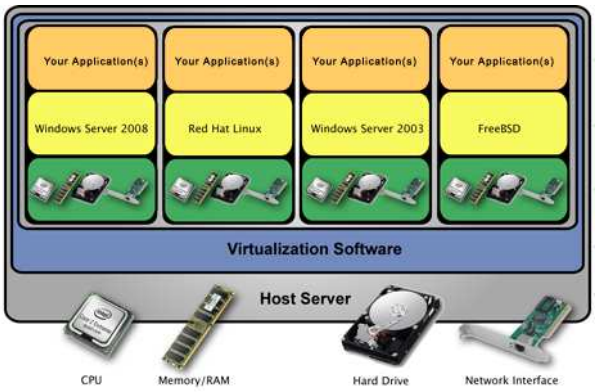
Laboratorio di Reti di Calcolatori

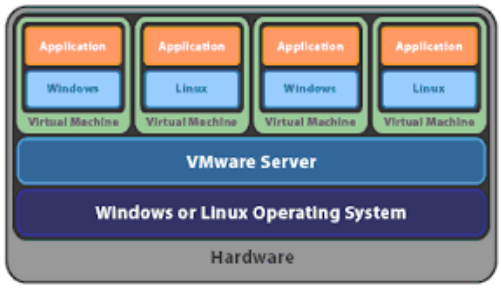
```
Destination     Gateway         Genmask         Flags Iface
loopback        *               255.0.0.0       U     lo
link-local      *               255.255.0.0     U     eth0
192.168.0.0     *               255.255.255.0   U     eth0
default         192.168.0.1     0.0.0.0         UG    eth0
```

Laboratorio di Reti di Calcolatori

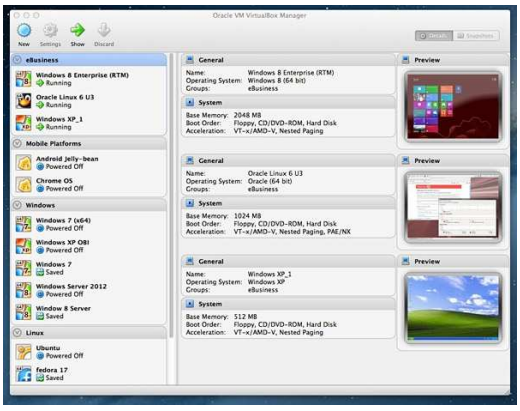
Virtualizzazione



Virtualizzazione



Virtualizzazione



Laboratorio di Reti di Calcolatori

Debian 9

Requisiti per l'istallazione:
- RAM 1GB
- HD 2GB

Dopo l'istallazione:
- RAM 256 (128) MB
- HD 800MB

Consistent Network Device Naming

Consistent Network Device Naming is a convention for naming Ethernet adapters in Linux, that replace the old standard ethX which caused problems on multihomed machines because the network interface controllers (NICs) would get named based on the order in which they were found by the kernel as it booted. Added new interfaces could cause the previously added ones to change names

Scheme 3: Names incorporating physical location of the connector of the hardware (example: enp2s0), are applied if applicable

ethx

A volte il S.O. rinomina le interfacce di rete.

Nel file **`/etc/udev/rules.d/70-persistent-net.rules`** sono le indicazioni su come il S.O. sta attualmente rinominando le interfacce di rete.

70-persistent-net.rules

```
user@nodeA /etc/udev/rules.d $ cat 70-persistent-net.rules
# This file was automatically generated by the
/lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules
file.
#
# You can modify it, as long as you keep each rule on a
single
# line, and change only the value of the NAME= key.

# PCI device 0x10ec:0x8168 (r8169)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="e0:db:55:cf:1d:d6", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
```


ifconfig vs ip

Deprecated command	Replacement command(s)
arp	ip n (ip neighbor)
ifconfig	ip a (ip addr), ip link, ip -s (ip -stats)
iptunnel	ip tunnel
iwconfig	iw
nameif	ip link, ifrename
netstat	ss, ip route (for netstat-r), ip -s link (for netstat -i), ip maddr (for netstat-g)
route	ip r (ip route)

ip addr

`ip addr`
Show information for all addresses

`ip addr show dev em1`
Display information only for device em1ip addr

`ip addr add 192.168.1.1/24 dev em1`
Add address 192.168.1.1 with netmask 24 to device em1

`ip addr del 192.168.1.1/24 dev em1`
Remove address 192.168.1.1/24 from device em1

ip route

`ip route`
List all of the route entries in the kernel

`ip route add default via 192.168.1.1 dev em1`
Add a default route (for all addresses) via the local gateway 192.168.1.1 that can be reached on device em1

`ip route add 192.168.1.0/24 via 192.168.1.1`
Add a route to 192.168.1.0/24 via the gateway at 192.168.1.1

`ip route delete 192.168.1.0/24 via 192.168.1.1`
Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.1

`ip route replace 192.168.1.0/24 dev em1`
Replace the defined route for 192.168.1.0/24 to use device em1

ip neigh

ip neigh
Display neighbour objects

ip neigh show dev em1
Show the ARP cache for device em1

ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev em1
Add address 192.168.1.1 with MAC 1:2:3:4:5:6 to em1

ip neigh del 192.168.1.1 dev em1
Invalidate the entry for 192.168.1.1 on em1

ip neigh replace 192.168.1.1 lladdr 1:2:3:4:5:6 dev em1
Replace the entry for address 192.168.1.1 to use MAC 1:2:3:4:5:6 on em1

ip neigh

ip neigh
Display neighbour objects

ip neigh show dev em1
Show the ARP cache for device em1

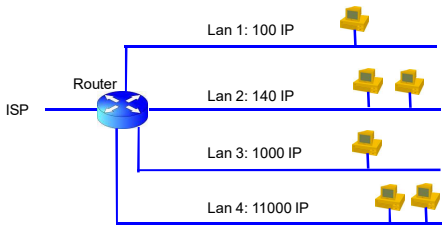
ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev em1
Add address 192.168.1.1 with MAC 1:2:3:4:5:6 to em1

ip neigh del 192.168.1.1 dev em1
Invalidate the entry for 192.168.1.1 on em1

ip neigh replace 192.168.1.1 lladdr 1:2:3:4:5:6 dev em1
Replace the entry for address 192.168.1.1 to use MAC 1:2:3:4:5:6 on em1

Esercizio 1

Una società ottiene la subnet 145.67.0.0/16
Configurare 4 subnet interne con i seguenti requisiti:



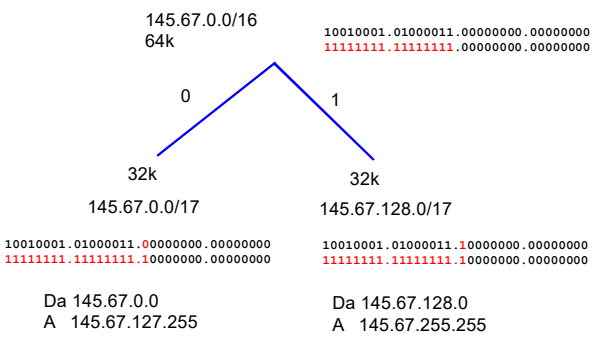
Garantire la massima espandibilità (ulteriori subnet interne)

Esercizio 1 - soluzione

145.67.0.0/16

145.67.0.0 = 10010001.01000011.00000000.00000000
255.255.0.0 = 11111111.11111111.00000000.00000000

Esercizio 1 - soluzione



10010001.01000011.00000000.00000000
11111111.11111111.00000000.00000000

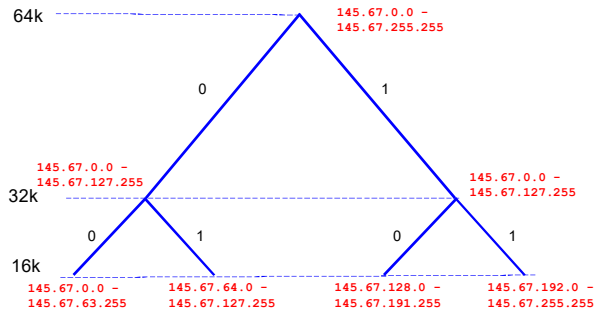
10010001.01000011.00000000.00000000
11111111.11111111.10000000.00000000

10010001.01000011.10000000.00000000
11111111.11111111.10000000.00000000

Da 145.67.0.0
A 145.67.127.255

Da 145.67.128.0
A 145.67.255.255

Esercizio 1 - soluzione



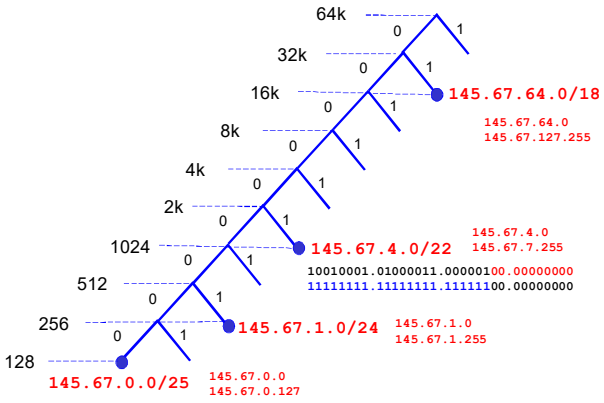
10010001.01000011.00000000.00000000
10010001.01000011.00111111.11111111

10010001.01000011.10000000.00000000
10010001.01000011.10111111.11111111

10010001.01000011.01000000.00000000
10010001.01000011.01111111.11111111

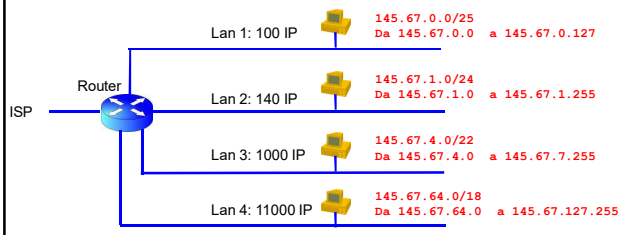
10010001.01000011.11000000.00000000
10010001.01000011.11111111.11111111

Esercizio 1 - soluzione



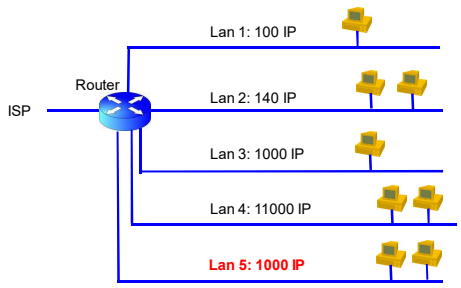
Esercizio 1 - soluzione

IP: 145.67.0.0/16

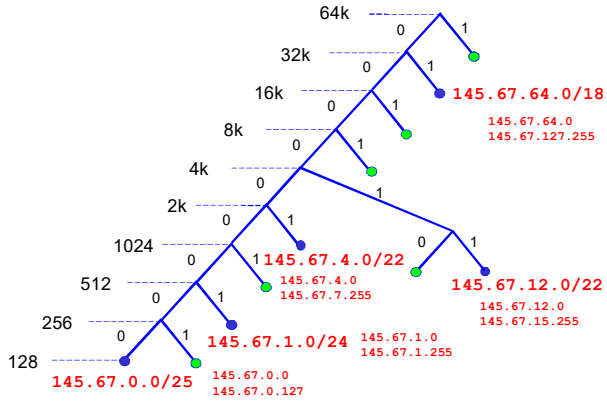


Esercizio 1 - espansione

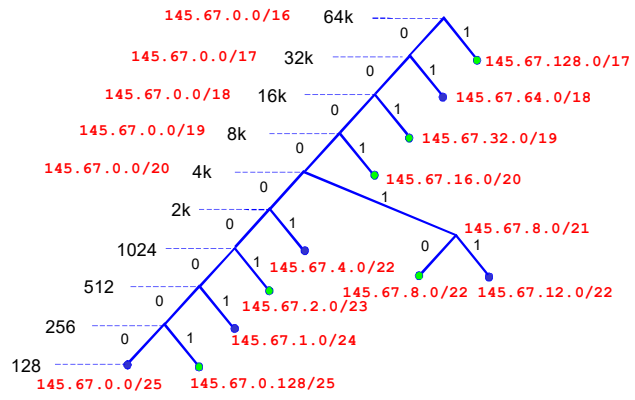
IP: 145.67.0.0/16



Esercizio 1 - espansione

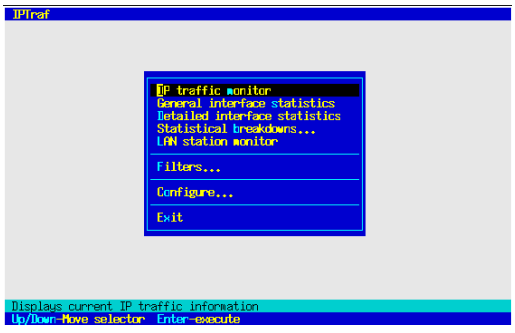


Esercizio 1 - Indirizzi e maschere

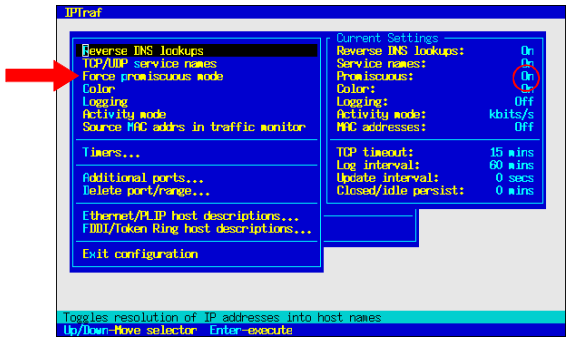


Tool iptraf

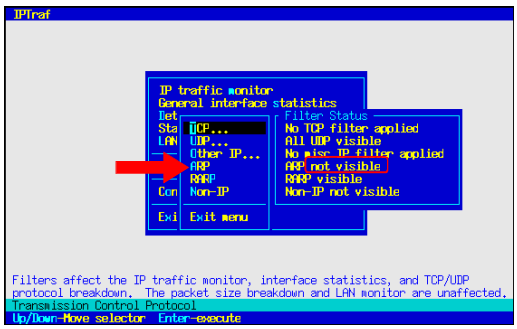
Questo tool (da istallare) consente di analizzare il traffico su un nodo o una interfaccia di rete



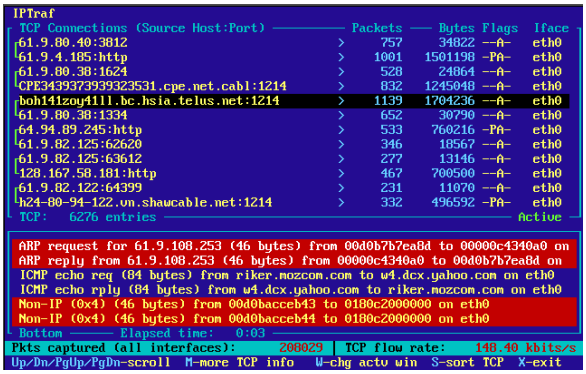
iptraf



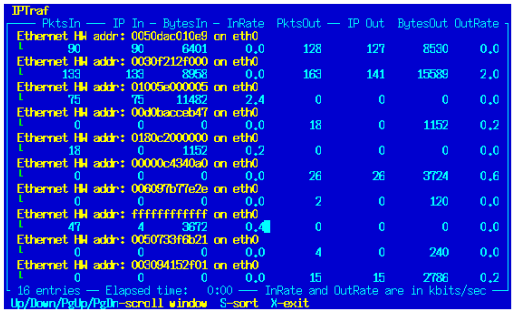
iptraf



iptraf



iptraf



IPv4 forwarding

Abilitazione IPv4 forwarding (di default è disabilitato). Se non è abilitato un host non si comporta da router.

/etc/sysctl.conf

cambiare 0 in 1 nella riga:

net.ipv4.conf.ip_forward = 1

Oppure per una modifica temporanea

echo "1" > /proc/sys/net/ipv4/ip_forward

ip addr

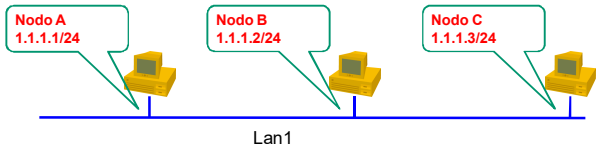
```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 00:0c:29:1b:06:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.49/24 brd 192.168.0.255 scope global dynamic noprefixroute
ens33
       valid_lft 84081sec preferred_lft 84081sec
    inet 10.0.0.1/24 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::77c2:e823:81c5:e36f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

ip route

```
#ip route
default via 192.168.0.1 dev ens33 proto dhcp metric 100
10.0.0.0/24 dev ens33 proto kernel scope link src 10.0.0.1
192.168.0.0/24 dev ens33 proto kernel scope link src 192.168.0.49 metric 100
```


routing

```
ip addr add 1.1.1.1/24 dev enp0s3
```

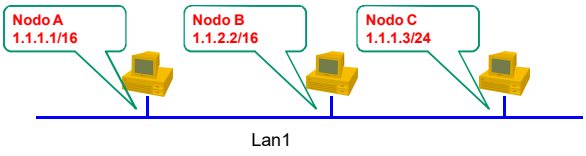


In automatico sarà generata la seguente riga di routing:

```
#ip route
10.1.1.0/24 dev enp0s3 proto kernel scope link src 10.1.1.1
```

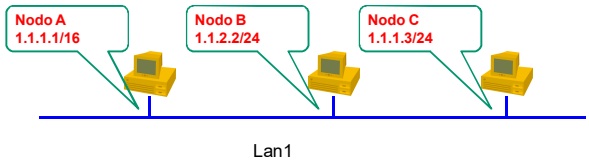

Esercizio 2a

Cosa succede in questa rete? Verificare sperimentalmente



Esercizio 2b

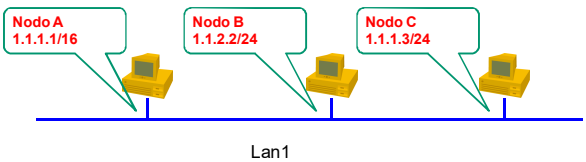
Cambiamo la maschera del nodo B. Cosa succede?



Modalità promiscua

Il seguente comando configura la scheda di rete in modalità promiscua (accetta tutte le frame, anche con MAC address diverso dal proprio).

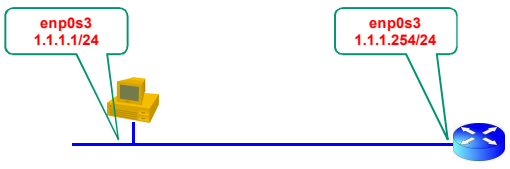
```
ip link set [interface] promisc on
```



Router

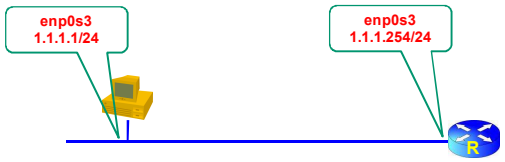
Inseriamo un router di uscita per la lan

```
enp0s3
IP address 1.1.1.1
Netmask 255.255.255.0
Gateway 1.1.1.254
```



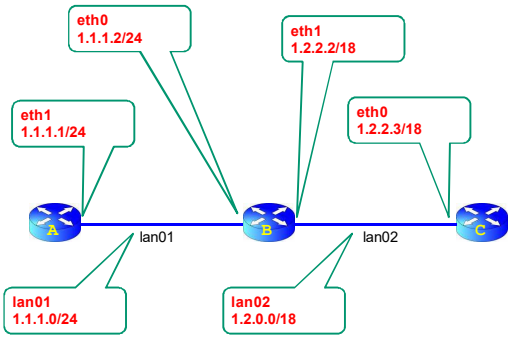
Gateway di default

```
ip addr 1.1.1.1/24 dev enp0s3
ip route add default via 1.1.1.254 dev enp0s3
```



Lan 1.1.1.0/24

Routing statico



Routing statico

Aggiungere una regola di routing specifica

```
ip route add x.x.x.x/n via g.g.g.g
```

Nel nodo A

```
ip route add 1.2.0.0/18 via 1.1.1.2
```

Nel nodo C

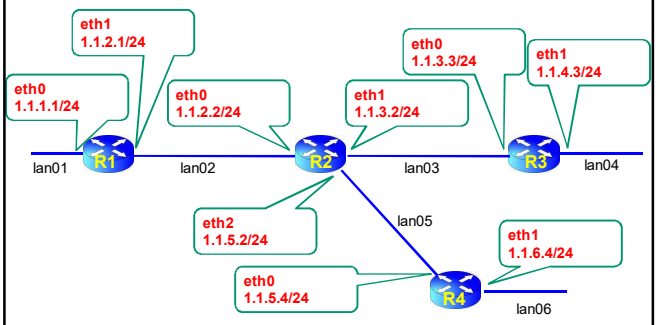
```
ip route add 1.1.1.0/24 via 1.2.2.2
```

Per cancellare una entry

```
ip route delete x.x.x.x/n via g.g.g.g
```

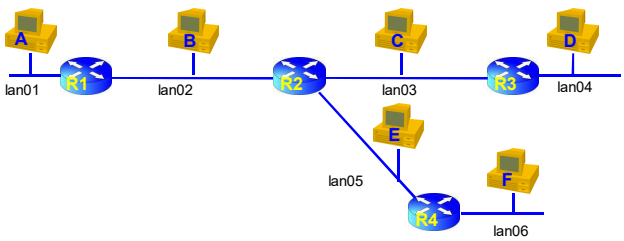

Esercizio 3

Configurare la seguente rete



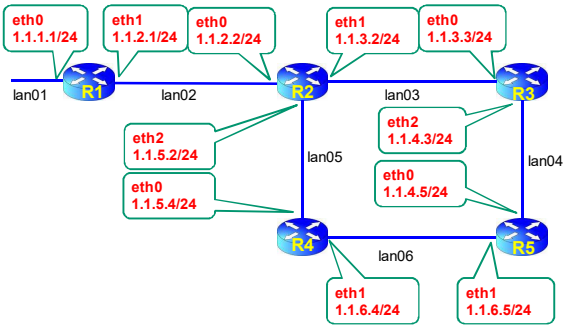
Esercizio 3a

Verificare che tutti gli host possano comunicare tra loro



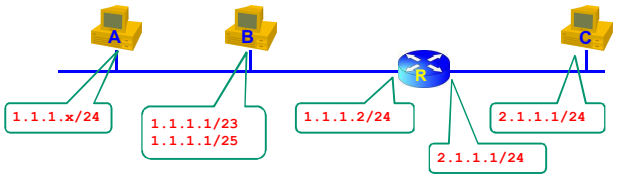
Esercizio 3b

Configurare la seguente rete e verificarne il corretto funzionamento

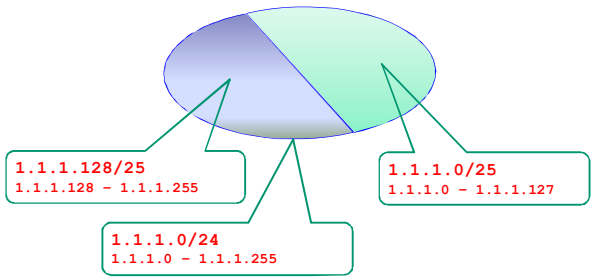


Esercizio 4

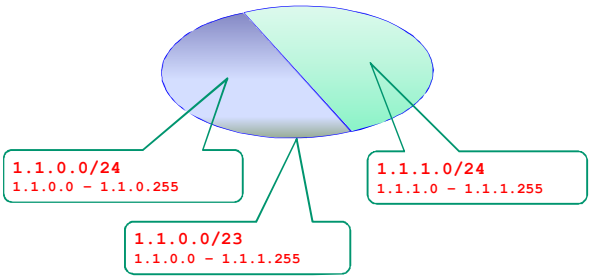
Un nodo in una lan ha la maschera di rete sbagliata. Descrivere teoricamente cosa può succedere nei vari scenari possibili. Successivamente verificare quanto detto.



Ipotesi 1: maschera più grande (24 -> 25)

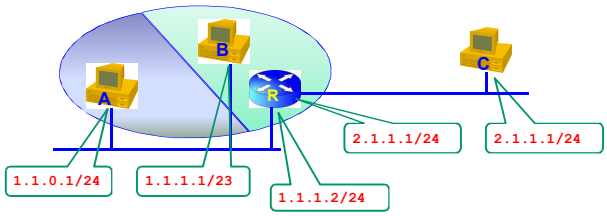


Ipotesi 2; maschera più piccola (24 -> 23)



Maschera più piccola (24 -> 23)

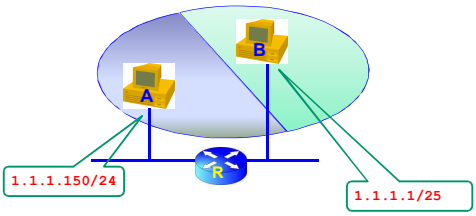
Rete vista dal nodo B



1.1.1.0/24 -> raggiungibile
1.1.0.0/24 -> irraggiungibile
Tutto il resto -> raggiungibile

Maschera più grande (24 -> 25)

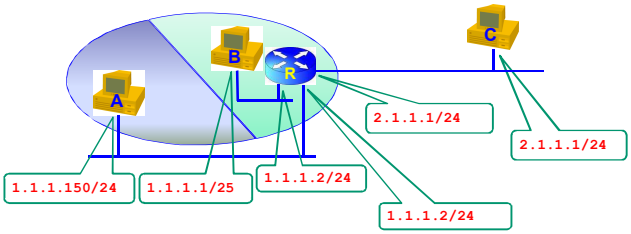
Rete vista dal nodo B



1.1.1.0/25 -> raggiungibile
Il nodo A vede il router R?

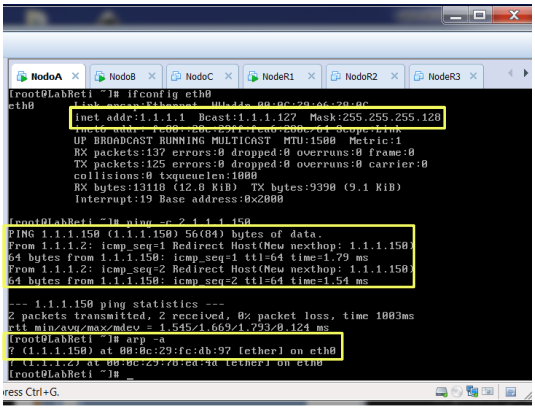
Maschera più grande (24 -> 25)

Rete vista dal nodo B



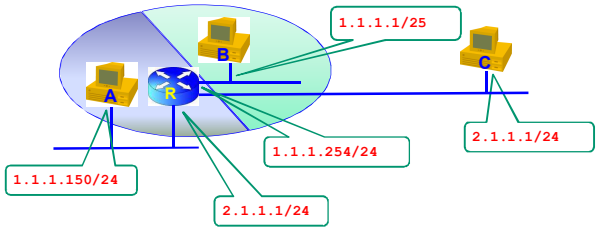
1.1.1.0/25 -> raggiungibile
1.1.1.128/25 -> raggiungibile tramite il router
Tutto il resto -> raggiungibile

Maschera più grande (24 -> 25)



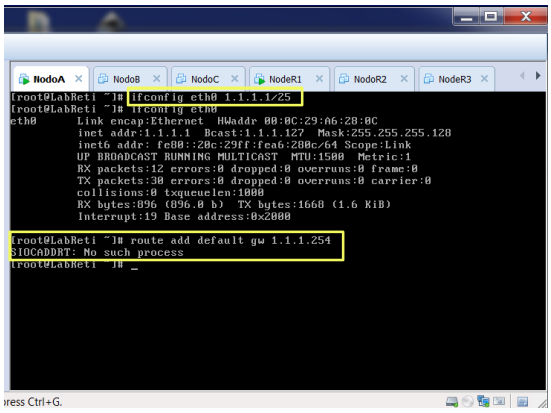
Maschera più grande (24 -> 25)

Rete vista dal nodo B



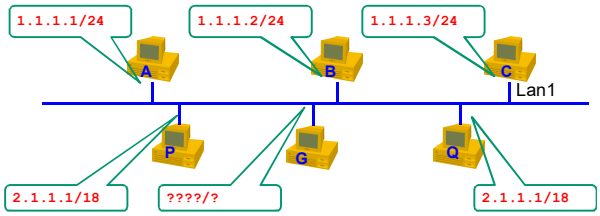
Il nodo B non raggiunge il router R

Maschera più grande (24 -> 25)



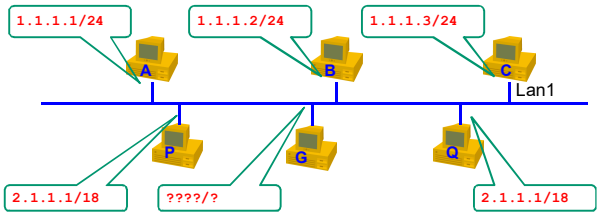
Esercizio 5a

Data la lan in figura (unica per tutti i nodi) configurare il nodo G in modo che gli altri nodi possano comunicare tra loro.



Esercizio 5a

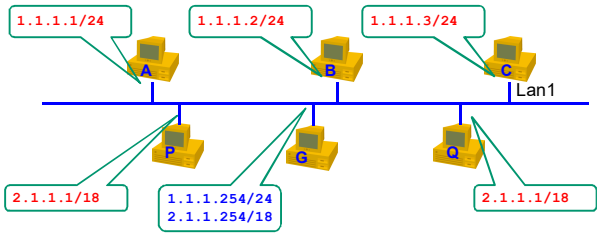
Tutti i nodi saranno configurati con un default GW della propria subnet




```
route add default gw 1.1.1.254 eth0

route add default gw 2.1.1.254 eth0
```


Esercizio 5a

Il nodo G deve avere due indirizzi



Virtual network interfaces

È possibile creare una virtual network interface su una interfaccia reale.

```
# ifconfig eth0:0
eth0:0    Link encap:Ethernet  HWaddr 3c:97:0e:02:98:c8
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Interrupt:20 Memory:f1600000-f1620000

# ifconfig eth0:0 123.123.22.22/24
# ifconfig eth0:0
eth0:0    Link encap:Ethernet  HWaddr 3c:97:0e:02:98:c8
          inet addr:123.123.22.22  Bcast:123.123.22.255
Mask:255.255.255.0
```

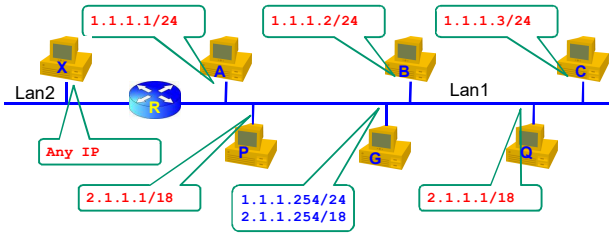
`eth0:0` avrà lo stesso MAC address di `eth0`

Virtual network interfaces

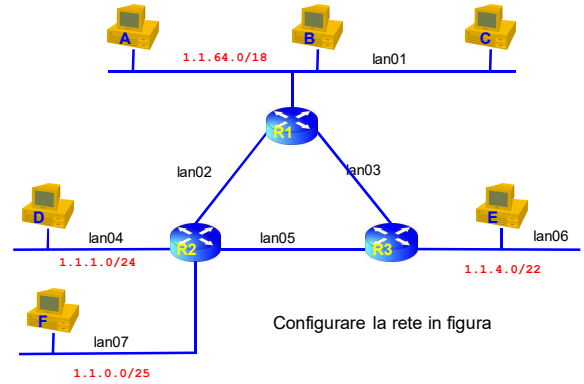
```
[root@localhost ~]# ifconfig eth0:0 2.2.2.2/18
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:32:B2:7B
          inet addr:1.1.1.1  Bcast:1.1.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe32:b27b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:468 (468.0 b)

eth0:0    Link encap:Ethernet  HWaddr 08:00:27:32:B2:7B
          inet addr:2.2.2.2  Bcast:2.2.63.255  Mask:255.255.192.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```


Esercizio 5b

Aggiungere un collegamento verso il mondo esterno

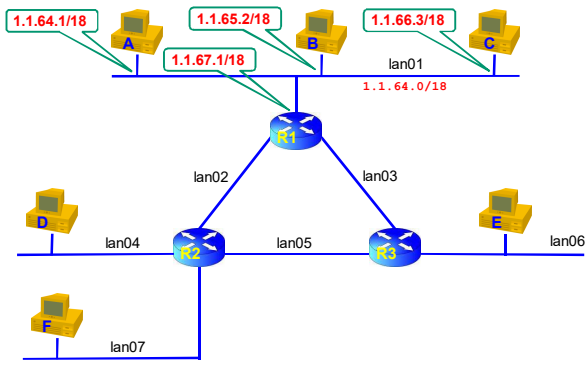


Esercizio 6a

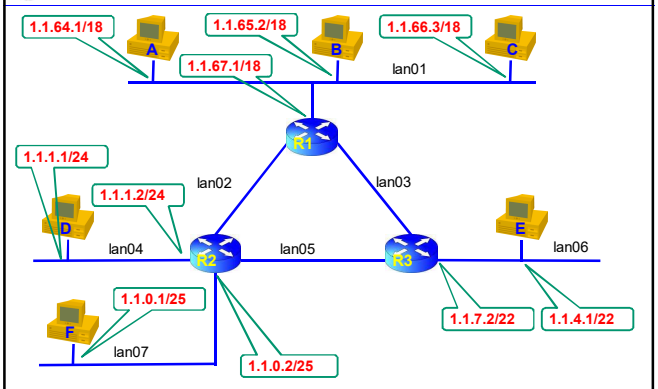


Configurare la rete in figura

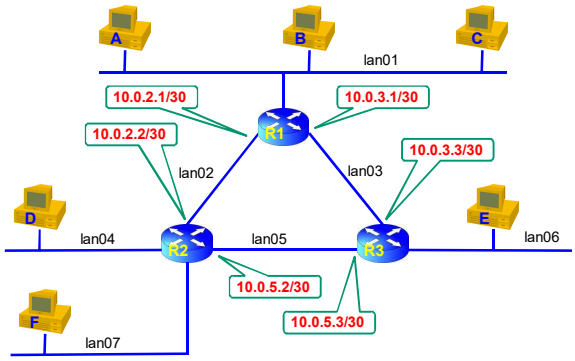
Esercizio 6a



Esercizio 6a

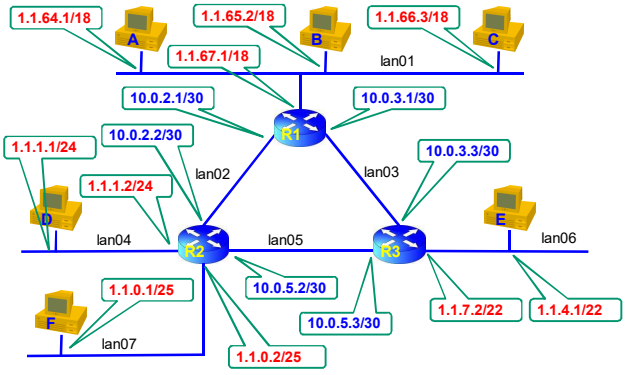


Esercizio 6a

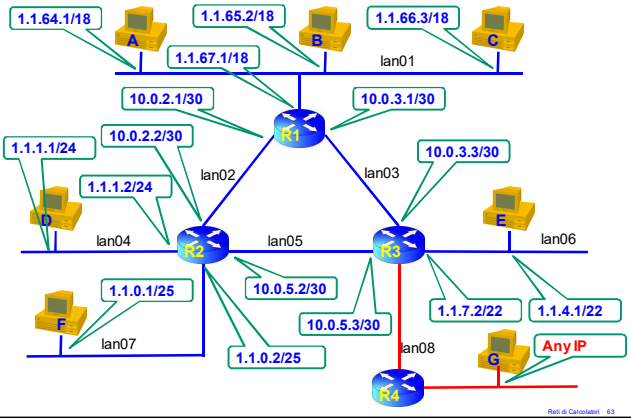


Nota: perché /30 e non /31?

Esercizio 6a

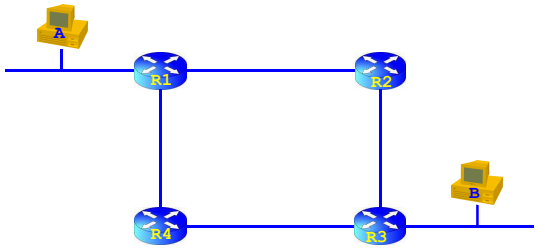


Esercizio 6b

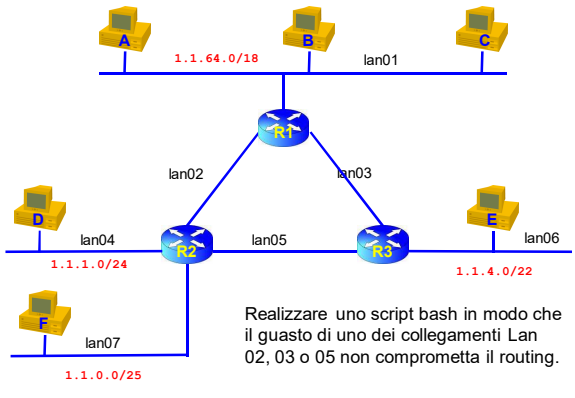


Esercizio 7

Configurare i router R1 R2 R3 R4 in modo che il traffico sui link tra loro sia unidirezionale (in senso orario)

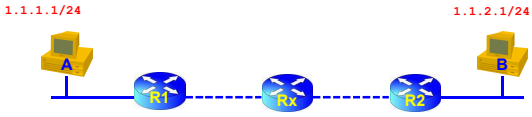


Esercizio 8



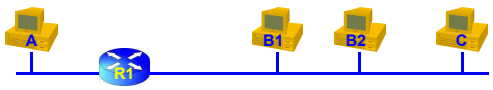
Realizzare uno script bash in modo che il guasto di uno dei collegamenti Lan 02, 03 o 05 non comprometta il routing.

Esercizio 9



Configurare la rete in modo che Rx non sia raggiungibile tramite ping da A o B
Tramite Rx passa il traffico da A verso B

Esercizio 10

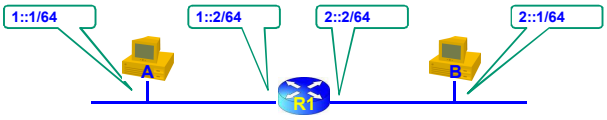


Configurare la rete in modo che, quando B1 è spenta, il traffico diretto a B1 vada in automatico a B2.

Esercizio 11

Configurare la rete utilizzando solo indirizzi IPv6

Esercizio 11



ifconfig eth0 inet6 add 1::1/64
ifconfig eth0 up

route -n -A inet6

ping6 1::2

Esercizio 11

```
[root@localhost ~]# route -n -A inet6
Kernel IPv6 routing table
Destination                    Next Hop
   Flags Metric Ref    Use Iface
1::/64                         ::
   U     256    0        0 eth0
fe80::/64                      ::
   U     256    0        0 eth0
::1/128                        ::
   U     0      9        1 lo
1::1/128                       ::
   U     0      8        1 lo
fe80::a00:27ff:fe32:b27b/128   ::
   U     0      3        1 lo
ff00::/8                       ::
   U     256    0        0 eth0
[root@localhost ~]# _
```