Switch - Reset

- Delete config.text and vlan.dat

```
switch: delete flash:config.text
switch: delete flash:vlan.dat
```

- Boot

```
switch: boot
...
Continue with the configuration dialog? [yes/no]:  N

Switch>
```

# Try the Real Switch - SSH Configuration

- Set a Switched Virtual Interface (SVI) to switch

```
switch(config)# interface vlan 1
switch(config-if)# ip address 192.168.<Team-ID>.69 255.255.0.0
switch(config-if)# no shutdown
```

- Create a user

```
switch(config)# username username secret passwd
```

# Try the Real Switch - SSH Configuration

- Configure hostname and domain name

```
switch(config)# hostname hostname
CCNA-01(config)# ip domain-name domain-name
```

- Configure RSA key pair

```
CCNA-01(config)# crypto key generate rsa
! Choose modulus length = 1024
```

# Try the Real Switch - SSH Configuration

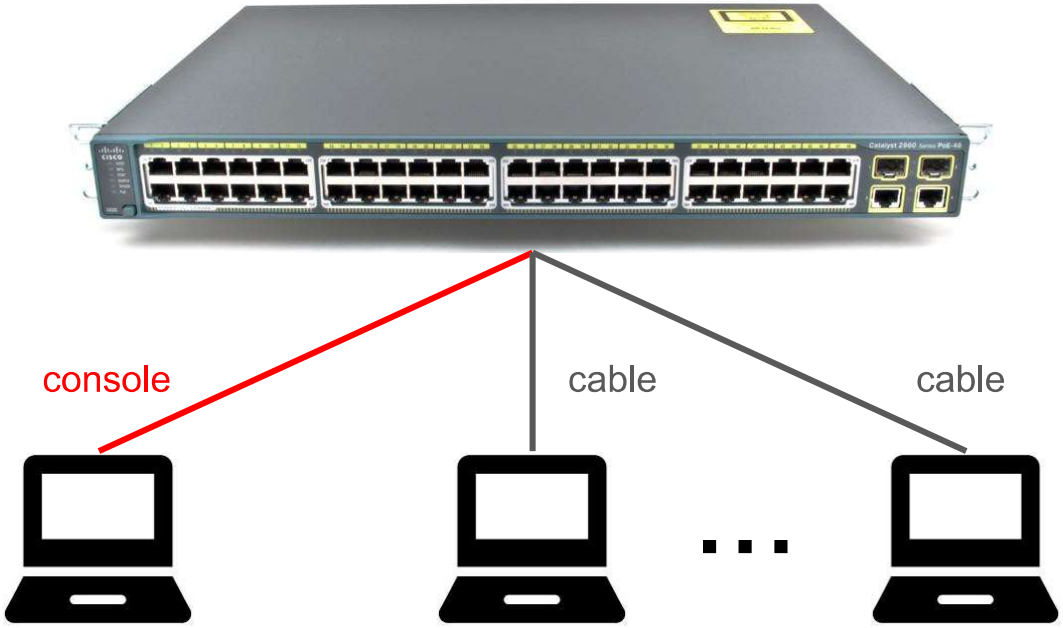
- Choose SSH version

```
CCNA-01(config)# ip ssh version 2
```

- Allow users to login by SSH
  - Allow SSH only and use user authentication

```
CCNA-01(config)# line vty 0 15
CCNA-01(config-line)# transport input ssh
CCNA-01(config-line)# login local
CCNA-01(config-line)# exit
```

# Try the Real Switch - Cable



# Try the Real Switch - Cable

1. Plug in the ethernet cable.
2. Set 192.168.<Team_ID>.x/16 on your laptop.
3. Gateway will not be used for now, set it to 192.168.<Team_ID>.254

# Try the Real Switch - SSH

- Ping to check connectivity
- Connect to the switch
  - Use PuTTY if you have
  - PowerShell / CMD: `C:\> ssh -l username 192.168.<Team-ID>.69`

# Try the Real Switch - SSH issue

- Unable to negotiate with `x.x.x.x` port `22`: no matching cipher found

```
C:\> ssh -l username 192.168.<Team-ID>.69 -c cipher
```

- Their offer: diffie-hellman-group1-sha1
  - Create a file `C:\User\<username>\.ssh\config`
  - Copy the content below to it and save

```
HostKeyAlgorithms +ssh-rsa
PubkeyAcceptedKeyTypes +ssh-rsa
KexAlgorithms +diffie-hellman-group1-sha1
Ciphers +3des-cbc
```

# Force Other Users to Logout

- Check current connections

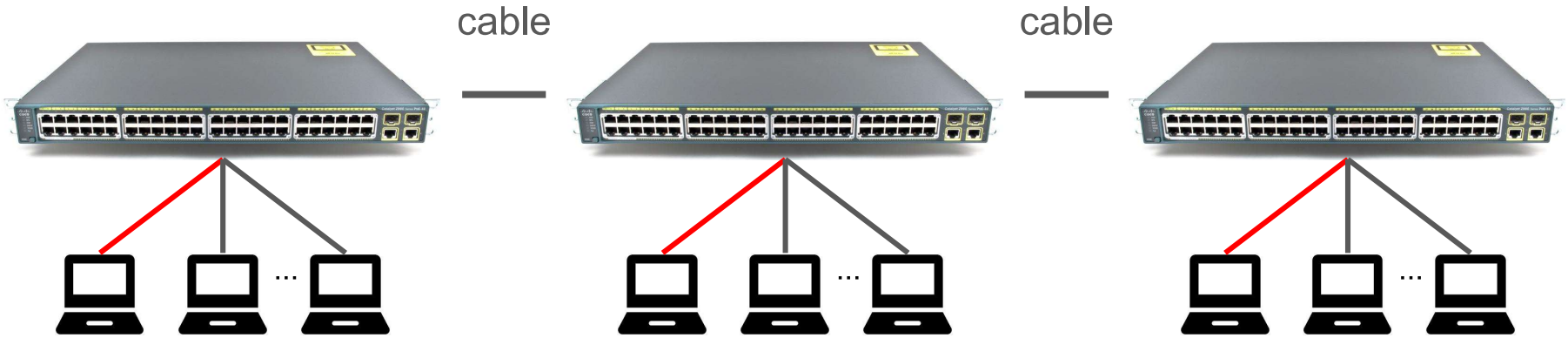
```
CCNA-01# show users
    Line       User   Host(s)       Idle       Location
* 0 con 0            idle          00:00:00
  2 vty 0     user2  idle          00:00:45
  3 vty 1     user3  idle          00:00:23
```

- Choose a lucky number and kick him/her out!

```
CCNA-01# clear line vty x
[confirm]
 [OK]
```

# Try the Real Switch - Interconnection

- Connect to your neighbor team



# Try the Real Switch - Interconnection

- Use CDP to check your neighbor(s)

```
CCNA-01# show cdp neighbors
CCNA-01# show cdp entry Device-ID
```

- Ping your neighbor(s)

```
CCNA-01# ping 192.168.x.y
```

- Use Wireshark to see what can be captured

# Try the Real Switch - Password Recovery

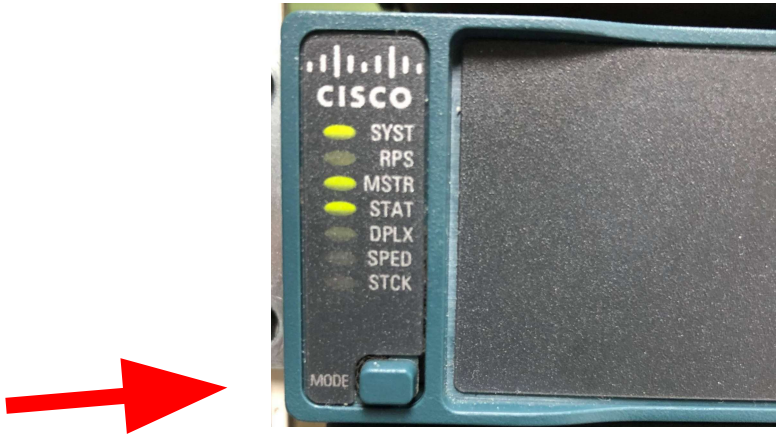
- Set a password for console port and save

```
CCNA-01(config)# line console 0
CCNA-01(config-line)# password passwd
CCNA-01(config-line)# login
CCNA-01(config-line)# ^z
CCNA-01# copy running-config startup-config
```

- Forgot password...

# Try the Real Switch - Password Recovery

- Unplug the power cord
- Plug in the power cord and immediately hold down the“MODE”button for a while



# Try the Real Switch - Password Recovery

- Mount flash file system

```
The system has been interrupted ...
...

switch: flash_init
```

- Show files

```
switch: dir flash:

Directory of flash:/
   13  drwx         192   Mar 01 1993 22:30:48  c2960-lanbase-mz.122-25.FX
   11  -rwx        5825   Mar 01 1993 22:31:59  config.text
   18  -rwx         720   Mar 01 1993 02:21:30  vlan.dat
```

# Try the Real Switch - Password Recovery

- Rename config.text

```
switch: rename flash:config.text flash:config.bak
```

- Boot

```
switch: boot
```

# Try the Real Switch - Password Recovery

- Enter no to reject initial configuration dialog

```
In order to access the device manager, ...
...
Would you like to enter the initial configuration dialog?
[yes/no]: no

Switch>
```

- Restore config.text and running-config

```
Switch> enable
Switch# copy running-config startup-config
Switch# copy flash:config.bak flash:config.text
Switch# copy startup-config running-config
```

# Try the Real Switch - Password Recovery

- Change the password

```
CCNA-01(config)# line console 0
CCNA-01(config-line)# password passwd
CCNA-01(config-line)# login
CCNA-01(config-line)# exit
CCNA-01(config)#
```

- Save to startup-config

```
CCNA-01# copy running-config startup-config
```