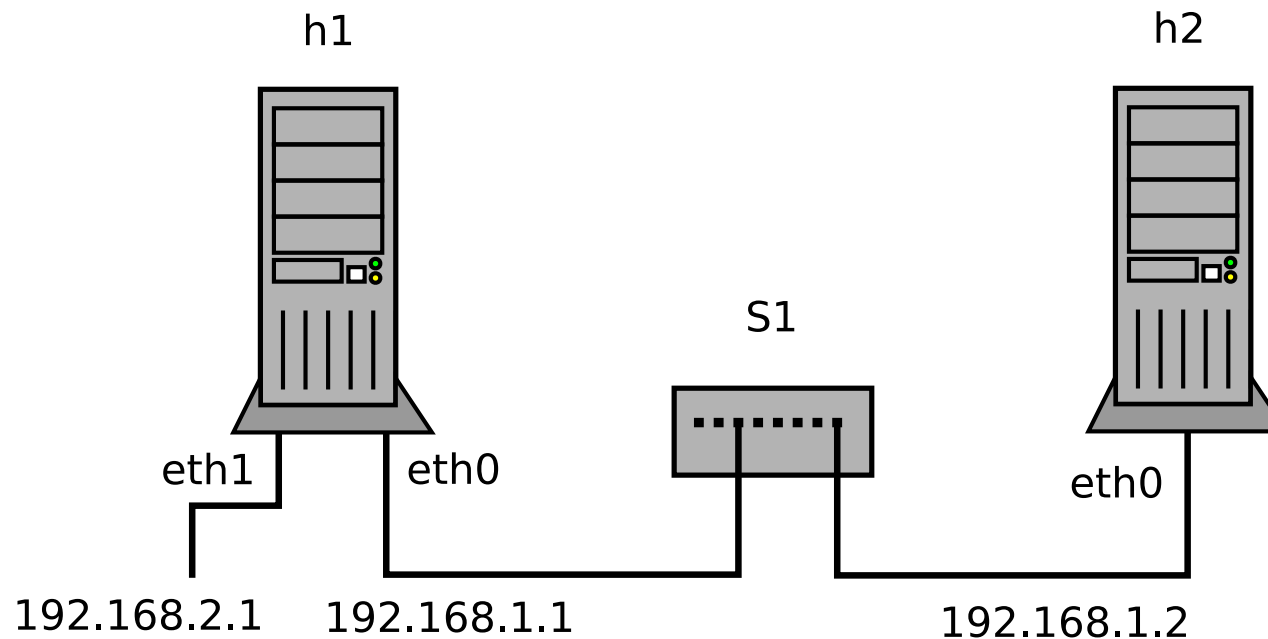


Esercizio di approfondimento (1)

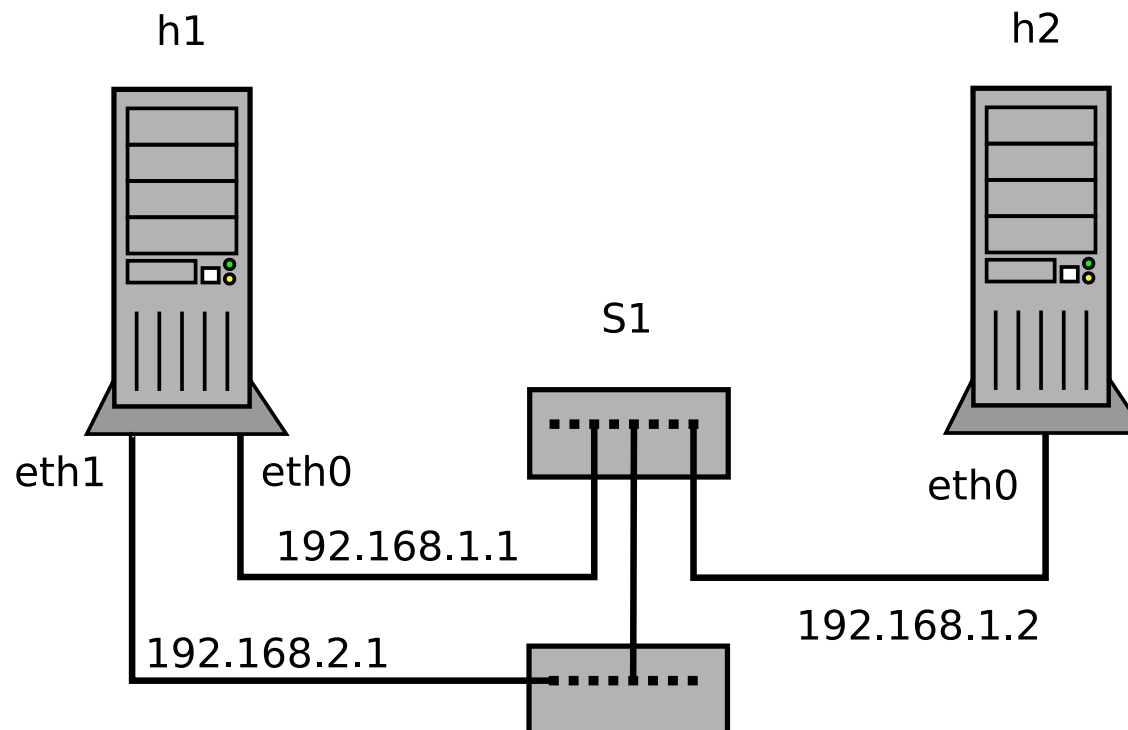
- Provare ad aggiungere un'interfaccia di rete ad **h1**:
 - lasciare scollegata l'interfaccia
 - configurare un indirizzo IP (192.168.2.1)
 - cercare di risolvere il nuovo indirizzo assegnato tramite **arping** da **h2**



Che risultato ci aspettiamo?

Esercizio di approfondimento (2)

- Provare ora collegando la nuova interfaccia allo stesso switch (o a un secondo switch collegato al primo, per ottenere uno schema migliore in Marionnet)
 - cercare di risolvere il nuovo indirizzo assegnato tramite **arping** da **h2**



Anche considerando il risultato ottenuto in precedenza, che risultato ci aspettiamo?

Soluzione approfondimento (1)

Scenario 1: richiesta ARP verso un indirizzo IP assegnato ad un host che appartiene ad un host della nostra sottorete, ma ad un'interfaccia diversa rispetto a quella utilizzata dall'host per comunicare sulla nostra sottorete

```
(h2) # arping -i eth0 -c 3 192.168.2.1
```

```
ARPING 192.168.2.1
```

```
42 bytes from 02:04:06:64:66:db (192.168.2.1): index=0 time=1.012 sec
```

```
42 bytes from 02:04:06:64:66:db (192.168.2.1): index=1 time=1.019 sec
```

```
42 bytes from 02:04:06:64:66:db (192.168.2.1): index=2 time=1.019 sec
```

```
--- 192.168.2.1 statistics ---
```

```
3 packets transmitted, 3 packets received,    0% unanswered (0 extra)
```

```
rtt min/avg/max/std-dev = 1012.330/1016.717/1019.080/3.105 ms
```

→ L'host **h1** risponde alle richieste ARP per l'indirizzo assegnato ad **eth1**, anche se la richiesta è arrivata all'interfaccia **eth0**.

Soluzione approfondimento (2)

Scenario 2: un host ha più interfacce di rete collegate allo stesso dominio di broadcast Ethernet

```
(h2) # arping -i eth0 -c 3 192.168.2.1
```

```
ARPING 192.168.2.1
```

```
42 bytes from 02:04:06:64:66:db (192.168.2.1): index=0 time=1.020 sec
```

```
42 bytes from 02:04:06:59:72:27 (192.168.2.1): index=1 time=1.023 sec
```

```
42 bytes from 02:04:06:64:66:db (192.168.2.1): index=2 time=1.016 sec
```

```
42 bytes from 02:04:06:59:72:27 (192.168.2.1): index=3 time=1.017 sec
```

```
42 bytes from 02:04:06:64:66:db (192.168.2.1): index=4 time=1.019 sec
```

```
42 bytes from 02:04:06:59:72:27 (192.168.2.1): index=5 time=1.019 sec
```

```
--- 192.168.2.1 statistics ---
```

```
3 packets transmitted, 6 packets received,    0% unanswered (3 extra)
```

```
rtt min/avg/max/std-dev = 1016.136/1018.958/1022.690/2.218 ms
```

→ L'host **h1** risponde alle richieste ARP sia tramite l'interfaccia **eth1** sia tramite l'interfaccia **eth0**.

Soluzione approfondimento (3)

Il comportamento predefinito dei sistemi Linux nel gestire il protocollo ARP è abbastanza permissivo:

- risponde per tutte le interfacce configurate, a prescindere da quale interfaccia ha ricevuto la richiesta ARP
 - nel primo caso, **h1** risponde alla richiesta ARP per l'ip assegnato su **eth1**, anche se la richiesta arriva su **eth0**. *Di fatto, questo può rendere più semplice la configurazione di certe reti.*
 - nel secondo caso, entrambe le interfacce **eth0** ed **eth1** di **h1** rispondono alla richiesta ARP, creando un potenziale conflitto nell'host che ha ricevuto la risposta.

In generale, quando un host ha più di un'interfaccia collegata allo stesso dominio di broadcast Ethernet, è spesso preferibile impostare regole più restrittive riguardo la gestione del protocollo ARP.

```
# sysctl -w net.ipv4.conf.all.arp_announce=1
```

```
# sysctl -w net.ipv4.conf.all.arp_ignore=1
```

- Provare a ri-eseguire i test di rete con la nuova configurazione.