

Università degli Studi di Roma "Tor Vergata"
Laurea in Informatica

Sistemi Operativi e Reti
(modulo Reti)
a.a. 2024/2025

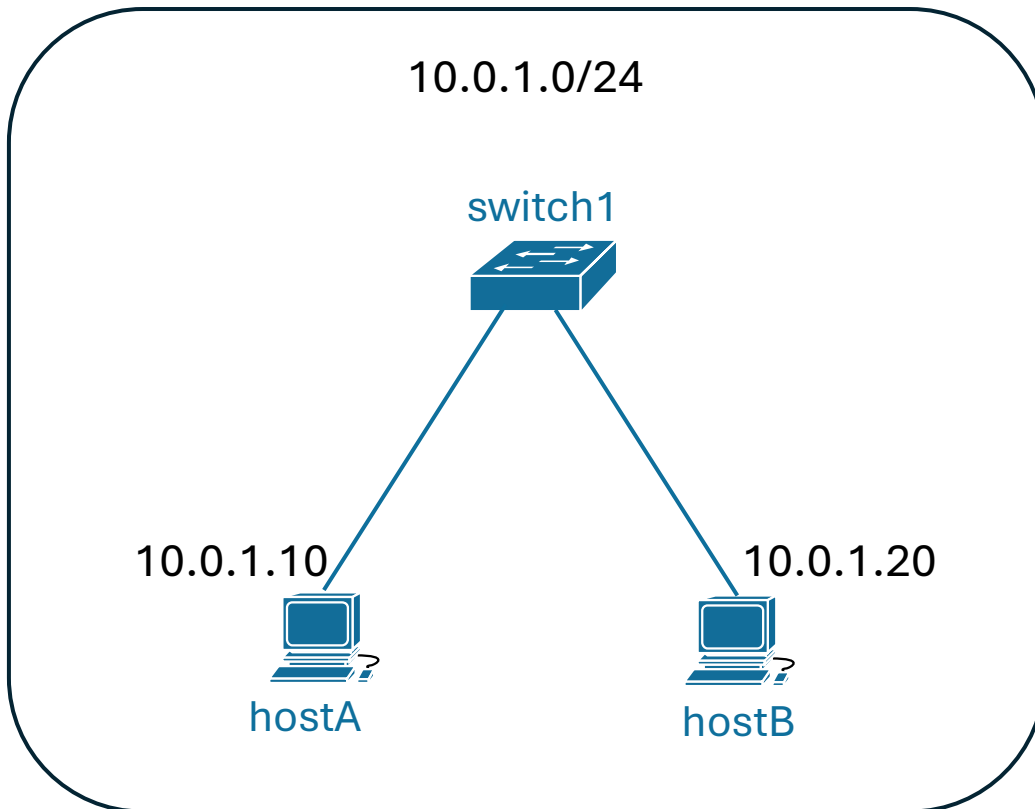
Esercitazione: virtual networking e comandi vari

dr. Manuel Fiorelli

manuel.fiorelli@uniroma2.it

<https://art.uniroma2.it/fiorelli>

Sottorete 1: virtual networking e namespace

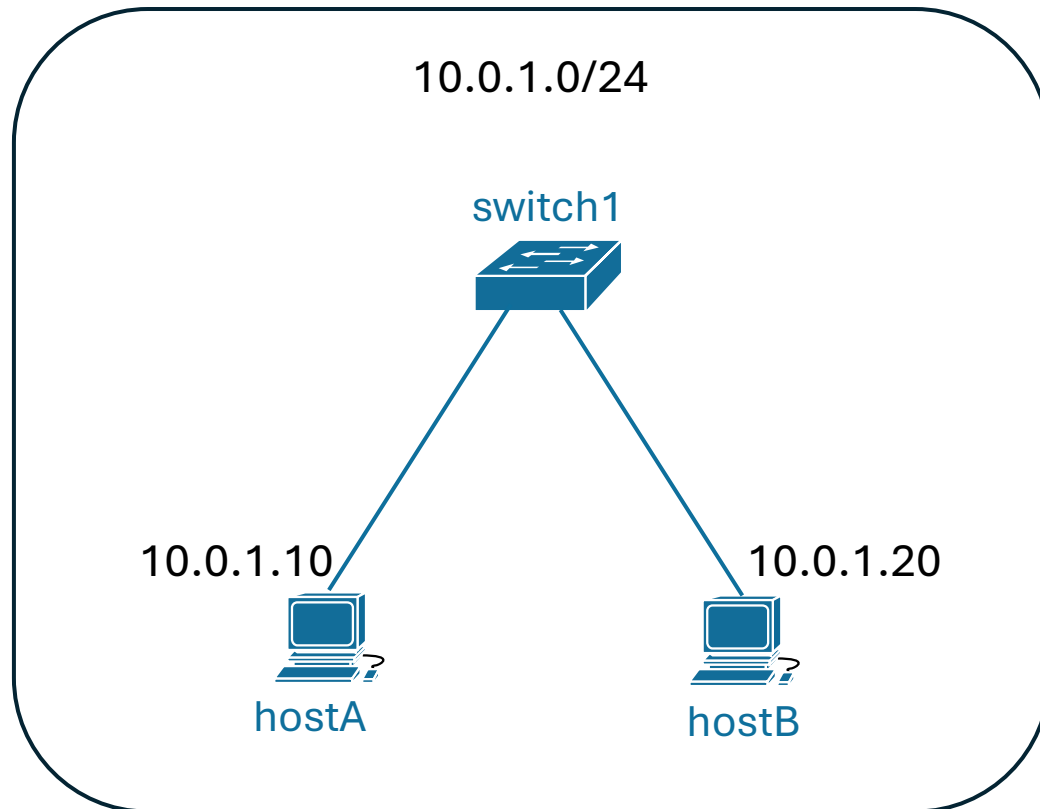


Implementeremo questa topologia di rete usando i **network namespace** di Linux e (alcune) delle sue funzionalità per il **virtual networking**.

Un **network namespace** è uno stack di rete virtuale con associate interfacce (fisiche o virtuali), indirizzi, tabelle di instradamento, etc. Ogni interfaccia di rete può appartenere a un solo network namespace.

- Ogni host sarà dunque associato a un network namespace differente (in realtà, avremmo dovuto utilizzare ulteriori namespace per isolare altre risorse)

Sottorete 1: *bridge* e *veth*

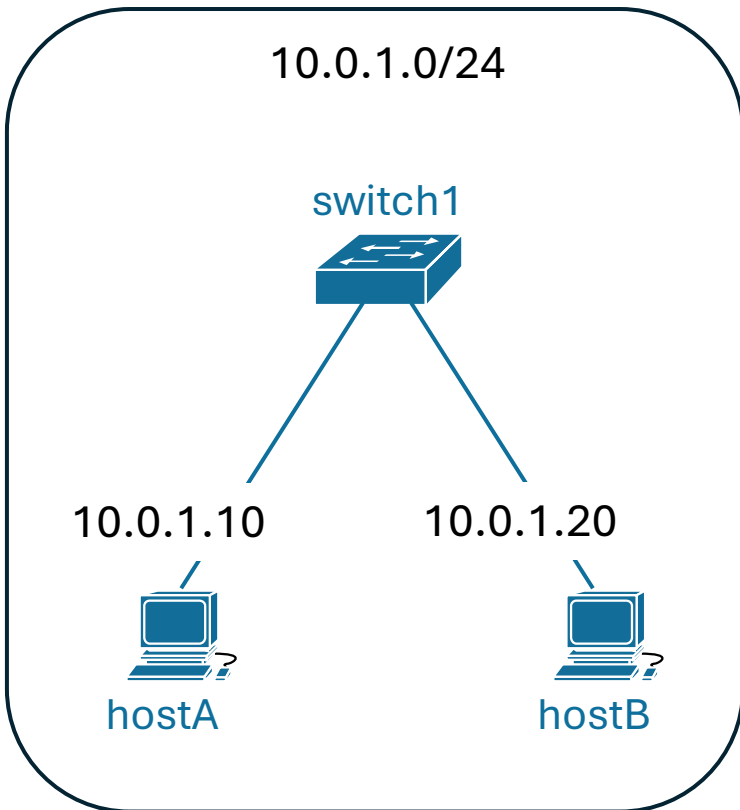


Implementeremo questa topologia di rete usando i **network namespace** di Linux e (alcune) delle sue funzionalità per il **virtual networking**.

Il support al **virtual networking** in Linux include un vasto assortimento di dispositivi di rete virtuali, tra cui:

- **bridge**: si comportano in *modo simile* agli switch (Open vSwitch è un'alternativa più completa per la NFV, ma non supportata da WSL)
- **veth**: coppie di adattatori di rete Ethernet virtuali direttamente connessi tra loro

Sottorete 1: creazione *netns*



Creo un *network namespace* (uno per ciascun host)

```
$ sudo ip netns add hostA
```

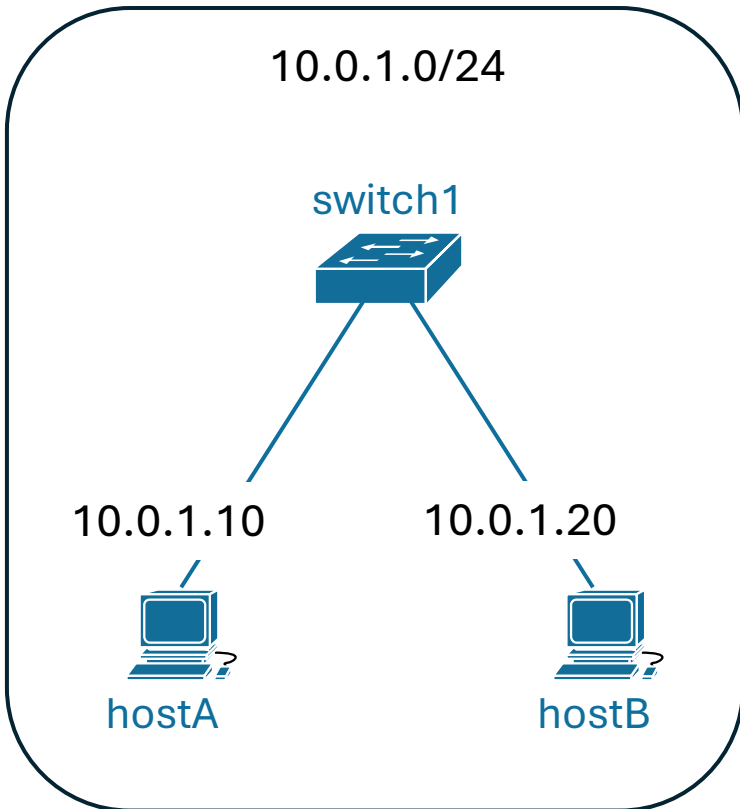
gestione network
namespace (netns)

nome del namespace
creazione di un nuovo namespace

Per il nuovo namespace viene creato un file speciale
dentro `/run/netns/`

```
$ ls /run/netns  
hostA  hostB
```

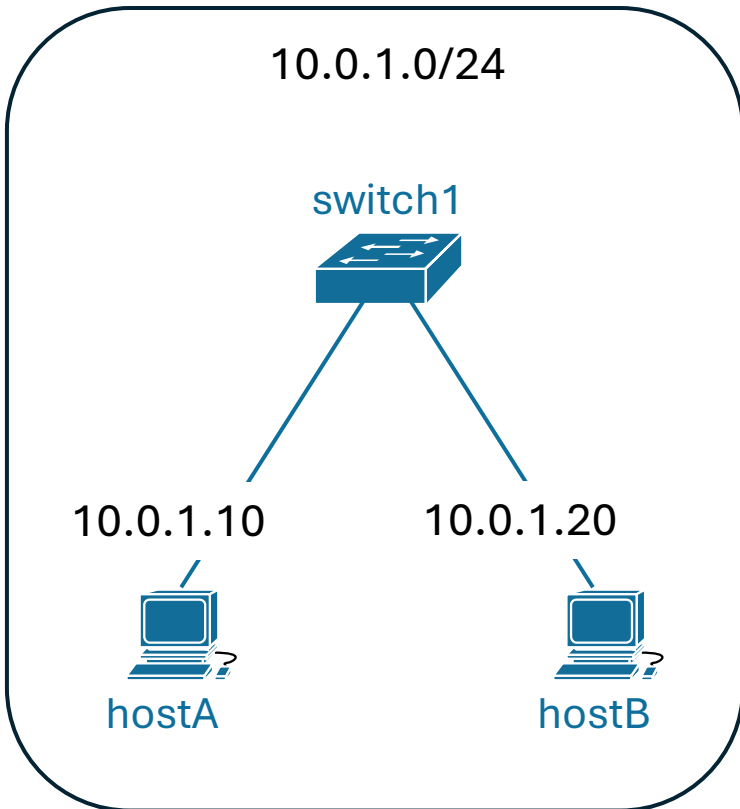
Sottorete 1: enumerazione netns



Elenco i network namespace

```
$ ip netns list  
hostB (id: 2)  
hostA (id: 1)
```

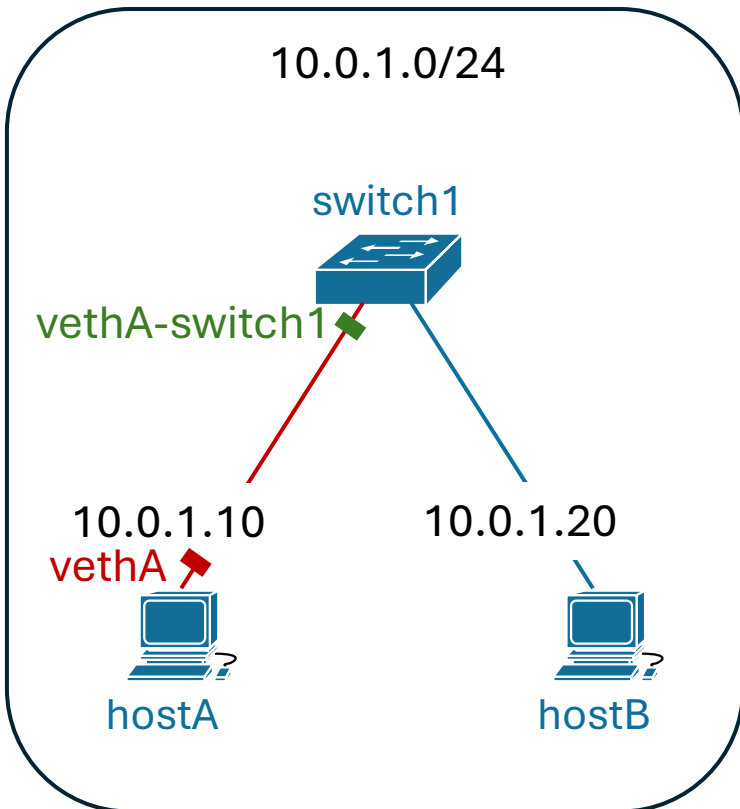
Sottorete 1: cancellazione netns



Rimuove il file speciale associata a un namespace da `/run/netns`: quando non ci saranno più utilizzatori del namespace (ovvero processi che usano quel namespace), questo verrà distrutto (le interfacce virtuali sono distrutte, mentre fisiche sono riassegnate al namespace predefinito)

```
$ sudo ip netns del hostA
```

Sottorete 1: creazione/distruzione veth



Aggiungo un collegamento virtuale

```
$ sudo ip link add vethA type veth peer name vethA-switch1
```

nome collegamento

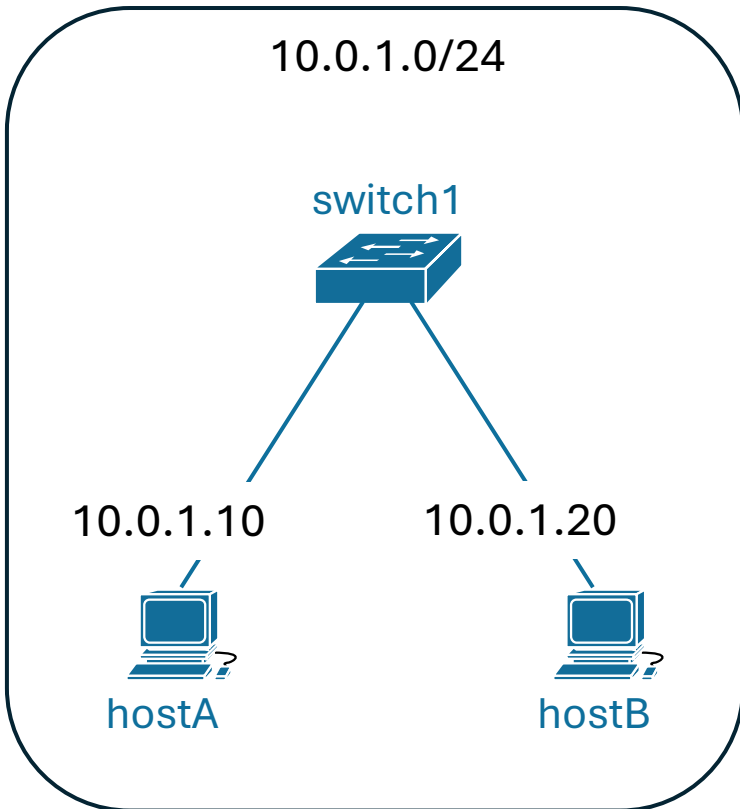
tipo di collegamento

altra estremità del
collegamento veth

Se voglio distruggere un collegamento?

```
$ sudo ip link del vethA-switch1
```

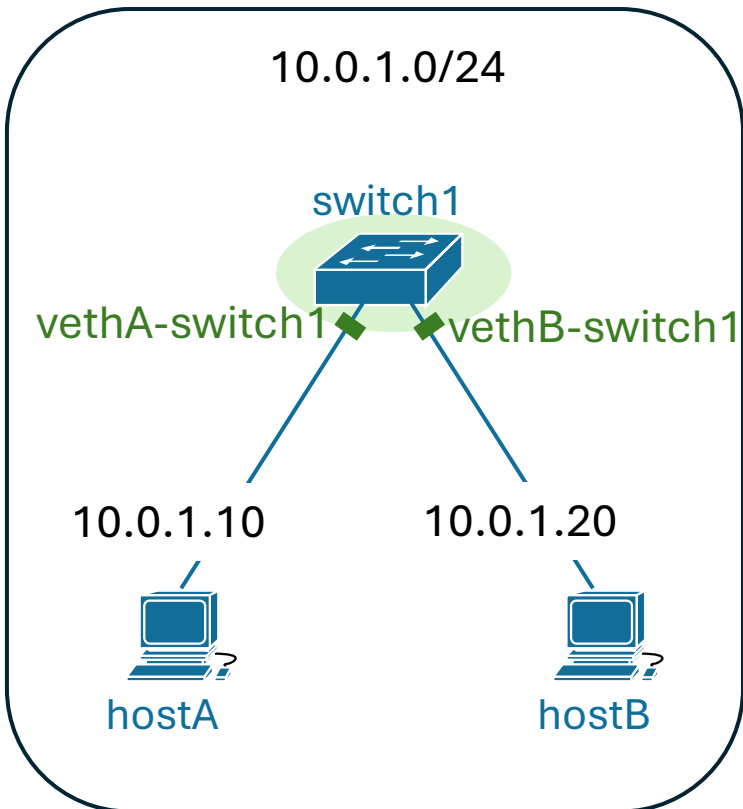
Sottorete 1: spostare un device in un netns



Prima di procedere oltre (es. impostare gli indirizzi IP), occorre spostare i collegamenti "lato host" nel network namespace creato per essi.

```
$ sudo ip link set vethA netns hostA  
$ sudo ip link set vethB netns hostB
```


Sottorete 1: creazione di un bridge



Creo un bridge (si noti il type del collegamento che si sta creando)

```
$ sudo ip link add name switch1 type bridge
```

Collega le estremità "bridge" al bridge

```
$ sudo ip link set vethA-switch1 master switch1
```

```
$ sudo ip link set vethB-switch1 master switch1
```

Attiva il bridge e le interfacce lato bridge

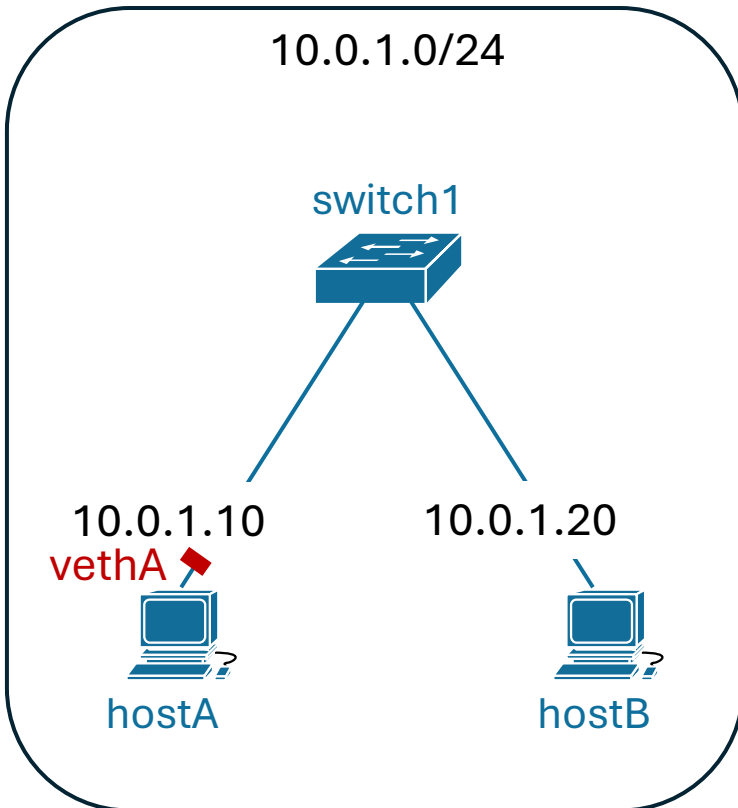
```
$ sudo ip link set switch1 up
```

```
$ sudo ip link set vethA-switch1 up
```

```
$ sudo ip link set vethB-switch1 up
```

Nota: il bridge non può essere spostato in un network namespace.

Sottorete 1: Eseguire un comando in un netns



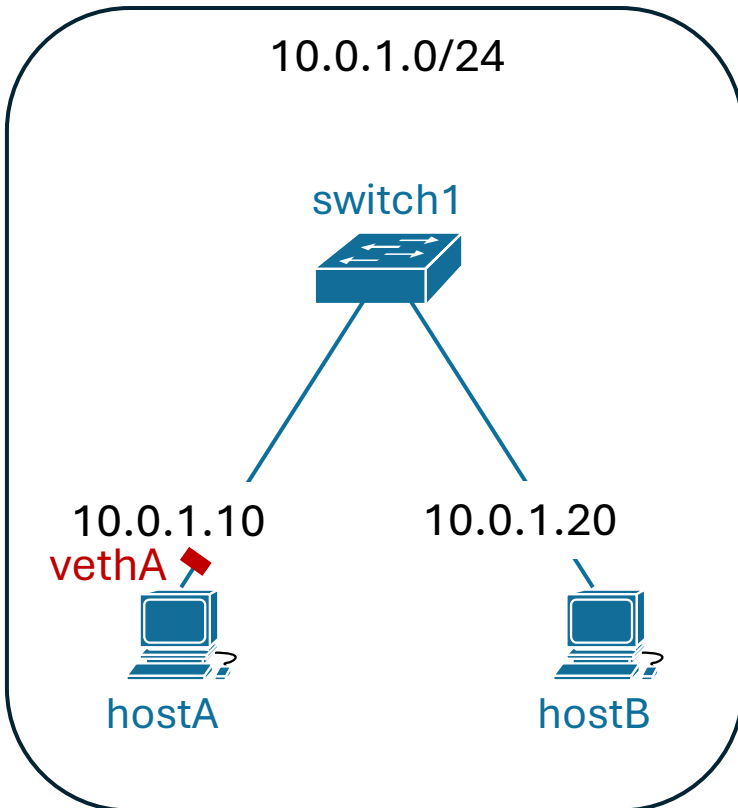
```
$ sudo ip netns exec <nome netns> <comando>
```

Nel seguito useremo questo comando per eseguire diversi comandi ip dentro un network namespace!

Se vogliamo avere una shell in nel network namespace hostA (per uscire eseguire il comando exit).

```
$ sudo ip netns exec hostA /bin/bash
```

Sottorete 1



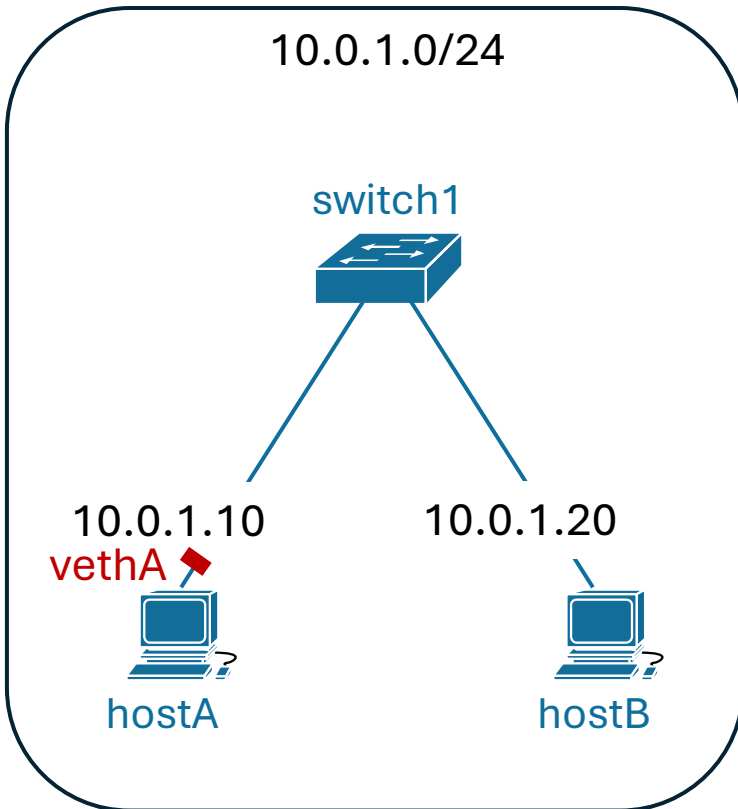
rinomina vethA in eth0: i namespace hanno appunto funzione di namespace, ovvero di spazio di nomi disgiunto dagli altri

```
$ sudo ip netns exec hostA ip link set vethA name eth0
$ sudo ip netns exec hostA ip addr add 10.0.1.10/24 br + dev eth0
$ sudo ip netns exec hostA ip link set eth0 up
$ sudo ip netns exec hostA ip link set lo up
```

attiva i collegamenti

Imposta l'indirizzo IP comprensivo di maschera di rete e imposta l'indirizzo di broadcast al valore corrispondente calcolato in automatico (vedi il +)

Sottorete 1: crea_prima_sottorete.sh



Potete usare lo script `crea_prima_sottorete.sh` per automatizzare la creazione della sottorete 1. Sono anche forniti script per creare ulteriori due sottoreti (vedi dopo).

Assumendo che abbiate reso lo script eseguibile (con `chmod`), potete eseguirlo con `sudo`:

```
$ sudo ./crea_prima_sottorete.sh
```

Le modifiche apportate non sono permanenti e sono perse appena si riavvia la macchina. Tuttavia, onde evitare di creare problemi nell'uso della macchina o di introdurre problemi di sicurezza, si rinnova l'invito a **eseguire tutti i comandi descritti in un ambiente virtuale**.

Più sottoreti

Eseguiamo un ping dal network namespace dell'host A

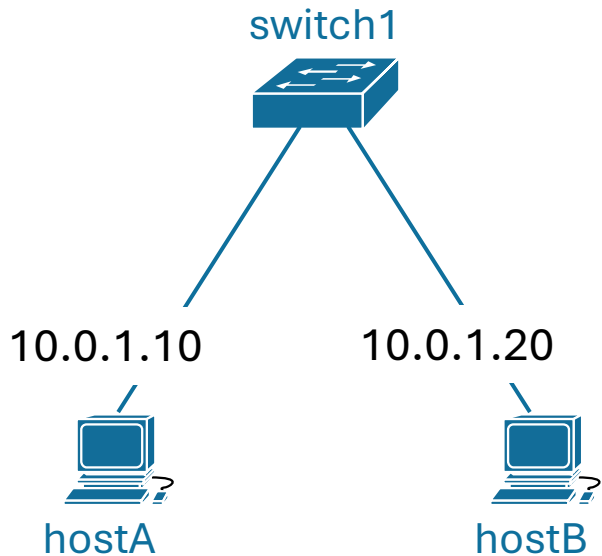
```
$ ping 10.0.3.10
```

```
ping: connect: Network is unreachable
```

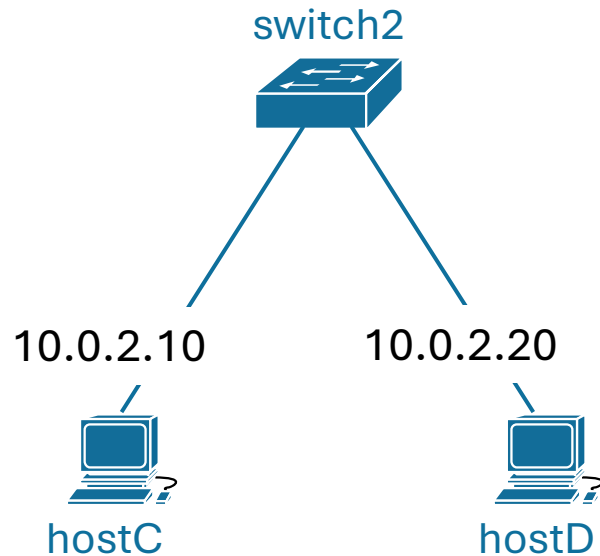
```
$ ip route get 10.0.3.10
```

```
RTNETLINK answers: Network is unreachable
```

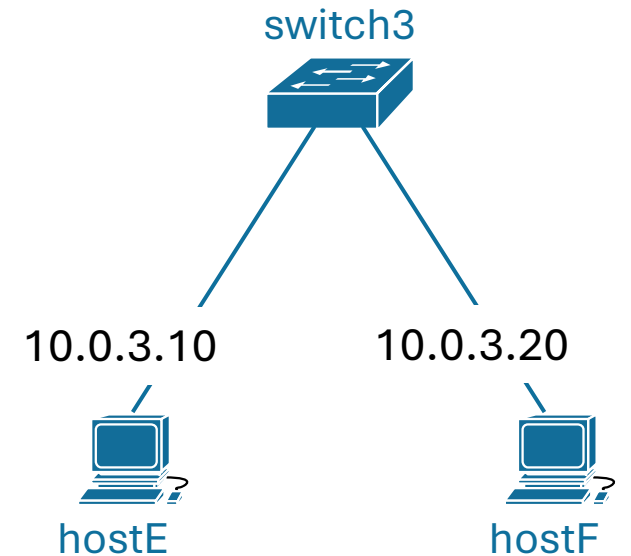
10.0.1.0/24



10.0.2.0/24

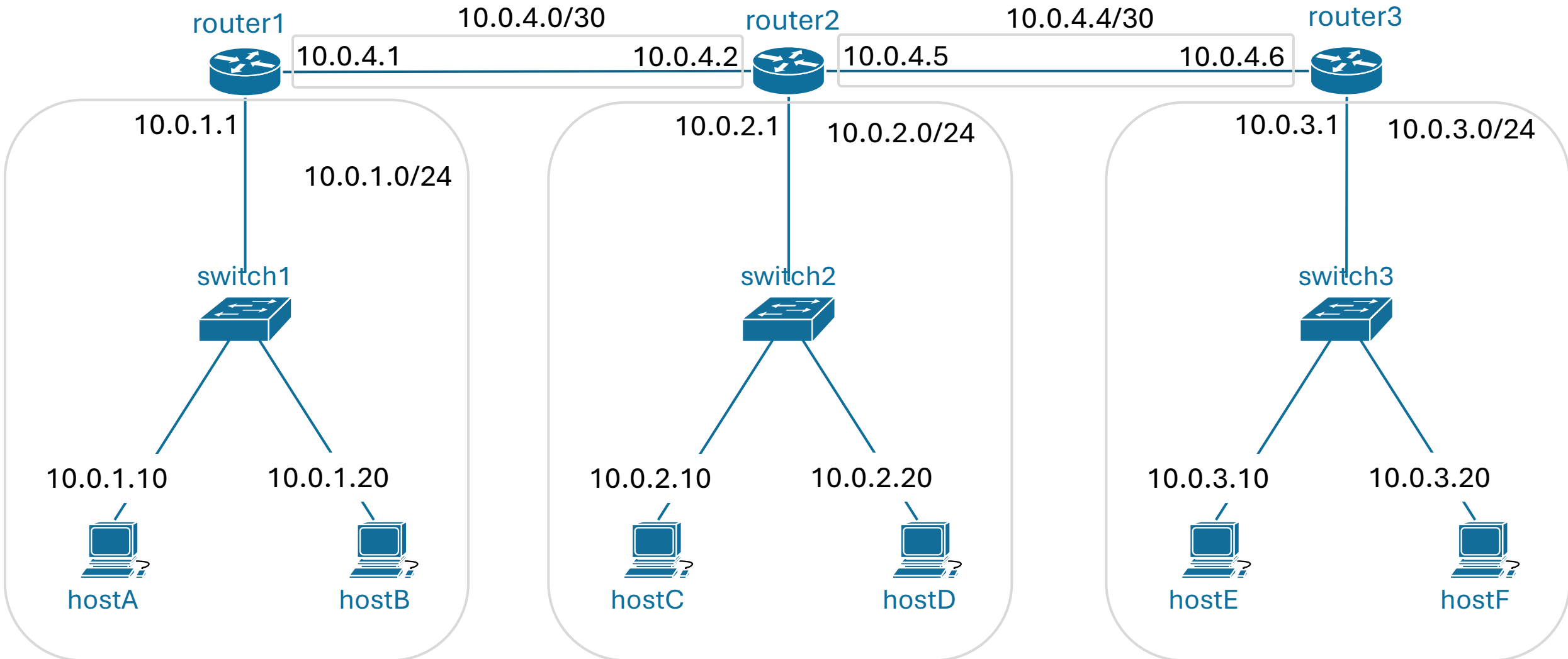


10.0.3.0/24



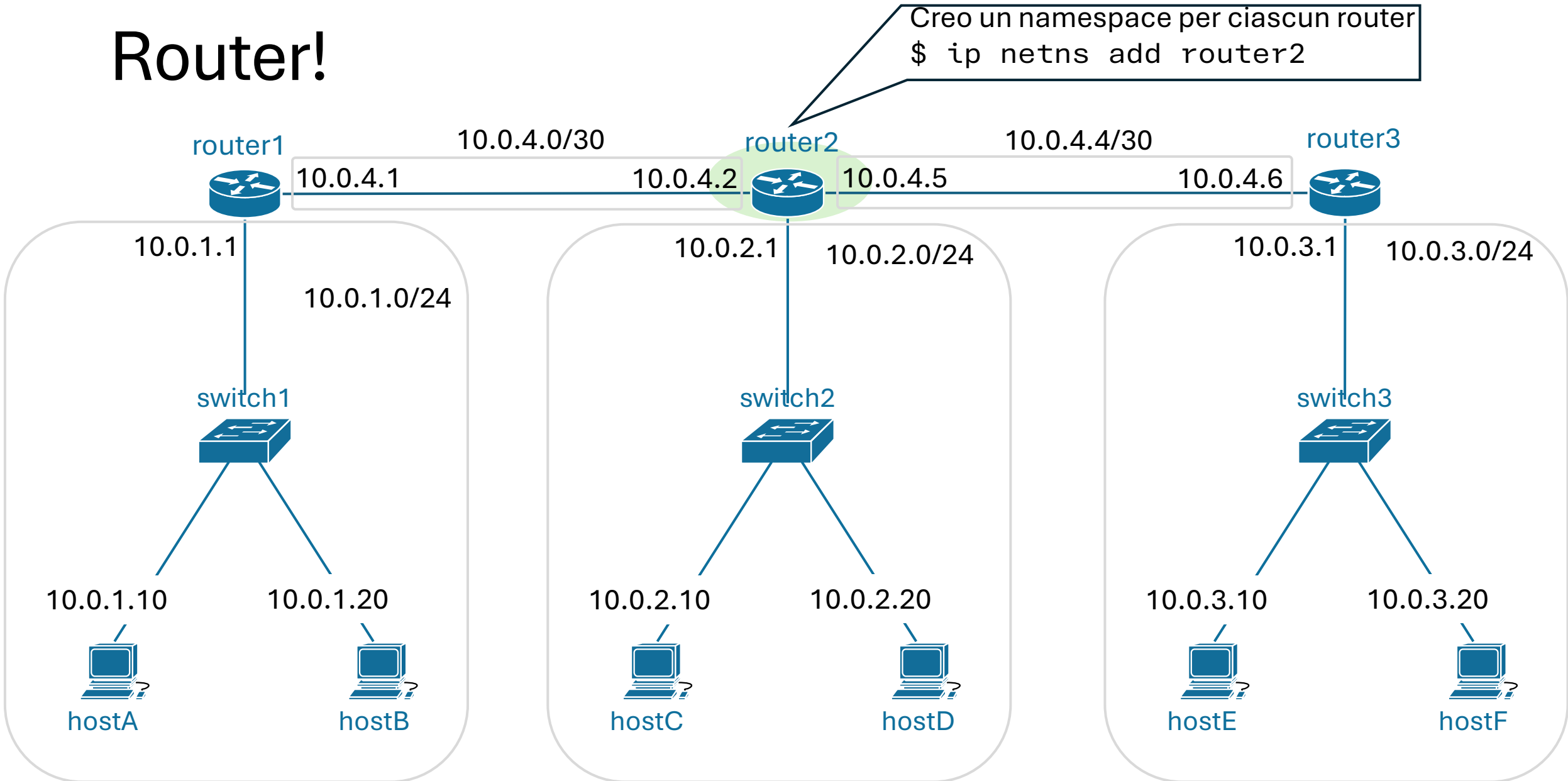
Router!

Viene fornito lo script `crea_router.sh` per automatizzare la creazione del tutto



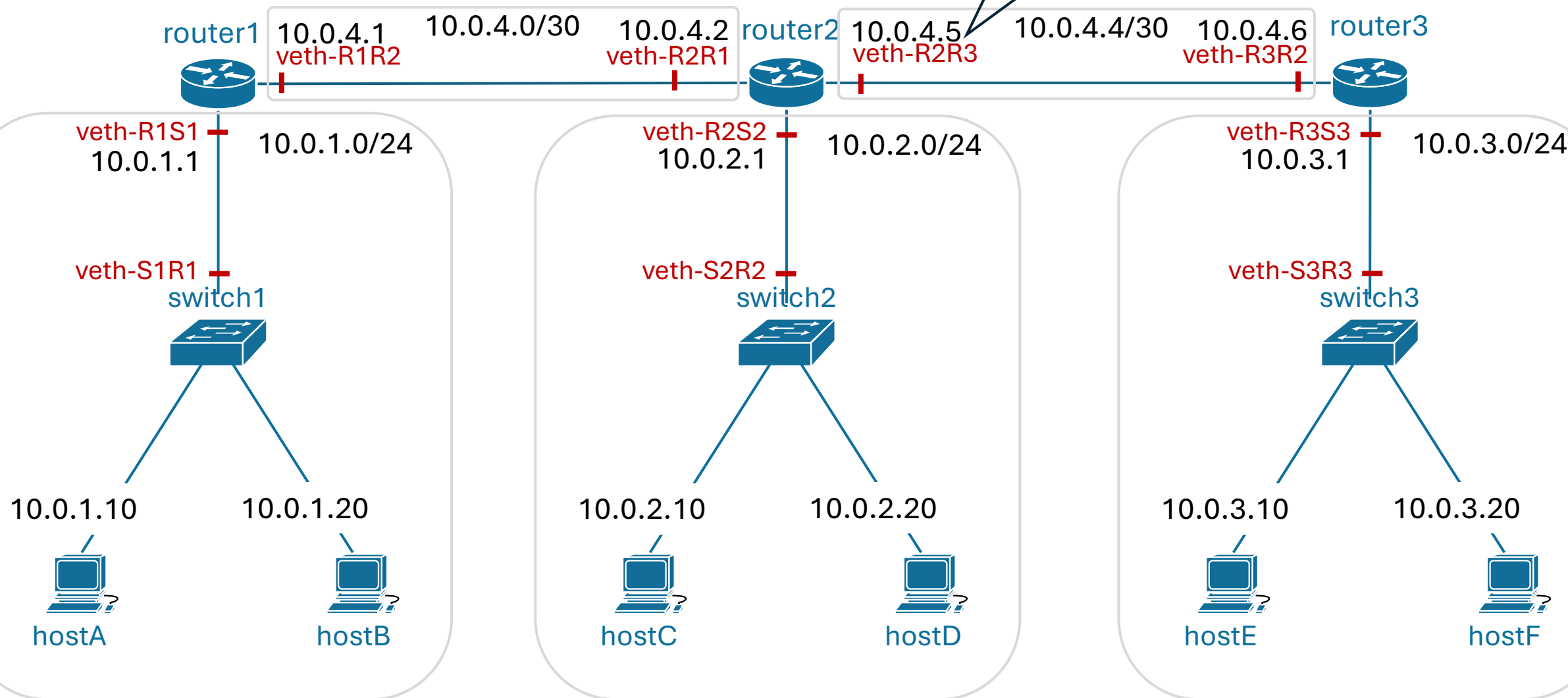
Router!

Creo un namespace per ciascun router
\$ ip netns add router2



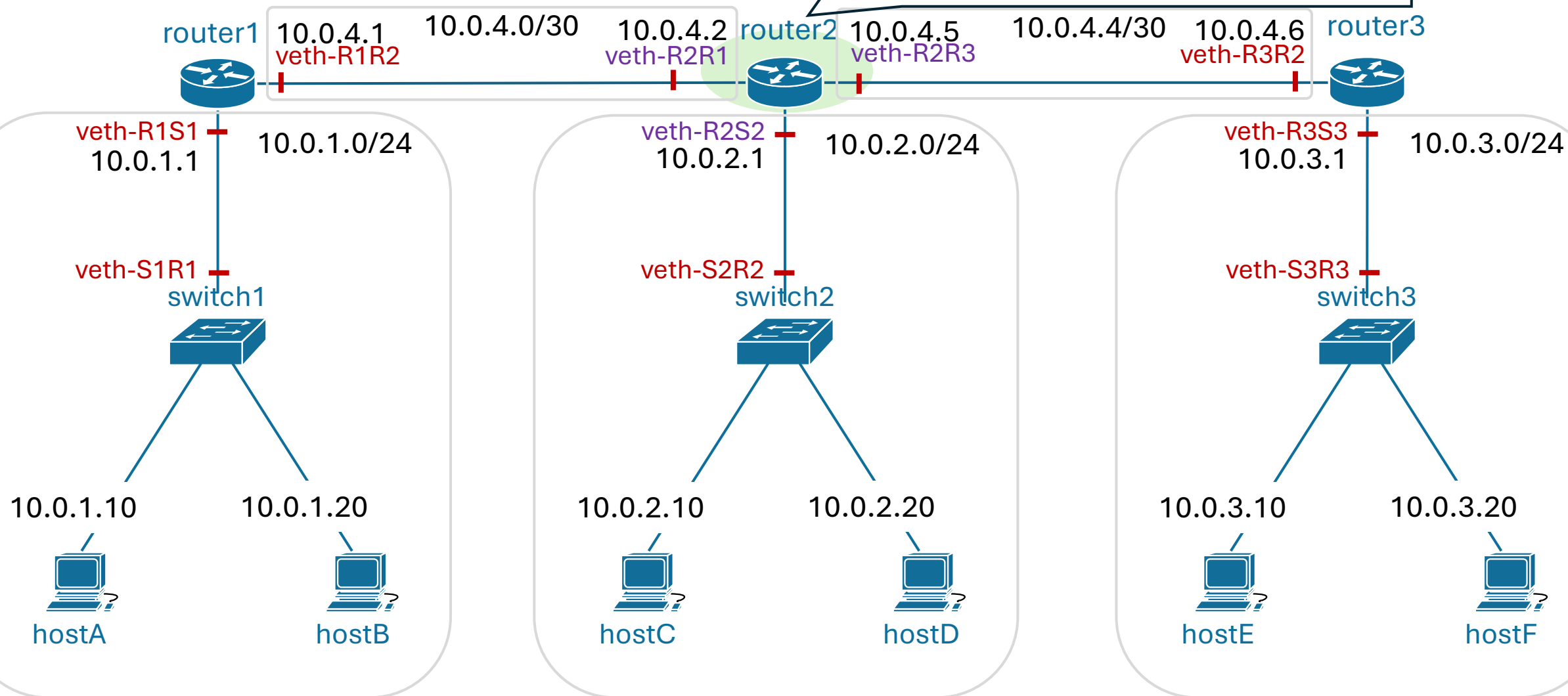
Router!

Dobbiamo creare 5 coppie di veth!

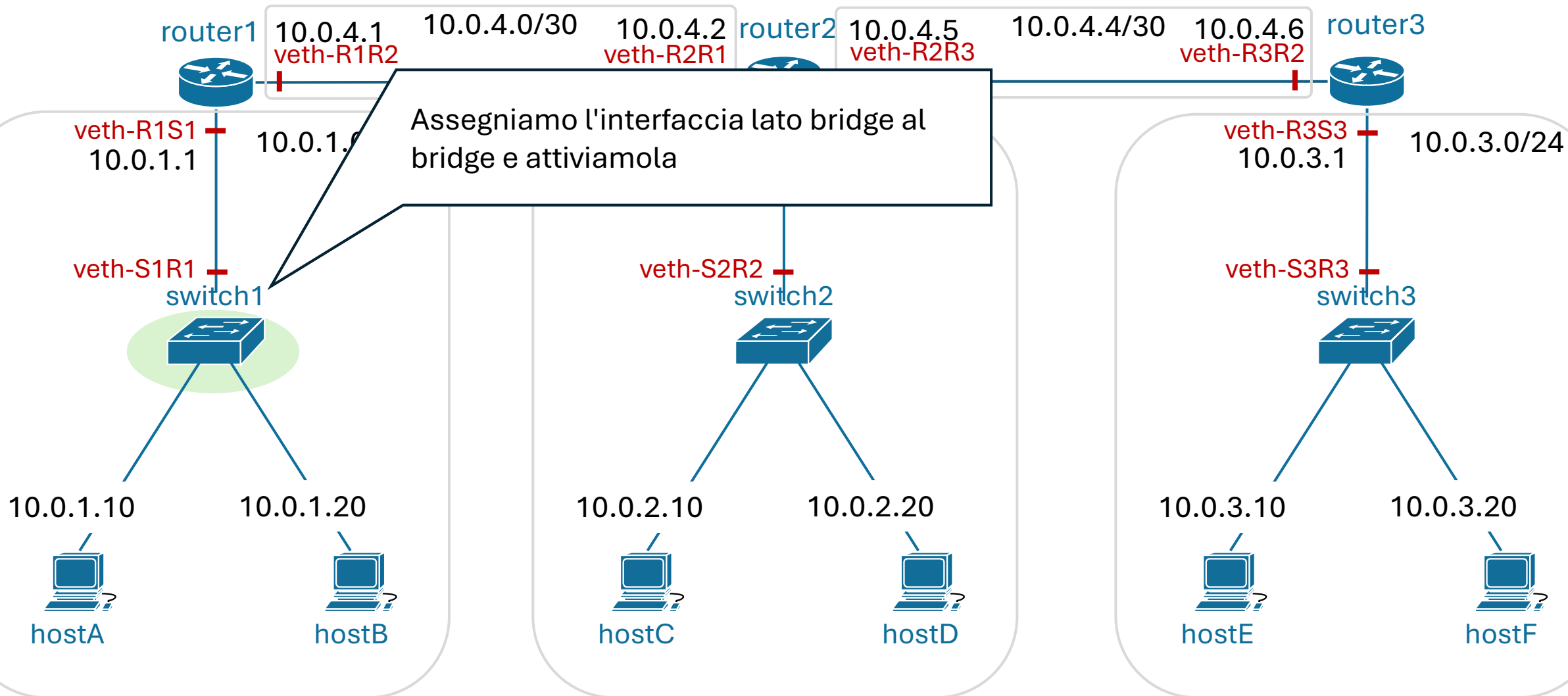


Router!

Dobbiamo assegnare al namespace associato a ciascun router i dispositivi veth lato router

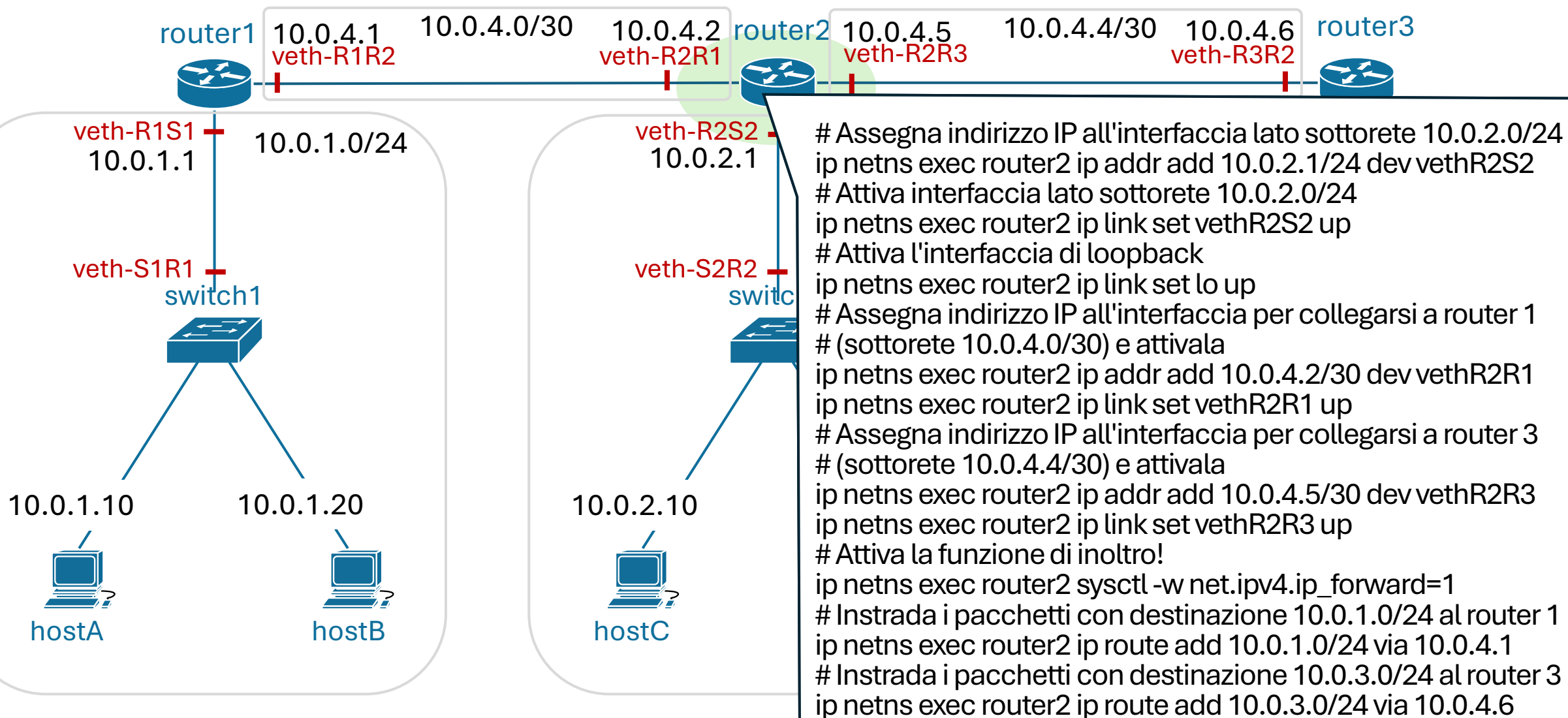


Router!



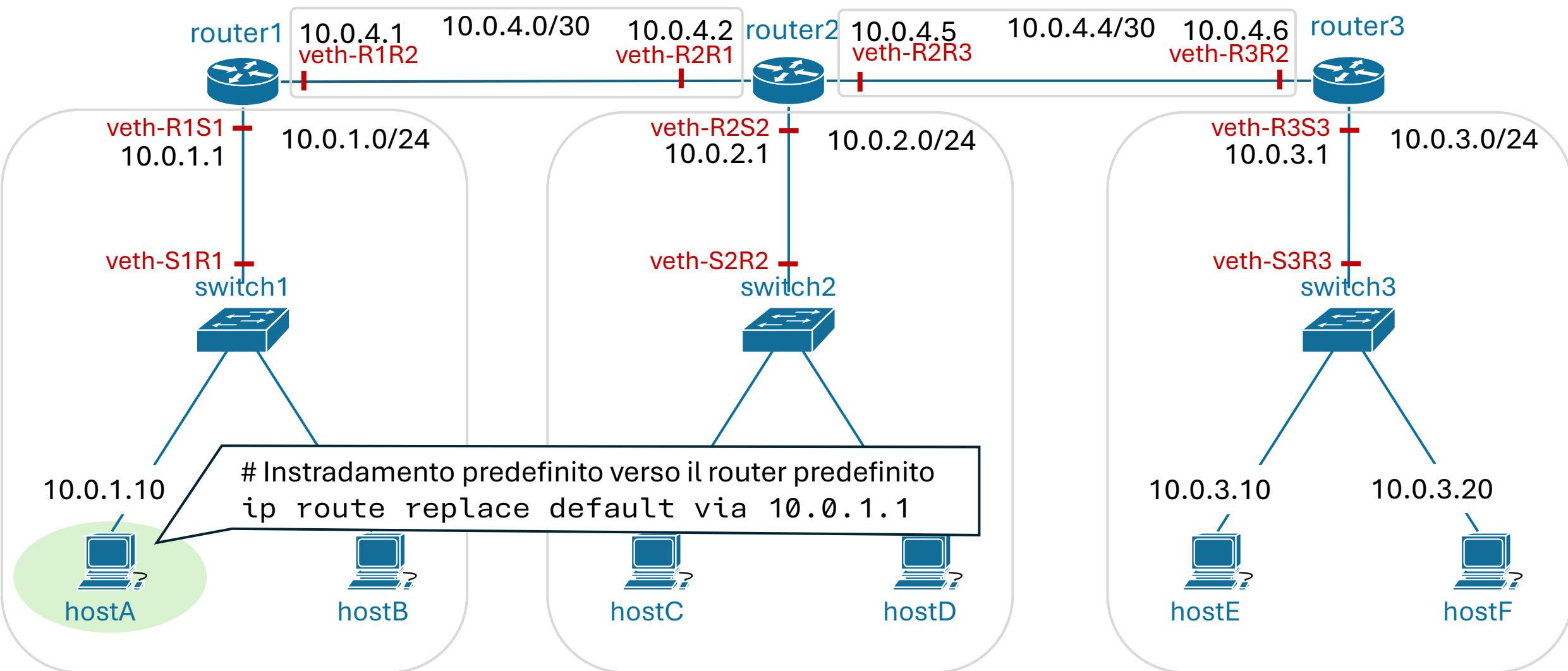
Router!

Configurare in modo simile gli altri router, aggiungendo una rotta per ciascuna sottorete, comprese quelle associate ai collegamenti tra coppie di router.



Router!

Configurare in modo simile tutti gli host



Riferimenti

- <https://manpages.ubuntu.com/manpages/noble/man8/ip-route.8.html>
- <https://manpages.ubuntu.com/manpages/noble/man8/ip-link.8.html>