Livelli di privilegio del modello Cisco

1° livello **user**
2° livello **privileged**
3° livello **global**

Per sapere in quale livello mi trovo devo guardare il carattere e il simbolo.
1° livello simbolo '**>**'
2° livello simbolo '**#**'
3° livello simbolo **'config'**

Per passare dal livello 1 al livello 2 c'è il comando **'enable**'(en)
Per passare dal livello 2 al livello 3 c'è il comando '**configure terminal**'(conf t)
Per passare dal livello 3 al livello 2 c'è il comando '**exit**'(ex)
Per passare dal livello 2 al livello 1 c'è il comando '**disable**'

COMANDI IOS

Comandi SHOW: servono per verificare delle cose del mio apparato.
Il comando show non **funziona** nell'ultimo livello del Cisco **solo nelle prime due**.

- **show version** (mi mostra la versione e informazioni dell'IOS) l'apparato ha principalmente due memoria, contiene una NVRam(non volatile) e una DRam(volatile).
    - startup-config modalità con la quale l'apparato parte.
    - running-config modalità di funzionamento del pc
- **show startup-config**: mostra il contenuto della nvram ovvero la configurazione di avvio
- **show running-config**: mostra il contenuto della dram volatile ovvero la configurazione di funzionamento. OGNI MODIFICA EFFETTUATA UTILIZZANDO I COMANDI IOS VIENE APPLICATA ALLA RUNNING-CONFIG.
- **copy run start**: per salvare la configurazione running sulla memoria nvram.(ora funziona anche il comando show startup-config)
- **erase startup-config**: cancello la configurazione di startup, quindi riavviando il dispositivo si cancellerà anche la running config
- **Show ip interface**(brief, per le informazioni più importanti):
- **hostname**: consente di modificare il nome dell'apparato(utile per differenziare i diversi apparati tra loro).
- Per annullare un comando per tornare indietro bisogna scrivere "no *configurazione da annullare*" es. no hostname.
- **banner motd "stringa"**: (in modalità config) per dare il messaggio di accesso all'apparato.

SICUREZZA

- **enable password** psw faccio in modo che dal livello user a quello privilege solo con una psw personalizzata (tuttavia se entro in privilege e faccio show running-config vedo la password in chiaro)
- **enable secret** psw (una psw diversa): si fa in modalità configure(global) e censura la password per accedere al privileged, si salva criptata.
- **line console 0**: serve per proteggere l'accesso tramite il cavo console(proprio per entrare nella modalità user), poi si imposta la psw con l'istruzione: **password** "psw" e infine digitare il comando **login**.
- **line vty 0 4**: Per proteggere l'accesso al terminale utilizzando telnet o ssh(collegamento tramite cavo UTP) poi si imposta la password con il comando password "psw" e si digita il comando **login**.
- **service password-encryption**: cifra tutte le password memorizzate nella configurazione running.

COMANDI SICUREZZA INFORMATICA DI BASSO LIVELLO

- **interface fa0/1**(gig0/1): scelgo una interfaccia(una porta dello switch)
- **switchport mode access**:
- **switchport port-security**:
- **switchport port-security maximum 1**: do l'accesso a quella porta(fa0/1) solo ad un host(in caso di range sarebbe 1 per ogni porta)
- **switchport port-security mac-address [mac]**: per fare in modo che solo il pc con quel mac possa entrare in quella porta(nel range al posto del mac mettiamo "sticky" ovvero memorizza i mac in maniera automatica)
- **interface range fa0/2-3**: proteggo dalla 2 alla 3 di porta dello switch
- **switchport port-security violation** "(restrict, shutdown(default), protect)": mi fa vedere come posso configurare in caso di violazione del sistema.

COMANDI VLAN

- **vlan [num]** → num > 1 In modalità global
- **name [nome]**
- **exit**
  interface range fa0/1-5
- **switchport access vlan [num]** → Assegna le porte alla VLAN del numero selezionato.
- **switchport mode access** → è facoltativo e permette di non far si che la porta o il range di porte non faccia mai trunking

- **show vlan brief** → (In modalità privilege) consente di verificare l'assegnazione del gruppo di porte alle VLAN create.

TRUNKING - ROUTER ON A STICK (VLAN) :

router on-a-stick/trunking per utilizzare un solo collegamento per mettere in comunicazione più vlan
Nello switch:

1. **interface fa0/1 (entriamo nella porta che farà da trunk)**
2. **switchport mode trunk (Creiamo una dorsale)**
3. **switchport trunk allowed vlan add [numero vlan] (tutto il traffico delle vlan passa per quella porta)**

Nel router:

- **interface gig0/0.[nlan] -> creazione sottointerfaccia per la VLAN**
- **encapsulation dot1Q [nlan]**
- **ip address [indirizzo default gateway] [subnet mask]**

COMANDI INSTRADAMENTO E ROUTING

- **interface fa0/0** → Seleziona l'interfaccia, posso mettere anche vlan [num] al posto della fast se per caso sono su uno switch
- **ip address [IP] [SUBNET MASK]** → Assegna l'indirizzo ip e la subnet all'interfaccia scelta. Es. ip address 192.168.1.1 255.255.255.0
- **ip route [IP RETE DEST.] [SUBNET MASK] [NEXT HOP]** → Crea una regola di routing.
- **no shutdown** → Attiva l'interfaccia

COMANDI IPV6

- **ipv6 enable → Abilità l'IPV6 e aggiunge il link-local**
- **ipv6 unicast-routing (in modalità global) → Abilità l'IPV6**
- **interface GigabitEthernet 0/0 → Entro nell'interfaccia dove devo inserire l'ind. ip.**
- **ipv6 address "indirizzo ipv6" → Assegna l'indirizzo Global IPV6 all'interfaccia scelta.**
- **ipv6 address "indirizzo ipv6" link-local → Assegna l'indirizzo Link Local IPV6 all'interfaccia scelta.**
- **no shutdown → attivo l'interfaccia.**
- **ipv6 route "network destination" "ip collegamento router"**

Con il comando: ipv6 address address/prefix-length eui-64 genero in maniera automatica un indirizzo ip partendo dal MAC

Per assegnare gli indirizzi in maniera automatica:

```
! This interface uses DHCP to learn its IPv6 address
interface FastEthernet0/0
 ipv6 address dhcp
!
! This interface uses SLAAC to learn its IPv6 address
interface FastEthernet0/1
 ipv6 address autoconfig
```

COMANDI CISCO 5°

Comandi fatti in quinta:

ROUTING STATIC

nella sezione STATIC del router:

- nel Network => ind. ip dell'altra rete (quella da raggiungere);
- nel Mask => subnet mask della rete da raggiungere;
- nel Next Hop => l'ind. del router da cui passare.

Oppure nella CLI usare il comando:

ip route [ind IP] [mask] [next hop] [distanza amministrativa]

La dist. amm. è facoltativa

In base all'ind. che metto la rotta può essere:

- **network**: fa match con tutta la rete;
- **predefinita**: fa match con tutti quelli che non hanno altri match;
- **host**: singolo host.

ROUTING SWITCH LAYER 3

- **sdm prefer lanbase-routing**
- **reload**
- **ip routing**

creo la vlan, metto l'ind e mask ed eventualmente la accendo

- **interface vlan [nlan]**
- **ip address [indirizzo] [mask]**
- **no shutdown**

COMANDI TELNET

bisogna sempre fare enable secret ma le password viaggiano in chiaro con telnet con SSH invece no. Utile per programmare da lontano lo switch.

Comandi base router/switch (in conf t):

- **line vty [range porte es. 0 15]** **nella config verrà segnato come line vty 0 4 + line vty 5 15**

- **password [“password”]**
- **login**
- **exit**
- **enable password/secret [“password”]**

Per lo switch con VLAN:

- **interface vlan 1 (vlan di default)**
- **ip address [“indirizzo ip della stessa rete ma nuovo”] [mask] (come se fosse un host)**
- **no shutdown**
- **line vty [range porte]**
- **password [password]**
- **login**
- **exit**
- **ip default-gateway [“indirizzo ip”]**
- **enable password/secret [“password”]**

Nel laptop nel cmd:

- **telnet [IP dello switch messo prima]**
- **eventuale password**

COMANDI SSH

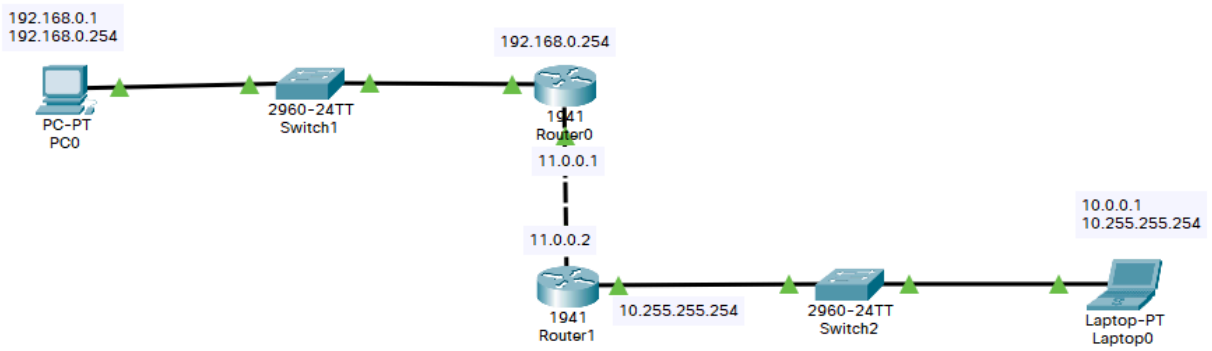
Nel router/switch:

- **ip domain name [nome]**
- **hostname [nome]**
- **crypto key generate rsa**
- **1024 o 512 dipende da quanto sicura è la key**
- **username [username] password [password]** **distinguiamo chi accede, creiamo un user+psw per ogni persona che dovrà accedere**
- **ip ssh version [numero versione (2)]**
- **line vty 0 15**
- **transport input ssh**
- **login local**

Con **show ip ssh** mostriamo la configurazione

Nel cmd:

- **ssh -L [username] [IP (o il nome messo nell’ip domain name)]**



esempio rete con ssh/telnet

COMANDI FTP CLIENT E SERVER

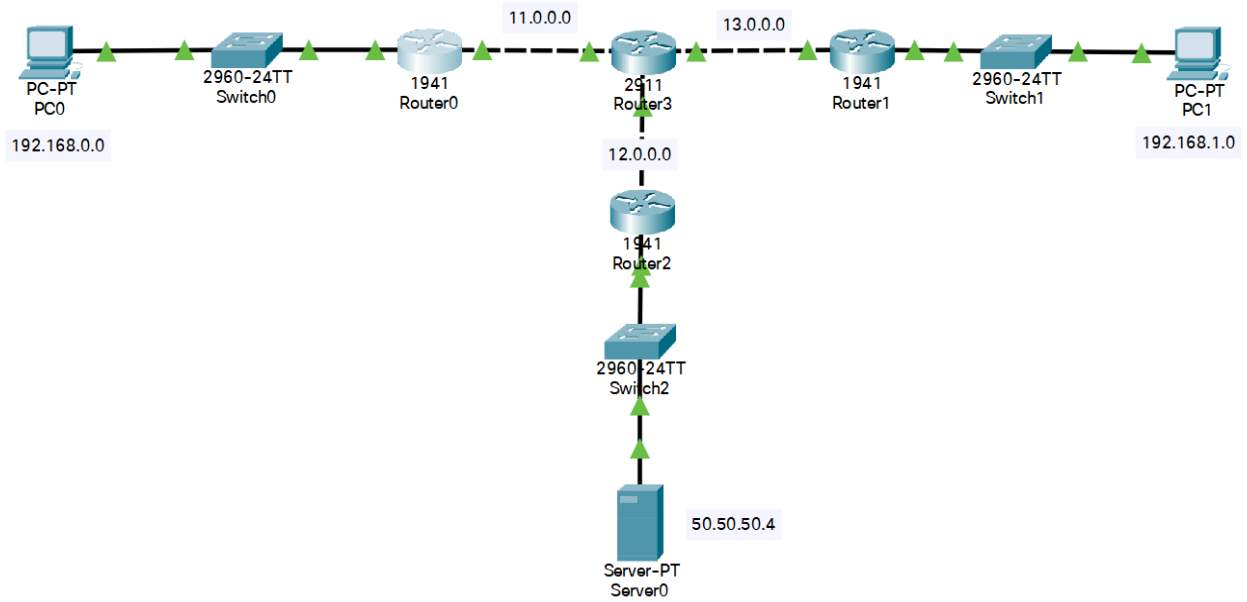
Creare tot reti con un server in comune dove andremo a prendere i file; andare nei services nel server, poi nella sezione FTP e aggiungere un utente.

Nel cmd per fare upload sul Server:

- **ftp [indirizzo server]**
- **put [nome file]**

Nel cmd per fare download

- **ftp [indirizzo server]**
- **get [nome file]**



esempio rete con FTP lato server e RIP

RIP (Router popolamento)

Nei vari router in conf t:

- **router rip**
- **version 2** (La versione con Subnet Mask)
- **network [ind ip rete1]**
- **network [ind ip rete2]**

(Ripetere network tante quante sono le reti da collegare)

DHCP

In config: DG

- **ip dhcp pool [Lan]**
- **network [indirizzo di rete] [SM]**
- **default-router [ind router]**
- **ip dhcp excluded-address [indirizzo]** (per non far inserire a dhcp alcuni indirizzi)

Successivamente nel PC possiamo fare **ip address dhcp** per assegnare uno degli indirizzi nel pool nella macchine

COMANDI FIREWALL E ACL

ACL STANDARD SENZA ACCESSO ALLE PORTE

In config: DG

- **access-list [numero lista es.1] [deny/permit] [indirizzo] [wildcard ovvero il NOT della subnetmask (255.255.255.0 → 0.0.0.255)** dice se vado a isolare una rete o un **host (0.0.0.0)]**
- **Esempio1 => access list 1 permit any** (per consentire a tutti di comunicare).
- **Esempio2 => access list 1 permit 192.168.2.0 0.0.0.255**(permette alla LAN con quell'indirizzo di rete di comunicare).
- Bisogna specificare dove si applicano queste regole: scegliamo l'interfaccia di output con: **interface [interfaccia]**
- **ip access-group [numero lista] [in/out]** si sceglie **in** se l'ACL deve essere applicata su un host all'interno della LAN, al contrario si sceglie **out**

ACL ESTESE

- **access-list [numero lista es.101] [deny/permit] [protocollo] host [indirizzo mittente] host [indirizzo destinatario] eq [numero porta lv7]**

- **Esempio1 =>** **access-list 101 permit tcp host 192.168.2.1 host 192.168.1.100 eq 80** (il PC con ind. 192.168.2.1 può comunicare con il server 192.168.1.100 sulla porta 80)
- **access-list [numero lista] deny ip any any** (per negare tutti gli altri tipi di traffico)
- aggiungere la lista all'interfaccia come nel passo per le acl standard

ACL NAMED

- **ip access-list [standard/extended] [name]** -> dopo questo comando tutte le acl fatte saranno già nel gruppo [name].

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# ip access-list extended barney
Router(config-ext-nacl)# permit tcp host 10.1.1.2 eq www any
Router(config-ext-nacl)# deny udp host 10.1.1.1 10.1.2.0 0.0.0.255
Router(config-ext-nacl)# deny ip 10.1.3.0 0.0.0.255 10.1.2.0 0.0.0.255
Router(config-ext-nacl)# deny ip 10.1.2.0 0.0.0.255 10.2.3.0 0.0.0.255
Router(config-ext-nacl)# permit ip any any
Router(config-ext-nacl)# interface serial1
Router(config-if)# ip access-group barney out
```

NAT STATICO

In config: DG

Sezione inside(da nascondere) da cui arriva il traffico e sezione outside.

- Scegliere l'interfaccia inside **(interface [interfaccia])**
- **ip nat inside**(nella interfaccia della sezione inside)
- exit
- Scegliere l'interfaccia outside **(interface [interfaccia])**
- **ip nat outside**(nella interfaccia della sezione esterna)
- **ip nat inside source static [ip del computer dell'inside] [ip pubblico che assegniamo in maniera statica]** (sempre nella interfaccia dell'outside)
- **Esempio1 =>** ip nat inside source static 192,168,1,1 200.100.50.254
- **do wr** (per applicare la conf. al router, da fare in config)
- per vedere il corretto funzionamento fare una request http dal browser e usare la simulation

NAT DINAMICO

- **ip nat pool [nome pool] [indirizzo pubblico di partenza pool] [indirizzo pubblico di fine pool] netmask [SM]**
- **Esempio1 =>** ip nat pool nomePool 200.100.50.1 200.100.50.10 255.255.255.0
- **access-list 10 permit [rete] [SM al contrario]**
- **Esempio2 =>** access-list 10 permit 192.168.1.0 0.0.0.255
- **ip nat inside source list [source access list] pool [nomepool]**
- **Esempio3 =>** ip nat inside source list 10 pool internetworking
- Scegliere l'interfaccia inside **(interface [interfaccia])**
- **ip nat inside**(nella interfaccia della sezione inside)
- exit
- Scegliere l'interfaccia outside **(interface [interfaccia])**
- **ip nat outside**(nella interfaccia della sezione esterna)
- exit
- **do wr** (per applicare la conf. al router, da fare in config)

PROTOCOLLI

Nome	Porta	TCP o UDP
HTTP/HTTPS	80/443	TCP
DHCP	68(client)/67(server)	UDP
FTP	21(comandi) e 20(dati)	TCP
DNS	53	UDP/TCP
SMTP	25	TCP
POP3	110	TCP

COMANDI FATTI ALLA VEM

AAA

Se in una azienda sono presenti innumerevoli device con utenti replicati su ognuno, è possibile adottare un server per l'autenticazione.

"Authentication, authorization, and accounting (AAA) server"

Vediamo come configurare uno switch (in config) se è presente un server AAA RADIUS:

- **aaa new-model**
- **radius-server host [IP AAA] key [key]**

- **aaa authentication login default group radius local** **lista di metodi di autenticazione (group radius e local)**
- **line vty 0 5**
- **login authentication default**

Address table

Nello switch:

`show mac address-table`
Shows all MAC table entries of all types

`show mac address-table dynamic`
Shows all dynamically learned MAC table entries

`show mac address-table dynamic vlan vlan-id`
Shows all dynamically learned MAC table entries in that VLAN

`show mac address-table dynamic address mac-address`
Shows the dynamically learned MAC table entries with that MAC address

`show mac address-table dynamic interface interface-id`
Shows all dynamically learned MAC table entries associated with that interface

`show mac address-table count`
Shows the number of entries in the MAC table and the total number of remaining empty slots in the MAC table

`show mac address-table aging-time`
Shows the global and per-VLAN aging timeout for inactive MAC table entries

`clear mac address-table dynamic`
Empties the MAC table of all dynamic entries

`show interfaces status`
Lists one line per interface on the switch, with basic status and operating information for each

Nel router:

Command	Lines of Output per Interface	IP Configuration Listed	Interface Status Listed?
show ip interface brief	1	Address	Yes
show protocols [*type number*]	1 or 2	Address/mask	Yes
show interfaces [*type number*]	Many	Address/mask	Yes

Interfaces configuration

```
Emma# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Emma(config)# interface FastEthernet 0/1
Emma(config-if)# duplex full
Emma(config-if)# speed 100
Emma(config-if)# description Printer on 3rd floor, Preset to 100/full
Emma(config-if)# exit
Emma(config)# interface range FastEthernet 0/11 - 20
Emma(config-if-range)# description end-users connect here
Emma(config-if-range)# ^Z
Emma#
```

Command	Mode/Purpose/Description
interface *type port-number*	Changes context to interface mode. The type is typically Fast Ethernet or Gigabit Ethernet. The possible port numbers vary depending on the model of switch—for example, Fa0/1, Fa0/2, and so on.
interface range *type port-number - end-port-number*	Changes the context to interface mode for a range of consecutively numbered interfaces. The subcommands that follow then apply to all interfaces in the range.
shutdown | no shutdown	Interface mode. Disables or enables the interface, respectively.
speed {10 | 100 | 1000 | auto}	Interface mode. Manually sets the speed to the listed speed or, with the auto setting, automatically negotiates the speed.
duplex {auto | full | half}	Interface mode. Manually sets the duplex to half or full, or to autonegotiate the duplex setting.
description *text*	Interface mode. Lists any information text that the engineer wants to track for the interface, such as the expected device on the other end of the cable.
no duplex **no speed** **no description**	Reverts to the default setting for each interface subcommand of **speed auto**, **duplex auto**, and the absence of a **description** command.

Command	Purpose
show running-config	Lists the currently used configuration
show running-config | interface *type number*	Displays the running-configuration excerpt of the listed interface and its subcommands only
show mac address-table dynamic [interface *type number*] [**vlan** *vlan-id*]	Lists the dynamically learned entries in the switch's address (forwarding) table, with subsets by interface and/or VLAN
show mac address-table static [interface *type number*]	Lists static MAC addresses and MAC addresses learned or defined with port security
show interfaces [*type number*] **status**	Lists one output line per interface (or for only the listed interface if included), noting the description, operating state, and settings for duplex and speed on each interface
show interfaces [*type number*]	Lists detailed status and statistical information about all interfaces (or the listed interface only)
show interfaces description	Displays one line of information per interface, with a two-item status (similar to the **show interfaces** command status), and includes any description that is configured on the interfaces

Il comando **show interfaces** restituisce tanti valori nel dettaglio:

Runts: Frames that did not meet the minimum frame size requirement (64 bytes, including the 18-byte destination MAC, source MAC, type, and FCS). Can be caused by collisions.

Giants: Frames that exceed the maximum frame size requirement (1518 bytes, including the 18-byte destination MAC, source MAC, type, and FCS).

Input Errors: A total of many counters, including runts, giants, no buffer, CRC, frame, overrun, and ignored counts.

CRC: Received frames that did not pass the FCS math; can be caused by collisions.

Frame: Received frames that have an illegal format, for example, ending with a partial byte; can be caused by collisions.

Packets Output: Total number of packets (frames) forwarded out the interface.

Output Errors: Total number of packets (frames) that the switch port tried to transmit, but for which some problem occurred.

Collisions: Counter of all collisions that occur when the interface is transmitting a frame.

Late Collisions: The subset of all collisions that happen after the 64th byte of the frame has been transmitted. (In a properly working Ethernet LAN, collisions should occur within the first 64 bytes; late collisions today often point to a duplex mismatch.)

VTP - VLAN Trunking Protocol

Serve per propagare in automatico uno VLAN in ogni switch della rete senza doverle inserire manualmente in ciascuna.

Ogni switch può assumere una modalità:

VTP server mode – a switch using this mode can create and delete VLANs. A VTP server switch will propagate VLAN changes. **This is the default mode for Cisco switches.**

VTP client mode – a switch using this mode can't change its VLAN configuration. That means that a VTP client switch cannot create or delete VLANs. However, received VTP updates are processed and forwarded.

VTP transparent mode – a switch using this mode doesn't share its VLAN database, but it forwards received VTP advertisements. You can create and delete VLANs on a VTP transparent switch, but these changes will not be sent to other switches.

VTP mode off – similar to VTP transparent mode, with a difference that a switch using this mode will not forward received VTP updates. This command is supported only in VTP V3.

Gli switch server possono coesistere

Negli switch server ogni volta che modifichiamo il database delle VLAN il revision number aumenta e le VLAN vengono propagate in base dallo switch (anche se non in server mode) con il revision number più grande.

*Se aggiungo uno switch alla rete con un revision number sbagliato introduco degli errori nelle tabelle degli altri switch, questo switch sbagliato si chiama **VTP BOMB***

```
Switch(config)#vtp ?
domain      Set the name of the VTP administrative domain.
file        Configure IFS filesystem file where VTP configuration is stored.
interface   Configure interface as the preferred source for the VTP IP updater address.
mode        Configure VTP device mode
password    Set the password for the VTP administrative domain
pruning     Set the administrative domain to permit pruning
version     Set the administrative domain to VTP version
```

show vtp status -> monitoriamo il funzionamento.

Il VTP Pruning migliora le prestazioni di rete diminuendo il traffico non necessario, andando a bloccare frame broadcast verso VLAN che non hanno interesse in quel messaggio.

STP/RSTP

```
SW1(config)# spanning-tree mode ?
  mst         Multiple spanning tree mode
  pvst        Per-Vlan spanning tree mode
  rapid-pvst  Per-Vlan rapid spanning tree mode
SW1(config)#
```

Essendo su apparati Cisco usiamo i loro protocolli che funzionano in ugual modo a STP/RSTP/MSTP.

Si può inserire una priorità ad ogni switch per decidere quale diventerà lo switch root.

```
SW1(config)# spanning-tree vlan 1 priority ?
  <0-61440>  bridge priority in increments of 4096
SW1(config)#
```

Elenco di comandi:

Command	Description
spanning-tree mode {pvst | rapid-pvst | mst}	Global configuration command to set the STP mode.
spanning-tree [vlan *vlan-number***] root primary**	Global configuration command that changes this switch to the root switch. The switch's priority is changed to the lower of either 24,576 or 4096 less than the priority of the current root bridge when the command was issued.
spanning-tree [vlan *vlan-number***] root secondary**	Global configuration command that sets this switch's STP base priority to 28,672.
spanning-tree vlan *vlan-id* **priority** *priority*	Global configuration command that changes the bridge priority of this switch for the specified VLAN.
spanning-tree [vlan *vlan-number***] cost** *cost*	Interface subcommand that changes the STP cost to the configured value.
spanning-tree [vlan *vlan-number***] port-priority** *priority*	Interface subcommand that changes the STP port priority in that VLAN (0 to 240, in increments of 16).

- **show spanning-tree** => mostro tutti i dettagli

PortChannel/EtherChannel

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface fa 0/14
SW1(config-if)# channel-group 1 mode on
SW1(config)# interface fa 0/15
SW1(config-if)# channel-group 1 mode on
SW1(config-if)# ^Z
SW1# show spanning-tree vlan 3

VLAN0003
 Spanning tree enabled protocol ieee
 Root ID     Priority    28675
             Address     0019.e859.5380
             Cost        12
             Port        72 (Port-channel1)
             Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID  Priority    28675 (priority 28672 sys-id-ext 3)
             Address     0019.e86a.6f80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Po1                 Root FWD 12        128.64   P2p Peer(STP)
```

In questo modo STP contera i due canali come uno unico e in caso di guasti non si bloccherà tutto.

Per far funzionare il PortChannel le porte devono essere uguali, nel dettaglio:

- Speed
- Duplex
- Modalità access o trunking (tutte in access tutte in trunk)
- Se in access, la access VLAN
- Se in trunk, la lista di VLAN consentite (comando: switchport trunk allowed)
- Se in trunk, la VLAN nativa
- I setting dell'interfaccia relativi a STP

```
port-channel load-balance method
```

Configuration Keyword	Math Uses...	Layer
src-mac	Source MAC address	2
dst-mac	Destination MAC address	2
src-dst-mac	Both source and destination MAC	2
src-ip	Source IP address	3
dst-ip	Destination IP address	3
src-dst-ip	Both source and destination IP	3
src-port	Source TCP or UDP port	4
dst-port	Destination TCP or UDP port	4
src-dst-port	Both source and destination TCP or UDP port	4

Con questo metodo decidiamo che il traffico verrà bilanciato su tutti i canali in base a determinati vincoli.

Root Guard

Diciamo che una certa interfaccia eviti i problemi che senza root guard si verificherebbero all'inserimento di un nuovo switch:

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# spanning-tree rootguard
```

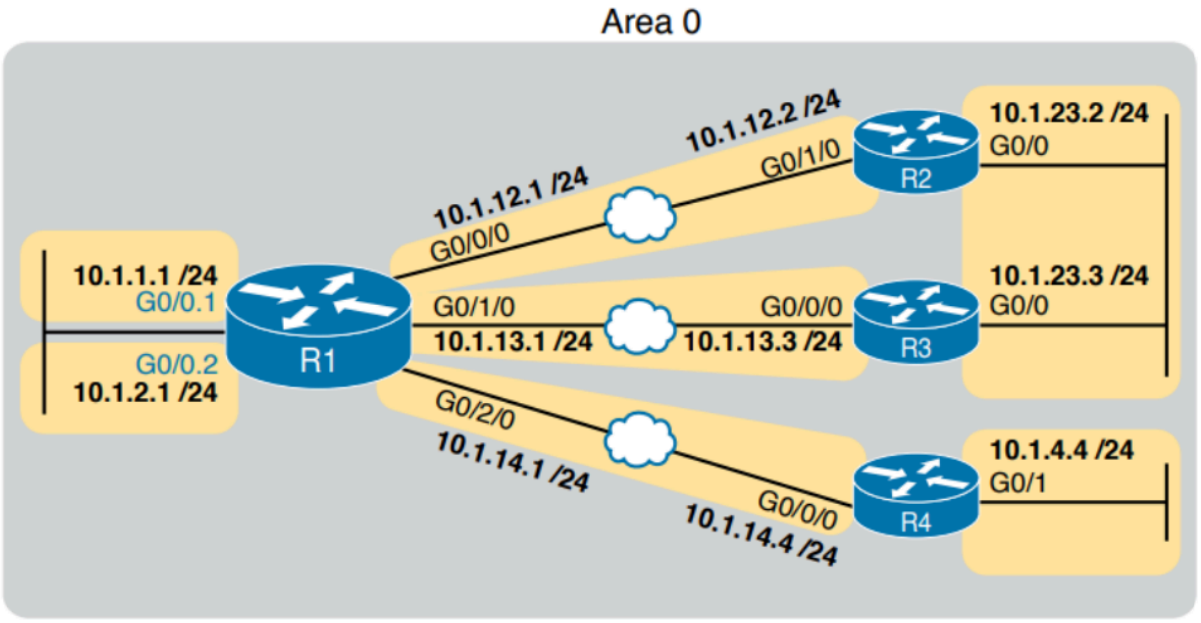
BPDU Guard

```
interface GigabitEthernet1/0/1
switchport mode access
switchport access vlan 2
spanning-tree portfast
spanning-tree bpduguard enable
```

OSPF

Single Area

Implementare Single-Area OSPF



Su R1:

```
interface GigabitEthernet0/0.1
 encapsulation dot1q 1 native
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet0/0.2
 encapsulation dot1q 2
 ip address 10.1.2.1 255.255.255.0
!
interface GigabitEthernet0/0/0
 ip address 10.1.12.1 255.255.255.0
!
interface GigabitEthernet0/1/0
 ip address 10.1.13.1 255.255.255.0
!
interface GigabitEthernet0/2/0
 ip address 10.1.14.1 255.255.255.0
```

La configurazione OSPF inizia con il comando globale

```
ospf process-id
```

che porta l'utente in modalità **configurazione OSPF** e imposta il valore del process-id OSPF.

Il numero di **process-id** consente di supportare più processi OSPF in un singolo router utilizzando ID di processo diversi.

Il process-id non deve necessariamente corrispondere su ogni router e può essere un numero intero qualsiasi compreso tra 1 e 65.535.

Esempio su R2

```
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
```

Il comando network di OSPF confronta il primo parametro del comando con ogni indirizzo IP delle interfacce del router locale, cercando di trovare una corrispondenza.

Tuttavia, anziché confrontare l'intero numero del comando network con l'intero indirizzo IPv4 dell'interfaccia, il router confronta un sottoinsieme di ottetti, in base alla **wildcard mask**

```
! R2 configuration next - one network command enables OSPF on both interfaces
interface GigabitEthernet0/0
 ip address 10.1.23.2 255.255.255.0
!
interface GigabitEthernet0/1/0
 ip address 10.1.12.2 255.255.255.0
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
```

La configurazione degli altri router (R3/4) è simile cambiano gli indirizzi e le network

Verifying OSPF Operation

- `show ip ospf neighbor`
- `show ip ospf database`
- `show ip route`

All’invio del primo comando ci uscirà una tabella con una sezione State, questi sono gli stati disponibili:

FULL/ -: Lo stato dei vicini è **full**, con "-" che significa che il collegamento non utilizza un DR/BDR.

FULL/DR: lo stato del vicino è **full** e il vicino è il **DR**.

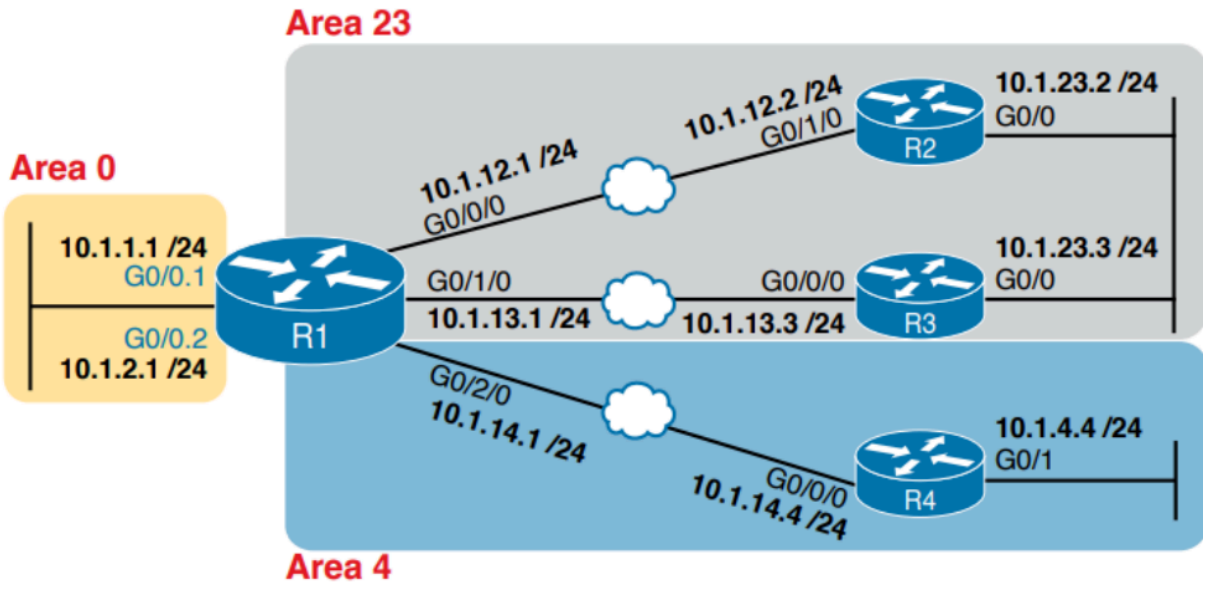
FULL/BDR: lo stato del vicino è **full** e il vicino è il DR di riserva (**BDR**).

FULL/DROTHER: lo stato del vicino è **full** e il vicino non è né il DR né il BDR. (Implica anche che il router locale è un DR o un BDR perché lo stato è FULL).

2WAY/DROTHER: lo stato del vicino è **2-way** e il vicino non è né il DR né il BDR, cioè un router **DROther**. (Ciò implica che anche il router locale è un router DROther, perché altrimenti lo stato raggiungerebbe lo stato **full**).

Multi area

Implementare Multiarea OSPF




```
router ospf 1
 network 10.1.1.1 0.0.0.0 area 0
 network 10.1.2.1 0.0.0.0 area 0
 network 10.1.12.1 0.0.0.0 area 23
 network 10.1.13.1 0.0.0.0 area 23
 network 10.1.14.1 0.0.0.0 area 4
```

OSPF Passive Interfaces

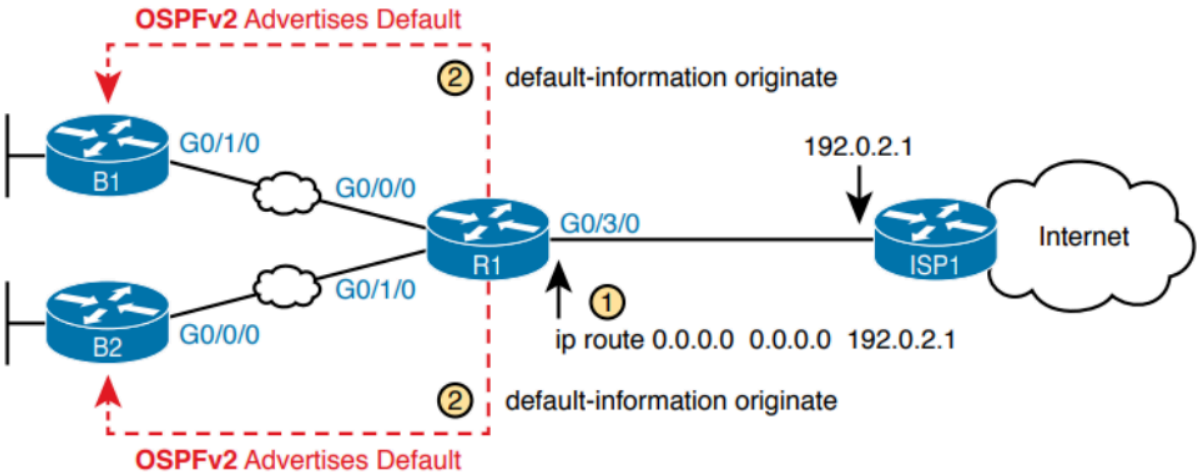
- OSPF continua a notificare la sottorete collegata all'interfaccia.
- OSPF non invia più OSPF Hello dall'interfaccia.
- OSPF non elabora più gli Hello ricevuti sull'interfaccia.

```
passive-interface type
passive-interface default
no passive-interface type number
```

OSPF Default Routes

Usare OSPF per propagare informazioni sul default gateway:

```
default-information originate
```



Vari

- **logging synchronous**: dice allo switch di aspettare che finiamo di scrivere prima di inviare log
- **history size [numero]**: indica quanti comandi la history tiene traccia
- **exec-timeout [minuti]**: indica quanto tempo l'utente può rimanere inattivo prima di essere kickato

CONFIGURAZIONE WIRELESS

PRIMO PASSO: config del router WI-FI

- Scegliamo il router WRT300N;
- Clicchiamo sull'apparato e **selezioniamo GUI**, andiamo in **Setup** → **Basic Setup** e andiamo in **Network Setup**;
- Qui scegliamo la configurazione degli indirizzi che vogliamo (l'ip del router wireless farà da DG per i dispositivi mobili);
- Terminato il setup salviamo con il bottone **Save Settings**;
- Poi apriamo **Wireless** → **Basic Wireless Settings** e impostiamo l'SSID con un nome a nostra scelta e salviamo;
- Ora apriamo la scheda **Wireless** → **Wireless Security** e impostiamo:
    - la protezione **WPA2-Enterprise** con crittografia AES;
    - l'indirizzo IP del server RADIUS: uno della stessa rete del router;
    - la Shared Secret.
- Dopo aver salvato la configurazione router è terminata.

SECONDO PASSO: config del server AAA

- Scegliamo un server e nei **Services** scegliamo il AAA e impostiamo nella **Network Configuration**:
    - come Client Name l'SSID della rete wireless;
    - come Client IP l'ind IP del router wireless;
    - come Secret la Shared Secret impostata sul router wireless;
    - come Service Type selezioniamo RADIUS.

  ora con **ADD** aggiungiamo la configurazione creata;
- Nella parte sotto (**User Setup**) configuriamo le tot. utenze previste e clicchiamo **ADD** per aggiungerle.

TERZO PASSO: config dei dispositivi wireless

- Clicchiamo nel dispositivo scelto andiamo in **Physical** lo spegniamo e cambiamo il modulo Ethernet con uno wireless (**WMP300N**) e lo riaccendiamo;
- In **Config** selezioniamo Wireless0 e impostiamo:
    - come SSID quello che abbiamo impostato del AAA e nel router;
    - come metodo di Authentication la **WPA2** specificando le credenziali;
    - e come Encryption Type **AES**.
- Questa configurazione è da fare per ogni dispositivo della rete wireless.

MACCHINE VIRTUALI

Prerequisiti Macchine Virtuali (Vanno fatti in entrambe le macchine):

- Macchina → Impostazioni → Rete
- Impostare la rete a “Rete interna”, lasciare intnet come nome.
- Configurare gli IP delle due macchine.
- Verificare il ping tra le due.

Procedimento XAMPP

La sigla ci dice che l'ambiente che stiamo creando è un ambiente server con particolari caratteristiche: è un server APACHE (utilizza MariaDB, PhP e Perl);
X: sta ad indicare che è multipiattaforma;
A: ci dice apache;
M: tipo di database su cui fare le operazioni come mariaDB;
P: PHP;
P: Perl è un linguaggio;
Apache se facciamo start creeremo un server web in locale.
se facciamo start in MySQL avremo un database.
Infine attiveremo FileZilla e useremo FTP
C:\XAMPP\HTdocs (c’è la pagina web) → al posto del file index.php creare un file index.html. Questa è la nostra pagina
Per consultare il proprio server nella barra dell’URL di un browser scrivere 127.0.0.1 oppure localhost/ o anche l’indirizzo IP della macchina

Creare un nuovo index.html si può scorrere anche dentro le cartelle VA MESSO IL NOME DOPO LA BARRA.

Procedimento FileZilla

FileZilla: FTP server.

- Andiamo su admin; metti password admin dentro XAMPP
- Edit setting e mettiamo 0 in tutti timeout i timeout dalle impostazioni dell'admin da XAMPP
  per collegarci da un client dobbiamo creare un utente.
- edit → users → add e password
- Shared folders → ADD → mettiamo la cartella che vogliamo condividere abilita tutto
- Avviamo Filezilla client mettiamo host: 127.0.0.1 (Se non usiamo due macchine diverse) Nome utente: nome dato alla user e password porta lasciarla vuota

Configurazione di Rete Windows

Comandi DNS nel cmd

ipconfig /displaydns vediamo il contenuto della cache
ipconfig /flushdns cancelliamo il contenuto della cache

Abilitare connessione desktop remoto windows

- Pannello di controllo
- Sistema e sicurezza
- Sistema
- Impostazioni di connessione remota (a sinistra)
- Fare la spunta su "Consenti connessioni di Assistenza remota al computer"
- Avanzate:
- fare la spunta su "Consenti il controllo del computer da postazioni remote"
- Ok
- Selezionare "Consenti connessioni remote al computer". (In basso)

SUBNETTING

1) Trovare i bit per sottoreti e per host, per i bit della sottorete fare 2^n >= del numero di sottoreti (l' n sarà i bit), per gli host prendo la sottorete con più host e faccio 2^n >= del numero massimo di host -3 (l' n sarà i bit);
2) Faccio la somma dei due bit trovati e con la somma trovo la classe che devo usare andando a vedere i bit dedicati agli host (Classe A = 24, Classe B = 16,

Classe C = 8), distribuisco eventuali bit restanti (facendo la somma dei bit trovati meno bit degli host di ogni classe e andando a scegliere una combinazione);

3) Prendiamo la combinazione scelta nel passaggio precedente (Es. 9+7 con classe B) e decidiamo un indirizzo, preso a caso, della classe scelta (Es. 172.16.0.0), iniziamo facendo al subnet mask andando a prendere quella della classe scelta (Es. classe B => 255.255.0.0) e mettiamo a 1 tutti i bit della rete ottenendo così la subnet mask (Es. 255.255.255.128 => 111111111 | 0000000).

   Per gli indirizzi degli host prendiamo il numero dell'host (Es. 30) e il numero della sottorete (Es. 4° sottorete) e dividiamo così:
   172.16.1.158 => 000000011 | 0011110 Nella parte sinistra dove ci sono i bit della rete metto il numero della sottorete -1 in binario, in quella a destra dove ci sono i bit destinati agli host metto il numero dell'host in binario.

   Per il D.G. metto a 1 tutti i bit dedicati agli host tranne l'ultimo che resta 0
   (Es.1 => 172.16.0.126 => 00000000 | 1111110 Lo 0 finale nella parte a sinistra perchè 1° sottorete.
   Es.2 => 172.16.0.254 => 00000001 | 1111110 L' 1 finale nella parte a sinistra perchè 2° sottorete.)