

# Progetto GNS3: Configurazione di una Rete con VLAN e Routing Inter-VLAN

## Sommario

[Obiettivo del Progetto](#)

[Descrizione topologia](#)

[Configurazione Switch1](#)

[Configurazione Switch2](#)

[Instradamento inter-VLAN](#)

[Assegnazione ai PC dei rispettivi indirizzi e gateway](#)

[Collegamento R1 e R2](#)

[Verifica della connettività della rete](#)

---

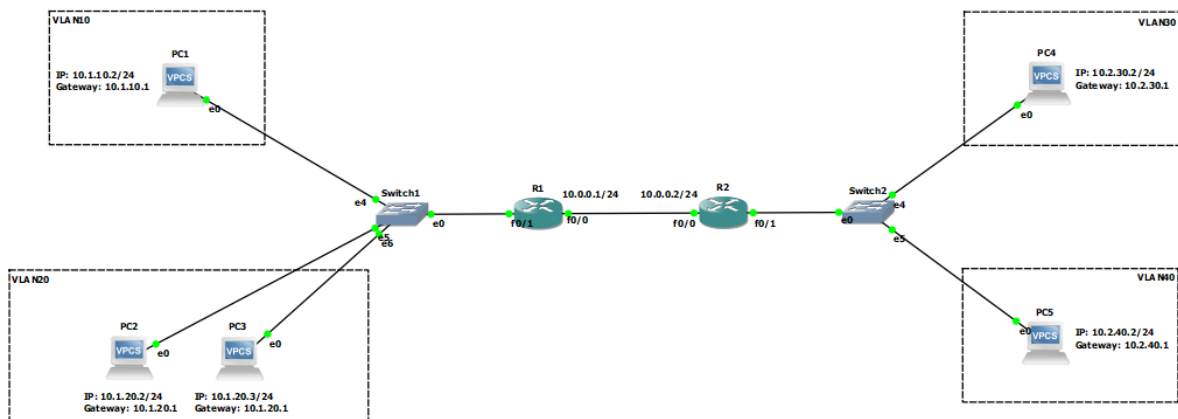
## Obiettivo del Progetto

Il progetto prevede la progettazione e la configurazione di una rete che include due LAN separate in due VLAN su GNS3.

Una VLAN è una rete LAN realizzata logicamente (isolata quindi virtualmente, ma non fisicamente, da altre LAN virtuali). Tale tecnologia permette di segmentare il dominio broadcast di una LAN, suddividendola in più reti LAN Virtuali, isolate tra loro, ma che condividono la stessa infrastruttura fisica.

## Descrizione topologia

- VLAN 10 associata a PC1 tramite interfaccia Ethernet4
- VLAN 20 associata a PC2 e PC3 tramite interfaccia Ethernet5 e Ethernet6
- LAN 1 composta dai terminali PC1, PC2, PC3 collegati a R1 tramite switch
- Router1 collegato a Router2 tramite interfaccia fa0/0
- Router2 collegato a LAN 2 tramite switch
- LAN 2 composta dai terminali PC4 e PC5
- VLAN 30 associata a PC4 tramite interfaccia Ethernet4
- VLAN 40 associata a PC4 tramite interfaccia Ethernet5



## Configurazione Switch1

1. Creazione di VLAN10 e VLAN20 a livello topologico.
2. Assegnazione delle porte alle VLAN appropriate specificandone la modalità (accesso o trunk)

Le porte di accesso collegano le macchine virtuali allo switch, trasmettendo dati all'interno della singola VLAN.

Le porte trunk sono utilizzate per realizzare la connessione tra switch e router, trasmettendo dati da più VLAN contemporaneamente. Il trunking consente ai dispositivi di rete di trasmettere numerose VLAN tramite un singolo canale fisico, consolidando efficacemente il traffico VLAN su un'infrastruttura condivisa.

Port	VLAN	Type	EtherType
0	1	dot1q	
1	1	access	
2	1	access	
3	1	access	
4	10	access	
5	20	access	
6	20	access	
7	1	access	

Nel nostro caso la porta Ethernet0 è configurata in modalità dot1q per permettere la connessione con il router.

La porta Ethernet4 è configurata in modalità di accesso, ed è adibita alla connessione di un terminale appartenente alla VLAN 10. La configurazione è la medesima per le porte Ethernet5 e Ethernet6, utilizzate per connettere lo switch ai terminali appartenenti alla VLAN 20.

## Configurazione Switch2

1. Creazione di VLAN30 e VLAN40 a livello topologico.
2. Assegnazione delle porte alle VLAN appropriate specificandone la modalità (accesso o trunk)

Port	VLAN	Type	EtherType
0	1	dot1q	
1	1	access	
2	1	access	
3	1	access	
4	30	access	
5	40	access	
6	1	access	
7	1	access	

Come per la LAN1, la porta Ethernet0 è configurata in modalità dot1q e riservata alla connessione con il router.

Per quanto riguarda le porte Ethernet4 e Ethernet5, invece, sono entrambe configurate in modalità di accesso per permettere il collegamento coi dispositivi terminali appartenenti, rispettivamente, alle VLAN 30 e 40.

## Instradamento inter-VLAN

Per inter-VLAN routing si intende l'inoltro di pacchetti tra host appartenenti a VLAN diverse.

Nella teoria, per consentire la comunicazione tra reti distinte, occorre definire per ognuna un gateway di riferimento che consenta il routing. Per ovviare a questo problema si utilizzano, quindi, delle sotto-interfacce virtuali delle interfacce fisiche. Ogni sub-interface deve appartenere ad una VLAN distinta.

---

Creazione delle sub-interfaces in fa0/1

1. Situazione iniziale delle interfacce di rete

```
R1#show ip int br
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES unset    administratively down down
Serial0/0           unassigned      YES unset    administratively down down
FastEthernet0/1     unassigned      YES unset    administratively down down
Serial0/1           unassigned      YES unset    administratively down down
Serial0/2           unassigned      YES unset    administratively down down
Serial0/3           unassigned      YES unset    administratively down down
FastEthernet1/0     unassigned      YES unset    administratively down down
```

2. Attiviamo prima di tutto l'interfaccia fisica principale.  
Successivamente creiamo la prima sub-interface associandola alla VLAN 10 (fa0/1.10). Configuriamo poi il router per utilizzare il protocollo 802.1Q (dot1q) come

metodo di incapsulamento per questa sottointerfaccia, associandola alla VLAN con id 10. Infine assegniamo l'indirizzo con la subnet mask alla sotto interfaccia.

```
R1(config)#int fa0/1
R1(config-if)#no shut
R1(config-if)#
*Mar 1 00:02:55.159: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:02:56.159: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1(config-if)#exit
R1(config)#int fa0/1.10
R1(config-subif)#encapsulation dot1q 10
R1(config-subif)#ip
*Mar 1 00:03:33.919: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
R1(config-subif)#ip add 10.1.10b
R1(config-subif)#ip add 10.1.10.1 255.255.255.0
R1(config-subif)#no shut
R1(config-subif)#end
R1#
*Mar 1 00:04:06.219: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip int br

```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	administratively down	down
Serial0/0	unassigned	YES	unset	administratively down	down
FastEthernet0/1	unassigned	YES	unset	up	up
FastEthernet0/1.10	10.1.10.1	YES	manual	up	up
Serial0/1	unassigned	YES	unset	administratively down	down
Serial0/2	unassigned	YES	unset	administratively down	down
Serial0/3	unassigned	YES	unset	administratively down	down
FastEthernet1/0	unassigned	YES	unset	administratively down	down

3. Ripetiamo la stessa operazione per la sotto interfaccia associata alla VLAN 20 (fa0/1.20).

```
R1(config)#int fa0/1.20
R1(config-subif)#encapsulation dot1q 20
R1(config-subif)#ip add 10.1.20.1 255.255.255.0
R1(config-subif)#no shut
R1(config-subif)#end
R1#show ip int br
*Mar 1 00:06:26.735: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip int br

```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	administratively down	down
Serial0/0	unassigned	YES	unset	administratively down	down
FastEthernet0/1	unassigned	YES	unset	up	up
FastEthernet0/1.10	10.1.10.1	YES	manual	up	up
FastEthernet0/1.20	10.1.20.1	YES	manual	up	up
Serial0/1	unassigned	YES	unset	administratively down	down
Serial0/2	unassigned	YES	unset	administratively down	down
Serial0/3	unassigned	YES	unset	administratively down	down
FastEthernet1/0	unassigned	YES	unset	administratively down	down

4. Il procedimento è analogo per la configurazione delle sotto interfacce associate alle VLAN con identificativo 30 e 40. Per questo motivo riportiamo direttamente la configurazione finale.

```
R2#show ip int br
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES NVRAM    administratively down down
Serial0/0          unassigned      YES NVRAM    administratively down down
FastEthernet0/1    unassigned      YES NVRAM    up          up
FastEthernet0/1.30 10.2.30.1       YES manual    up          up
FastEthernet0/1.40 10.2.40.1       YES manual    up          up
Serial0/1          unassigned      YES NVRAM    administratively down down
Serial0/2          unassigned      YES NVRAM    administratively down down
Serial0/3          unassigned      YES NVRAM    administratively down down
FastEthernet1/0    unassigned      YES NVRAM    administratively down down
```

## Assegnazione ai PC dei rispettivi indirizzi e gateway

1. PC1 appartenente alla VLAN 10, PC2 e PC3 appartenenti alla VLAN 20

```
PC1> ip 10.1.10.2 255.255.255.0 10.1.10.1
Checking for duplicate address...
PC1 : 10.1.10.2 255.255.255.0 gateway 10.1.10.1
```

```
PC2> ip 10.1.20.2 255.255.255.0 10.1.20.1
Checking for duplicate address...
PC2 : 10.1.20.2 255.255.255.0 gateway 10.1.20.1
```

```
PC3> ip 10.1.20.3 255.255.255.0 10.1.20.1
Checking for duplicate address...
PC3 : 10.1.20.3 255.255.255.0 gateway 10.1.20.1
```

2. PC4 appartenente alla VLAN 30, PC5 appartenente alla VLAN 40

```
PC4> ip 10.2.30.2 255.255.255.0 10.2.30.1
Checking for duplicate address...
PC4 : 10.2.30.2 255.255.255.0 gateway 10.2.30.1
```

```
PC5> ip 10.2.40.2 255.255.255.0 10.2.40.1
Checking for duplicate address...
PC5 : 10.2.40.2 255.255.255.0 gateway 10.2.40.1
```

3. Verifichiamo che tutto sia andato a buon fine. Vediamo se i terminali appartenenti a VLAN differenti possono pingarsi a vicenda.

```

PC1> ping 10.1.20.2

10.1.20.2 icmp_seq=1 timeout
10.1.20.2 icmp_seq=2 timeout
84 bytes from 10.1.20.2 icmp_seq=3 ttl=63 time=15.808 ms
84 bytes from 10.1.20.2 icmp_seq=4 ttl=63 time=20.765 ms
84 bytes from 10.1.20.2 icmp_seq=5 ttl=63 time=14.314 ms

PC1> ping 10.1.20.3

10.1.20.3 icmp_seq=1 timeout
84 bytes from 10.1.20.3 icmp_seq=2 ttl=63 time=14.028 ms
84 bytes from 10.1.20.3 icmp_seq=3 ttl=63 time=18.344 ms
84 bytes from 10.1.20.3 icmp_seq=4 ttl=63 time=16.746 ms
84 bytes from 10.1.20.3 icmp_seq=5 ttl=63 time=17.276 ms

PC2> ping 10.1.10.2

84 bytes from 10.1.10.2 icmp_seq=1 ttl=63 time=21.618 ms
84 bytes from 10.1.10.2 icmp_seq=2 ttl=63 time=18.082 ms
84 bytes from 10.1.10.2 icmp_seq=3 ttl=63 time=21.259 ms
84 bytes from 10.1.10.2 icmp_seq=4 ttl=63 time=16.700 ms
84 bytes from 10.1.10.2 icmp_seq=5 ttl=63 time=16.609 ms

PC4> ping 10.2.40.2

10.2.40.2 icmp_seq=1 timeout
84 bytes from 10.2.40.2 icmp_seq=2 ttl=63 time=13.261 ms
84 bytes from 10.2.40.2 icmp_seq=3 ttl=63 time=17.690 ms
84 bytes from 10.2.40.2 icmp_seq=4 ttl=63 time=12.222 ms
84 bytes from 10.2.40.2 icmp_seq=5 ttl=63 time=18.222 ms

PC5> ping 10.2.30.2

84 bytes from 10.2.30.2 icmp_seq=1 ttl=63 time=19.170 ms
84 bytes from 10.2.30.2 icmp_seq=2 ttl=63 time=19.530 ms
84 bytes from 10.2.30.2 icmp_seq=3 ttl=63 time=19.562 ms
84 bytes from 10.2.30.2 icmp_seq=4 ttl=63 time=11.162 ms
84 bytes from 10.2.30.2 icmp_seq=5 ttl=63 time=18.252 ms

```

## Collegamento R1 e R2

1. Configuriamo l'interfaccia di rete FastEthernet0/0 su entrambi i router.

```

R1(config)#int fa0/0
R1(config-if)#ip add 10.0.0.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#end

*Mar 1 00:01:42.115: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:01:43.115: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1#conf t
*Mar 1 00:01:44.031: %SYS-5-CONFIG_I: Configured from console by cshow ip int br
R1#show ip int br

```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	10.0.0.1	YES	manual	up	up
Serial0/0	unassigned	YES	NVRAM	administratively down	down
FastEthernet0/1	unassigned	YES	NVRAM	up	up
FastEthernet0/1.10	10.1.10.1	YES	NVRAM	up	up
FastEthernet0/1.20	10.1.20.1	YES	NVRAM	up	up
Serial0/1	unassigned	YES	NVRAM	administratively down	down
Serial0/2	unassigned	YES	NVRAM	administratively down	down
Serial0/3	unassigned	YES	NVRAM	administratively down	down
FastEthernet1/0	unassigned	YES	NVRAM	administratively down	down

```

R2(config)#int fa0/0
R2(config-if)#ip add 10.0.0.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#end
R2#
*Mar 1 00:22:32.307: %SYS-5-CONFIG_I: Configured from console by console
R2#wr
*Mar 1 00:22:32.863: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:22:33.863: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2#show ip int br

```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	10.0.0.2	YES	manual	up	up
Serial0/0	unassigned	YES	NVRAM	administratively down	down
FastEthernet0/1	unassigned	YES	NVRAM	up	up
FastEthernet0/1.30	10.2.30.1	YES	manual	up	up
FastEthernet0/1.40	10.2.40.1	YES	manual	up	up
Serial0/1	unassigned	YES	NVRAM	administratively down	down
Serial0/2	unassigned	YES	NVRAM	administratively down	down
Serial0/3	unassigned	YES	NVRAM	administratively down	down
FastEthernet1/0	unassigned	YES	NVRAM	administratively down	down

2. Configuriamo le rotte statiche su ciascun router per consentire la comunicazione

```

R1(config)#ip route 10.2.30.0 255.255.255.0 10.0.0.2
R1(config)#end
R1#
*Mar 1 00:09:54.247: %SYS-5-CONFIG_I: Configured from console by
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip route 10.2.40.0 255.255.255.0 10.0.0.2

```

```

R2(config)#ip route 10.1.10.0 255.255.255.0 10.0.0.1
R2(config)#ip route 10.1.20.0 255.255.255.0 10.0.0.1

```

3. Visualizziamo le tabelle di routing di R1 e R2

```

10.0.0.0/24 is subnetted, 5 subnets
C    10.1.10.0 is directly connected, FastEthernet0/1.10
C    10.0.0.0 is directly connected, FastEthernet0/0
S    10.2.30.0 [1/0] via 10.0.0.2
C    10.1.20.0 is directly connected, FastEthernet0/1.20
S    10.2.40.0 [1/0] via 10.0.0.2

```

```

10.0.0.0/24 is subnetted, 5 subnets
S    10.1.10.0 [1/0] via 10.0.0.1
C    10.0.0.0 is directly connected, FastEthernet0/0
C    10.2.30.0 is directly connected, FastEthernet0/1.30
S    10.1.20.0 [1/0] via 10.0.0.1
C    10.2.40.0 is directly connected, FastEthernet0/1.40

```

## Verifica della connettività della rete

Se viene effettuato un ping da PC2 a PC3, entrambi appartenenti alla VLAN 20, notiamo che il pacchetto non viene indirizzato verso il router.

```

PC2> ping 10.1.20.3 -c 1

84 bytes from 10.1.20.3 icmp_seq=1 ttl=64 time=0.305 ms

```

3	0.001026	10.1.20.2	10.1.20.3	ICMP	98 Echo (ping) request	id=0x5f30, seq=1/256, ttl=64 (reply in 4)
4	0.001182	10.1.20.3	10.1.20.2	ICMP	98 Echo (ping) reply	id=0x5f30, seq=1/256, ttl=64 (request in 3)

Al contrario, quando viene effettuato un ping fuori dalla VLAN di provenienza, pur rimanendo nella stessa LAN (in questo caso, da PC2 a PC1), il pacchetto viene instradato in direzione del router, in quanto sarà necessario individuarne il percorso.

```

PC2> ping 10.1.10.2 -c 1

84 bytes from 10.1.10.2 icmp_seq=1 ttl=63 time=42.022 ms

```

7	59.465649	10.1.20.2	10.1.10.2	ICMP	98 Echo (ping) request	id=0x9a30, seq=1/256, ttl=64 (reply in 8)
8	59.506411	10.1.10.2	10.1.20.2	ICMP	98 Echo (ping) reply	id=0x9a30, seq=1/256, ttl=63 (request in 7)
10	118.214201	10.1.20.2	10.1.10.2	ICMP	102 Echo (ping) request	id=0x9a30, seq=1/256, ttl=64 (no response found!)
11	118.221917	10.1.20.2	10.1.10.2	ICMP	102 Echo (ping) request	id=0x9a30, seq=1/256, ttl=63 (reply in 14)
14	118.237859	10.1.10.2	10.1.20.2	ICMP	102 Echo (ping) reply	id=0x9a30, seq=1/256, ttl=64 (request in 11)
15	118.254629	10.1.10.2	10.1.20.2	ICMP	102 Echo (ping) reply	id=0x9a30, seq=1/256, ttl=63