Analizzare la topologia della rete

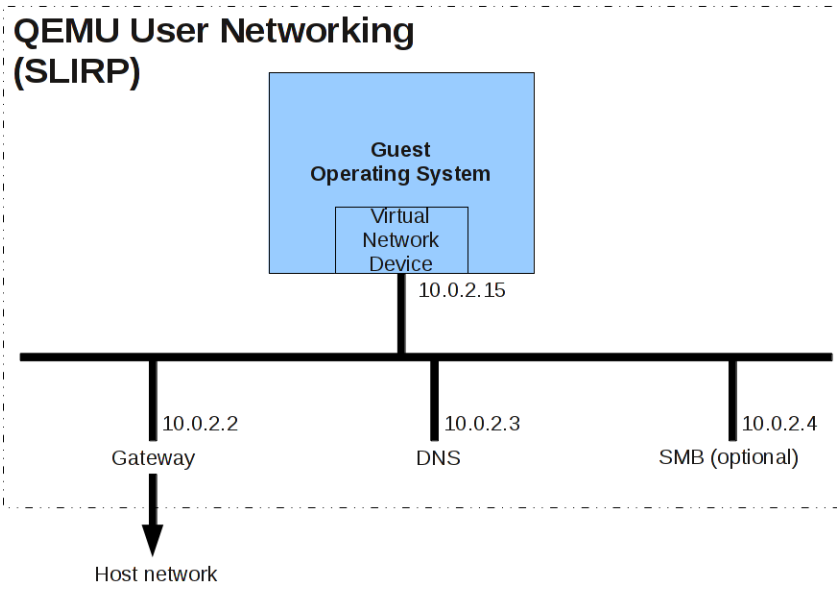

Figura 1.1: La rete virtuale creata da Qemu in modalità user networking

Qemu implementa una rete virtuale minimale con la topologia descritta nella figura 1.1. La macchina *guest* è un nodo che generalmente ha il numero IP `10.0.2.15` (assegnato tramite DHCP dal nodo `10.0.2.2`, che svolge anche le funzioni di *gateway*), anche se naturalmente l'amministratore del sistema può cambiare la configurazione.

In questa modalità detta "user networking", dalla macchina *guest* è possibile stabilire comunicazioni TCP e UDP verso l'esterno sfruttando lo stack di rete della macchina *host*. Non è invece possibile il traffico nella direzione opposta (occorre lanciare Qemu con particolari opzioni per la redirezione del traffico o usare altre modalità di networking, vedi manuale). Inoltre il traffico ICMP da e verso l'esterno viene scartato.

1.1 ping e traceroute

Poiché il protocollo ICMP non è trasmesso dalla rete, `ping` e `traceroute` non possono essere usati per tracciare nodi esterni dalla macchina virtuale: `traceroute` (che usa ICMP per ricevere i "timeout" dei pacchetti TCP spediti con TTL crescenti) risulta del tutto inutile in modalità user networking (per i nodi della rete locale non ha senso), ma è stato incluso comunque nel caso `sicureti.iso` fosse utilizzato in modo diverso: inoltre può essere interessante dare un'occhiata al manuale.

`ping` invece funziona con i nodi della rete locale.

1.2 Esercizi

- Quali nodi della rete locale virtuale rispondono a un `ping`?
- Significa necessariamente che nessuno altro nodo è attivo sulla rete?

*R*isponde solo `10.0.2.2`, ma è attivo anche `10.0.2.3` il DNS server.

1.3 Netcat

Netcat (`http://nc110.sourceforge.net/`) è un programma che permette di stabilire comunicazioni TCP e UDP senza alcun livello applicativo: fornisce sostanzialmente il servizio di creazione di un socket a linea di comando.

L'uso di base prevede che il server si metta in ascolto (−l listen) su di una porta (−p): per quelle < 1024 servono i privilegi di root. La porta non deve naturalmente essere già in uso da un altro programma.

```
1 nc −l −p 12345
```

Il client si connetterà con un parallelo

```
1 nc 10.0.2.15 12345
```

Cosí però non è molto utile: bisogna far viaggiare dei dati sulla connessione.

```
1 echo ciao | nc 10.0.2.15 12345
```

In questo caso il server vede il messaggio "ciao". Vale anche nell'altro senso.

```
1 echo ciao | nc −l −p 12345
```

Il client riceverà il messaggio al momento del collegamento.

Per trasferire un file `file.pdf`

```
1 nc 10.0.2.15 12345 < file.pdf
```

si può anche realizzare un piccolo web server che serve un unico file sulla porta `8080` (in questo caso il client potrà essere un browser):

```
1 { echo −ne "HTTP/1.0 200 OK\r\n\r\n"; cat file.html; } | nc −l −p 8080
```

Oppure si può specificare quale programma deve interpretare i messaggi in arrivo. Provate per esempio a mandare il messaggio `ls` al server:

```
1 nc −l −p 12345 −e /bin/bash
```

1.4 Esercizi

- Leggere il manuale di `nc`
- Collegarsi alla porta 22 (SSH) di `localhost` con netcat
- Collegarsi alla porta 53 UDP (DNS) di `10.0.2.3`
- Collegarsi alla porta 55 (connection refused) di `10.0.2.3`