Elementi di amministrazione di rete da shell Linux e Windows

Prof. Pizzutilo, dott.ssa Novielli,

Accordo:

I concetti trattati in queste slide sono indipendenti dal sistema operativo utilizzato;tuttavia i programmi che li realizzano, vengono presentati per sistemi GNU/Linux e sono basati sul modello TCP/IP(v4), se non indicato diversamente.

Convenzione

Usiamo questo logo per indicare che stiamo parlando di comandi della shell di Windows XP

Configurare un host per la rete

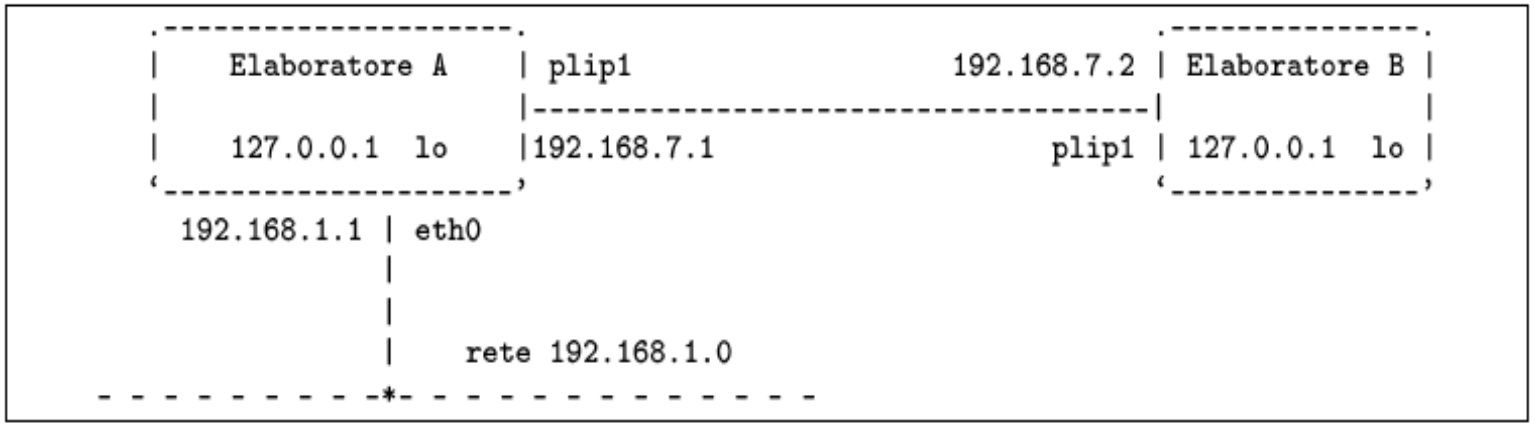
Passi

1. Assegnare un nome all'host
   - `# hostname nome`
2. Assegnare un indirizzo IP a ciascuna interfaccia di rete
3. Configurare la tabella di routing (instradamento)

Configurazione delle interfacce di rete

- La connessione in una rete basata su IP necessita inizialmente dell'assegnazione di indirizzi IP e quindi di un instradamento per determinare quale percorso devono prendere i pacchetti per raggiungere la destinazione.
- Generalmente:
  - ogni interfaccia di rete ha un proprio indirizzo IP;
  - un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se gli indirizzi di queste interfacce appartengono alla stessa rete.

Indirizzo di interfaccia e di rete



- L'indirizzo di rete si calcola applicando all'indirizzo IP di una interfaccia una maschera di rete
- IR = IP AND M
  - Es. IP = 192.168.7.1, M=255.255.255.0 ==> IR = 192.168.7.0

Loopback

- Un elaboratore deve avere una connessione virtuale a una rete immaginaria interna allo stesso elaboratore.
- A questa rete virtuale inesistente si accede per mezzo di un'interfaccia immaginaria, che in Linux è denominata '**lo**',
- l'indirizzo utilizzato è sempre lo stesso, 127.0.0.1

```
# ifconfig lo 127.0.0.1
```

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broa-dcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Interfaccia ethernet

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broa-dcast	Indirizzo punto-punto
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

ifconfig

- `ifconfig [`*`interfaccia`*`]`
- `ifconfig [`*`interfaccia ...`*
  `[`*`famiglia_indirizzamento`*`] [`*`indirizzo`*`] `*`opzioni`*`]`
- '**ifconfig**' viene utilizzato per attivare e mantenere il sistema delle interfacce di rete residente nel kernel.
- Viene utilizzato al momento dell'avvio
- Dopo, viene utilizzato di solito solo a scopo diagnostico o quando sono necessarie delle regolazioni.
- Se non vengono forniti argomenti, oppure se vengono indicate solo delle interfacce, '**ifconfig**' visualizza semplicemente lo stato delle interfacce specificate, oppure di tutte se non sono state indicate.

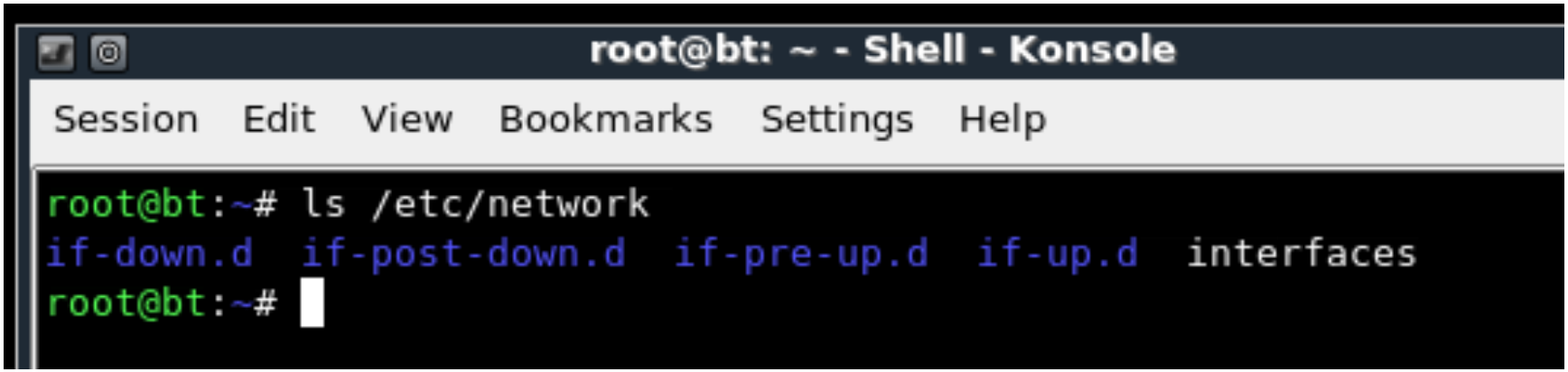
ifconfig - esempi

- # **ifconfig lo 127.0.0.1**
  - Attiva l'interfaccia '**lo**' corrispondente al *loopback* con il noto indirizzo IP 127.0.0.1.
- # **ifconfig eth0 192.168.1.1 netmask 255.255.255.0**
  - Attiva l'interfaccia '**eth0**' corrispondente alla prima scheda Ethernet, con l'indirizzo IP 192.168.1.1 e la maschera di rete 255.255.255.0.
- $ **/sbin/ifconfig eth0**
  - Emette la situazione dell'interfaccia '**eth0**' corrispondente alla prima scheda Ethernet.
- $ **/sbin/ifconfig**
  - Emette la situazione di tutte le interfacce di rete attivate.

/etc/network

Contiene le informazioni per le configurazioni delle schede di rete

Provando a fare ls…



Le configurazioni di rete sono raccolte in /etc/network/interfaces. Se non è presente alcun dispositivo ethernet, in questo file è elencata solo l'interfaccia di loopback

ipconfig

- `ipconfig [/all] [/renew [`*`Adapter`*`]] [/release [`*`Adapter`*`]] [/displaydns]`
- Mostra i valori correnti di configurazione di rete e rinfresca le impostazioni Dynamic Host Configuration Protocol(DHCP) e Domain Name System(DNS).
- Usato senza parametri, **ipconfig** mostra indirizzo IP, subnet mask e gateway di dafault per tutti gli adapter
- Questo programma può solo mostrare informazioni. Non c'è nessun tool da riga di comando per configurare le interfacce(adpter) di rete su Win ma bisogna ricorrere ai tool per interfaccia grafica.

Esempi

To display the basic TCP/IP configuration for all adapters, type:

ipconfig

To display the full TCP/IP configuration for all adapters, type:

ipconfig /all

To renew a DHCP-assigned IP address configuration for only the Local Area Connection adapter, type:

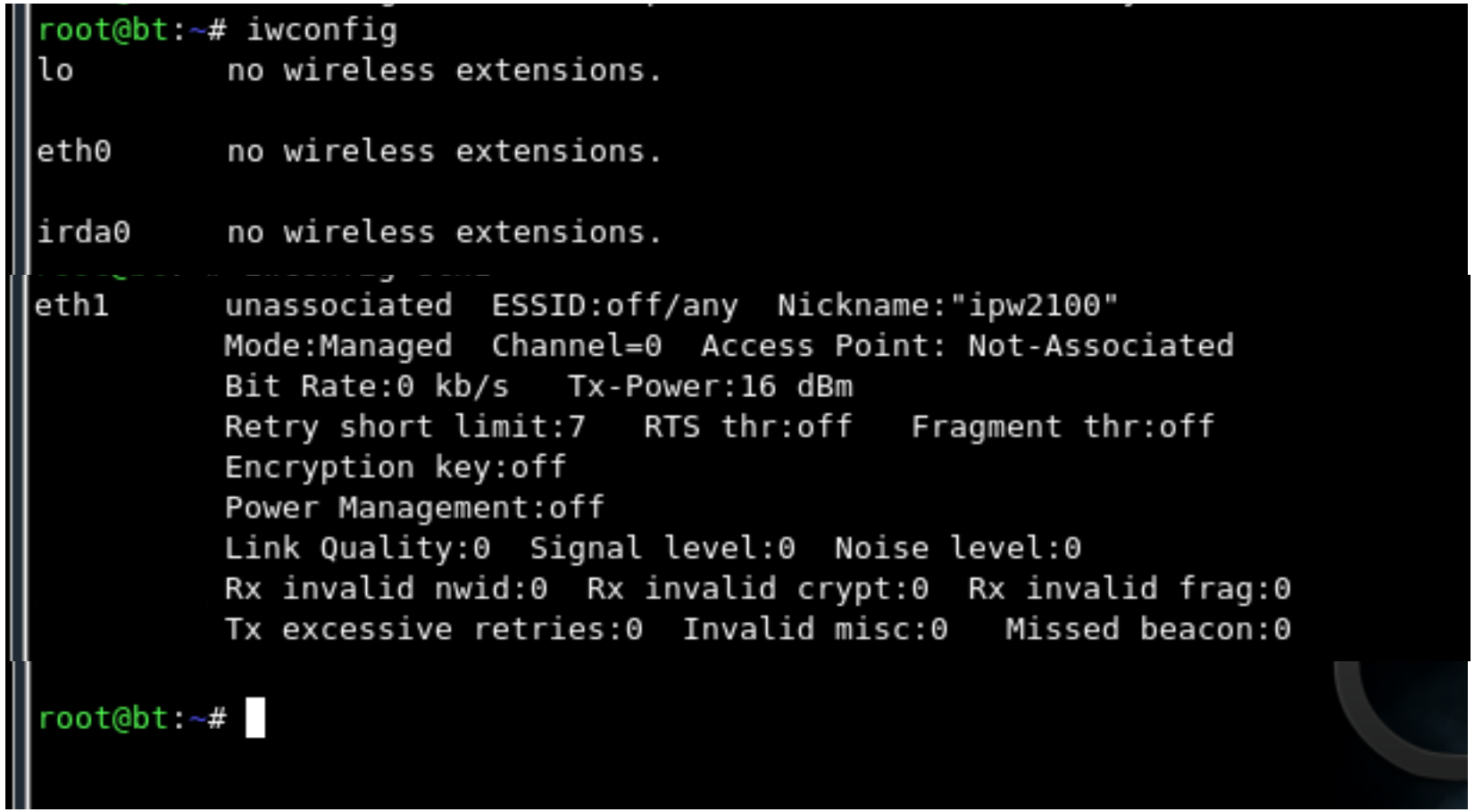
ipconfig /renew "Local Area Connection"

iwconfig

Utilizzato per configurare interfacce di reti wireless

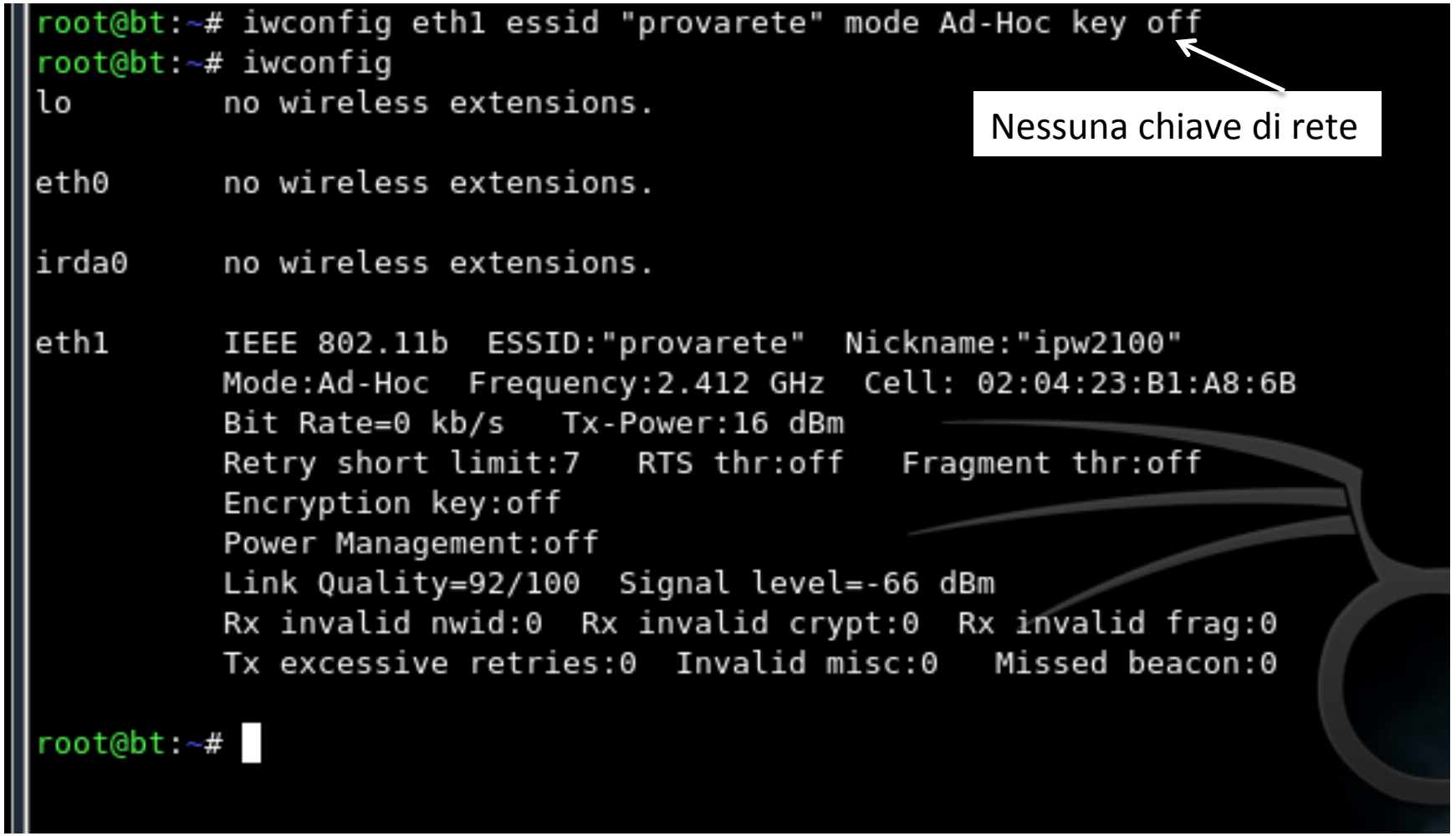
SINTASSI
iwconfig [interface]
iwconfig interface [essid X] [nwid N] [freq F] [channel C]
[sens S] [mode M] [ap A] [nick NN]
[rate R] [rts RT] [frag FT] [txpower T]
[enc E] [key K] [power P] [retry R] [commit]

Iwconfig --help
iwconfig --version



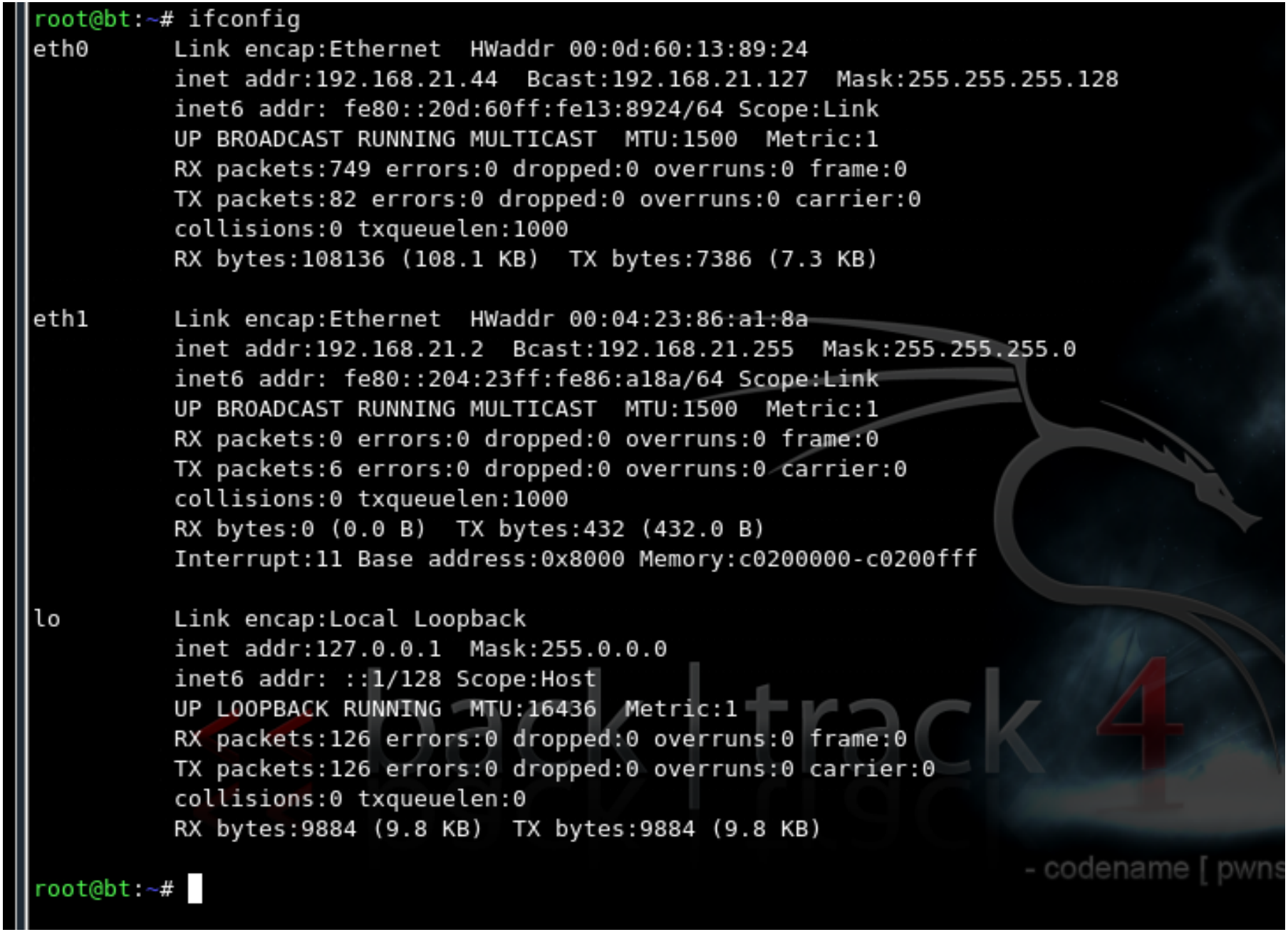
Chiamando iwconfig verifico lo stato delle interfacce wireless
In questo caso l'interfaccia (eth1) non è associata ad alcuna rete e va configurata

Creazione di una rete ad-hoc



“provarete” indica il nome della rete ad-hoc: se la rete non esiste viene creata altrimenti il risultato di iwconfig è che il mio dispositivo è connesso a “provarete”

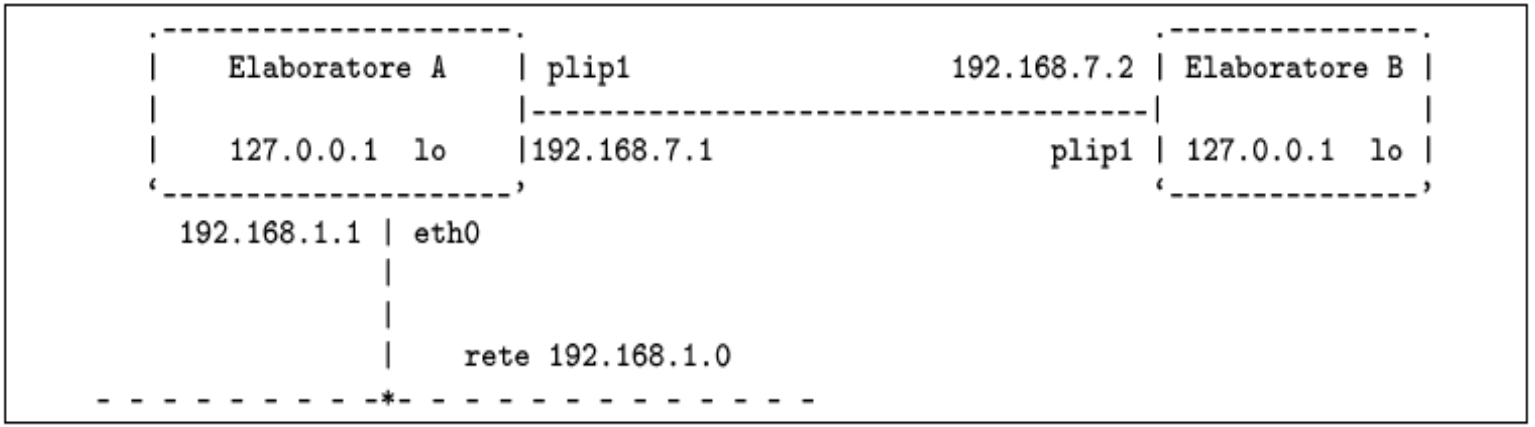
Ifconfig mi restituisce quindi lo stato di tutte le interfacce di rete



Instradamento locale

- Ogni elaboratore che utilizza la rete deve sapere quali percorsi di partenza siano possibili, in funzione degli indirizzi utilizzati.
- Gli eventuali percorsi successivi, vengono definiti da altri elaboratori nella rete.
- Si tratta di costruire una *tabella di instradamento*, attraverso la quale, ogni elaboratore sa quale strada deve prendere un pacchetto a partire da quella posizione.

Tabelle di routing



Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Intestazioni della tabella di instradamento.

Nome	Descrizione
Destination	La rete o il nodo di destinazione.
Gateway	Il router. Se appare un asterisco ('`*`') o l'indirizzo 0.0.0.0 significa che non si tratta di un instradamento attraverso un router.
Genmask	In linea di massima corrisponde alla maschera di rete; in particolare, se è un instradamento verso un nodo appare 255.255.255.255, se invece è l'instradamento predefinito appare 0.0.0.0 ('`default`').
Flags	Indica diversi tipi di informazioni utilizzando lettere o simboli.
Metric	La distanza o il costo della strada. Rappresenta la distanza (espressa solitamente in *hop* o salti) per raggiungere la destinazione.
Ref	Il numero di riferimenti all'instradamento. Questa informazione non viene utilizzata dal kernel Linux e, di conseguenza, l'informazione appare sempre azzerata.
Use	Conteggio del numero di volte in cui la voce è stata visionata.
Iface	Il nome dell'interfaccia da cui partono i pacchetti IP.

Il comando ROUTE

route è un comando che permette di vedere e modificare la tabella di routing.

Un esempio dell'output di tale comando può essere:

route -n

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.101.0	192.168.102.102	255.255.255.0	UG	0	0	0	eth0
192.168.102.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
192.168.103.0	192.168.102.102	255.255.255.0	UG	0	0	0	eth0
192.168.12.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
0.0.0.0	192.168.12.1	0.0.0.0	UG	0	0	0	eth0

*Nell'esempio indicato**, il computer dove è stato lanciato il comando***
- *si connette tramite l'interfaccia eth0 direttamente alle reti 192.168.102.0 e 192.168.12.0 (il campo gateway è indicato con 0.0.0.0) (righe 2 e 4);*
- si connette alle reti 192.168.101.0 e 192.168.103.0 tramite il gateway 192.168.102.102 (righe 1 e 3);
- per tutte le altre destinazioni (0.0.0.0) si connette mediante il gateway 192.168.12.1 (riga 5).

route

`route [-f] [-p] [`*`Command`* `[`*`Destination`*`]` `[mask` *`Netmask`*`]` `[`*`Gateway`*`]` `[metric` *`Metric`*`]]` `[if` *`Interface`*`]]`

- Mostra e modifica le istanze della tabella di routing IP locale.
- Usato senza parametri, route mostra l'help.

Command	Purpose
add	Adds a route.
change	Modifies an existing route.
delete	Deletes a route or routes.
print	Prints a route or routes.

Gestione delle tabelle di routing

- Le voci elementari sono inserite automaticamente da `ifconfig`
- Per una gestione più completa, si può usare `route`.

Esempi

 # route add -host 127.0.0.1 dev lo

- Attiva l'instradamento verso l'interfaccia locale *loopback*.

 # route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0

- Attiva l'instradamento della rete 192.168.1.0 che utilizza la maschera di rete 255.255.255.0, specificando che riguarda l'interfaccia di rete '**eth0**'.

$ route

- Mostra la tabella di instradamento attuale.

Examples

To display the entire contents of the IP routing table, type:

route print

To display the routes in the IP routing table that begin with 10., type:

route print 10.*

To add a default route with the default gateway address of 192.168.12.1, type:

route add 0.0.0.0 mask 0.0.0.0 192.168.12.1

To add a persistent route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0 and the next hop address of 10.27.0.1, type:

route -p add 10.41.0.0 mask 255.255.0.0 10.27.0.1

To add a route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0, the next hop address of 10.27.0.1, and the cost metric of 7, type:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 metric 7

To add a route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0, the next hop address of 10.27.0.1, and using the interface index 0x3, type:

route add 10.41.0.0 mask 255.255.0.0 10.27.0.1 if 0x3

To delete the route to the destination 10.41.0.0 with the subnet mask of 255.255.0.0, type:

route delete 10.41.0.0 mask 255.255.0.0

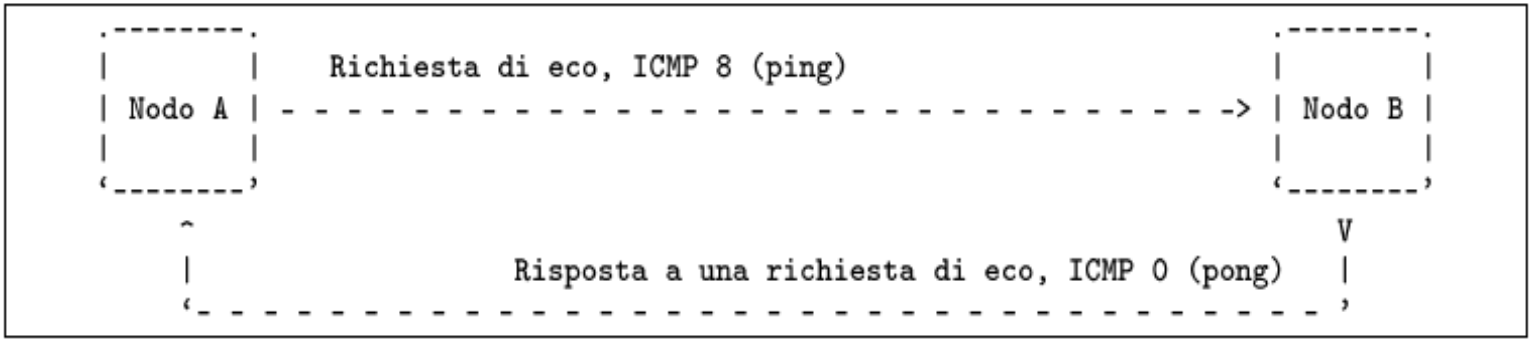
ping

`ping [`*`opzioni`*`] `*`indirizzo`*

- '**ping**' permette di inviare una richiesta di eco a un indirizzo, utilizzando il protocollo ICMP, verificando di ricevere tale eco in modo corretto ==> instradamento funzionante
- '**ping**' viene usato quasi sempre senza opzioni, in modo da ottenere una richiesta di eco continuo, a intervalli di un secondo, che può essere interrotta attraverso la tastiera con la combinazione [*Ctrl+c*].

Esempio

- $ **ping 192.168.1.1**
  - Invia una richiesta di eco all'indirizzo 192.168.1.1, a intervalli regolari di un secondo, fino a che riceve un segnale di interruzione.



ping

```
ping [-t] [-a] [-n Count] [-l Size]  [-i TTL]  [-r
Count] [{-j HostList }] [-w Timeout] [TargetName]
```

- Verifica a livello IP la connettività con un altro computer inviando messaggi ICMP (Internet Control Message Protocol) di *Echo Request.* Mostra la ricevuta dei msg di *Echo Reply* corrispondenti, insieme ai round-trip times.
- Ping è il comando primario usato per verificare connettività, raggiungibilità, e risoluzione dei nomi.
- Usato senza parametri, **ping** mostra l'help.

Examples

The following example shows ping command output:

C:\>ping example.microsoft.com

Pinging example.microsoft.com [192.168.239.132] with 32 bytes of data:

Reply from 192.168.239.132: bytes=32 time=101ms TTL=124

Reply from 192.168.239.132: bytes=32 time=100ms TTL=124

Reply from 192.168.239.132: bytes=32 time=120ms TTL=124

Reply from 192.168.239.132: bytes=32 time=120ms TTL=124

To ping the destination 10.0.99.221 and resolve 10.0.99.221 to its host name, type:

ping -a 10.0.99.221

To ping the destination 10.0.99.221 with 10 Echo Request messages, each of which has a Data field of 1000 bytes, type:

ping -n 10 -l 1000 10.0.99.221

To ping the destination 10.0.99.221 and record the route for 4 hops, type:

ping -r 4 10.0.99.221

To ping the destination 10.0.99.221 and specify the loose source route of 10.12.0.1-10.29.3.1-10.1.44.1, type:

ping -j 10.12.0.1 10.29.3.1 10.1.44.1 10.0.99.221

ARP

Gestisce corrispondenze tra indirizzi di livello rete (IP) e di livello dati (ethernet).

'**arp**' permette di ispezionare e di modificare la tabella ARP del sistema.

 # arp -a
Elenca tutte le voci accumulate nella tabella ARP.

 # arp -a 192.168.1.2
Mostra le voci riferite esclusivamente al nodo 192.168.1.2.

 # arp -n -a 192.168.1.2
Cancella le voci riferite al nodo 192.168.1.2 contenute nella tabella ARP.

 # arp -s 192.168.1.2 00:01:02:03:04:05
Assegna permanentemente (per la durata del funzionamento del sistema) l'indirizzo Ethernet 00:01:02:03:04:05 all'indirizzo IP 192.168.1.2.

arp

```
arp [-a [InetAddr] [-N IfaceAddr]] [-d InetAddr
[IfaceAddr]] [-s InetAddr EtherAddr [IfaceAddr]]
```

- Mostra e modifica le istanze presenti nella cache ARP(Address Resolution Protocl), che contiene una o più tabelle usate per salvare indirizzi IP e risp. Indirizzi fisici – Ethernet o Token Ring – risolti.
- C'è una tabella per ogni adapter Ethernet o Token Ring installato sul computer.
- Usato senza parametri, **arp** mostra l'help.

Examples

To display the ARP cache tables for all interfaces, type:

arp -a

To display the ARP cache table for the interface that is assigned the IP address 10.0.0.99, type:

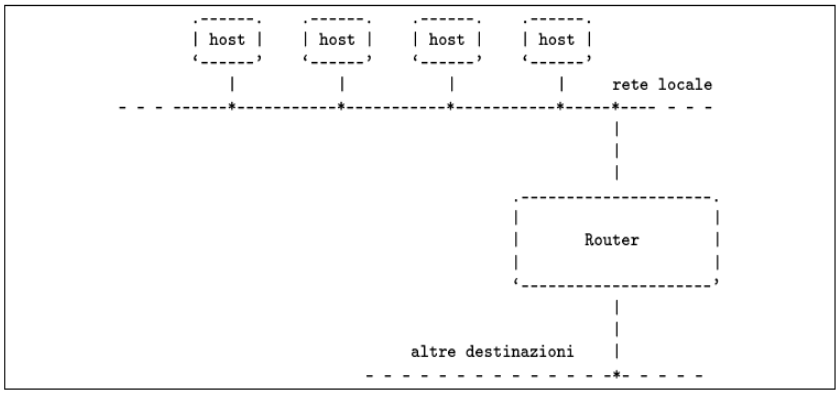
arp -a -N 10.0.0.99

To add a static ARP cache entry that resolves the IP address 10.0.0.80 to the physical address 00-AA-00-4F-2A-9C, type:

arp -s 10.0.0.80 00-AA-00-4F-2A-9C

Instradamento verso altre reti

- Quando si ha la necessità di raggiungere una destinazione che non si trova a essere connessa con la rete fisica a cui si accede, c'è bisogno di un intermediario, ovvero un elaboratore connesso alla stessa rete fisica a cui accede l'elaboratore locale, che sia in grado di inoltrare i pacchetti alle destinazioni richieste.
- Questo elaboratore è il **router** o **gateway**



accedere ad altre reti

Una rete locale potrebbe essere articolata in sottoreti in modo da evitare di sovraffollare di traffico un'unica rete. Per fare in modo che le sottoreti possano comunicare tra loro si devono utilizzare i router, che funzionano come ponti tra una sottorete e un'altra.

Con l'istruzione seguente, il router 192.168.1.254 viene utilizzato per accedere alla rete 192.168.7.0 da un host della rete 192.168.7.0 :

```
# route add -net 192.168.7.0 netmask 255.255.255.0 ↩
↪gw 192.168.1.254 dev eth0
```

Routing predefinito

Per indicare un instradamento che permette di raggiungere tutte le destinazioni non diversamente specificate, si utilizza l'indirizzo IP **0.0.0.0**, corrispondente al nome simbolico '**default**'. L'approccio più comune consiste nel definire l'instradamento 'default' come passante per un router:

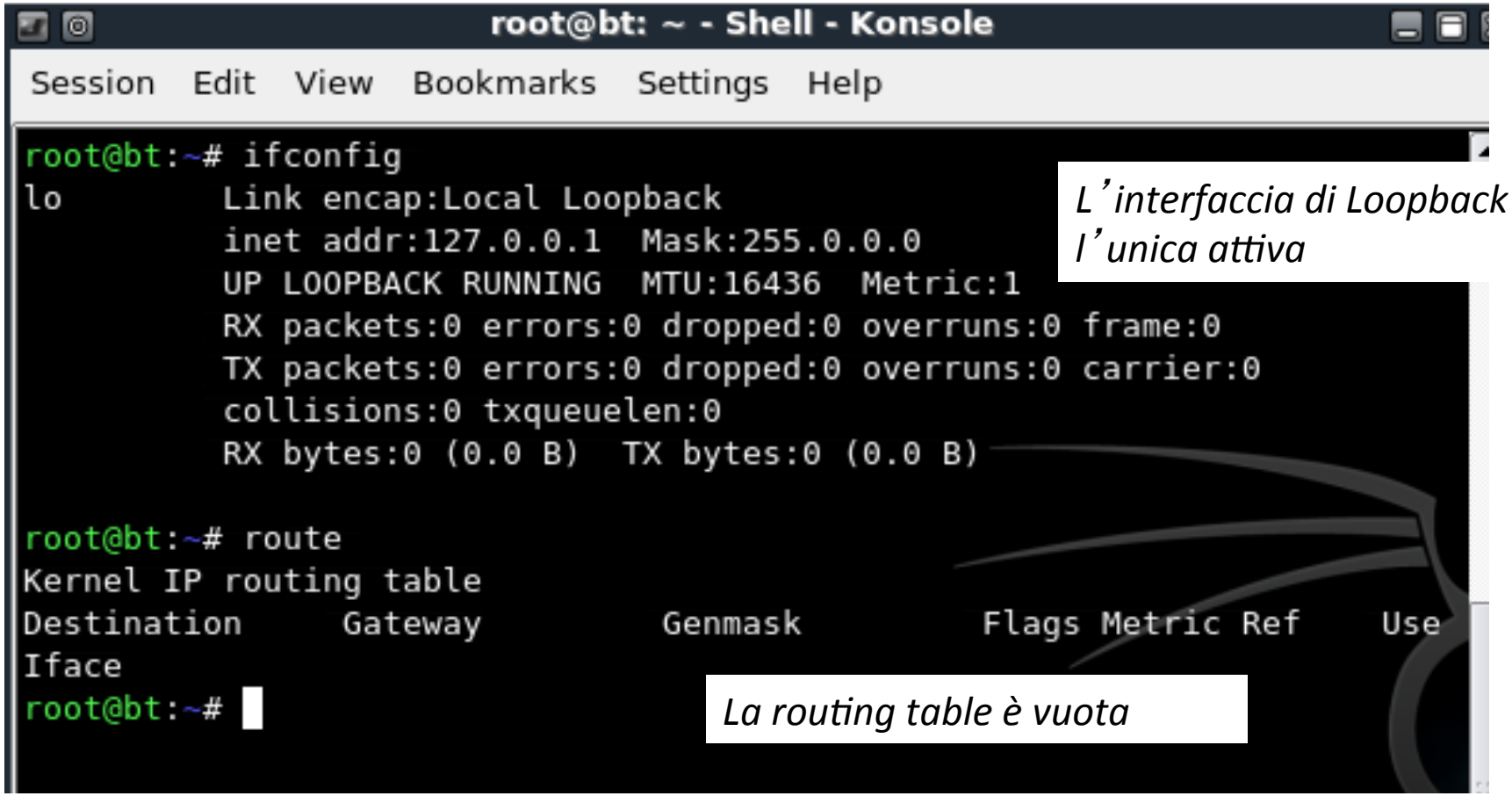
```
# route add -net default gw 192.168.1.254 dev eth0
```

Supponendo già definito l'instradamento verso la rete locale 192.168.1.0, in modo da poter raggiungere il router, si può ottenere il risultato seguente:

```
$ route -n[ Invio ]

Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        1 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth0
```

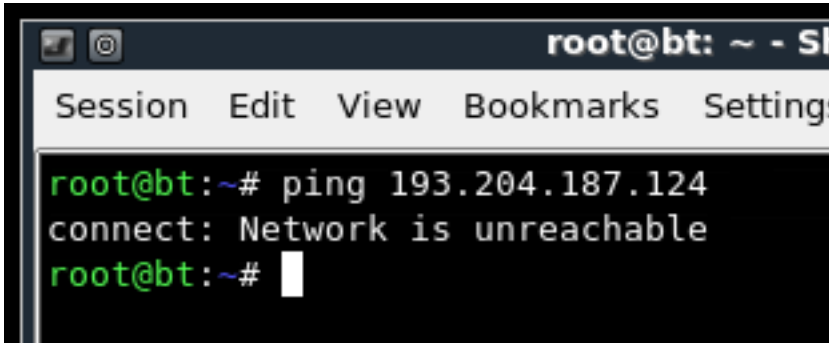
Un esempio



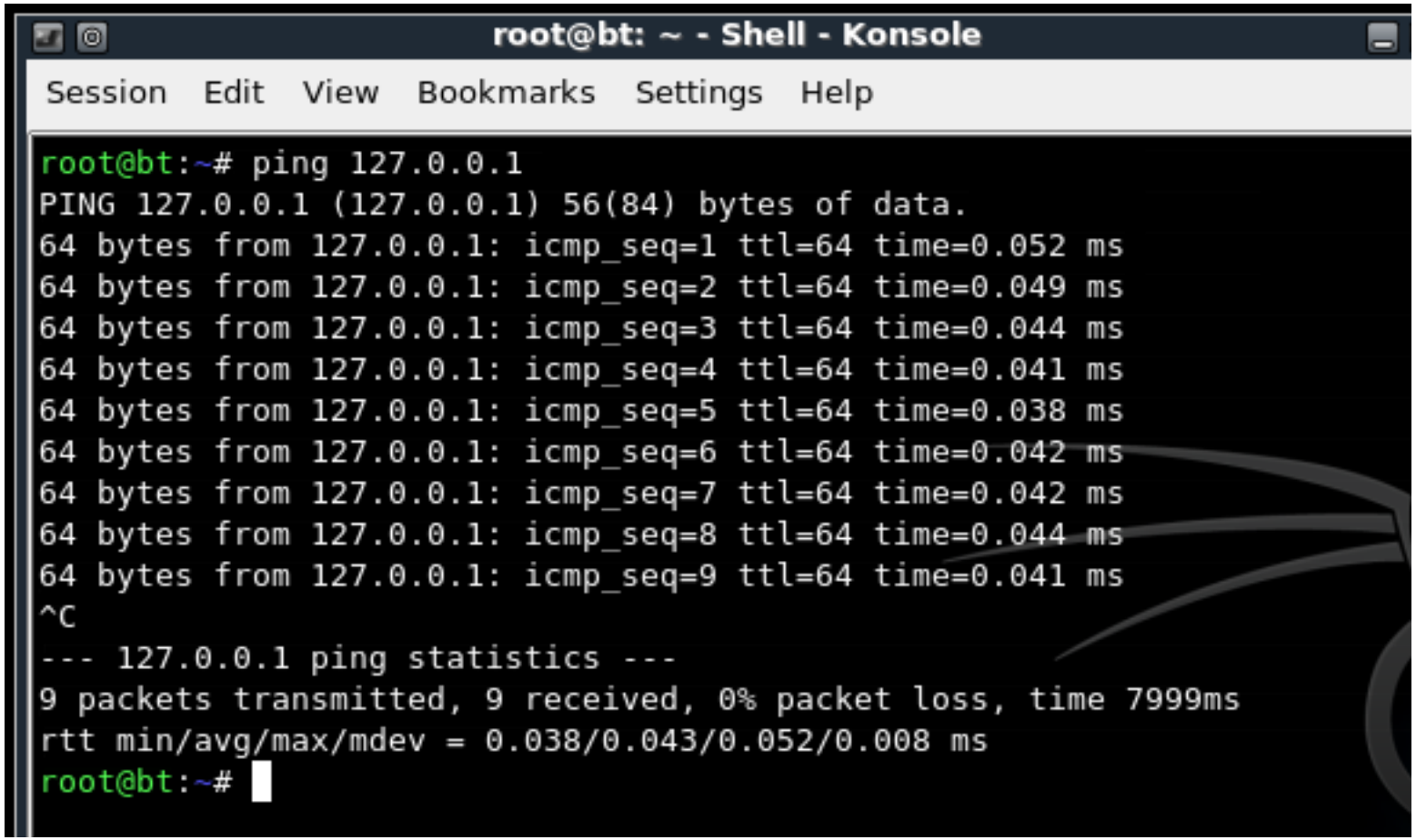
L'interfaccia di Loopback è l'unica attiva

La routing table è vuota

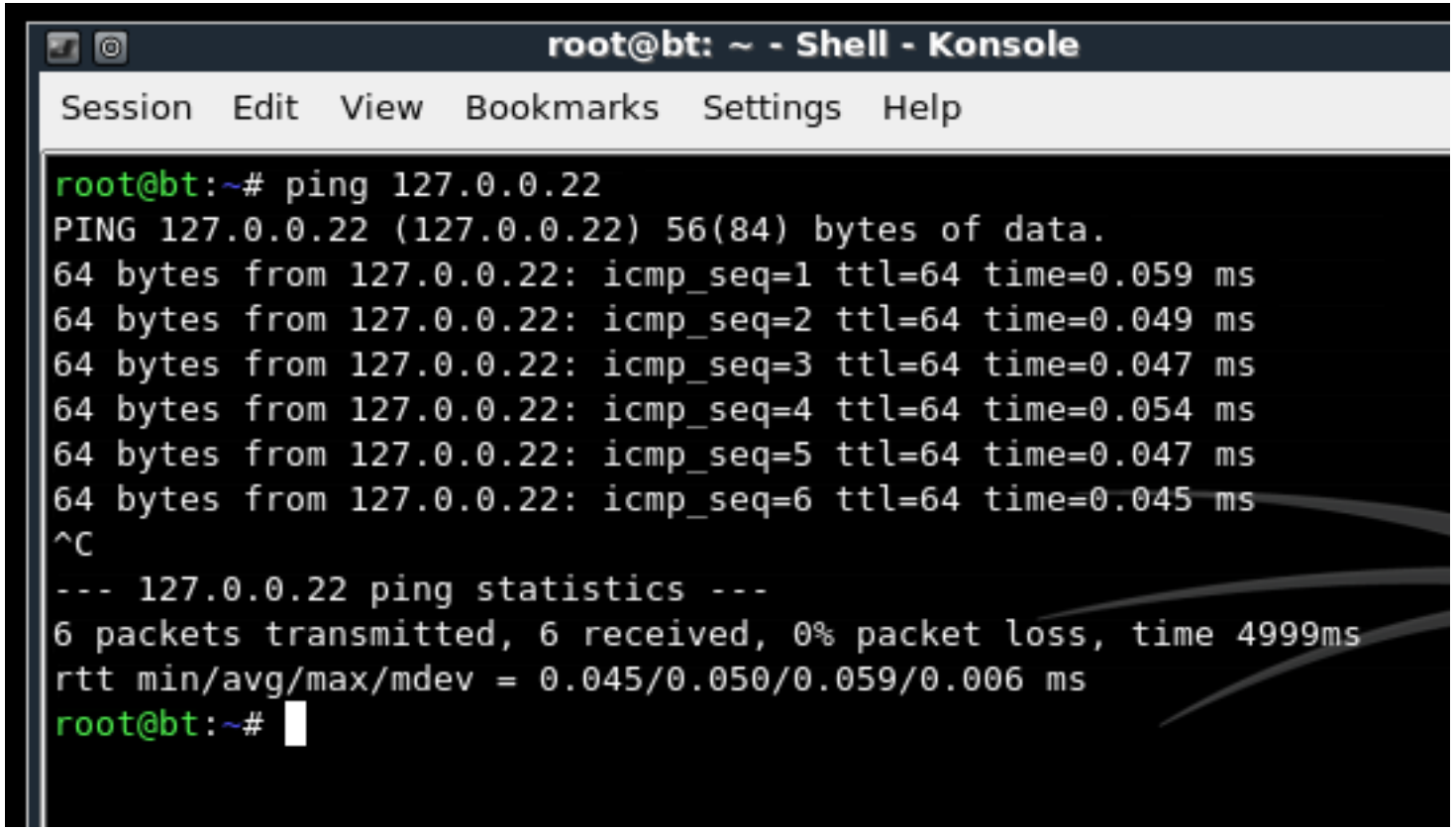
Infatti…



Ovviamente riesco a fare ping 127.0.0.1



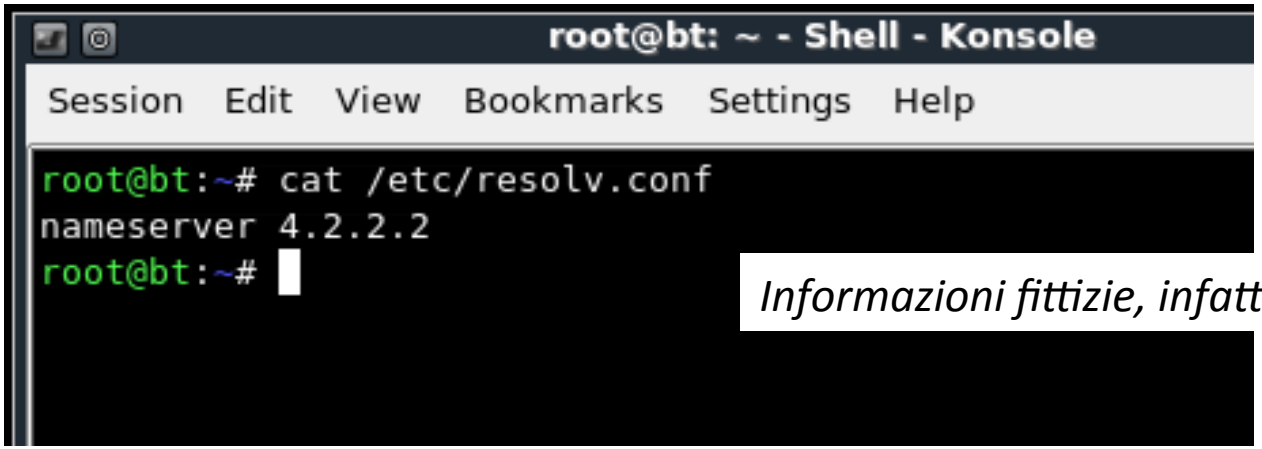
E ovviamente riesco a fare *ping* verso tutti gli indirizzi della ‘famiglia’ identificata come loopback

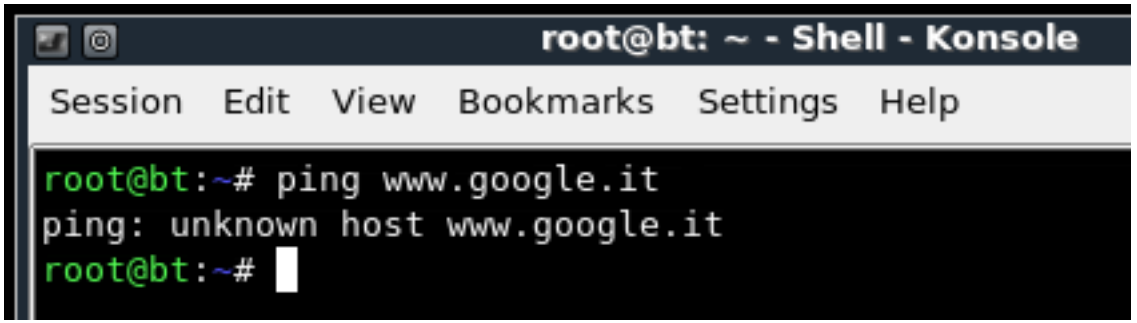


/etc/resolv.conf

In Linux il file ***resolv.conf*** contiene le informazioni principali relative alla risoluzione dei nomi, come il dominio locale e l'IP del nameserver.

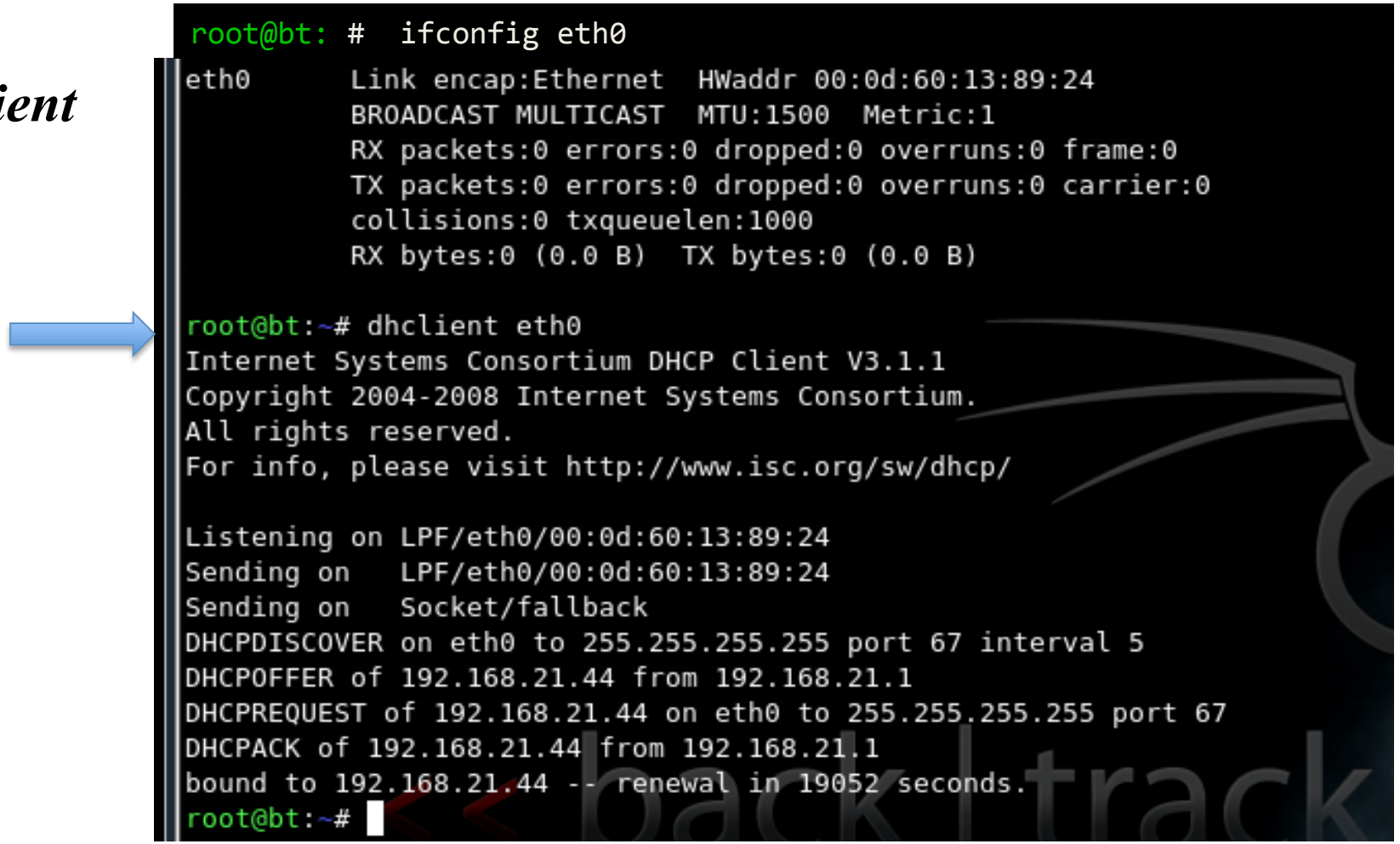
Il file controlla il comportamento delle funzioni del *resolver* il quale esegue la risoluzione dei nomi (vedremo in seguito in maniera più approfondita)





Nota: l'errore questa volta è diverso, eth0 è configurata ma il problema questa volta è la risoluzione del nome (DNS server).

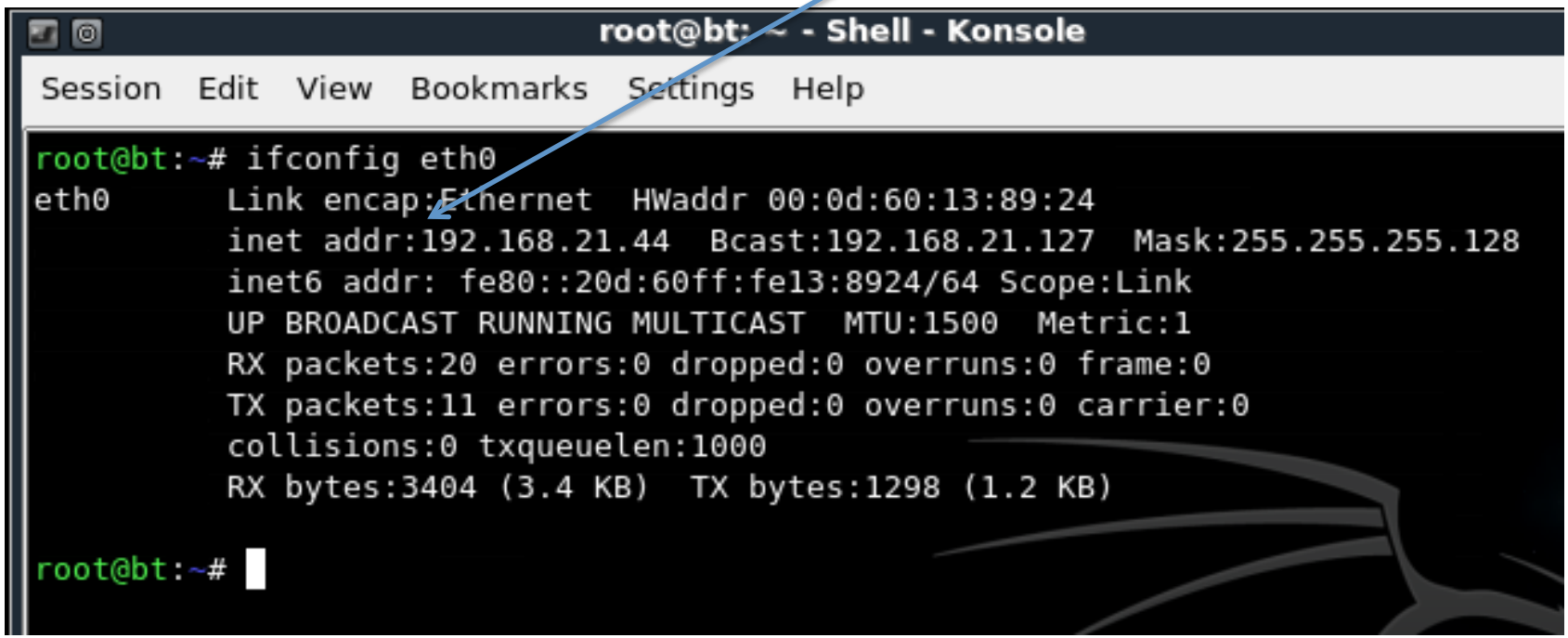
dhclient



Dhclient *è un programma che permette di ottenere automaticamente tutte le informazioni necessarie per la connessione alla rete sfruttando il protocollo* ***DHCP****.*
Manda una richiesta per ottenere in prestito un indirizzo di rete a tutti i computer della rete locale. Questa richiesta viene riconosciuta da un router che possiede un agente di inoltro DHCP, e viene dirottata al server DHCP, che risponde prestando un indirizzo di rete valido.

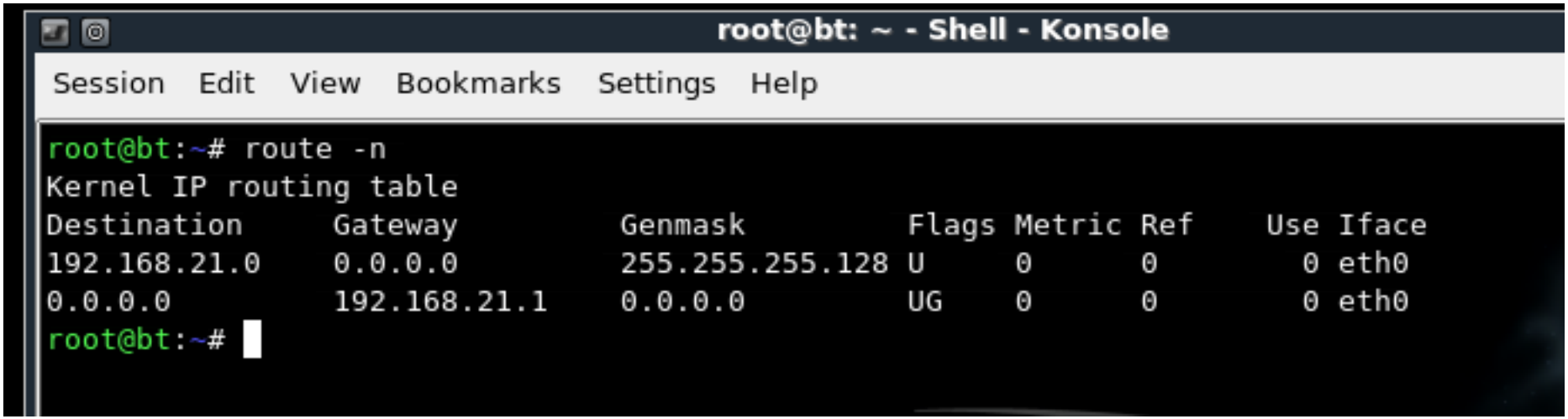
dhclient *(continua)*

Ottengo quindi un indirizzo di rete…



dhclient *(continua)*

Provo a consultare la tabella di routing: è stata aggiornata a seguito dell'esecuzione della richiesta a ***dhcp***



Il risultato è lo stesso che avrei ottenuto facendo :

```
route add default gw 192.168.21.1
```

(dhcp aggiunge automaticamente la regola nell'ultima riga)

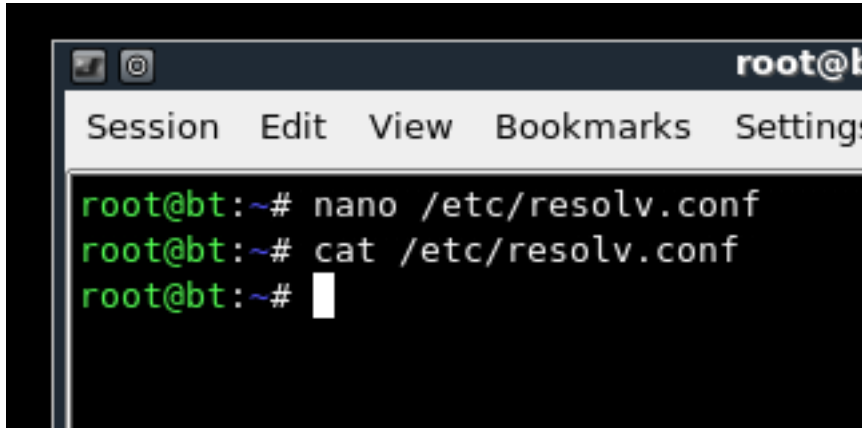
Proviamo a vedere cosa succede se cancello i riferimenti dal file ***/etc/resolv.conf***

Per modificare il file ho semplicemente invocato un editor
Nell'esempio riportato nella figura: nano /etc/resolv.conf

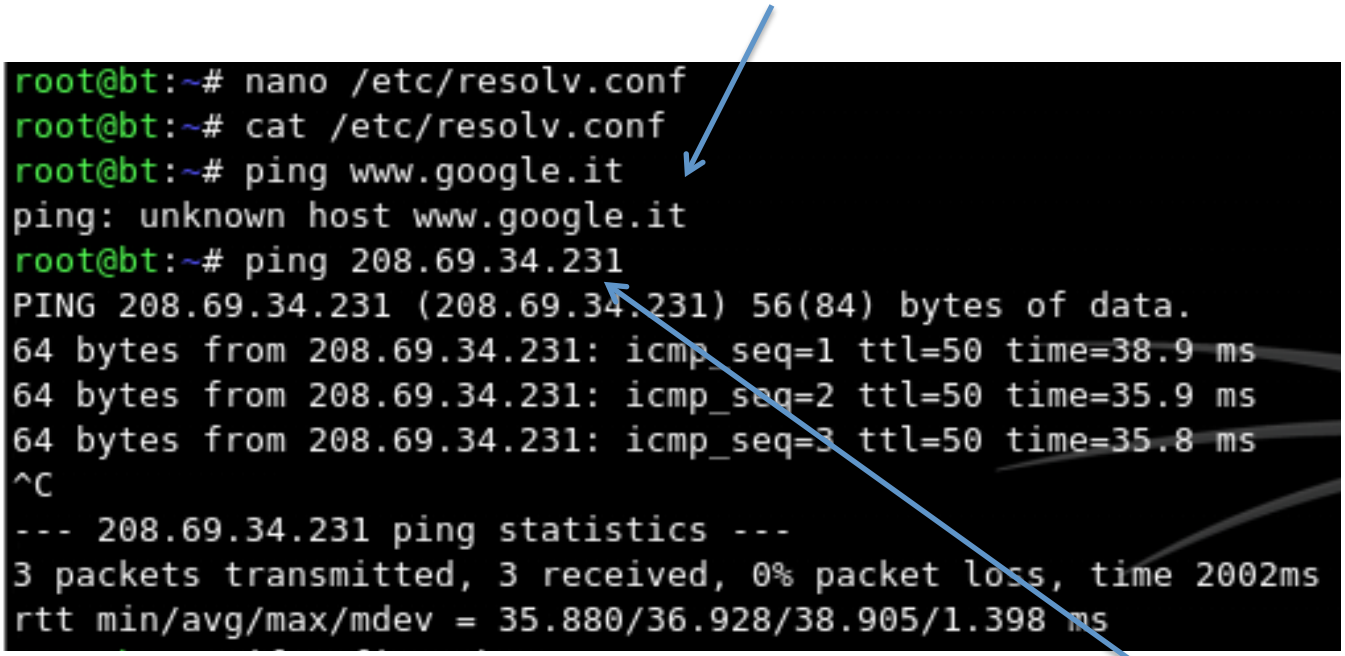
-Cancello entrambe le righe dal file
-Salvo ed esco

Cosa accadrà se provo ad eseguire il comando 'ping www.google.it'? Perché?

A questo punto il file è vuoto…



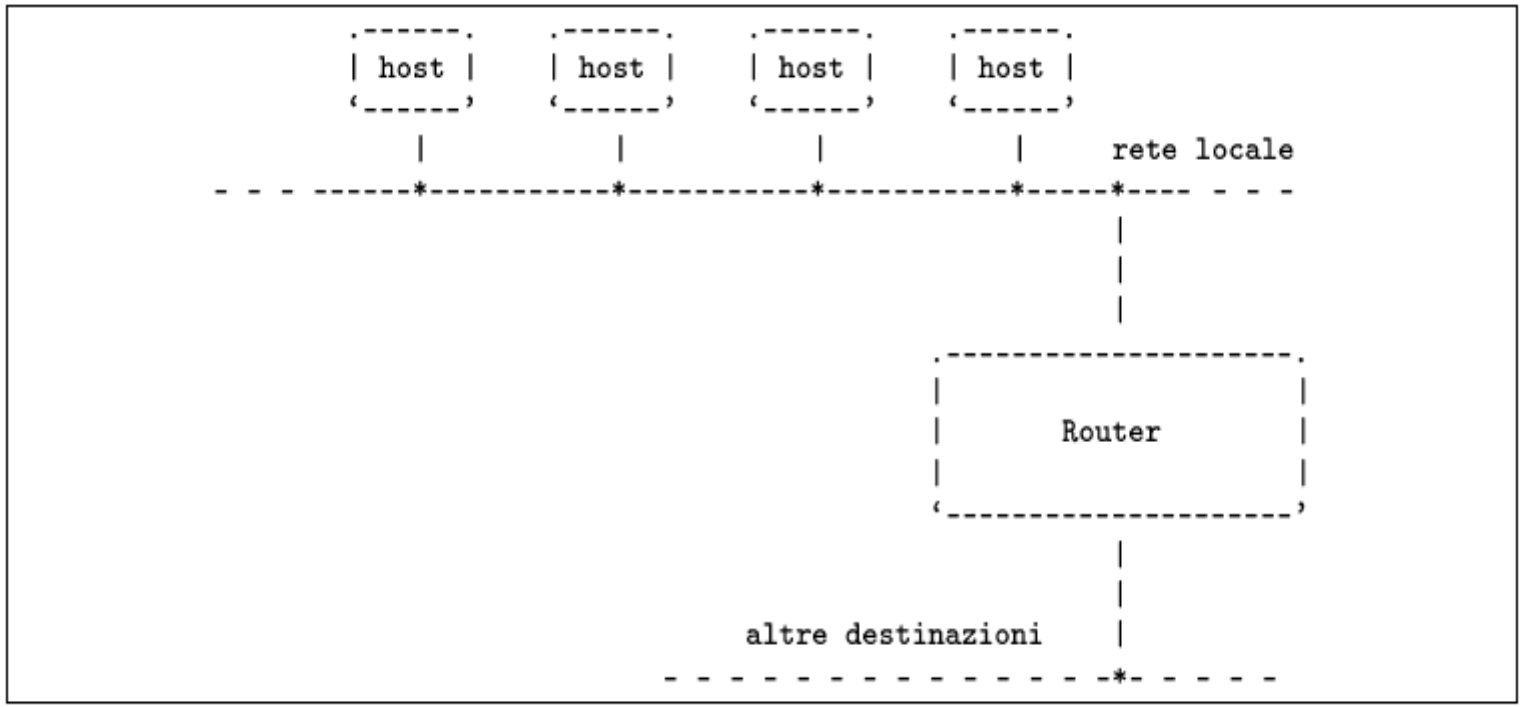
….e non riesco ad uscire dalla mia sottorete



…a meno che non faccia riferimento esplicitamente all'ip dell'host

Instradamento verso altre reti

- Quando si ha la necessità di raggiungere una destinazione che non si trova a essere connessa con la rete fisica a cui si accede, c'è bisogno di un intermediario, ovvero un elaboratore connesso alla stessa rete fisica a cui accede l'elaboratore locale, che sia in grado di inoltrare i pacchetti alle destinazioni richieste.
- Questo elaboratore è il **router** o **gateway**



accedere ad altre reti

Una rete locale potrebbe essere articolata in sottoreti in modo da evitare di sovraffollare di traffico un'unica rete. Per fare in modo che le sottoreti possano comunicare tra loro si devono utilizzare i router, che funzionano come ponti tra una sottorete e un'altra.

Con l'istruzione seguente, il router 192.168.1.254 viene utilizzato per accedere alla rete 192.168.7.0 da un host della rete 192.168.1.0

```
# route add -net 192.168.7.0 netmask 255.255.255.0 ↩
↪gw 192.168.1.254 dev eth0
```

Routing predefinito

Per indicare un instradamento che permette di raggiungere tutte le destinazioni non diversamente specificate, si utilizza l'indirizzo IP **0.0.0.0**, corrispondente al nome simbolico '**default**'. L'approccio più comune consiste nel definire l'instradamento 'default' come passante per un router:

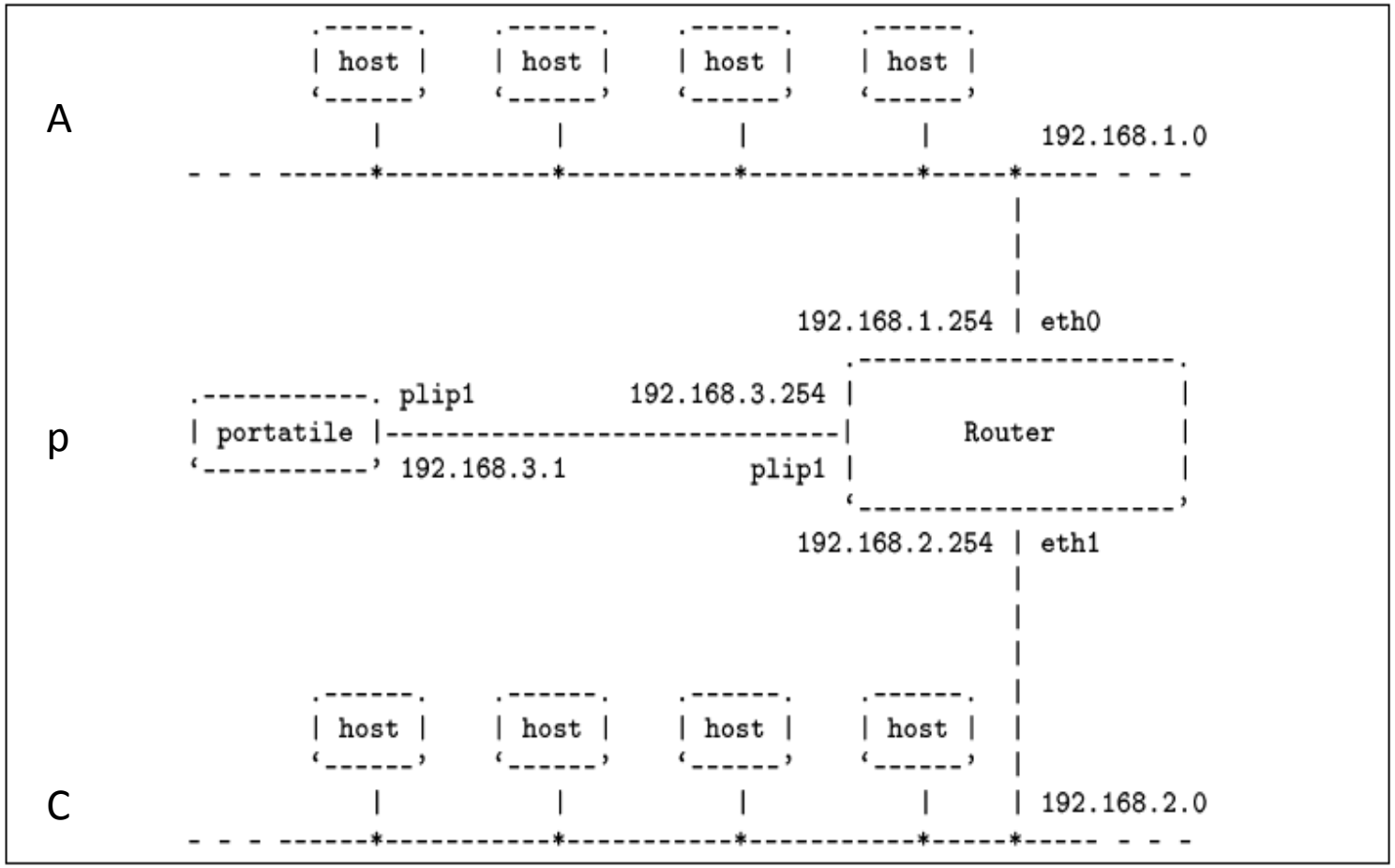
```
# route add -net default gw 192.168.1.254 dev eth0
```

Supponendo già definito l'instradamento verso la rete locale 192.168.1.0, in modo da poter raggiungere il router, si può ottenere il risultato seguente:

```
$ route -n[ Invio ]

Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        1 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth0
```

Router unico per più reti



Configurazione del router

interfacce

```
# ifconfig eth0 192.168.1.254 netmask 255.255.255.0
# ifconfig eth1 192.168.2.254 netmask 255.255.255.0
# ifconfig plip1 192.168.3.254 pointopoint 192.168.3.1
```

instradamenti

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0°
# route add -net 192.168.2.0 netmask 255.255.255.0 dev eth1°
# route add -host 192.168.3.1 dev plip1
# route add -host 192.168.3.254 dev plip1
```

°*Questo instradamento sarà stato automaticamente definito da Ifconfig*

Configurazione degli host in A e p

Rete A:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0°
# route add -net default gw 192.168.1.254
```

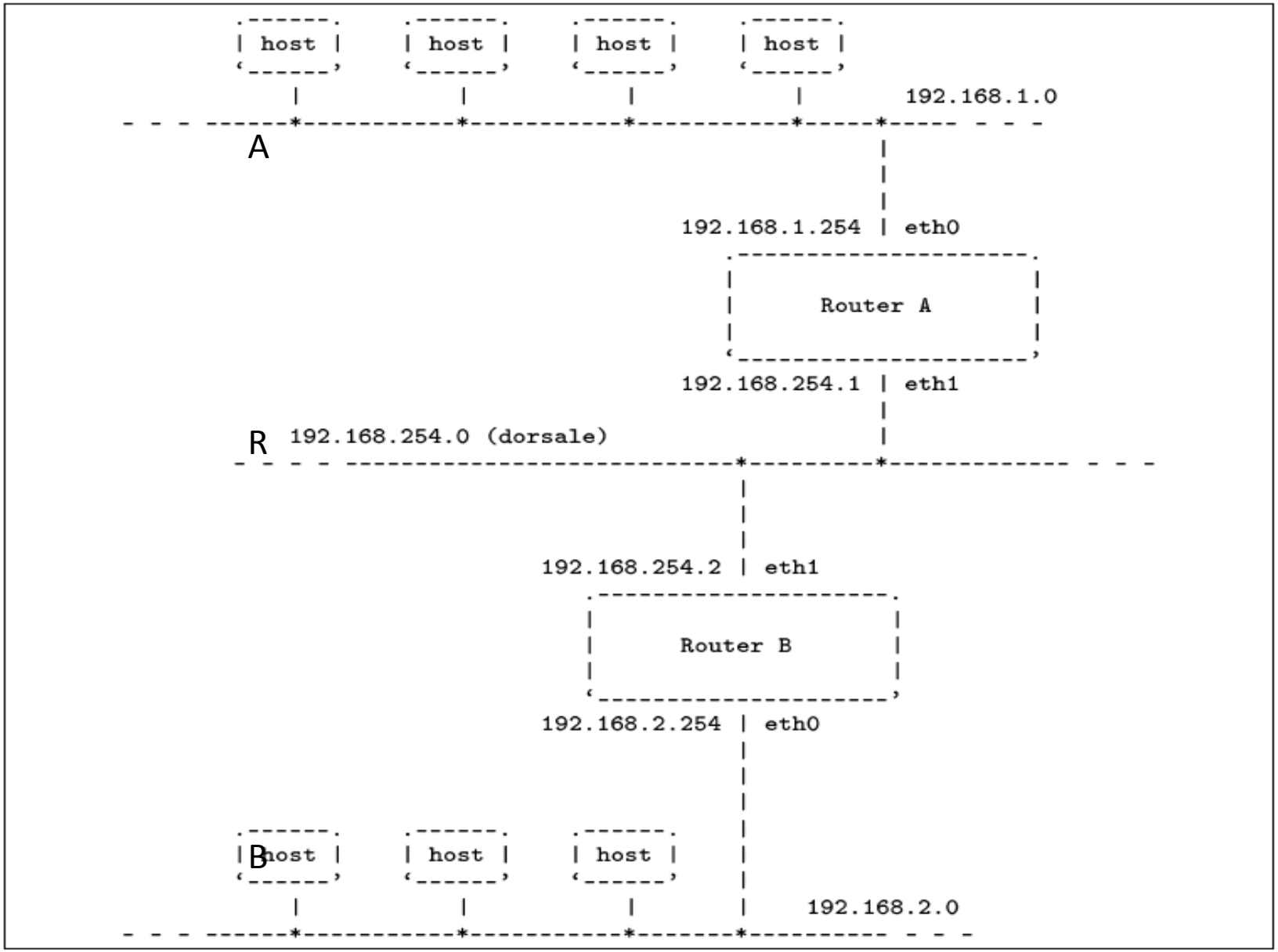
Rete p

(l'elaboratore portatile è connesso con un cavo laplink al router):

```
# route add -host 192.168.3.254 dev plip1
# route add -host 192.168.3.1 dev plip1
# route add -net default gw 192.168.3.254
```

'default' nella regola indica che qualsiasi altra richiesta che non soddisfi le regole precedenti sarà instradata attraverso il gateway con ip 192.168.3.254

Router verso router



Configurazione del router A

Il router A deve poter raggiungere tutte e tre le reti: sulla rete A e R è connesso direttamente, mentre per la rete B deve fare affidamento sul router B:

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0°
# route add -net 192.168.254.0 netmask 255.255.255.0 dev eth1°
# route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.254.2
```

Nota: quando il kernel Linux dispone della funzionalità di forward/gateway, questa può essere controllata attraverso un file del file system virtuale '/proc/'.

Per motivi di sicurezza, alcune distribuzioni GNU/Linux sono predisposte in modo da disattivare questa funzionalità, attraverso uno dei comandi inseriti nella procedura di inizializzazione del sistema.

Per riattivare il forwarding/gatewaying, si può agire nel modo seguente:

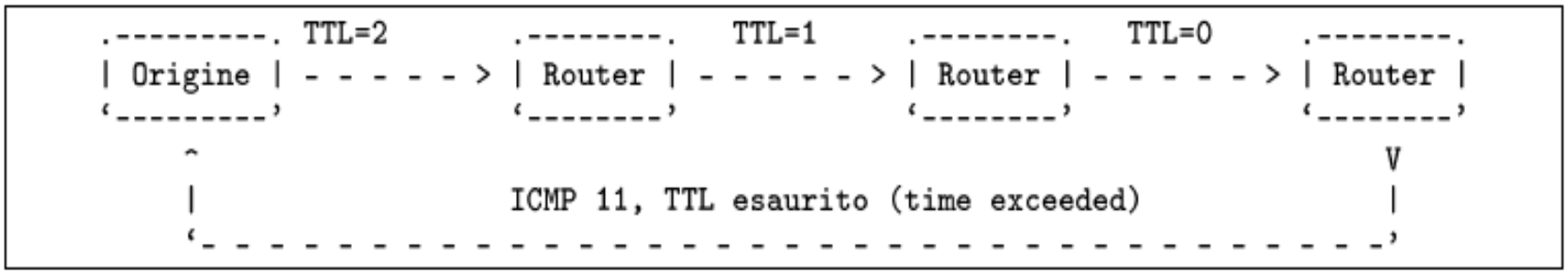
```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Verifica dell'instradamento tra router: ping

Lo strumento fondamentale per la verifica degli instradamenti è sempre 'ping'.

In presenza di router si introduce un concetto nuovo, quello del nodo da attraversare; l'attraversamento di un nodo viene definito comunemente salto(hop). 255 è il max di questi salti: oltre questo numero, detto TTL, i pacchetti vengono scartati.

In effetti, i pacchetti IP contengono l'indicazione del valore TTL massimo, che viene decrementato all'attraversamento di ogni router a opera dello stesso. Quando si raggiunge lo zero il router invia al computer di origine un messaggio ICMP di errore.



..e traceroute

Per verificare quale sia il percorso utilizzato dai pacchetti per raggiungere una destinazione, si utilizza il comando 'traceroute:

`traceroute [opzioni] destinazione [lunghezza ]`

Traceroute inizia la trasmissione di pacchetti (utilizzando il protocollo UDP) con un TTL basso. In tal modo si aspetta di ricevere un messaggio di errore, attraverso il protocollo ICMP, dal nodo in cui il valore TTL raggiunge lo zero. Incrementando di 1 e fino a 30 il TTL, traceroute riesce a conoscere gli indirizzi dei nodi attraversati. Nell'ultimo nodo, traceroute genera un errore differente, per ottenere un messaggio ICMP differente.

```
# traceroute portatile.plip.dg
traceroute to portatile.plip.dg (192.168.254.1), 30 hops max, 40 byte packets
 1 dinkel.brot.dg (192.168.1.1) 0.433 ms 0.278 ms 0.216 ms
 2 router.brot.dg (192.168.1.254) 2.335 ms 2.278 ms 3.216 ms
 3 * * *
 4 * * *
 5 portatile.plip.dg (192.168.254.1) 10.654 ms 13.543 ms 11.344 ms
```

tracert

- `tracert [-d] [-h MaximumHops] [TargetName]`
- Determina il percorso per una destinazione inviando messaggi ICMP di *Echo Request* verso la destinazione ed incrementando di volta in volta il TTL. Il percorso mostrato è la lista di interfacce dei router più vicini tra loro nel percorso tra host sorgente e destinazione; un asterisco viene stampato in caso di router che non restituiscono messaggi di *time Exceeded* per i pacchetti con TTL scaduto. Usato senza parametri, **tracert** mostra l'help.

Un esempio

traceroute www.google.com (72.14.234.104), 64 hops max, 52 byte packets

1 192.168.21.1 (192.168.21.1) 1.193 ms 0.926 ms 0.791 ms
2 campusone-gw.di.uniba.it (193.204.184.33) 2.034 ms 2.849 ms 3.926 ms
3 193.204.180.17 (193.204.180.17) 1.892 ms 1.212 ms 1.936 ms
4 ru-uniba-rt-ba1.ba1.garr.net (193.206.137.89) 2.088 ms 2.297 ms 1.746 ms
5 rt-ba1-rt1-bo1.bo1.garr.net (193.206.134.77) 10.186 ms 9.707 ms 12.725 ms
6 rt1-bo1-rt1-mi1-l1.mi1.garr.net (193.206.134.193) 13.126 ms 13.610 ms 13.644 ms
7 rt1-mi1-rt-mi2.mi2.garr.net (193.206.134.190) 13.842 ms 13.659 ms 12.918 ms
8 193.206.129.134 (193.206.129.134) 14.026 ms 13.602 ms 13.188 ms
9 209.85.249.54 (209.85.249.54) 28.985 ms 13.597 ms 14.296 ms
10 72.14.232.63 (72.14.232.63) 13.818 ms 13.571 ms 13.963 ms
11 mil01s07-in-f104.1e100.net (72.14.234.104) 13.443 ms 13.914 ms 13.840 ms

L'ip della macchina da cui ho fatto traceroute è 192.168.21.41

Un esempio

traceroute www.google.com (72.14.234.104), 64 hops max, 52 byte packets

1192.168.21.1 (192.168.21.1) 1.193 ms 0.926 ms 0.791 ms

Sono ancora nella mia sottorete, l'ip della mia macchina è 192.168.21.41

2campusone-gw.di.uniba.it (193.204.184.33) 2.034 ms 2.849 ms 3.926 ms
3193.204.180.17 (193.204.180.17) 1.892 ms 1.212 ms 1.936 ms

NB: attenzione agli ip adesso, sono uscito dalla sottorete cui appartiene l'host

Un esempio

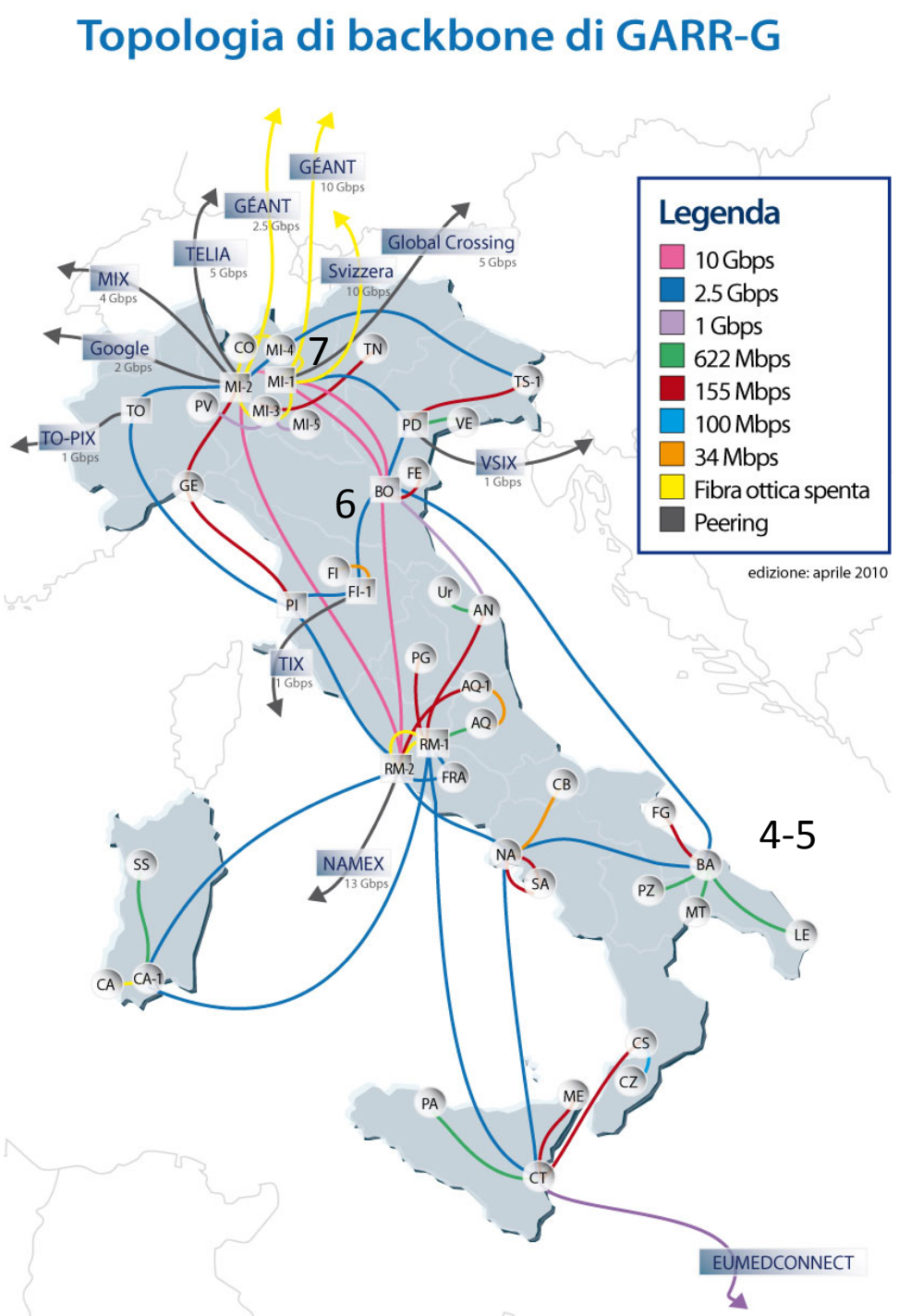
traceroute www.google.com (72.14.234.104)

4ru-uniba-rt-ba1.ba1.garr.net (193.206.137.89) 2.088 ms 2.297 ms 1.746 ms

5rt-ba1-rt1-bo1.bo1.garr.net (193.206.134.77) 10.186 ms 9.707 ms 12.725 ms

6rt1-bo1-rt1-mi1-l1.mi1.garr.net (193.206.134.193) 13.126 ms 13.610 ms 13.644 ms

7rt1-mi1-rt-mi2.mi2.garr.net (193.206.134.190) 13.842 ms 13.659 ms 12.918 ms

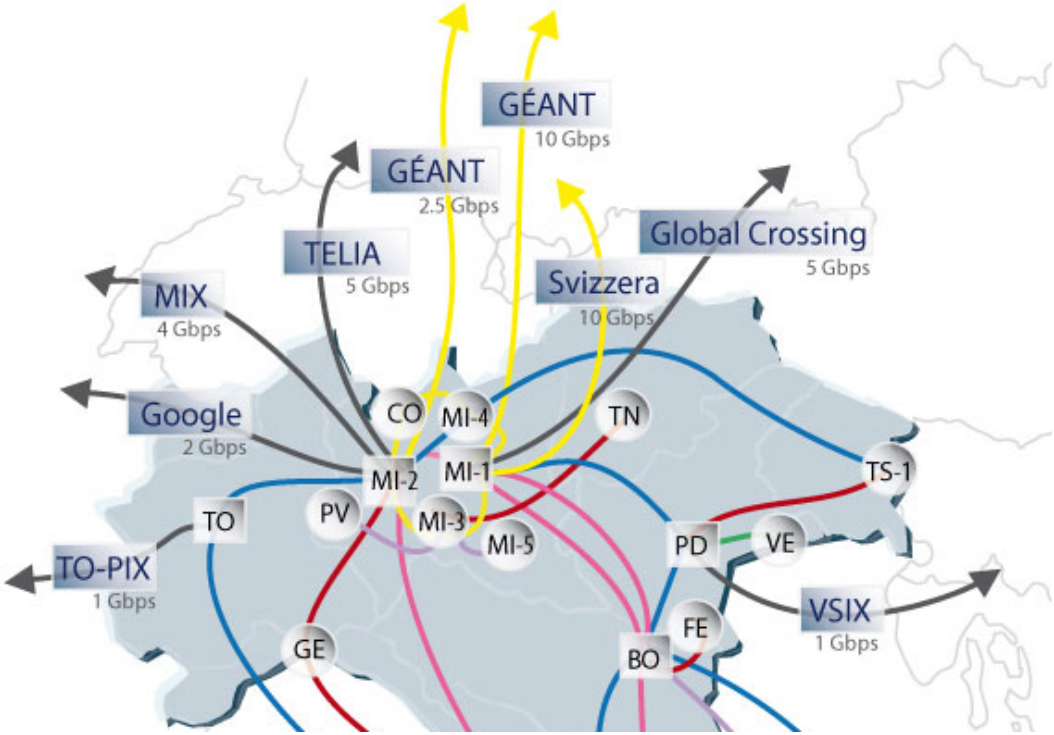


Un esempio

traceroute www.google.com
(72.14.234.104)

8193.206.129.134 (193.206.129.134)
14.026 ms 13.602 ms 13.188 ms
9209.85.249.54 (209.85.249.54)
28.985 ms 13.597 ms 14.296 ms

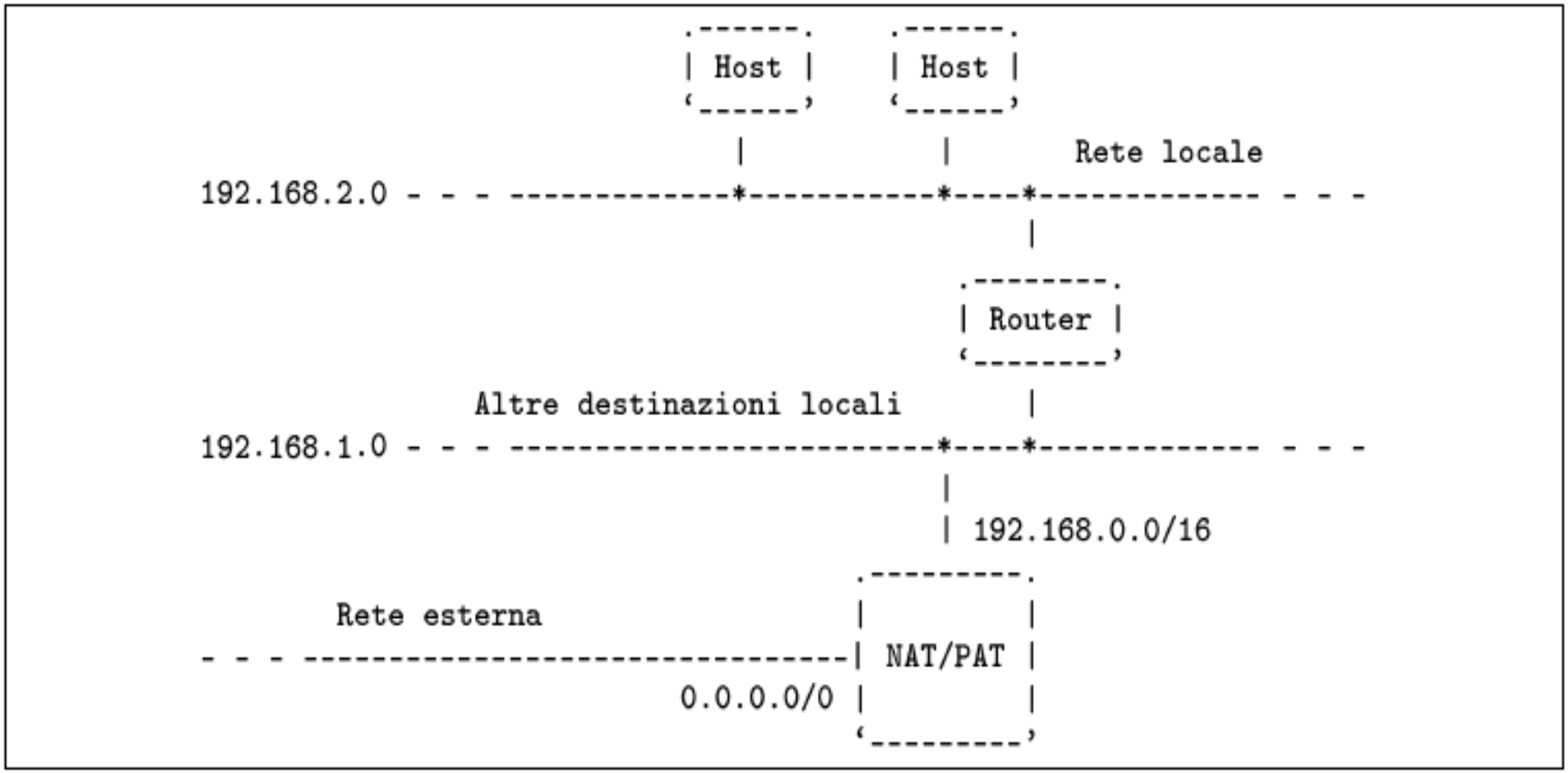
La richiesta viene quindi inoltrata a mi-2 che ha la connessione con Google



10 72.14.232.63 (72.14.232.63) 13.818 ms 13.571 ms 13.963 ms
11 mil01s07-in-f104.1e100.net (72.14.234.104) 13.443 ms 13.914 ms 13.840 ms

Quindi il pacchetto giunge a destinazione

Inoltro verso l'inter-rete



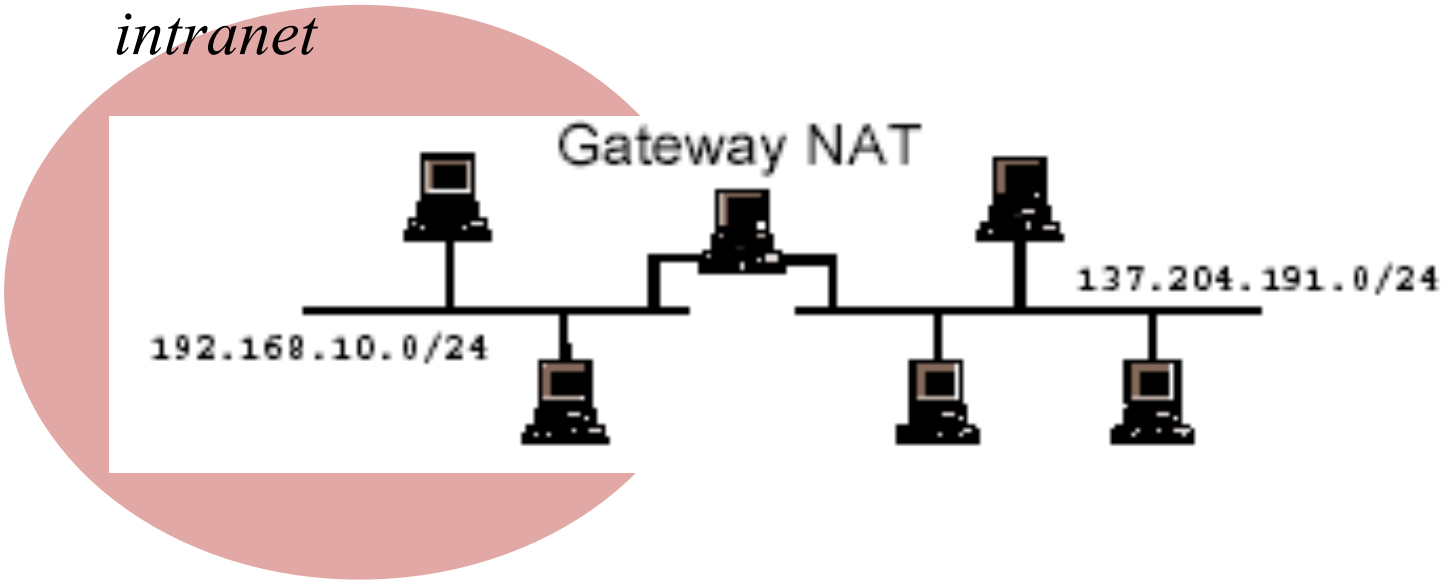
Router NAT

- Il meccanismo NAT/PAT - *Network address translation, Port address translation* - permette ad una rete locale che utilizza indirizzi IP riservati alle reti private di accedere all'esterno
- In tal caso, tutto il traffico con la rete esterna viene smistato dal router NAT, che si preoccupa di inoltrare le risposte all'interno della rete locale
- Ciò significa che all'esterno appare sempre solo un elaboratore, **il router NAT**, e dall'esterno non c'è modo di accedere agli elaboratori della rete locale, non avendo questi un indirizzo IP pubblico:
  - Dalla rete locale verso l'esterno: il router prende il pacchetto IP dell'host della rete locale e gli sostituisce l'indirizzo IP privato con il proprio.
  - Dall'esterno verso la rete locale: in ricezione l'ip dei pacchetti viene gestito in maniera analoga dal router NAT, che ripristina l'ip dell'host destinatario

Network Address Translation (NAT)

- ***Tecnica per il filtraggio di pacchetti IP con sostituzione degli indirizzi*** *(o network masquerading)*

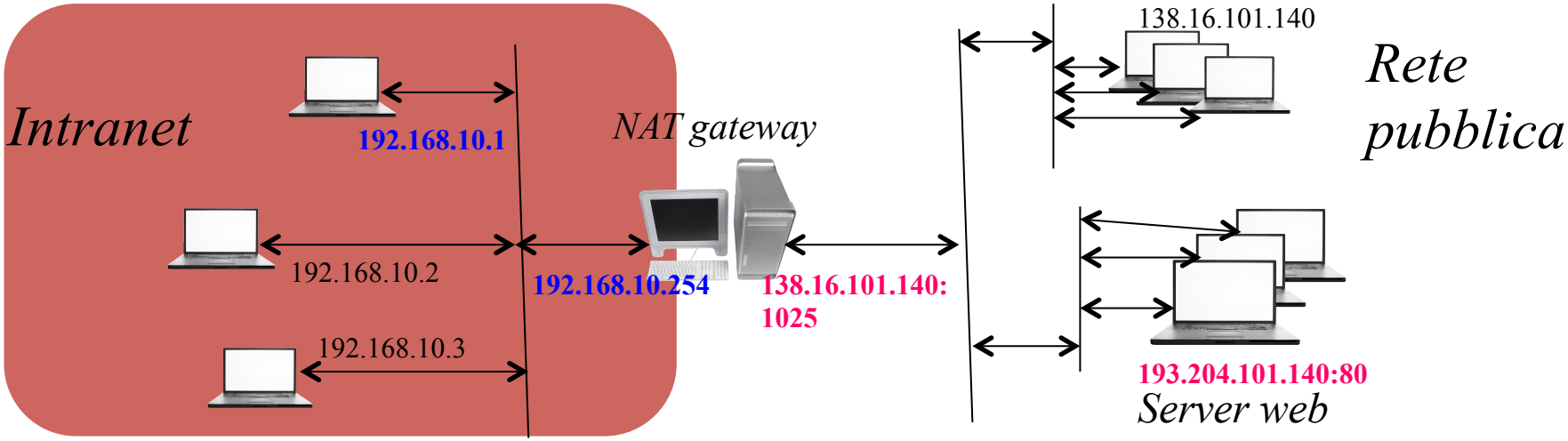
- ***Definito nella RFC 1631 per permettere a reti IP private l'accesso a reti IP pubbliche tramite un apposito gateway che modifica gli indirizzi IP (del sorgente e del destinatario) dei pacchetti in transito sul sistema (router o firewall).***



NAT – Conversione di indirizzo sorgente

Il NAT può fornire una semplice conversione di indirizzo IP *o* ***conversioni contemporanee di indirizzi IP e numero di porta*** *per presentare all'esterno le macchine di una rete interna con l'indirizzo e la porta del gateway.*

L'host interno 192.168.10.1 vuole contattare un web server pubblico 193.204.101.140
Il client interno dalla porta 3123 contatta il gateway NAT 192.168.10.254.
Il gateway 192.168.10.254 modifica l'IP sorgente con l'IP del gateway 138.16.101.140:1025 e come destinatario conserva l'IP del web server 137.204.191.140:80.
Il gateway inoltra il pacchetto della richiesta web con i nuovi indirizzi sorgente e destinazione e memorizza queste informazioni nella propria tabella NAT dinamica .
Il server web risponde al gateway con il proprio indirizzo sorgente e con l'indirizzo destinatario del gateway, il quale , in base alla tabella NAT provvede a girare al client 192.168.10.1 la pagina html ricevuta.



NAT con IPTABLES

Il kernel Linux implementa funzionalità di NAT tramite il pacchetto applicativo IPTABLES

- Il NAT è definito nella tabella "**nat**"
- IPTABLES definisce nella tabella "**nat**" le catene
  - **PREROUTING**: contiene le regole da usare per sostituire l'indirizzo di destinazione dei pacchetti in arrivo (Destination NAT o **DNAT**)
  - **POSTROUTING**: contiene le regole da usare per sostituire l'indirizzo di origine dei pacchetti in uscita (Source NAT o **SNAT**)
  - **OUTPUT**: contiene le regole da usare per sostituire l'indirizzo di destinazione dei pacchetti generati localmente (**DNAT**)
- Se un pacchetto non soddisfa nessuna regola, viene applicata la regola di default, o "policy", di quella catena
- La policy "**ACCEPT**" vuol dire assenza di conversione

Instradamento da e verso il router NAT

- Il router NAT/PAT, prima di poter compiere il suo lavoro, deve essere instradato attraverso le sue interfacce di rete.

- Nell'esempio prima raffigurato

  – Per Il router si preparano gli instradamenti verso le varie parti della rete locale, e l'instradamento verso l'esterno corrisponde a quello predefinito

  – Per il resto della rete locale, l'instradamento predefinito deve portare al router NAT/PAT, perché solo lui è in grado di gestire il traffico con gli indirizzi esterni alla rete locale.

Traduzione degli indirizzi

- Il meccanismo NAT/PAT deve essere impostato definendo i gruppi di indirizzi (cioè le sottoreti) di origine e di destinazione.

- Nell'esempio
  - il router è connesso a due sottoreti, 192.168.1.0 e 192.168.2.0, ma potrebbero essercene altre (192.168.3.0,...).
    In tal senso, gli indirizzi da inoltrare all'esterno sono tutti quelli della rete 192.168.0.0/16, dove il secondo indirizzo è la maschera di rete.

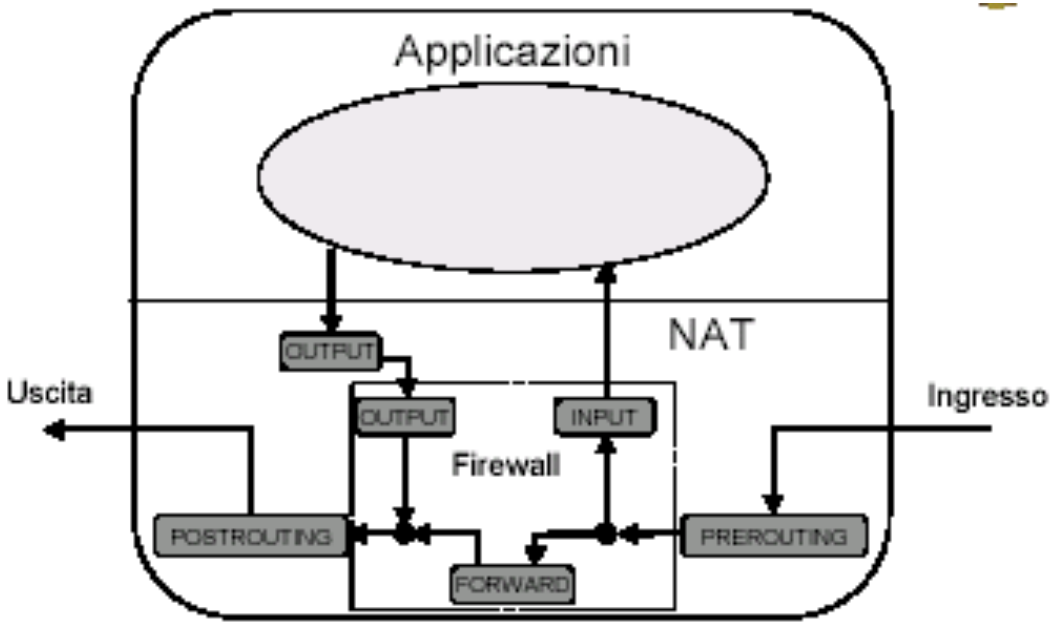
  - gli indirizzi di destinazione sono semplicemente tutti i restanti, cosa che si indica con 0.0.0.0/0.

Inner firewall

Un ***firewall*** che filtri pacchetti IP è **iptables** (lo stesso usato per la NAT).

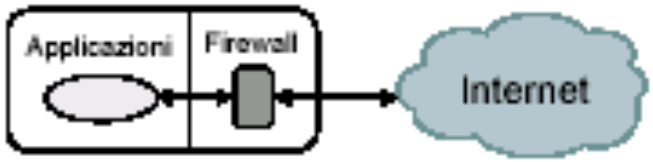
Questa **applicazione** permette di creare regole – organizzate in catene di comandi – che specificano a quali combinazioni di indirizzi IP e porte è consentito il passaggio per le interfacce del firewall.

Le regole tengono in considerazione lo stato request/reply di una applicazione IP, e vengono eseguite nel kernel.

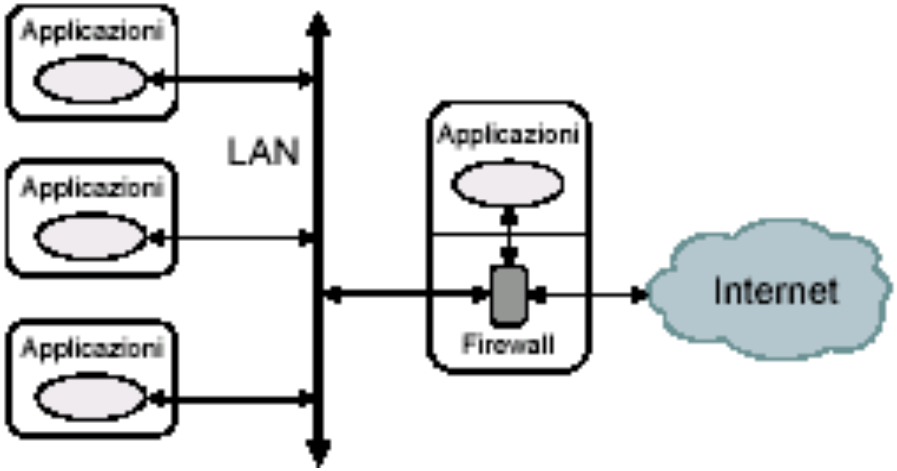


Servizi di Firewall e NAT (IPTABLES)

Un Firewall è un "filtro" software che serve a proteggersi da accessi indesiderati provenienti dall'esterno, cioè da Internet.



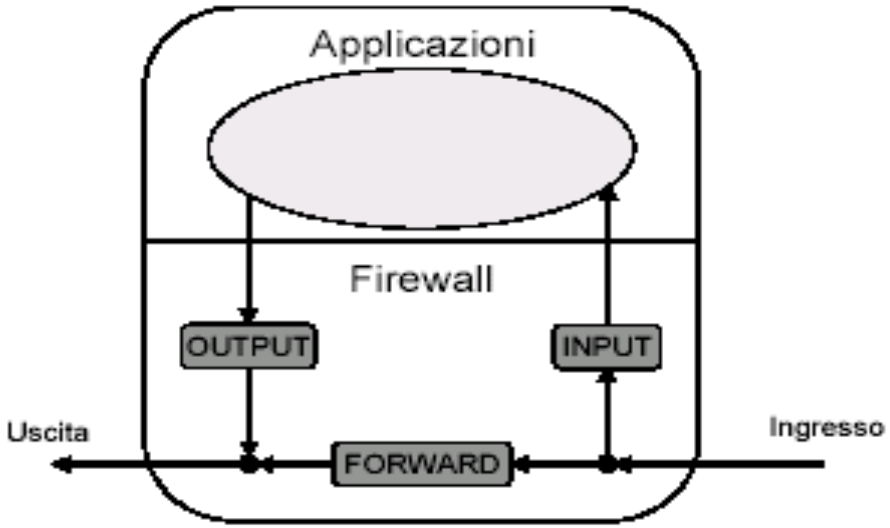
- **Può essere semplicemente un programma installato sul proprio PC che protegge quest'ultimo da attacchi esterni.**
- **Tipicamente usato in accessi domestici a larga banda (ADSL, FTTH).**
- **Oppure può essere una macchina dedicata che filtra tutto il traffico da e per una rete locale Internet**



IPTABLES

- **Il kernel Linux implementa funzionalità di firewall tramite il pacchetto applicativo IPTABLES**
- IPTABLES definisce nella tabella "**filter**" tre gruppi di regole di controllo, chiamate "**catene**"

– **INPUT**: contiene le regole per i pacchetti in arrivo al firewall e destinati all'host locale
– **OUTPUT**: contiene le regole per i pacchetti in uscita dal firewall e originati dall'host locale
– **FORWARD**: contiene le regole da usare per i pacchetti in arrivo al firewall e destinati ad altri host



Regole di IPTABLES

- **Quando un pacchetto viene processato da una catena, esso è soggetto alle regole specificate in essa, secondo l'ordine di inserimento**
- **Una regola può stabilire di scartare (DROP), rifiutare (REJECT) o di accettare (ACCEPT) un pacchetto in base a**
  - ***Interfaccia di rete coinvolta***
  - ***Indirizzo IP di origine***
  - ***Indirizzo IP di destinazione***
  - ***Protocollo (TCP, UDP, ICMP)***
  - ***Porta TCP o UDP di origine***
  - ***Porta TCP o UDP di destinazione***
  - ***Tipo di messaggio ICMP***
- **Se un pacchetto non soddisfa nessuna regola, viene applicata la regola di default, o "policy", di quella catena**

Blocking Incoming Traffic

You want to block all incoming network traffic, except from your system itself. Do not affect outgoing traffic.

```
# iptables -F INPUT
# iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
# iptables -A INPUT -j REJECT
```

dal man di iptables:

-F, --flush [chain] Flush the selected chain (all the chains in the table if none is given). This is equivalent to deleting all the rules one by one.
-A, --append chain rule-specificationAppend one or more rules to the end of the selected chain.
-state: ESTABLISHED meaning that the packet is associated with a connection which has seen packets in both directions
- -j, --jump target This specifies the target of the rule; i.e., what to do if the packet matches it.

Inner firewall

- La shell di windows non fornisce comandi equivalenti ad iptables. Per creare applicazioni firewall si può adoperare il Windows Driver Kit e creare un Filter-Hook Driver.
- Chi è curioso può cliccare questo sito: http://msdn.microsoft.com/it-it/library/aa504968(en-us).aspx