Come funziona Linux: Reteeeeeeeeeeeee !

Undicesima parte

di Giuseppe Zanetti

Il networking è senza ombra di dubbio l'aspetto in cui Linux maggiormente eccelle nel confronto con quasi tutti gli altri sistemi operativi, grazie all'esperienza trentennale ereditata da UNIX. Su questo sistema infatti è nata, ed ancora maggiormente si basa, la Rete di tutte le reti...".

Non stiamo per affrontare un argomento semplice o che possa essere svolto completamente in una o più puntate di un corso su una rivista. Perciò in questa sede mi limiterò a mostrarne gli aspetti più importanti e specifici per Linux. E' richiesta perciò una conoscenza di base del protocollo TCP/IP, di cui si è già parlato in altre rubriche di questa rivista e per cui esistono decine di documenti introduttivi, sia in biblioteca che su Internet.

In questa prima puntata dedicata alla rete vedremo cosa sono e come configurare le interfacce di rete in Linux.

Protocolli di rete supportati da Linux

Linux supporta diversi protocolli di rete, adatti sia per lavorare in LAN che per collegamenti su distanze geografiche (di tipo WAN, Wide Area Network). Fra i primi vale la pena ricordare NetBIOS (implementato come client nel kernel e, per quanto riguarda la parte server, mediante il programma Samba), AppleTalk, IPX (Novell), NFS e CODA. Anche per quanto riguarda le reti geografiche, Linux non è certamente avaro di protocolli: oltre ai classici PPP e SLIP per il collegamento ai provider Internet, le nuove versioni del kernel supportano anche i protocolli tipici dei router, come Fame Relay. Ovviamente il protocollo (per meglio dire la "famiglia" di protocolli) su cui concentreremo la nostra attenzione in questo articolo è il TCP/IP, il quale è ormai lo standard de facto per il collegamento in rete, sia locale che geografica.

Per potere utilizzare il TCP/IP, il kernel deve essere stato compilato con il supporto per il software di rete e per questo protocollo. Ciò si ottiene in fase di compilazione rispondendo in modo affermativo alle domande

```
Networking support   (CONFIG_NET) : Y
TCP/IP networking (CONFIG_INET) : Y
```

Interfacce di rete TCP/IP

Una macchina Linux su cui è installato il protocollo TCP/IP non ha un solo indirizzo di rete, bensí tanti quante sono le "interfacce di rete" attive. In Linux per "interfaccia di rete" si intende un oggetto logico a cui è associato un indirizzo IP. Nella maggior parte dei casi ad essa è associato un dispositivo fisico, ad esempio una scheda ethernet o una connessione PPP su modem, tuttavia esistono interfacce di rete che non hanno alcun hardware associato, ad esempio l'interfaccia "lo" che permette il collegamento alla macchina locale mediante l'indirizzo 127.0.0.1 (indirizzo di "loopback").

Date le loro particolari caratteristiche, le interfacce di rete non hanno un corrispondente file speciale in /dev ma vengono create e gestite direttamente dal kernel.

Le interfacce di rete più comunemente usate sono le seguenti:

lo — Si tratta dell'interfaccia di loopback utilizzata per collegamenti TCP/IP all'interno dello stesso calcolatore. Per convenzione ad essa viene assegnato l'indirizzo IP 127.0.0.1 e generalmente le viene anche associato il nome simbolico "localhost". Una vecchia battuta è la seguente: volete accedere ad un archivio vastissimo di immagini pornografiche ? Collegatevi usando il vostro nome e la vostra password all'indirizzo FTP 127.0.0.1...

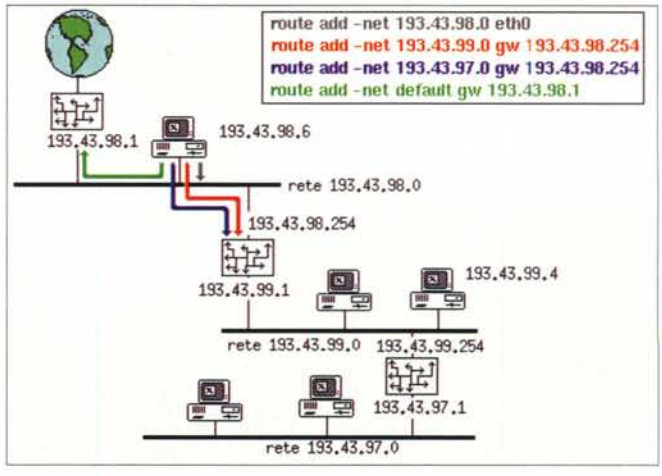


Figura 1: Schema relativo all'esempio di rete proposto.

eth0, eth1, ..., ethn	Si tratta delle interfacce di rete collegate alle schede ethernet del computer. Nel caso siano presenti più dispositivi hardware, essi vengono numerati in ordine crescente in base al loro MAC address (un indirizzo di 48 bit unico per ogni scheda ethernet prodotta e cablato direttamente nell'hardware).
eth0:1, ..., ethn:m	Mediante queste interfacce logiche è possibile assegnare più indirizzi IP alla stessa scheda ethernet. Tale operazione, se supportata da un software adeguato, permette ad esempio di gestire siti FTP o WWW virtuali (www.pippo.it e www.pluto.it) con indirizzi diversi sulla stessa macchina.
ppp0, ppp1, ..., pppn	Si tratta delle interfacce logiche che gestiscono i collegamenti PPP. Vengono create nel sistema quando si lancia il demone pppd che gestisce la connessione. Di solito sono collegate ad una interfaccia seriale e ad un modem, ma per alcune applicazioni è possibile utilizzarle anche senza hardware, ad esempio per realizzare collegamenti PPP che si appoggiano sopra connessioni TCP/IP (tunnel).
ippp0, ..., ipppn	Sono simili a pppn, solamente che si riferiscono alla versione sincrona del PPP (SyncPPP) usata per i collegamenti ISDN.
plip0	Permette un collegamento TCP/IP punto-a-punto fra due PC utilizzando la porta parallela ed un cavo di tipo "laplink". E' compatibile con un analogo software funzionante in ambiente Windows.
tap0	Si tratta di una interfaccia logica di rete (ethertap), introdotta dalla versione 2.2 di Linux. Essa viene vista a livello di sistema come una comune scheda ethernet, ma non è associata ad un dispositivo fisico: i frame di dati invece di essere trasmessi su un cavo possono essere spediti/letti da software. In questo modo è possibile realizzare facilmente gateway fra protocolli di rete diversi (ad esempio fra TCP-IP ed Ethertalk) oppure alcune applicazioni molto interessanti, come è spiegato nel file /usr/src/linux/Documentation/networking/ethertap.txt incluso nei sorgenti di Linux 2.2.x.

Configurazione delle interfacce di rete in Red Hat

A seconda dell'utilizzo, le interfacce di rete possono essere configurate una volta per tutte in fase di boot mediante uno degli script in /etc/rc.d (ad esempio le schede ethernet o una connessione PPP dedicata), oppure possono essere invece create solamente per il tempo necessario durante l'utilizzo, come per il PPP nel caso di una connessione dial-up

Nella distribuzione Red Hat, lo script che si occupa della attivazione delle interfacce di rete è /etc/rc.d/init.d/network. Nel caso si utilizzi il runlevel 3, esso avrà un link simbolico come /etc/rc.d/rc3.d/S10network. Per attivare la rete si userà pertanto:

```
# /etc/rc.d/rc3.d/S10network start
Enabling IPv4 packet forwarding          [  OK  ]
Bringing up interface lo                 [  OK  ]
Bringing up interface eth0               [  OK  ]
```

Bibliografia essenziale

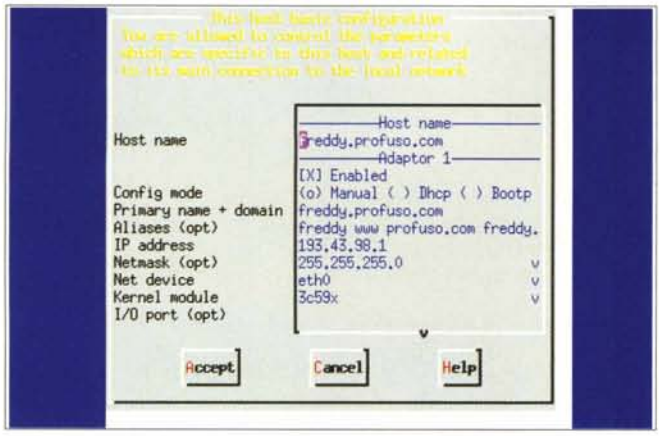
Per ulteriori informazioni e dettagli sul funzionamento della rete in Linux, è possibile fare riferimento ai seguenti testi, accessibili gratuitamente (ooppsss... liberamente) nel sito http://www.linuxdoc.org/:

Titolo	Descrizione
Net-HOWTO	Riassunto delle principali funzioni di networking di Linux con puntatori verso informazioni più dettagliate
PPP-HOWTO	Come configurare il PPP
Virtual-services-HOWTO	Uso dell'IP aliasing per creare server virtuali
DHCP-Mini-HOWTO	Funzionamento del DHCP
Home-Network-Mini-HOWTO	Come configurare Red Hat 6 come server per una Intranet domestica o per un ufficio. Comprende argomenti come DNS, DHCP, sicurezza base, ...
IP-Alias-Mini-HOWTO	Descrive il funzionamento dell'IP aliasing in Linux
PLIP-Mini-HOWTO	Come collegare due macchine Linux usando un cavo parallelo
The Linux Network Administrator Guide (2nd Edition)	Libro che spiega come amministrare il software di rete in Linux e come configurare i principali servizi di rete
The Linux System Administrator Guide	Guida per l'amministratore di sistema. Comprende un capitolo sul networking

I seguenti documenti in italiano si possono invece prelevare nel sito del progetto Italian Linux Documentation Project (http://www.pluto.linux.it/ildp/), dove è anche possibile trovare la traduzione in italiano di alcuni dei documenti appena citati.

Titolo	Descrizione
AppuntiLinux	Guida (2000 pagine) completamente in italiano alla configurazione dei principali servizi in Linux. Contiene una ampia sezione dedicata al networking.
LDR (Linux Domande e Risposte)	Una raccolta di domande e risposte su Linux in italiano

Fra i libri disponibili in libreria, un buon libro di alto livello sul TCP/IP è il seguente:
Douglas E. Comer, Internetworking with TCP/IP, Volume 1: principles, protocols and architecture, Prentice Hall International Editions, ISBN 0-13-227836-7



La rete può essere facilmente configurata anche utilizzando linuxconf.

mentre per fermarla sarà sufficiente scrivere:

```
# /etc/rc.d/rc3.d/S10network stop
Shutting down interface eth0                [  OK  ]
Disabling IPv4 packet forwarding            [  OK  ]
Disabling IPv4 automatic defragmentation    [  OK  ]
```

Le singole interfacce possono essere abilitate e disabilitate singolarmente mediante i comandi ifup e ifdown:

```
# /sbin/ifup eth0
# /sbin/ifdown eth0
```

Gli script ricavano i dati di configurazione delle diverse interfacce dal file /etc/sysconfig/network e dalla directory /etc/sysconfig/network-scripts/. Di solito tali file vengono modificati mediante uno degli strumenti di configurazione resi disponibili dalla distribuzione, ad esempio linuxconf. Tuttavia è anche possibile ed interessante provare a gestirli a mano. Per rendere attive le modifiche è necessario far ripartire la rete con uno dei metodi appena visti.

Il file /etc/sysconfig/network contiene i seguenti dati, generici per il sistema:

```
NETWORKING=yes
FORWARD_IPV4=true
HOSTNAME=freddy.profuso.com
DOMAINNAME=profuso.com
GATEWAY=193.43.98.254
```

Le prime due linee indicano rispettivamente che deve essere attivato il software di rete e che la macchina deve eseguire il routing del protocollo TCP/IP (in seguito vedremo ulteriori dettagli). La riga seguente indica invece il FQDN (Full Qualified Domain Name) della macchina, ovvero il suo nome completo del dominio a cui essa appartiene. Seguono il solo nome del dominio e l'indirizzo IP del "default gateway" (nel caso la rete a cui è collegata la macchina ne abbia uno, altrimenti il campo è vuoto).

Nella directory /etc/sysconfig/network-scripts/ invece sono presenti i file contenenti i dati di configurazione relativi alle interfacce presenti nel sistema. Ad esempio ifcfg-eth0 contiene i dati relativi alla prima scheda ethernet eth0:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=193.43.98.255
IPADDR=193.43.98.1
NETMASK=255.255.255.0
NETWORK=193.43.98.0
ONBOOT=yes
```

Al nome della interfaccia (eth0) segue l'indicazione che essa deve essere configurata in modo statico mediante i dati di seguito indicati. In alternativa è possibile configurarla in modo automatico ricavando i dati necessari da un server DHCP (BOOTPROTO=dhcp) oppure bootp (BOOTPROTO=bootp).

Trattandosi di una interfaccia configurata in modo statico, è necessario inserire di seguito nel file i dati relativi all'indirizzo IP, alla netmask, all'indirizzo di broadcast e a quello di rete. Per il calcolo di questi ultimi si può fare riferimento al riquadro presente in questo articolo.

L'ultima riga indica che l'interfaccia deve essere attivata al boot.

Nel caso si utilizzi un server DHCP, la maggior parte dei dati di configurazione diventa inutile ed è necessario specificare solamente le righe che seguono:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

Il file ifcfg-lo, relativo all'interfaccia di loopback invece è preconfigurato secondo la convenzione di usare l'indirizzo IP 127.0.0.1 per indirizzare la macchina locale. Salvo esigenze particolari, questa impostazione non deve essere modificata.

```
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
BOOTPROTO=none
```

Routing (instradamento di pacchetti IP)

Il "routing" indica la possibilità di instradare pacchetti IP verso interfacce o percorsi (route) diversi a seconda dell'indirizzo IP di destinazione. Nell'esempio rappresentato in figura 1, tutti i pacchetti per la rete a cui appartiene la macchina (193.43.98.0) vengono mandati direttamente sulla scheda di rete eth0, mentre quelli per le reti 193.43.97.0 e 193.43.99.0 vengono passati ad un'altra macchina (interfaccia 193.43.98.254 del router interno) che si occupa di smistarli verso la giusta destinazione. Ovviamente, essendo il router un ponte fra reti diverse, avrà più interfacce su reti diverse, di cui una sulla stessa rete del PC, in modo da poter comunicare con esso.

I pacchetti per le reti non esplicitamente indicate possono essere mandati verso un instradamento di default (default route). Se l'instradamento di default avviene passando per un'altra macchina, questa prende anche il nome di "default gateway".

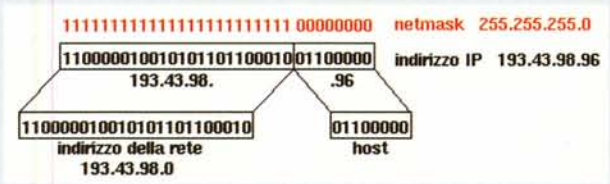
Il funzionamento del meccanismo di routing si basa su una "tabella di routing" (in realtà, in un utilizzo avanzato, la versione 2.2 di Linux permette di definire più tabelle), la quale indica i percorsi che i dati devono seguire per raggiungere la rete/macchina di destinazione. In realtà non viene indicato il percorso completo, bensí solamente l'interfaccia a cui mandare i dati (nel caso la macchina di destinazione sia su una rete accessibile direttamente) oppure il prossimo router a cui passare i pacchetti IP, nel caso di macchine non direttamente collegate. Una volta passato il pacchetto di dati, saranno problemi del router mandare il pacchetto ricevuto verso la giusta destinazione, eventualmente passandolo ad un altro router.

Un nuovo instradamento può essere creato mediante il comando route, il quale inserisce nella tabella di routing una entry di tipo statico. Esiste anche la possibilità, che di solito si usa solo nelle reti complesse, di far cercare ai router in modo autonomo i percorsi e di fare in modo che essi si scambino fra loro le informazioni sulla topologia della rete.

Nel nostro esempio le route da creare sono le seguenti:

Reti e indirizzi TCP/IP

Il protocollo TCP/IP presuppone una topologia di networking composta da reti, ognuna composta da una o più macchina (chiamate anche "host"). Le diverse reti sono connesse fra loro mediante apposite macchine dette router, che svolgono la funzione di instradatori del traffico fra le diverse reti. Un esempio di rete è una LAN formata da tutte le macchine connesse allo stesso spezzone di cavo ethernet. In TCP/IP comunque il concetto di rete deve essere visto come una suddivisione logica, necessaria all'indirizzamento, prima che come una suddivisione fisica.



Anche se per una persona la cosa sembra difficile, per una macchina si tratta di conti banali, che permettono di capire molto velocemente se un indirizzo si riferisca o meno ad un host presente nella stessa rete.

Ad ogni macchina - o meglio ad ogni sua interfaccia di rete (ethernet, connessione PPP, ...) - viene assegnato un unico indirizzo a 32 bit, che ha lo scopo di distinguerla univocamente da ogni altra macchina. Nella notazione più utilizzata, detta "dotted-decimal format", l'indirizzo viene espresso spezzando i 32 bit in 4 otteti (gruppi di 8 bit) e scrivendo i valori risultanti in decimale (un numero da 0 a 255).

```
11000001.00101011.01100010.01100000 -> 193.43.98.96
```

Dei 32 bit che formano un indirizzo TCP/IP, alcuni indicano la rete, mentre i rimanenti indicano il numero della macchina all'interno di essa. Per distinguere i bit utilizzati per l'indirizzo della rete, si utilizza la netmask, un numero binario a 32 bit in cui sono posti a 1 solamente gli n bit più significativi dedicati all'indirizzamento della rete (anche per esprimere la netmask si usa di solito la notazione dotted decimal).

L'indirizzo della rete è facilmente ottenibile facendo un AND bit a bit fra l'indirizzo IP e la netmask:

```
indirizzo IP   11000001.00101011.01100010.01100000
               193.43.98.96
                              AND
netmask        11111111.11111111.11111111.00000000
               255.255.255.0
               ------------------------------------
rete           11000001.00101011.01100010.00000000
               193.43.98.0
```

Eseguendo invece un AND fra l'indirizzo IP e il complementare a 32 bit della netmask (ovvero con i bit 1 posti a 0 e viceversa) si ottiene l'indirizzo dell'host all'interno della rete:

```
indirizzo IP      11000001.00101011.01100010.01100000
                  193.43.98.96
                                 AND
~netmask          00000000.00000000.00000000.11111111  0.0.0.255
                  ------------------------------------
numero host       00000000.00000000.00000000.01100000  0.0.0.96
```

Classi di reti

A seconda di quanti bit vengono dedicati all'indirizzamento della rete, originariamente è stata fatta una suddivisione in reti di classe A (8 bit di netmask, ovvero 255.0.0.0), B (255.255.0.0) o C (255.255.255.0).

Ovviamente un numero maggiore di bit a 1 nella netmask significa la possibilità di avere più reti ma con un numero minore di host. Una rete di classe C ad esempio permette di indirizzare fino a 254 macchine (in teoria 256, ma il primo e l'ultimo indirizzo sono riservati per identificare la rete stessa e per il broadcast, ovvero per la trasmissione di un pacchetto IP a tutte le macchine della rete.). Prima dell'esplosione di Internet c'erano indirizzi a volontà e purtroppo sono state assegnate reti di classe C anche a chi aveva bisogno solo di pochi indirizzi e ciò ha portato al quasi esaurimento dello spazio di indirizzamento. Per questo da alcuni anni non vengono più assegnate intere reti di classe C, ma solamente sottoreti composte da un numero minore di indirizzi (ad esempio 32 indirizzi con netmask 255.255.255.224). In questo caso si parla di indirizzamento "classless". La soluzione definitiva al problema sarà probabilmente il passaggio al protocollo IPv6, già perfettamente supportato da Linux, il quale aumenta notevolmente lo spazio di indirizzamento.

Calcolo dell'indirizzo di broadcast

Configurando a mano la rete in Linux c'è la necessità di calcolare a mano i due valori per l'indirizzo della rete a cui la macchina appartiene e per l'indirizzo di broadcast. Abbiamo già visto come calcolare il primo. Anche il broadcast è facile da calcolare, tenendo conto che è l'indirizzo della rete con tutti i bit dell'host posti ad uno, il che in aritmetica binaria equivale a fare un'operazione di OR bit a bit fra l'indirizzo della rete e il complementare della netmask:

```
rete              11000001.00101011.01100010.00000000
                  193.43.98.0
                         OR
~netmask          00000000.00000000.00000000.11111111
                  0.0.0.255
                  ------------------------------------
broadcast         11000001.00101011.01100010.11111111
                  193.43.98.255
```

```
# route add -net 193.43.98.0 netmask 255.255.255.0 eth0
# route add -net 193.43.99.0 netmask 255.255.255.0 gw 193.43.98.254
# route add -net 193.43.97.0 netmask 255.255.255.0 gw 193.43.98.254
# route add default gw 193.43.98.1
```

Come si vede, la prima delle reti è accessibile direttamente mandando i pacchetti IP alla scheda ethernet collegata all'interfaccia logica eth0, mentre tutte le altre sono accessibili sfruttando dei router.

Ovviamente deve essere presente anche l'instradamento verso l'indirizzo di loopback 127.0.0.0. Di solito si dirige verso l'interfaccia "lo" l'intera rete 127.0.0.0, usando il comando:

```
# route add -net 127.0.0.0 netmask 255.0.0.0 lo
```

Per vedere la tabella di routing si utilizzano i comandi "route -n" oppure "netstat -rn". L'opzione -n fa in modo che il comando non tenti di risolvere all'indietro (reverse lookup) gli indirizzi IP in nomi simbolici, il che potrebbe bloccare il comando nel caso il resolving dei nomi (lo vedremo fra poco) non fosse configurato correttamente o il DNS non raggiungibile

```
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
193.43.98.6     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
193.43.98.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
193.43.97.0     193.43.98.254   255.255.255.0   UG    0      0        0 eth0
193.43.99.0     193.43.98.254   255.255.255.0   UG    0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         193.43.98.1     0.0.0.0         UG    0      0        0 eth0
```

Il significato dei valori presentati è il seguente: Destination indica la rete o la macchina di destinazione (0.0.0.0 indica la route di default); Gateway (sinonimo di router) indica l'indirizzo del router da usare (0.0.0.0 nel caso di connessione diretta); Genmask corrisponde alla netmask; Flags presenta varie informazioni, fra cui se si tratta di una route verso una rete o verso un singolo indirizzo (H=host), se si sta usando un router (G) e se l'instradamento è al momento attivo (U); Metric indica il "costo" del percorso (di solito la "distanza" espressa in hops, ovvero il numero di router che occorre passare per raggiungere la destinazione) e permette di scegliere degli instradamenti preferenziali (non usato dal kernel, ma da alcuni protocolli di routing dinamico); Ref indica il numero di riferimenti per la route; Use indica il numero di volte in cui la route è stata visionata; infine Iface è il nome dell'interfaccia di rete. Nel caso di un singolo PC connesso ad Internet mediante un modem, saranno attive solamente le interfacce di rete di loopback e ppp0. La route di default sarà in questo caso il collegamento ad Internet.

```
# route add default ppp0
```

Per verificare che le tabelle di routing funzionino in modo corretto si può utilizzare il comando ping, che verifica la connettività verso un dato indirizzo spedendo dei pacchetti di tipo ICMP ed aspettando una risposta dalla macchina remota.

Esso mostra inoltre il tempo impiegato dai pacchetti per compiere il tragitto di andata e ritorno (RTT, Round Trip Time). PS: per terminare usate CTRL-C:

Reti private

Per risolvere temporaneamente ed in modo elegante il problema dell'esaurimento degli indirizzi, si è stabilito che le reti private (ad esempio le reti aziendali o casalinghe) non debbano avere assegnati degli indirizzi Internet validi, ma che debbano usare degli indirizzi convenzionali. Per questo scopo sono stati riservati i gruppi di indirizzi

```
10.0.0.0 - 10.255.255.255       netmask 255.0.0.0
172.16.0.0 - 172.31.255.255     netmask 255.255.0.0
192.168.0.0 - 192.168.255.255   netmask 255.255.255.0
```

rispettivamente per reti private di classe A, B oppure C. Essendo reti private, nessuno vieta di spezzarle a loro volta in reti più piccole (subnetting).

Poichè gli indirizzi di rete privata non sono usabili in Internet, di solito il collegamento ad Internet di una rete siffatta avviene mediante un proxy (una macchina che si occupa di fare da tramite dalla rete locale ai server Internet) oppure mediante meccanismi di traslazione di indirizzi come NAT (Network Address Translation) o PAT (Port Address Translation). Nel primo caso il router che collega la rete privata ad Internet dispone di n indirizzi che usa per rimappare in modo statico oppure dinamico (di volta in volta) i pacchetti in entrata o in uscita, in modo da associare un indirizzo valido per Internet ad un indirizzo di rete privata. E' ovvio che se il numero di indirizzi reali di cui dispone il server è minore del numero di macchine nella rete privata, qualcuno finisce col non poter navigare. Se, al contrario, il numero di indirizzi reali è maggiore o uguale a quello delle macchine interne, allora l'operazione non comporta alcun beneficio in termini di risparmio di indirizzi Internet. Per questo motivo spesso si preferisce usare (o affiancare al NAT) una tecnica di PAT, che in Linux si chiama IP masquerading, in cui ad ogni indirizzo reale vengono fatti corrispondere più indirizzi interni, grazie al fatto che una connessione non è identificata solamente dagli indirizzi delle due macchine coinvolte, ma anche dal numero delle porte TCP/UDP (i due protocolli maggiormente usati della famiglia TCP/IP). Linux è in grado di gestire sia NAT che IP masquerading e perciò può essere utilizzato facilmente come un router per una rete privata.

```
# ping www.pluto.linux.it
PING www.pluto.linux.it (147.162.126.3) from 62.98.87.226 :
56(84) bytes of data.
64 bytes from 147.162.126.3: icmp_seq=0 ttl=242 time=105.4 ms
64 bytes from 147.162.126.3: icmp_seq=1 ttl=242 time=141.1 ms
64 bytes from 147.162.126.3: icmp_seq=2 ttl=242 time=100.5 ms

--- www.pluto.linux.it ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 100.5/115.6/141.1 ms
```

Se si vuole conoscere il percorso esatto che compiono i pacchetti per arrivare a destinazione si utilizza invece traceroute. Esso spedisce un tipo particolare di pacchetto ICMP percui tutti i router per cui il esso transita spediscono indietro una risposta. La "mappa" così ottenuta viene mostrata sullo schermo e permette di capire se e dove ci sono problemi di connettività o quale router è la causa di eventuali rallentamenti nella connessione. Poichè in Internet il routing viene eseguito in modo dinamico, ogni pacchetto non segue necessariamente lo stesso percorso di quello precedente o di quello successivo (ad esempio perchè una tratta è congestionata o interrotta). Traceroute di solito manda 3 pacchetti ICMP. Nell'esempio che segue, gli hop (salti) 12 e 13 mostrano che i pacchetti ad un certo punto hanno preso strade leggermente diverse.

```
# traceroute www.pluto.linux.it
traceroute to www.pluto.linux.it (147.162.126.3), 30 hops max, 38 byte packets
 1  gw3a-62.wind.it (212.245.79.229)  22.303 ms  21.660 ms  21.615 ms
 2  c-ve1-fe3a.wind.it (212.245.64.129)  21.453 ms  21.339 ms  21.352 ms
 3  c-rm2-ve1-atm1.wind.it (212.245.248.117)  32.792 ms  32.730 ms  32.572 ms
 4  c-rm4-fe6a.wind.it (212.245.158.132)  33.528 ms  33.553 ms  33.178 ms
 5  212.245.158.122 (212.245.158.122)  40.152 ms  44.528 ms  38.558 ms
 6  garr-nap.inroma.roma.it (194.242.224.15)  38.028 ms  46.223 ms  37.464 ms
 7  roma-rix.garr.net (193.206.134.225)  118.376 ms  58.960 ms  54.713 ms
 8  bo-rm-2.garr.net (193.206.134.37)  54.743 ms  433.767 ms  131.212 ms
 9  pd-bo-2.garr.net (193.206.134.78)  88.991 ms  121.562 ms  99.836 ms
10  unipd-rc.pd.garr.net (193.206.132.222)  159.118 ms  103.492 ms  89.186 ms
11  147.162.250.251 (147.162.250.251)  87.472 ms  54.550 ms  58.323 ms
12    147.162.252.173  (147.162.252.173)    73.298  ms  sito-180.unipd.it
(147.162.251.14)  76.696 ms  72.514 ms
13  147.162.252.173 (147.162.252.173)  92.581 ms 147.162.254.5 (147.162.254.5)
      104.462 ms  62.019 ms
14  147.162.254.5 (147.162.254.5)  255.819 ms  74.544 ms  75.587 ms
15  ip126003.psy.unipd.it (147.162.126.3)  86.291 ms  79.725 ms  104.960 ms
```

Forwarding in Linux 2.2

Dal punto di vista concettuale, la maggiore differenza fra un router ed una macchina normale è la possibilità di passare i pacchetti di dati ricevuti su una interfaccia verso un'altra (forwarding). Nelle versioni 2.2.x del kernel il forwarding di pacchetti IP non è abilitato di default per ragioni di sicurezza. Red Hat permette di scegliere tale possibilità all'interno del file /etc/sysconfig/network. Il comando che viene eseguito nel caso sia presente la linea FORWARD_IPV4=true è il seguente:

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

Esso mostra un esempio di come sia possibile utilizzare il proc filesystem per modificare il funzionamento del kernel a run time. Altri esempi di "tuning" del kernel sono documentati nei file /usr/src/linux/net/TUNABLE e /usr/src/linux/Documentation/networking/ip-sysctl.txt.

Driver dell'hardware (ethernet)

Nel caso l'interfaccia di rete si appoggi ad un dispositivo hardware, prima di configurarla è ovviamente necessario caricare il device driver opportuno. Nel caso di una vecchia scheda ethernet su bus ISA, è necessario caricare il relativo driver mediante insmod specificando indirizzo di I/O e interrupt:

```
# insmod /lib/modules/2.2.18/net/ne.o io=0x300 irq=7
```

Tale operazione può essere automatizzata inserendo i parametri che occorrono in /etc/conf.modules:

```
alias eth0 ne
options ne io=0x300 irq=7
```

I parametri accettati dai driver per le diverse schede di rete sono descritti nel file /usr/src/linux/Documentation/networking/net-modules.txt.

La maggior parte dei driver delle schede ethernet (tutte le PCI e alcune ISA) è capace di cercare automaticamente la presenza dell'hardware su diversi indirizzi I/O e di individuare da solo l'IRQ da usare. In questo caso sarà sufficiente specificare solamente il nome del driver:

```
alias eth0 3c59x
```

Se tutto sarà andato per il verso giusto, la scheda verrà riconosciuta ed evidenziata nei messaggi che appaiono al boot (ottenibili a posteriori mediante il comando dmesg):

```
# dmesg|more
...
3c59x.c:v0.99H 11/17/98 Donald Becker
http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html
eth0: 3Com 3c900 Boomerang 10Mbps Combo at 0x6400,
00:60:97:ac:34:a4, IRQ 12
8K word-wide RAM 3:5 Rx:Tx split, autoselect/10baseT interface.
Enabling bus-master transmits and whole-frame receives.
...
```

Ulteriori informazioni sulle schede ethernet possono esse-

re reperita nell'HOWTO specifico (Ethernet-HOWTO) su http://www.linuxdoc.org/.

Configurazione manuale delle interfacce di rete

Nel caso si volesse configurare a mano una interfaccia di rete, la procedura da seguire sarà quella di attivarla mediante il comando apposito ifconfig e poi di creare mediante route gli opportuni percorsi di instradamento dei pacchetti. Tale procedura vale anche per le interfacce di tipo ethernet. Usando i dati indicati nell'esempio precedente avremo pertanto:

```
# ifconfig eth0 193.43.98.6 netmask 255.255.255.0
  broadcast 193.43.98.255 up
```

E' ora necessario creare una route (percorso) di instradamento dei pacchetti verso l'interfaccia

```
# route add 193.43.98.6 eth0
```

e uno verso la rete a cui è collegata l'interfaccia ethernet:

```
# route add -net 193.43.98.0 netmask 255.255.255.0 eth0
```

Se ora proviamo ad usare il comando ping per testare l'interfaccia di rete, dovremmo vederla attiva:

```
# ping 193.43.98.6
PING 193.43.98.6 from 193.43.98.6 : 56(84) bytes of data.
64 bytes from 193.43.98.6: icmp_seq=0 ttl=255 time=0.2 ms
--- 193.43.98.6 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
```

Per l'interfaccia di loopback useremo invece:

```
# ifconfig lo 127.0.0.1 netmask 255.0.0.0 up
# route add -net 127.0.0.0 lo
```

Per vedere lo stato di tutte le interfacce di rete si usa il comando ifconfig:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:60:97:AC:34:A4
          inet addr:193.43.98.6  Bcast:193.43.98.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:12 Base address:0x6400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:593 errors:0 dropped:0 overruns:0 frame:0
          TX packets:593 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Le configurazioni possono essere rese permanenti inserendo i comandi appena descritti in uno dei file di inizializzazione della macchina in /etc/rc.d. In questo caso è buona norma indicare il path completo dei comandi (/sbin/ifconfig, /sbin/route, ...), per evitare che il sistema rischi di non trovarli se la directory in cui risiedono non è inclusa nel path che esso "vede" al momento del boot.

Per disabilitare una interfaccia si utilizza l'opzione "down" di ifconfig:

```
# ifconfig eth0 down
```

Ovviamente è bene anche cancellare le route relative:

```
# route del 193.43.98.0
```

IP Aliasing

Abbiamo già detto che è possibile assegnare ad una stessa interfaccia più indirizzi IP. Per poter sfruttare tale possibilità è necessario avere un kernel che supporti tale funzionalità. Ciò si ottiene in fase di compilazione del kernel (make config) rispondendo in modo affermativo alle seguenti domande:

```
Network aliasing (CONFIG_NET_ALIAS): Y
IP: aliasing support (CONFIG_IP_ALIAS): Y
```

La procedura di configurazione è poi del tutto analoga a quella appena vista per le ethernet, con la differenza che si utilizza una interfaccia di rete del tipo eth0:n:

```
#   ifconfig   eth0:1   193.43.98.200   netmask
255.255.255.0 up
# route add 193.43.98.200 eth0:1

#   ifconfig   eth0:2   193.43.98.201   netmask
255.255.255.0 up
# route add 193.43.98.201 eth0:2
```

Non occorre ripetere l'instradamento verso la rete 193.43.98.0.

Nel caso di Red Hat, i file a cui fare riferimento per le interfacce relative all'IP aliasing sono /etc/sysconfig/network-scripts/ifcfg-eth0-1 ed /etc/sysconfig/network-scripts/ifcfg-eth0-2.

La maggior parte dei software che forniscono servizi Internet (WWW, FTP, ...) sono configurabili per rispondere in modo diverso a seconda dell'IP mediante cui vi ci si accede. Si veda il documento Virtual-Services-HOWTO.

Resolving dei nomi

La risoluzione dei nomi in Linux (ovvero la conversione da un nome simbolico come www.pluricom.it al corrispondente indirizzo IP 195.110.137.254) può essere effettuata in vari modi: file hosts, DNS, NIS, ... I primi due sono quelli più usati.

Nel primo caso si tratta di creare un file /etc/hosts contenente le associazioni fra nomi ed indirizzi, nel seguente

formato:

```
indirizzo nome [eventuali alias]
```

Vediamo un esempio di file /etc/hosts

```
# indirizzo di loopback
127.0.0.1    localhost.localdomain localhost

# me stesso

193.43.98.1  freddy.profuso.com freddy

# macchine della mia rete

193.43.98.2  proteus proteus.profuso.com
193.43.98.3  ulisse ulisse.profuso.com portatile
193.43.98.4  alien alien.profuso.com

# altri host conosciuti (per evitare query inutili al
DNS)

212.216.176.74  mail.tin.it
212.141.196.152 news.inwind.it
```

Nel caso si disponesse di uno o più server DNS (massimo 3), essi possono essere utilizzati per la risoluzione dei nomi inserendo i relativi indirizzi IP nel file /etc/resolv.conf

```
domain pippo.com
search pippo.com intranet.pippo.com
nameserver 209.237.162.197
nameserver 212.245.255.2
nameserver 194.20.24.1
```

La riga contenente la parola "domain" definisce il dominio predefinito per le interrogazioni al DNS, da usare quando si esegue una query senza specificare il dominio (con il valore dell'esempio, "www" viene cercato come "www.pippo.com"). Analogamente "search" definisce un elenco di possibili domini alternativi. Entrambe le linee "search" e "domain" sono opzionali.

L'ordine standard con cui Linux risolve i nomi è quello di provare prima a cercare una corrispondenza in /etc/hosts e, solo se questa operazione fallisce, di consultare i DNS. Per evitare richieste inutili al DNS (più lente e necessitano di un collegamento alla rete) è perciò possibile inserire i nomi che si usano più spesso in /etc/hosts. Il rovescio della medaglia è dato dal fatto che se gli indirizzi cambiano è necessario tenere aggiornato a mano il file.

Per variare l'ordine di ricerca si può modificare la configurazione standard in /etc/host.conf:

```
order hosts,bind
multi on
```

La prima riga indica l'ordine di consultazione dei servizi (bind è il nome del software che gestisce il DNS). La riga "multi on", abilita la possibilità di trovare in /etc/hosts più indirizzi IP per lo stesso nome simbolico. Per verificare se il resolving dei nomi funziona correttamente è sufficiente provare a raggiungere con un ping l'indirizzo che si vuole testare. Se l'indirizzo non viene risolto, viene segnalato un errore:

```
# ping www.nonesisto.it
ping: unknown host www.nonesisto.it
```

Esistono degli strumenti pensati apposta per verificare la funzionalità dei DNS creando delle opportune query. I più utilizzati sono nslookup e dig:

```
# nslookup www.pippo.it
Server:  dns.wind.it
Address:  212.245.255.2

Non-authoritative answer:
Name:    hosting.pol.it
Address:  212.131.155.25
Aliases:  www.pippo.it
```

Per ulteriori dettagli sul loro funzionamento, si consultino i manuali in linea oppure il documento DNS-HOWTO che si può trovare nel sito http://www.linux-doc.org/.

Configurazione mediante DHCP/bootp

Nel caso la rete locale disponesse di un server DHCP, è possibile configurare le interfacce ethernet in modo da ottenere mediante esso i dati necessari (indirizzi IP, netmask, indirizzi dei DNS, nome della macchina e del dominio, default gateway, ...). Abbiamo già visto come configurare un'interfaccia in modo che usi DHCP in Red Hat, vediamo ora come funziona la procedura.

Al momento del boot viene eseguito il programma pump, il quale è un demone che si occupa di configurare mediante DHCP o bootp una particolare interfaccia e di gestire il file /etc/resolv.conf. Poichè al momento del boot la macchina non ha ancora un indirizzo IP, la richiesta al server viene fatta mandando un frame ethernet di tipo broadcast.

E' possibile lanciare pump mediante una linea di comando simile alla seguente, dove hostname è il nome che abbiamo assegnato alla nostra macchina:

/sbin/pump -i eth0 -h name

Per conoscere lo stato di una interfaccia gestita da pump si può usare il comando

/sbin/pump -i eth0 –status

E' possibile creare un file di configurazione /etc/pump.conf in cui specificare alcuni parametri fra cui il timeout dopo cui considerare la richiesta fallita o il numero di tentativi da eseguire in caso di insuccesso.

Conclusioni

Abbiamo iniziato a vedere come configurare la nostra macchina Linux per l'utilizzo in rete. Nelle prossime puntate vedremo come realizzare un collegamento ad Internet usando PPP e come utilizzare una macchina Linux per offrire servizi di rete, sia locale che su Internet.

MC