

Setup

- Scaricare l'immagine OVA presente sul sito
- Importarla dentro virtualbox
- Avviare la macchina virtuale con RAM > 8GB (credenziali `user:resu`)
- Aprire un terminale e lanciare il comando `marionnet`
- Fare clic su nuovo progetto

NB: - Cavi incrociati (crossover): connessioni fra dispositivi di rete dello stesso livello dello stack (e.g., switch-switch, host-host) - Cavi dritti (straight): connessioni fra dispositivi di livello differente (e.g., switch-host) - Attenzione al numero di interfacce e porte quando crei una macchina o uno switch - Attenzione a selezionare la porta e interfaccia giusta quando colleghi i cavi (double check coi numeri che vengono mostrati)

Setup macchine

Per setuppare le interfacce di una macchina host (indirizzi IP statici o con DHCP ecc.) si va a modificare il file `/etc/network/interfaces` al quale si devono aggiungere le configurazioni riportate di seguito a seconda delle necessità:

Per impostare tutte le interfacce con il loopback (su tutti)

```
auto lo
iface lo inet loopback
```

Per impostare un indirizzo IP statico sull'interfaccia `eth0` (si può ad esempio mettere `/24` nell'address senza specificare la netmask)

```
auto eth0
iface eth0 inet static
    address <ip_address>
    netmask <netmask>
    gateway <gateway_ip>
    TODO: post-up vari
```

Per impostare la futura recezione di indirizzo IP dinamico tramite DHCP, si deve impostare il MAC address (dove `02:04:06:c0:2d:9b` è il MAC address dell'interfaccia della macchina, non del server DHCP)

```
auto eth0
iface eth0 inet dhcp
    hwaddress ether 02:04:06:c0:2d:9b
```

VLAN

Per impostare le porte (tagged, untagged) sullo switch si deve, nella scheda di configurazione, attivare l'opzione `Startup Configuration` e fare clic su `Modifica`. Si aprirà un file di configurazione dove dovranno essere impostati i seguenti comandi a seconda delle necessità:

Creare una VLAN (con identificativo `10`)

```
vlan/create 10
```

Aggiungere la porta `3` alla VLAN `10` (tagged, alla porta `3` c'è solo la VLAN `10`)

```
# port/setvlan PORT VLAN
port/setvlan 3 10
```

Aggiungere la porta `4` alla VLAN `10` e `20` (untagged, alla porta `4` possono coesistere più VLAN)

```
# vlan/addport VLAN PORT
vlan/addport 10 4
vlan/addport 20 4
```

In generale il GW che farà da tramite per le 2 VLAN non dovrà avere delle configurazioni con `eth0` ma con `eth0.x` se necessario.

Ad esempio, se GW[1] gestisce le VLAN `10` e `20` ai quali sono collegati rispettivamente Srv1[2] e Srv2[3]

```
vlan/create 10
vlan/create 20

vlan/addport 10 1
vlan/addport 20 1

port/setvlan 2 10
port/setvlan 3 20
```

Setup DHCP

Sulla macchina che farà da server DHCP si deve installare il servizio con `apt install isc-dhcp-server`.

Per specificare su quali interfacce si deve attivare il DHCP si deve modificare il file `/etc/default/isc-dhcp-server` e modificare la chiave `INTERFACES` (es. `INTERFACES="eth0"` oppure `INTERFACES="eth0.10 eth0.20"`).

Per configurare i parametri DHCP si deve invece modificare il file `/etc/dhcp/dhcpd.conf` con le configurazioni riportate di seguito a seconda delle necessità:

Per impostare i parametri DHCP di una rete con indirizzo di rete `192.168.1.0/32`, gateway raggiungibile al `192.168.1.254` e pool di indirizzi da servire dal `.10` al `.19` (`authoritative`; da impostare una sola volta per tutto il file)

```
authoritative;

shared-network 192-168-1 {
    subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers 192.168.1.254;
    }
    pool {
        range 192.168.1.10 192.168.1.19;
    }
}
```

NB: - la sezione pool si può eliminare se non richiesta - nel caso di DHCP che deve servire su 2 VLAN si dovranno scrivere 2 blocchi di questo tipo all'interno del file di configurazione. Si deve inoltre fare attenzione a inserire l'indirizzo del gateway correttamente

Per impostare un indirizzo IP su una macchina tramite DHCP statico (dove `02:04:06:c0:2d:9b` è il MAC address della macchina in questione e `server1` il suo nome):

```
host server1 {
    hardware ethernet 02:04:06:c0:2d:9b;
    fixed-address 192.168.1.3;
}
```

Una volta configurato il tutto: - Per avviare il server DHCP: `service isc-dhcp-server start` - Per attivare il servizio all'avvio del server: `systemctl enable isc-dhcp-server`

Routing

Una volta impostati i vari indirizzi IP si possono impostare anche le tabelle di route, specificando dove trovare gli altri indirizzi IP. Si deve aggiungere un `post-up` nell'interfaccia alla quale si può raggiungere il secondo host.

Per aggiungere una route dall'host corrente, collegato con `eth0` verso uno switch al quale è collegato l'host destinatario `1.1.1.1` allora: `post-up route add -host 1.1.1.1 dev eth0`.

In maniera reciproca si dovrà aggiungere una route dall'host `1.1.1.1` verso l'host utilizzato.

Natting

SNAT -> postrouting DNAT -> prerouting

Per visualizzare le regole di NAT: `iptables -t nat -L -v -n` Per eliminare una regola di NAT: `iptables -t nat -D POSTROUTING <rule_number>`

Per utilizzare il natting sulla macchina che deve gestire il NAT si deve decommentare nel file `/etc/sysctl.conf` la riga `net.ipv4.ip_forward=1`. Devo poi rendere effettive le modifiche con `sysctl -p /etc/sysctl.conf`

SNAT

Per modificare il `source_address` di tutti i pacchetti in uscita sull'interfaccia `eth1` con l'indirizzo del gateway si deve aggiungere alla configurazione delle interfaces

```
post-up iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

DNAT

Per fare un port forwarding dall'interfaccia `eth1` dalla porta esterna `8080` (`external_port`) alla porta interna `80` (`internal_port`) dell'host `192.168.1.10` utilizzando `TCP`

```
post-up iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 8080 -j DNAT --to-destination 192.168.1.10:80
```

Per testare il port forwarding si può fare una chiamata dall'esterno verso un server privato tramite il comando netcat (`nc`). Ci si deve però mettere in ascolto sul server interno.

Sul server interno:

```
# Se TCP
nc -l -p <internal_port>

# Se UDP
nc -l -u -p <internal_port>
```

Sulla macchina esterna (dopo aver premuto invio si possono provare ad inviare stringhe):

```
# Se TCP
nc <gateway_address> <external_port>

Se UDP
nc -u <gateway_address> <external_port>
```

Traffic shaping

Easy: aggiungo un filtro di 1Mbit sull'interfaccia eth0:

```
tc qdisc add dev eth0 root tbf rate 1Mbit latency 50ms burst 1539
```

Complicato: creo 2 filtri, uno che permette più velocità (`1:20` , di default min 20Mbit, max 50Mbit) e uno che ne permette meno (`1:10`). Creiamo poi una regola che reindiriga i pacchetti con destinazione `192.168.1.2` verso il filtro `1:10` .

```
# Elimino la classe precedente
tc qdisc del root dev eth0

# Inseriamo la qdisk HBT nella root, specificando la default 20 e creiamo la classe associata
tc qdisc add dev eth0 root handle 1: htb default 20
tc class add dev eth0 parent 1: classid 1:1 htb rate 100Mbit burst 15k

# Aggiungiamo le 2 classi figlie 1:10 e 1:20
tc class add dev eth0 parent 1:1 classid 1:10 htb rate 1Mbit burst 15k
tc class add dev eth0 parent 1:1 classid 1:20 htb rate 20Mbit ceil 50Mbit burst 15k

# Facciamo collegamenti
tc qdisc add dev eth0 parent 1:10 handle 10: pfifo limit 50
tc qdisc add dev eth0 parent 1:20 handle 20: pfifo limit 50

# Creiamo la regola per dest 192.168.1.2
tc filter add dev eth0 protocol ip parent 1:0 prio 1 u32 match ip dst 192.168.1.2 flowid 1:10
```

Testing

Creo un file fuffa da usare per inviare traffico per fare le prove (in questo caso $1024b * 1000 = 1MB$, può anche essere `bs=1M`)

```
dd if=/dev/zero of=prova.bin bs=1024 count=1000
```

Mi metto in ascolto sulla macchina ricevente: `nc -l -p 8080 > /dev/null`

Calcolo il tempo di invio del file fuffa: `time sh -c "cat file.bin | nc 192.168.1.2 8080 -q1"`

Utils

Pulizia delle interfacce

```
ifconfig eth0 0
ifdown eth0
ifup eth0
```

Per assegnare temporaneamente l'hostname alla macchina: `hostname <nome>` Per assegnare in modo permanente l'hostname alla macchina modificare il file `/etc/hostname`

Per gestire temporaneamente le interfacce: - Attivare interfaccia: `ifup <iface>` - Disattivare interfaccia: `ifdown <iface>`