Le basi dell'amministrazione di rete.

Simone Piccardi

8 agosto 2002

Indice

1 I concetti base. 1
1.1 Introduzione. 1
1.2 Indirizzi IP 2
1.3 Il *routing* 4
1.4 Il DNS 4
1.5 I servizi e le porte. 5

2 La configurazione della rete 6
2.1 Configurazione del kernel 6
2.2 La configurazione degli indirizzi. 6
2.3 La configurazione dei servizi 8

3 La configurazione manuale della rete 8
3.1 Il comando `ifconfig` 8
3.2 Il comando `route` 10
3.3 Il comando `ping` 11
3.4 Il comando `traceroute` 12
3.5 Il comando `host` 13
3.6 Il comando `netstat` 13

4 I principali file di configurazione 14
4.1 I file di configurazione delle interfacce statiche. 14
4.2 Il file `modules` 16
4.3 Il file `resolv.conf` 16
4.4 Il file `services` 16
4.5 Il file `hosts` 18
4.6 Il file `hosts.conf` 18
4.7 I file `hosts.allow` e `hosts.deny` 18

5 Il collegamento da casa 19
5.1 Il settaggio del modem 19
5.2 Il programma `chat` 21
5.3 Il protocollo PPP 21

1 I concetti base.

Il campo della comunicazione via rete è uno dei più vasti e complessi di tutta l'informatica. Nel corso degli anni sono stati sviluppati molti protocolli che permettessero di far comunicare fra loro diversi computer (AppleTalk, IPX, ecc.); ma in questa lezione prenderemo in esame soltanto il caso di reti basate su TCP/IP, il protocollo[1] più diffuso, quello su cui si basa "Internet", e che ormai è diventato uno standard universale e disponibile su qualunque sistema operativo. Ci limiteremo ad introdurre i concetti fondamentali delle reti IP, la notazione e le terminologie principali.

1.1 Introduzione.

Dato che il protocollo è nato su macchine Unix, il supporto delle reti TCP/IP è la modalità di comunicazione nativa di Linux, e benché per compatibilità siano stati implementati nel kernel anche parecchi altri protocolli di comunicazione, questo resta il principale ed il più usato.

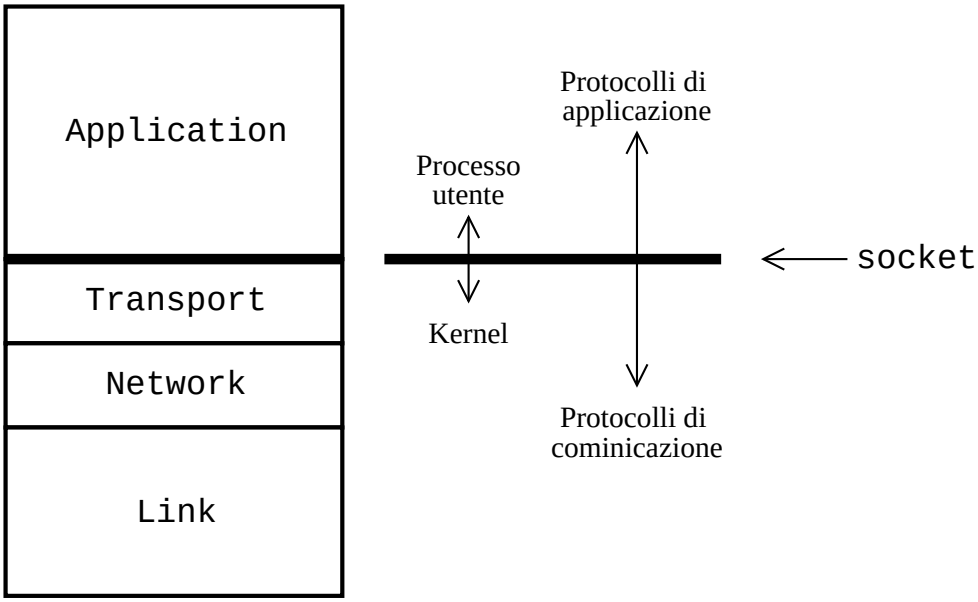

Figura 1: Struttura del sistema

La suite TCP/IP è strutturata su 4 livelli, mostrati in fig. 1, alla base c'è il livello del *collegamento* fisico, cioè il protocollo usato dall'hardware di comunicazione (nella maggior parte dei casi si tratta di ethernet). Questo livello dipende dall'hardware utilizzato, ed in genere viene usato in maniera trasparente all'utente.

Il livello immediatamente superiore è quello di IP, o Internet Protocol, è chiamato livello di *rete* dato che è quello che gestisce la trasmissione dei pacchetti sulla rete. È a questo livello che sono definiti gli indirizzi IP che identificano ogni macchina sulla rete.

Il terzo livello è quello detto di *trasporto*, e serve a gestire la creazione di canali di comunicazione diretta fra due computer. È a questo livello che vengono create le connessioni fra le macchine, ed qui che vengono implementate tutte le procedure che consentono di trasmettere i dati da un estremo all'altro in maniera affidabile.

Dal punto di vista dei programmi questo viene realizzato attraverso i cosiddetti *socket*, l'interfaccia usata in tutti i sistemi Unix per accedere alla comunicazione via rete; attraverso questa interfaccia (che è generica e non dipende dal protocollo specifico che si sta usando) i

[1] in realtà non si tratta di un solo protocollo, ma di una *suite* in cui sono inseriti vari protocolli, in modo da coprire in maniera sostanzialmente completa le più varie esigenze di comunicazione

programmi possono creare i necessari *canali di comunicazione* fra due macchine diverse attraverso la rete.

Il quarto livello, quello di *applicazione*, è quello usato dai vari programmi che forniscono servizi su internet per parlarsi fra di loro, dentro i canali di comunicazione creati con i *socket* che si innestano sul terzo livello. Esso dipende dalle esigenze specifiche del programma, e varia da servizio a servizio.

1.2 Indirizzi IP

Per poter comunicare fra loro due computer in rete devono potersi in qualche modo riconoscere. L'analogia più immediata è quella con il telefono; per poter telefonare occorre avere un numero e conoscere il numero di chi si vuole chiamare. Uno dei concetti base del TCP/IP è appunto di indirizzo IP: questo è l'equivalente del numero di telefono, solo che invece che di un numero decimale composto di un numero variabile di cifre è un numero binario (quindi espresso con soli 0 e 1) ed ha una dimensione fissa (di quattro byte).

Di solito, visto che scrivere i numeri in formato binario è poco comprensibile, si è soliti esprimere il numero IP usando una apposita notazione che viene chiamata*dotted decimal*, usando i normali numeri decimali; un esempio potrebbe essere qualcosa del tipo di `192.168.111.11`. È attraverso questo numero il vostro computer viene identificato univocamente su Internet.

Come per il telefono di casa ogni computer connesso ad Internet viene sempre considerato come facente parte di una *rete*; la cosa è vera anche per le reti private non connesse direttamente ad Internet (come quelle che collegano i computer di un ufficio), per proseguire nell'analogia si pensi alle linee telefoniche interne di una ditta, usate per parlare dalla portineria agli uffici. Dato che Internet, come dice il nome, è un insieme di reti, anche devono venire identificati, questo è fatto attraverso altrettanti indirizzi IP, che corrispondono alla parte *comune* di tutti gli indirizzi delle macchine sulla stessa rete.

Per proseguire nell'analogia con il telefono si può pensare all'indirizzo di rete come al prefisso che serve per parlare con un altra città o un'altro stato, e che è lo stesso per tutti i telefoni di quell'area. La differenza con i prefissi è che un indirizzo di rete IP, quando lo si scrive, deve essere completato da tanti zeri quanti sono necessari a raggiungere la dimensione di 32 bit degli indirizzi normali; per riprendere il precedente esempio di numero IP, un possibile indirizzo di rete ad esso relativo potrebbe essere `192.168.111.0`.

Questo meccanismo significa in realtà che ogni indirizzo su Internet, pur essendo espresso sempre come un singolo numero, nei fatti è composto da due parti, *l'indirizzo di rete*, che prende la parte superiore dell'indirizzo e identifica la particolare sezione di internet su cui detto computer si trova e *l'indirizzo della macchina finale* (il cosiddetto *host*), che prende la parte inferiore del numero e che identifica, all'interno della rete, il vostro computer.

La situazione dunque è ancora analoga a quella di un numero di telefono che è diviso in prefisso e numero locale. La notazione degli indirizzi IP però è diversa da quella dei numeri telefonici e, dato che si ha a che fare con indirizzi binari a dimensione fissa, non esiste l'equivalente della / per dividere il prefisso dal resto del numero; pertanto non è automatico sapere come effettuare questa divisione.

Per questo motivo quando si configura una macchina ad ogni indirizzo IP si associa sempre anche quella che viene chiamata una *netmask*: una maschera che permette di separare la parte di rete da quella di host. Questa viene espressa con la solita notazione dotted decimal, mettendo un 1 ad ogni bit dell'indirizzo corrispondente alla rete e uno zero a quello corrispondente all'host: nel caso dell'indirizzo in esempio si avrebbe allora `255.255.255.0`).

L'assegnazione degli indirizzi IP è gestita a livello internazionale dalla IANA[2], che ha delegato la gestione di parte delle assegnazioni ad altre organizzazioni regionali (come INTERNIC,

[2] *Internet Assigned Number Authority.*

RIPE NCC e APNIC). Per venire incontro alle diverse esigenze gli indirizzi di rete sono stati originariamente organizzati in *classi*, (riportate tab. 1), per consentire dispiegamenti di reti di dimensioni diverse.

Classe	Intervallo		
A	**0**.0.0.0	—	127.255.255.255
B	**128.0**.0.0	—	191.255.255.255
C	**192.0.0**.0	—	223.255.255.255
D	224.0.0.0	—	239.255.255.255
E	240.0.0.0	—	247.255.255.255

Tabella 1: Le classi di indirizzi IP.

Le classi usate per il dispiegamento delle reti di cui è attualmente composta Internet sono le prime tre; la classe D è destinata all'ancora non molto usato *multicast*[3], mentre la classe E è riservata per usi sperimentali e non viene impiegata. Oggigiorno questa divisione in classi non è più molto usata (perché come vedremo fra poco è inefficiente), ma la si trova riportata spesso, ed alcuni programmi cercano di calcolare automaticamente la netmask a seconda dell'IP che gli date, seguendo queste tabelle.[4]

Nella tabella si è riportata in grassetto la parte di indirizzo assegnata alla rete per ciascuna delle tre classi; una rete di classe A è una rete che comprende 16777216 (cioè 2^{24}) indirizzi di singoli computer ed ha una netmask pari a `255.0.0.0`, una rete di classe B comprende 65536 (cioè 2^{16}) indirizzi ed ha una netmask pari a `255.255.0.0` e una rete di classe C comprende 256 (cioè 2^8) indirizzi ed ha una netmask pari a `255.255.255.0`.

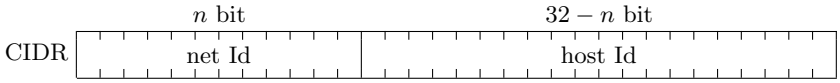

Tabella 2: Uno esempio di indirizzamento CIDR.

La suddivisione riportata in tab. 1 è largamente inefficiente in quanto se un utente necessita di anche solo un indirizzo in più dei 256[5] disponibili con una classe A occorre passare a una classe B, con un conseguente enorme spreco di numeri (si passerebbe da 256 a 65536).

Per questo nel 1992 è stato introdotto un indirizzamento senza classi (detto CIDR) in cui il limite fra i bit destinati a indicare il numero di rete e quello destinati a indicare l'host finale può essere piazzato in qualunque punto dei 32 bit totali (vedi tab. 2), permettendo così di accorpare più classi A su un'unica rete o suddividere una classe B. È stata così introdotta anche una nuova notazione, che permette di indicare la parte di rete appendendo all'indirizzo l'indicazione del numero di bit riservati alla rete; nel caso in esempio si avrebbe allora `192.168.111.11/24`.

Per concludere questa panoramica sugli indirizzi occorre accennare all'indirizzo di *broadcast* mostrato nello specchietto in tab. 3 in cui si è riassunta la struttura di un indirizzo IP. In ogni rete Internet infatti esiste un indirizzo riservato, che per convenzione è sempre ottenuto mettendo ad 1 tutti i bit della porzione dell'indirizzo riservata all'host, che serve ad inviare messaggi a tutti i computer presenti sulla rete.

Questo permette il funzionamento di una serie di protocolli ausiliari del TCP/IP che devono poter scambiare e ricevere informazione con tutti i computer presenti (ad esempio quelli che

[3] il *multicast* è progettato per la comunicazione simultanea verso più stazioni che si possono mettere in ascolto su uno di questi indirizzi.

[4] e questo alle volte può creare problemi, dato che non è detto che la rete in questione sia ancora classificabile in questo modo.

[5] in realtà gli indirizzi disponibili con una classe A sono 254, questo perché l'indirizzo `.0` è riservato per indicare la rete, mentre l'indirizzo `.255`, come vedremo fra poco, è quello di *broadcast*; questi indirizzi sono riservati per ogni rete e non possono essere usati per un singolo host.

Indirizzo	Esempio
Indirizzo completo	`192.168.111.11`
Maschera di rete	`255.255.255.0`
Porzione di rete	`192.168.111.`
Porzione dell'host	`.11`
Indirizzo di rete	`192.168.111.0`
Indirizzo broadcast	`192.168.111.255`

Tabella 3: Specchietto riassuntivo della struttura degli indirizzi IP.

permettono di scoprire quali sono gli indirizzi realmente attivi), senza doversi indirizzare a ciascuno di essi individualmente.

1.3 Il *routing*

Per capire il funzionamento di Internet continuiamo ad utilizzare la nostra analogia con i telefoni; quando chiamate qualcuno scrivendo il numero sul telefono fate in modo che si attivino una serie di connessioni (nelle varie centraline telefoniche che stanno fra voi e lui) che vi fanno entrare in contatto con il vostro interlocutore finale. Qualcosa di analogo avviene anche nel protocollo IP, che necessita di un lavoro di smistamento e reindirizzamento verso la loro destinazione finale dei pacchetti di dati che vengono trasmessi via rete; questo procedimento è quello che in termini tecnici viene chiamato *routing*.

L'argomento del *routing* di per sè è uno dei più complessi, per questo ci occuperemo solo della parte che ci riguarda più da vicino, e cioè quella dell'instradamento dei pacchetti che escono dal computer. Proseguendo con l'analogia telefonica si può pensare alla propria rete locale (al limite composta da un solo computer) come alla rete telefonica di una ditta; per poter uscire e telefonare all'esterno occorre in qualche modo passare dal centralino.

Questo è il significato del cosiddetto *default gateway*, questa è la macchina che nella vostra rete fa da ponte verso l'esterno. Tutte le telefonate dirette fuori (cioè tutti i pacchetti di dati che devono uscire dalla rete locale per andare su internet) devono passare da questa macchina; per questo quando installate una macchina in una rete locale dovete sempre sapere l'indirizzo del gateway.

La differenza coi numeri telefonici sta però nel fatto che questo non si può inserire all'interno del numero stesso (ad esempio mettendo un 0 o un 1 all'inizio dello stesso) ma deve essere proprio specificato l'indirizzo completo della macchina che fa da ponte (anch'essa avrà un suo numero IP).

Torneremo sull'argomento del *routing*, azzardando una spiegazione un po' più dettagliata di questi meccanismi, in sez. 3.5, quando prenderemo in esame il funzionamento del comando `traceroute`.

1.4 Il DNS

Proseguendo con la nostra analogia telefonica consideriamo l'agenda che sta sempre accanto al telefono. Ricordarsi a mente a quale sito, macchina o persona corrisponde un certo numero IP è assolutamente impraticabile; per questo occorre l'equivalente dell'agenda e dell'elenco telefonico che associa ogni numero ad un nome.

Questo è realizzato dal *Domain Name Service*, o DNS. Il DNS è un enorme database distribuito che associa ad un nome letterale (quello dei siti internet) un indirizzo IP. La struttura del funzionamento del DNS va al di la dello scopo di questa lezione, quello che è importante sapere è che questo servizio esiste e che è essenziale per il buon funzionamento della rete.

In effetti il DNS è più simile al servizio "12" che all'agenda telefonica (questa in realtà è realizzata dal file `/etc/hosts`, su cui torneremo in sez. 4.5), in quanto esso viene realizzato da

macchine che forniscono questo servizio. Esso ha anche il vantaggio (come il servizio 12 rispetto all'agenda) che viene aggiornato automaticamente qualora una associazione fra nome e numero IP dovesse cambiare.

Per questo motivo quando configurate la rete una delle informazioni che vi servono è l'indirizzo IP di una di queste macchine (ogni provider ne mette a disposizione una) che fa le veci del vostro "12" personale.

1.5 I servizi e le porte.

Finora abbiamo parlato quasi esclusivamente di IP; come accennato nell'introduzione (si ricordi quanto detto in sez. 1.1) questo è solo una parte dei protocolli di internet, ed copre soltanto il cosiddetto *livello di rete*. Per poter effetture delle comunicazioni in genere i programmi necessitano di creare delle connessioni, e per far questo si usano i protocolli del *livello di trasporto*, come TCP ed UDP.

Perciò occorre introdurre un'altra delle caratteristiche del protocollo TCP/IP, che riguarda solo il *livello di trasporto*, quella delle *porte*, che non è stata ancora presa in considerazione, ma che è essenziale per capire alcune caratteristiche generali senza le quali non si avrebbero le basi per andare avanti.

Anche in questo caso l'analogia telefonica ci viene, sia pure in maniera molto parziale, in aiuto. Finora infatti abbiamo parlato dei numeri IP come dei numeri di telefono, ma questo riguarda solo la parte del protocollo che viene usato per effettuare la trasmissione fra due computer, e cioè il protocollo IP.

Allora come su un numero di telefono può rispondere una persona (se solleva la cornetta), una segreteria telefonica, un fax, o un altro computer (se c'è attaccato un modem), lo stesso accade anche per internet; su un numero IP possono in realtà rispondere diversi *servizi* corrispondenti a forme di comunicazione diversa.

L'analogia usata è molto debole perché di solito per fare ognuno di questi compiti ci vogliono apparecchi diversi (anche se talvolta si trovano oggetti che assommano più di uno di essi). Per questo in realtà si potrebbe pensare alle *porte* come ai canali della filodiffusione,[6] cioè a delle specie di "frequenze" diverse su cui sintonizzate il vostro telefono, sulle quali trovate i contenuti più diversi.

In realtà non è neanche così, perché nel caso della filodiffusione il segnale non viene da un altro telefono, ma dal fornitore del servizio telefonico, potete solo ascoltare, ed un canale alla volta, mentre con internet potete sia ascoltare che trasmettere, da e verso qualunque altro telefono e su quanti canali volete[7] in contemporanea.

Questo avviene perché, come accennato, TCP/IP è un insieme di protocolli, ed IP (quello dei numeri) serve solo a gestire la trasmissione dei pacchetti attraverso una rete. Per poter effettuare uno scambio di dati occorre una modalità per stabilire una *connessione*, e per far questo occorre andare più in là di quanto si fa con IP, che serve solo ad inviare pacchetti da un computer all'altro, per introdurre ad esempio, dei meccanismi che garantiscano che i pacchetti arrivino davvero a destinazione, che non siano persi, modificati, ecc.

Per questo si usano i protocolli di livello superiore (quello di *trasporto*, si veda fig. 1) come UDP e TCP, i quali a loro volta introducono il concetto di *porta* per poter gestire la possibilità di avere più connessioni[8] in contemporanea, dedicate a servizi diversi, cioè allo scambio di dati specifici come la posta o il web, che vanno a costituire l'ultimo livello (quello di *applicazione*) della struttura mostrata in fig. 1.

[6] per chi non ha idea di che cosa sia, si tratta di una specie di radio via telefono, usata per trasmettere musica quando le radio avevano una pessima qualità, ma che oggi non esiste praticamente più.

[7] in realtà lo si può fare fino ad un numero massimo di 65535 porte, pari a $2^{16} - 1$.

[8] per UDP è più corretto parlare di canali di comunicazione in quanto non c'è una connessione.

Così quando si vuole inviare della posta elettronica si comunicherà attraverso una di queste porte, mentre quando si vuole leggere una pagina web se ne userà un'altra. Si tenga conto però che il concetto di porta è spesso fuorviante, in quanto con porta si intende una qualche forma di accesso permanente che può essere aperto o chiuso. In realtà non esiste nessuna forma di accesso permanente e lo scambio di dati avviene solo se si hanno da ambo le parti gli strumenti per effettuarlo[9] (il server ed il client); per questo sarebbe più chiaro parlare di frequenza, su cui si può trasmettere o ascoltare, ed in cui ciascuno può essere la trasmittente (il server) o il ricevente (il client) o anche entrambi allo stesso tempo (ad esempio nei sistemi *peer to peer*).

2 La configurazione della rete

La configurazione della rete è una materia alquanto complessa, sia per la difficoltà intrinseca del soggetto, sia perché, a causa della strutturazione vista in sez. 1, e dell'architettura modulare di Linux, sono molteplici i punti su cui si può dover intervenire.

2.1 Configurazione del kernel

In fig. 1 si è mostrato dove è situata la separazione fra kernel space e user space nei protocolli di rete. Il primo passo della configurazione è allora quello di prevedere il supporto dei protocolli necessari (per il trasporto, la rete e il collegamento) nel kernel.

In genere tutte le distribuzioni provvedono dei kernel standard che sono già predisposti a supportare tutto quello che serve nei casi più comuni. Nel caso il vostro hardware non sia supportato, o vi necessiti un protocollo non previsto nel kernel corrente vi occorrerà ricompilare il kernel[10].

Il procedimento per la compilazione del kernel è spiegato nella relativa lezione del corso (vedi sez. **??**). Le sezioni relativa alla rete sono due, *Networking options* e *Network device support*, come si può vedere dalla seguente schermata del menù di configurazione:

nella prima si possono attivare i protocolli di rete di alto livello (sopra il collegamento) necessari. Di solito per il TCP/IP tutto quello che serve è attivato di default, è necessario intervenire qui solo se si vuole supportare un altro protocollo (come IPX o Appletalk), o attivare alcune opzioni specialistiche (il cui scopo va al di là di quanto è possibile affrontare in questa sede).

Nella seconda parte si attivano invece i driver per le varie schede ed i protocolli di basso livello. In genere anche in questo caso la configurazione di default per la maggior parte delle distribuzioni fornisce il supporto per le condizioni di uso più comune; se però si ha una scheda di rete non supportata dal kernel di default può essere necessario accedere alla sottosezione *Ethernet (10 or 100Mbit)*[11] dove troverete le opzioni per una vasta scelta di schede.

In tal caso la difficoltà maggiore sarà quella di selezionare il modulo che corrisponde alla vostra scheda, cosa che potrete fare ricorrendo ad una lettura dell'Ethernet HOWTO che di solito si trova nella documentazione allegata con la vostra distribuzione o all'indirizzo del link precedente, nel quale è riportata una lunga lista di schede supportate.

2.2 La configurazione degli indirizzi.

Il primo passo per la configurazione della rete è quello della configurazione degli indirizzi. In generale questo tutte le distribuzioni provvedono dei tool per automatizzare questo processo,

[9] per questo non sarà mai possibile sfondare una "porta" sul vostro computer, se su di essa non c'è un server, così come non possono mandarvi offese sulla radio, se non siete in ascolto.

[10] nel caso di kernel modulare può bastare la compilazione dei moduli necessari

[11] a meno che non abbiate una gigabit, nel qual caso probabilmente potete pure permettervi un sistemista che può fare la configurazione per voi.

```
Linux Kernel v2.4.16 Configuration
--------------------------------------------------------------------------
+------------------------------- Main Menu -------------------------------+
|  Arrow keys navigate the menu.  <Enter> selects submenus --->.          |
|  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, |
|  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help.     |
|  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable     |
| +---------^(-)-------------------------------------------------------+ |
| |         Multi-device support (RAID and LVM)  --->                   | |
| |       * Networking options  --->                                     | |
| |         Telephony Support  --->                                      | |
| |         ATA/IDE/MFM/RLL support  --->                               | |
| |         SCSI support  --->                                           | |
| |         Fusion MPT device support  --->                              | |
| |         IEEE 1394 (FireWire) support (EXPERIMENTAL)  --->            | |
| |         I2O device support  --->                                     | |
| |       * Network device support  --->                                 | |
| +         Amateur Radio support  --->                                  | |
| +---------v(+)-------------------------------------------------------+ |
+--------------------------------------------------------------------------+
|                    <Select>    < Exit >    < Help >                     |
+--------------------------------------------------------------------------+
```

Figura 2: Schermata di configurazione per la compilazione del kernel; le opzioni di configurazione per il supporto di rete sono indicate con un asterisco

memorizzando i valori (che normalmente non cambiano) in opportuni file di configurazione (torneremo su questo in sez. 4) che vengono letti all'avvio.

Per questo motivo di solito la configurazione della rete viene effettuata una volta per tutte in fase di installazione, dove una opportuna applicazione vi richiederà tutte le informazioni necessarie. Di solito i casi che si presentano sono due:

- Un computer a casa, che si collega ad internet attraverso un provider (connessione via modem analogico, ISDN).
- Un computer connesso in rete locale (con una scheda di rete).

Nel primo caso vi verranno chiesti i dati del provider (numero telefonico, username e password del vostro account, modalità di autenticazione) ed eventualmente quelli relativi al vostro modem (anche se ormai tutte le distribuzioni sono in grado di eseguire il riconoscimento automatico), dopo di che sarà il programma di connessione che si preoccuperà di eseguire le relative operazioni per attivare la connessione.

In questo caso non dovete preoccuparvi del settaggio dell'indirizzo IP in quanto ci penserà il programma di connessione (che in genere è una qualche forma di frontend per `pppd`) a utilizzare opportunamente le informazioni che gli vengono fornite dal provider per eseguire le configurazioni opportune. Al più ci potrà essere da configurare a mano, per quei pochi provider che non forniscono l'informazione, il DNS (torneremo su questo in sez. 4.3).

Nel secondo caso invece è molto probabile che dobbiate eseguire voi il settaggio dell'IP chiedendo all'amministratore di fornirvi una serie di informazioni. In particolare vi occorrerà il numero IP da assegnare alla vostra macchina, la netmask, e l'indirizzo del gateway (oltre alla solita informazione sul DNS).

Questo è un passo necessario anche se volete creare la vostra piccola rete casalinga, in questo caso l'amministratore di rete siete voi, e i numeri li dovete decidere da soli; vi consiglio caldamente di usare le varie classi riservate per le reti locali come i numeri `192.168.x.x` e `10.x.x.x`, che la IANA ha destinato appositamente a questo uso e che non vengono mai usati per macchine pubbliche su Internet.[12]

Quasi tutte le distribuzioni hanno dei programmi per configurare la rete locale,[13] che permettono di settare questi valori in maniera semplice, in genere attraverso una interfaccia a finestre e campi (che può essere grafica o testuale). I comandi più comuni sono riportati in tab. 4, ed in

Distribuzione	**Comando**
Debian	`dpkg-reconfigure etherconf`
RedHat	`netcfg`
Slackware	`/sbin/netconfig`

Tabella 4: Comandi di configurazione della rete

genere si tratta di soltanto di specificare i valori richiesti.

2.3 La configurazione dei servizi

Come accennato nella lezione sulla struttura del sistema in GNU/Linux tutti i servizi sono forniti tramite appositi programmi. Per questo motivo ciascun servizio in genere va configurato singolarmente agendo sul relativo file di configurazione.

In un corso base non è possibile affrontare l'argomento della configurazione dei servizi che un sistema GNU/Linux può offrire; alcuni di essi (come la posta elettronica o il web) infatti, per la loro stessa complessità, meriterebbero un corso monografico.

Inoltre qualunque distribuzione quando installa uno di questi servizi si preoccupa di effettuare una configurazione di base che di solito è in grado di rispondere alle normali esigenze dell'utente (ad esempio ogni distribuzione installa e configura il demone di trasmissione della posta elettronica, che di norma è necessario alla gestione della posta inviata localmente).

Tutto quello che tratteremo qui perciò sarà solo la configurazione di alcuni servizi elementari, quelli che in genere vengono avviati attraverso l'intermediazione del cosiddetto *superdemone*.

3 La configurazione manuale della rete

In questa sezione tratteremo la configurazione manuale della rete. In particolare vedremo come è possibile far ripartire la rete in caso di necessità senza dover ricorrere all'uso degli script di avvio; questo può risultare utile in tutti quei casi in cui si vuole mettere su una configurazione provvisoria senza toccare quella standard funzionante.

3.1 Il comando `ifconfig`

Il primo passo per tirare su la rete è quella di assegnare ad una interfaccia di rete il suo numero IP. Questo corrisponde a mettere il numero sul vostro telefono. Per fortuna non vi serve Telecom (a meno che non la usiate come provider), ma basta diventare root.

[12] se con le vostre macchine non accedete mai ad internet potreste anche pensare di usare altri indirizzi; questo purtroppo è un errore sciocco, che va interamente a vostro scapito. Una volta infatti che uno di questi computer dovesse accedere ad internet, ad esempio attraverso un modem, automaticamente i siti che hanno i numeri che voi avete assegnato alle vostre macchine sarebbero irraggiungibili.

[13] nel caso di Debian non c'è un programma specifico, ma si può usare il pacchetto etherconf, che usa il sistema standard di `debconf` per effettuare la riconfigurazione.

Il vostro telefono corrisponde sul computer alla interfaccia di rete; perché questa funzioni dovrete avere attivato nel kernel il supporto per la relativa scheda secondo la procedura descritta in sez. 2.1. Il comando che vi consente di fare questo è `ifconfig` che permette di settare la varie caratteristiche delle interfacce di rete. Tutto quello che serve nella maggior parte dei casi sono soltanto le opzioni che permettono di *tirare su* e *tirare giù*[14] una interfaccia.

Se usato senza opzioni il comando elenca tutte le interfacce attive; questo è il primo passo da fare sempre prima di qualunque configurazione (ed anche dopo per controllare che sia tutto a posto) per vedere lo stato del sistema. Un risultato possibile è il seguente:

```
[root@havnor root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:01:02:2F:BC:40
          inet addr:194.177.127.234  Bcast:194.177.127.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:82075264 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51585638 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:2858378779 (2.6 GiB)  TX bytes:2524425895 (2.3 GiB)
          Interrupt:10 Base address:0x8800

eth0:0    Link encap:Ethernet  HWaddr 00:01:02:2F:BC:40
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:10 Base address:0x8800

eth1      Link encap:Ethernet  HWaddr 00:E0:7D:81:9C:08
          inet addr:192.168.168.1  Bcast:192.168.168.255  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
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:9 Base address:0x6000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10226970 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10226970 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1385547296 (1.2 GiB)  TX bytes:1385547296 (1.2 GiB)
```

come si vede sulla macchina sono presenti 3 interfacce di rete; due di esse (*eth0* e *eth1*) corrispondono a due schede di rete, la terza (*lo*) è una interfaccia logica, la cosiddetta interfaccia di *loopback* che viene usata per le comunicazioni locali, e che deve essere sempre attivata anche per i computer non connessi in rete.

Si noti anche come la prima interfaccia sia *multihomed*; abbia cioè assegnati due indirizzi diversi, (identificati da *eth0* ed *eth0:1*), cosa che è sempre possibile fare quando si sia abilitato il relativo supporto[15] nel kernel.

Si può notare inoltre come il comando ci riporti le varie caratteristiche delle interfacce, come gli IP ad esse associati (scritti nella linea `inet addr:`), lo stato della stessa (nella riga successiva)

[14] espressioni gergali per dire attivare e disattivare.
[15] l'opzione è indicata come *IP aliasing*.

e tutta una serie di altre informazioni i cui dettagli si possono trovare nella pagina di manuale del comando.

Prima di poter configurare una interfaccia occorre verificare che essa non sia già attiva. Supponiamo che si tratti di `eth0`. Il comando `ifconfig eth0` ci mostrerà lo stato dell'interfaccia (dando un errore nel caso il supporto nel kernel non sia attivato), qualora essa sia già attiva e debba essere riconfigurata può essere disattivata con il comando:

```
[root@havnor root]# ifconfig eth0 down
```

dopo di che si può assegnarle un indirizzo ed attivarla in un colpo solo con il comando:

```
[root@havnor root]# ifconfig eth0 192.168.1.100
```

in cui è sottintesa l'opzione `up`.

Si ricordi che ad ogni indirizzo è sempre associata una rete, nel caso specifico però essa sembra non comparire; questo è dovuto al fatto che se non viene specificato esplicitamente il comando appena visto assegna automaticamente all'interfaccia anche una netmask corrispondente all'indirizzo di classe C utilizzato (nel caso sarebbe pari a 255.255.255.0). In realtà la rete su cui si allaccia l'interfaccia può essere specificata esplicitamente indicando esplicitamente la netmask con una opzione del tipo:

```
[root@havnor root]# ifconfig eth0 192.168.1.100 netmask 255.255.0.0
```

oltre a queste ci sono poi molte altre opzioni possibili, per le quali rimandiamo alla lettura della man page.

3.2 Il comando `route`

Avere attivato l'interfaccia ed averle assegnato un numero di IP è solo il primo passo, perché sia possibile utilizzare la rete occorre anche settare *l'instradamento* (in sostanza attaccare un telefono alla presa non basta se poi non viene registrato sulla centralina di smistamento). Anche in questo caso non vi serve la Telecom, ma il comando `route`, che si chiama così proprio perché serve a specificare la strada che i pacchetti devono prendere per arrivare a destinazione.

In sostanza quello che dovete fare è dire al computer a chi deve rivolgersi quando vuole parlare con qualcuno (come qual'è il numero del centralino, quali numeri sono diretti, ecc.). Se invocato senza parametri il comando vi mostra lo stato attuale delle *tabelle di instradamento*. Ad esempio:

```
[root@havnor root]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
194.177.127.0   0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.168.0   0.0.0.0         255.255.255.0   U     0      0        0 eth1
0.0.0.0         194.177.127.1   0.0.0.0         UG    0      0        0 eth0
```

(si è specificata l'opzione `-n` per avere un output con i valori numerici). L'output del comando mostra le strade disponibili; la prima colonna identifica la destinazione, la seconda il gateway (cioè il centralino per quella destinazione), la terza la sottorete coperta dalla destinazione, l'ultima l'interfaccia usata per l'invio dei pacchetti (il significato delle altre colonne è più esoterico e vi lascio alla spiegazione della pagina del manuale).

In questo caso si può notare come ci siano tre diverse destinazioni associate a tre diverse sottoreti, di cui due fanno capo alla stessa interfaccia (quella che in sez. 3.1 abbiamo visto essere multihomed). Per tutte queste, essendo le strade accessibili direttamente da una interfaccia

locale, non esiste gateway (è come per le telefonate all'interno dell'ufficio, non c'è bisogno di usare il centralino) e questo è indicato dall'indirizzo nullo in seconda colonna.

Qualora l'accesso fosse condizionato al passaggio da una macchina specifica (come nel caso l'accesso alla sottorete 192.168.168.0 da parte di una macchina che sta sulla LAN attaccata ad eth0) si sarebbe dovuto specificare un gateway. L'ultima riga indica invece quello che è il default gateway, cioè l'indirizzo cui devono essere inviati i pacchetti che non hanno una strada specificata altrimenti; in tal caso la destinazione è indicata dall'indirizzo nullo (che fa le veci della wildcard).

In realtà se avete una sola interfaccia e non uscite dalla rete locale non c'è bisogno di chiamare esplicitamente `route`; infatti tutte le volte che tirate su una interfaccia la strada per la rete a cui l'indirizzo è associato viene inserita automaticamente.

Per questo in generale quello che si deve fare in più (a meno di non avere esigenze particolari) è di specificare qual'è il default gateway (cioè l'indirizzo che si usa per uscire in internet); questo si fa con il comando:

```
[root@havnor root]# route add default gw 194.177.127.1
```

(ovviamente dato da root, altrimenti non funziona).

Si tenga presente che, come nel caso precedente in cui il default gateway è specificato per qualunque indirizzo, la tabella di instradamento può contenere più strade per la stessa destinazione. In genere il kernel le riordina, eseguendo il controllo su quale direzione far prendere ad un pacchetto, a partire dalla strada più specifica, ed esegue l'instradamento non appena trova una strada valida. Così nel caso dell'esempio precedente i pacchetti destinati all'indirizzo 194.177.127.243 non arrivaranno mai al gateway, ma saranno mandati immediatamente sulla scheda di rete corrispondente a `eth0`.

3.3 Il comando `ping`

Una volta settate le interfacce il primo controllo da effettuare per vedere se la rete funziona è "*pingare*" un'altra macchina.

Il comando `ping` serve appunto ad inviare un pacchetto speciale (per i curiosi si chiama ICMP Echo Request) che chiede un risposta alla macchina di destinazione. É un po' come telefonare per sentire se dà il libero. Il comando si invoca specificando l'IP della macchina bersaglio,[16] altrimenti deve anche funzionare il DNS, e se la rete non va per qualche altro motivo che non vi consente di raggiungere quest'ultimo non otterreste nulla anche se magari il resto è a posto.

Un esempio di uso è il seguente:

```
[piccardi@havnor corso]$ ping 192.168.168.20
PING 192.168.168.20 (192.168.168.20): 56 data bytes
64 bytes from 192.168.168.20: icmp_seq=0 ttl=255 time=0.7 ms
64 bytes from 192.168.168.20: icmp_seq=1 ttl=255 time=0.3 ms
64 bytes from 192.168.168.20: icmp_seq=2 ttl=255 time=0.3 ms

--- 192.168.168.20 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.7 ms
```

Il comando invia un pacchetto al secondo, e nel caso riceve sempre risposta riportando il tempo che ci è voluto. Quando lo fermate (con `C-c`) vi stampa anche una statistica. Questo ci dice che la macchina con IP 192.168.168.20 è attiva ed è raggiungibile.

[16] anche se in realtà va bene pure il nome, ma se lo usate esso deve essere scritto in `/etc/hosts` (vedi sez. 4.5).

Si tenga presente che questo è un controllo preliminare, e se le cose non vanno può dipendere da molti fattori (incluso il fatto che la macchina che "*pingate*" può essere spenta), ma se vanno potete se non altro concludere che la rete è attiva e funzionante (ed evitare di mettervi a controllare se avete attaccato la spina).

3.4 Il comando `traceroute`

Un altro comando che permette di controllare il collegamento è `traceroute`, che come dice il nome serve a tracciare la strada che fanno i pacchetti per arrivare alla destinazione indicata.

In questo caso l'analogia telefonica non ci aiuta, perché di solito con il telefono questo tipo di controllo non può essere fatto. Infatti quello che succede con Internet è che, in realtà non esiste una connessione diretta fra due computer, anche se poi i programmi usano delle funzionalità che permettono di lavorare come se le cose fossero effettivamente così, ma i dati che inviate vengono passati da un computer all'altro fino ad arrivare alla loro destinazione, e viceversa.

Allora un'analogia che può spiegare un po' meglio la cosa, ed illustrare un po' più chiaramente i concetti del *routing* è più che quella delle reti telefoniche è quella delle reti autostradali. Ad esempio lavoravo per l'università capitava spesso di dover andare al CERN (a Ginevra). Per farlo si prendeva l'autostrada a Firenze Sud, a Firenze Nord si cambiava sulla Firenze Mare, uscendo a Lucca per fare il raccordo per prendere l'autostrada per Genova, da Genova si proseguiva per Alessandria, li si cambiava di nuovo per Torino, dove fatta la circonvallazione si prendeva l'autostrada per il traforo del Monte Bianco. Da li il raccordo ci portava sull'autostrada per Ginevra.

Come vedete si tratta di un bel percorso complicato, che comporta di entrare e uscire da diversi caselli; ora quando inviate un pacchetto su internet succede qualcosa di simile, e anche lui deve passare attraverso dei "caselli". Quello che succede ad esempio quando vi collegate con un modem e iniziate a chattare con qualcun'altro, è che i pacchetti che escono dal vostro computer vengono inviati al *router* (l'equivalente del casello) del vostro provider; da li prenderanno la strada opportuna per arrivare al router del provider a cui è collegato il computer del vostro interlocutore.

In tutto questo percorso i pacchetti passeranno per una serie di altri *router* che sanno che strada devono prendere i pacchetti per poter arrivare alla destinazione finale. La differenza fra i caselli ed i router è che questi ultimi sanno trovare da soli la strada su cui ti devono mandare per farti arrivare a destinazione. In realtà sono ancora più intelligenti, e sono in grado di farti prendere la strada più veloce, tenendo conto di eventuali ingorghi, incidenti ecc. Così se il tunnel del Monte Bianco viene chiuso, quando arrivate a Torino il router vi farà dirottare per il Frejus.

Usando un accorgimento previsto dal protocollo TCP/IP il comando `traceroute` si fa mandare un messaggio di ritorno da ciascuno dei router attraverso cui passa un pacchetto per arrivare alla destinazione che avete indicato, in questo modo si può avere tracciata tutta la strada che fa. Un esempio del funzionamento di `traceroute` è il seguente:

```
 1  gw5a-65.wind.it (212.245.127.230)  125.017 ms  108.475 ms  100.188 ms
 2  c-fi1-fe5a.wind.it (212.245.96.1)  99.628 ms  89.136 ms  100.163 ms
 3  c-rm6-fi1-pos.wind.it (212.245.248.81)  110.412 ms  109.086 ms  110.687 ms
 4  c-mix2-rm6-pos.wind.it (212.245.250.30)  119.925 ms  111.059 ms  108.968 ms
 5  inet-mix.mix-it.net (217.29.66.2)  119.797 ms  110.656 ms  119.937 ms
 6  ge0-0-0.milano1-cr10.net.inet.it (194.185.46.75)  119.939 ms  119.880 ms  110.663 ms
 7  s1-1.firenze1-ar1.net.inet.it (194.185.64.238)  129.952 ms  125.473 ms s2-0.firenze1-ar1.n
 8  e0.firenze1-r4.net.inet.it (194.185.128.104)  119.915 ms  119.684 ms  130.637 ms
 9  s1-4.gw-chlcpn2.inet.it (194.185.66.230)  129.236 ms !A s1-3.gw-chlcpn.inet.it (194.185.66
```

Così se per un qualche motivo non riuscite a raggiungere il vostro indirizzo di destinazione potete verificare se questo è dovuto al fatto che la strada che prendono i vostri pacchetti è interrotta da qualche parte.[17]

3.5 Il comando `host`

Il comando `host` permette di interrogare il DNS per avere l'associazione fra un indirizzo numerico ed uno simbolico. In genere è utile per verificare che il DNS funzioni davvero. Accade spesso infatti che la rete funzioni, ma non si riesca a fare nulla perché gli indirizzi simbolici non vengono risolti; un esempio tipico di questo comando è il seguente:

```
[piccardi@havnor piccardi]$ host firenze.linux.it
firenze.linux.it        A       195.110.99.218
```

questo ci dice appunto che l'indirizzo `firenze.linux.it`, il dominio cioè associato al FLUG, è risolto come corrispondente all'IP 195.110.99.218; il comando funziona anche alla rovescia, si può cioè trovare l'indirizzo simbolico a partire da quello numerico con:

```
[piccardi@havnor piccardi]$ host 195.110.99.218
Name: serverone.firenze.linux.it
Address: 195.110.99.218
```

e si noti come in questo caso il nome riportato sia `serverone.firenze.linux.it`, che è diverso dal precedente. Questo è normale in quanto ad un solo IP possono essere associati diversi nomi; ad esempio anche interrogando con:

```
[piccardi@havnor piccardi]$ host www.softwarelibero.it
www.softwarelibero.it        A       195.110.99.218
```

si ottiene di nuovo lo stesso numero, dato che la macchina che ospita il sito è la stessa.

3.6 Il comando `netstat`

Questo è un comando diagnostico molto utile ed estremamente complesso, che permette di visualizzare una grande quantità di informazioni relative alla rete. La trattazione di tutte le sue capacità comporterebbe un approfondimento dei concetti relativi alle reti che va al di là delle possibilità di un corso introduttivo.

Qui ne tratteremo solo un caso specifico, quello che permette di visualizzare tutte le connessioni attive sulla macchina (un po' come potrebbe fare il quadro di un centralino che mostra tutti i collegamenti in corso). Per questo è anche uno dei primi comandi che un eventuale intruso che si sia introdotto nella vostra macchina cercherà di sostituire con una sua versione "taroccata" perché può mostrare eventuali collegamenti "indesiderati".

Anche per capire questi risultati è comunque necessario di un minimo di conoscenza dei protocolli e dei servizi: è facile trovare persone che allarmano perché hanno "un collegamento strano sulla porta 25", e in realtà stanno semplicemente inviando la posta che hanno appena finito di scrivere.

Il comando usato con le opzioni di default mostra le informazioni riguardo a tutti i *socket* aperti; anche i collegamenti interni al sistema (usati da vari programmi per scambiarsi i dati con l'interfaccia dei *socket*) che di norma non han nulla a che fare con la rete. Per questo motivo è d'uopo specificare le opzioni `-t` per richiedere di visualizzare solo i *socket* TCP o `-u` per vedere quelli UDP, che sono quelli che riguardano le connessioni con la rete esterna.

Un possibile esempio del risultato di `netstat` è il seguente:

[17] sempre che qualcuno, preso da eccesso di zelo, non si sia messo a filtrare anche i pacchetti di controllo usati dal protocollo.

```
[piccardi@gont piccardi]$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:printer               *:*                     LISTEN
tcp        0      0 *:5865                  *:*                     LISTEN
tcp        0      0 *:webcache              *:*                     LISTEN
tcp        0      0 *:tproxy                *:*                     LISTEN
tcp        0      0 gont.earthsea.ea:domain *:*                     LISTEN
tcp        0      0 localhost:domain        *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 *:ipp                   *:*                     LISTEN
tcp        0      0 *:nntp                  *:*                     LISTEN
tcp        0      0 *:smtp                  *:*                     LISTEN
tcp        0      0 ppp-42-241-98-62.:32798 serverone.firenze:imaps ESTABLISHED
```

Il comando riporta una tabella con le indicazioni relative a ciascuna connessione. Il campo *Proto* riporta il protocollo della connessione. I campi *Local Address* e *Foreign Address* indicano gli indirizzi locale e remoto della stessa (che può essere stampato in forma numerica anziché usando lo switch `-n`), nella forma:

```
indirizzo:porta
```

dove un asterisco indica un indirizzo o una porta qualunque. Il campo *State* indica lo stato della connessione. Una spiegazione dettagliata del significato dei vari campi va di nuovo al di là delle possibilità di un corso introduttivo (specie per il campo *State*), per gli interessati mi limito a segnalare il capitolo *Socket TCP elementari* GaPiL, in cui questi concetti vengono spiegati in dettaglio.

Delle varie righe quelle che meritano attenzione sono quelle relative agli stati `LISTEN` ed `ESTABLISHED`. Lo stato `LISTEN` indica la presenza di un programma in ascolto sulla vostra macchina in attesa di connessione, nel caso ce ne sono vari corrispondenti a servizi come la posta, le news, il dns, la stampa via rete, gli indirizzi sono di norma non specificati in quanto la connessione può essere effettuata su uno qualunque degli indirizzi locali, da un qualunque indirizzo esterno. Lo stato `ESTABLISHED` indica le connessioni stabilite ed attive, e riporta nei campi degli indirizzi i due capi della connessione. Altri stati che possono essere riportati sono `FIN_WAIT`, `TIME_WAIT`, e si riferiscono a connessioni che si stanno chiudendo.

4 I principali file di configurazione

Come tutti gli altri file di configurazione anche quelli che riguardano la rete sono tenuti in `/etc`. In questa sezione vedremo quali sono i principali file che riguardano direttamente o indirettamente la configurazione della rete.

4.1 I file di configurazione delle interfacce statiche.

In generale tutte le distribuzioni avviano la rete all'interno di opportuni script di avvio, uno specchietto è riportato in tab. 5. In genere questi script non fanno altro che andare a leggere degli opportuni file di testo che contengono le informazioni necessarie (quante interfacce ci sono, quali IP devono essere assegnati, qual'è il default gateway, ecc.) ed eseguono tutti i comandi necessari.

Sul come fare per avviare la rete manualmente abbiamo già discusso in sez. 3, se però vogliamo effettuare un settaggio permanente dobbiamo andare a modificare i file in cui le informazioni sono memorizzate.

Distribuzione	Comando
Debian	`/etc/init.d/networking`
RedHat	`/etc/rc.d/init.d/network`
Slackware	`/etc/rc.d/rc.inet1`
Suse	`/etc/rc.d/network`

Tabella 5: Script di avvio della rete nelle varie distribuzioni

Questi variano da distribuzione a distribuzione, così come può essere diverso il loro formato; prenderemo in esame due dei casi più comuni, Debian e RedHat (Mandrake usa gli stessi file). In genere i programmi di configurazione automatica (o i vari programmi grafici per la configurazione) non fanno altro che leggere e modificare i valori che stanno su questi file; farlo a mano può essere comodo quando non avete la grafica a disposizione (ad esempio siete collegati con un modem).

Debian Il file di configurazione delle interfacce[18] è `/etc/network/interfaces`, il cui formato è descritto in dettaglio dalla omonima pagina di manuale. Per l'uso normale è sufficiente specificare i dati con un contenuto del tipo:

```
auto lo eth0
iface lo inet loopback
iface eth0 inet static
        address 194.177.127.234
        netmask 255.255.255.0
        gateway 194.177.127.1
```

la prima riga, introdotta dalla parola chiave `auto`, dice quali sono le interfacce attivare automaticamente all'avvio del sistema, che possono essere specificate sia insieme, come nell'esempio, che in altrettante righe distinte.

Le due righe seguenti, introdotte dalla parola chiave `iface`, servono a settare i parametri di ciascuna interfaccia. Per ciascuna interfaccia deve essere fornita una riga in cui specificarne il nome, la famiglia di protocolli usata,[19] e le modalità della stessa; esse sono sostanzialmente tre:

`loopback` si usa per l'interfaccia di loopback.

`dhcp` usa un server DHCP

`static` assegna i valori secondo i parametri specificati nelle righe seguenti.

e se si specifica `static` si possono settare i relativi parametri nelle righe seguenti, (che di solito si indentano, per maggiore chiarezza), come nell'esempio.

Si tenga presente che in generale le interfacce temporanee relative all'uso di `ppp` non vengono mai menzionate in questo file, ma vanno configurate a parte (vedi sez. 5.3).

RedHat i file di configurazione sono dentro `/etc/sysconfig/`, e ce n'è uno per interfaccia

[18] in realtà si tratta del file di configurazione usato dai comandi `ifup` e `ifdown` che sono quelli usati da Debian per gestire attivazione e disattivazione delle interfacce.

[19] `inet` indica l'usuale TCP/IP, ma sono possibili anche `ipx` per IPX, e `inet6` per IPv6.

4.2 Il file `modules`

Abbiamo già incontrato questo file in sez. **??** nella lezione sui file di configurazione e per il suo significato vi rimando a quella lezione. Qui mi preme sottolineare che è possibile usare questo file, quando si ha un kernel con il supporto per le schede di rete modulare, per assegnare in maniera univoca a quale interfaccia viene associata una certa scheda; in particolare una sezione del file tipo:

```
alias eth0 3c59x
alias eth1 eepro100
```

dice di usare la 3Com Vortex per la prima interfaccia e la Intel EtherExpress per seconda.

4.3 Il file `resolv.conf`

È qui che vengono memorizzate le informazioni principali relative alla risoluzione dei nomi, come il dominio locale e l'IP del nameserver che vengono di solito indicati in fase di installazione o settaggio iniziale della rete. In sostanza è in questo file che dovete dire qual'è il numero di telefono del servizio "12" a cui vi rivolgete (di solito ogni provider ha il suo).

Il file in realtà controlla il comportamento delle funzioni del cosiddetto *resolver*, quella parte delle librerie di sistema che esegue la risoluzione dei nomi, e che vengono usate da tutti i programmi.

Di norma contiene il nome del dominio e la lista degli IP dei nameserver; il formato, documentato in `man resolv.conf` è molto semplice, una serie di righe nella forma *keyword valore*; le righe vuote e quelle che iniziano per `#` vengono ignorate.

Un possibile esempio è il seguente:

```
nameserver 127.0.0.1
search firenze.linux.it
nameserver 195.110.99.218
```

La keyword `nameserver` serve ad indicare la specificazione del numero IP di un nameserver, la risoluzione di un nome verrà eseguita nell'ordine in cui esse sono state specificate; normalmente si possono specificare fino ad un massimo di 3 diversi nameserver.

La keyword `search` specifica quale è il nome del dominio locale, in questo modo si può specificare solo l'hostname del computer, e la ricerca assumerà che esso si trovi nella lista dei domini qui elencati.

Di solito i tool di connessione che si usano per collegarsi ad internet da casa ottengono dal provider questo numero durante la fase di negoziazione del collegamento, e riscrivono al volo questo file. Vi può capitare di volerci aggiungere qualche altro DNS (ce ne sono di pubblici) qualora quello del vostro provider avesse problemi.

4.4 Il file `services`

Questo è il file che contiene le corrispondenze fra nomi simbolici e numeri delle porte assegnate ai vari servizi (si ricordi quanto spiegato in sez. 1.5). Di norma non è un file che sia necessario modificare, ma è utile per avere un riferimento che permette di associare un numero di porta ad un determinato servizio.

Bussando a quella porta (o sintonizzandosi su quella frequenza a seconda dell'analogia che si preferisce) si potranno scambiare, attraverso l'opportuno protocollo, i dati relativi al servizio associato. In realtà dal punto di vista del TCP/IP si potrebbe usare un numero di porta qualsiasi, ma la standardizzazione ha portato ad associare alcuni numeri a dei servizi specifici (la porta 25 alla posta elettronica, la porta 80 al web, ecc.).

In genere in Unix le prime 1024 porte sono dette *riservate* in quanto solo l'amministratore può installarci sopra dei servizi; la corrispondenza fra queste porte ed i servizi che ci devono essere installati è regolata a livello internazionale: nessuno vi obbliga a rispettare la convenzione, ma se mettete la posta elettronica sulla porta 80 e il web sulla 25 avrete certamente delle grosse difficoltà a comunicare con gli altri, dato che in genere i browser cercano i siti sulla porta 80, ed i programmi di posta la mandano sulla 25.

Al di sopra della porta 1024 qualunque utente può mettere un suo servizio, alcuni però sono stati usati tradizionalmente da alcuni servizi, ed il file `/etc/services` tiene conto anche di questi. Esso contiene un elenco di numeri a ciascuno dei quali è associato un nome simbolico che individua il servizio ad esso associato dalle convenzioni internazionali. Un estratto del file è:

```
...
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
whois           43/tcp          nicname
re-mail-ck      50/tcp                          # Remote Mail Checking Protocol
re-mail-ck      50/udp                          # Remote Mail Checking Protocol
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
mtp             57/tcp                          # deprecated
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp                          # BOOTP client
bootpc          68/udp
tftp            69/udp
gopher          70/tcp                          # Internet Gopher
gopher          70/udp
rje             77/tcp          netrjs
finger          79/tcp
www             80/tcp          http            # WorldWideWeb HTTP
www             80/udp                          # HyperText Transfer Protocol
...
```

Al solito, righe vuote e tutto quello che segue un `#` viene ignorato; ogni riga ha il formato:

```
nome            numero/protocollo    alias
```

dove `nome` é l'identificativo simbolico del servizio, `numero` è il numero di porta ad esso assegnato, `protocollo` indica se si tratta di UDP o TCP, e `alias` è la lista di eventuali altri nomi associati allo stesso servizio.

È guardando in questo file che vari programmi attinenti alla rete (ad esempio `netstat`) che riportano o richiedono un numero di porta per un servizio possono utilizzare il nome simbolico di quest'ultimo (www, ftp, telnet), qui specificato, invece che il numero.

4.5 Il file `hosts`

Questo file contiene un elenco di nomi e dei relativi numero IP. Lo si usa come agenda del telefono per specificare gli indirizzi delle macchine per le quali si ha una mappa *statica* degli indirizzi (ad esempio le macchine di una rete privata che non vanno su internet, ma che volete risolvere col nome che gli avete assegnato).

Il formato del file è il seguente: le righe vuote od inizianti per `#` sono ignorate, le altre righe devono contenere il numero di IP, il nome completo ed un nome breve.

Un possibile esempio è il seguente:

```
# private nets
192.168.168.20  atuan.centrohl.it       atuan
192.168.168.10  oppish.centrohl.it      oppish
```

4.6 Il file `hosts.conf`

Questo file controlla le modalità di funzionamento delle routine che eseguono la risoluzione dei nomi (il *resolver*). Al solito le righe vuote od inizianti per `#` sono ignorate, le altre devono contenere una parola chiave seguita da un valore.

Un possibile esempio del contenuto di questo file è il seguente:

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

la prima parola chiave controlla la sequenza in cui viene effettuata la risoluzione dei nomi, e dice che deve prima essere usato il file `hosts` di sez. 4.5, e poi le interrogazioni ai DNS esterni. La seconda permette di avere indietro tutti gli indirizzi validi di un host che compare in `hosts` invece di avere solo il primo.

Altre parole chiavi valide si trovano, insieme alle relative spiegazione nella pagina di manuale del file.

4.7 I file `hosts.allow` e `hosts.deny`

Questi due file controllano i cosiddetti *tcp wrappers*, un insieme di funzioni che permettono di controllare come dall'esterno si può accedere al vostro computer (una specie di filtro telefonico che impedisce che ci possano chiamare da certi numeri, o a certe ore).

Questi file permettono, per i programmi che sono stati predisposti per usarli, di specificare un controllo degli accessi che permetta di collegarsi a certi servizi solo da parte di certi host, aumentando quindi la sicurezza ad essi relativa. Il loro formato è riportato nella pagina di manuale accessibile con `man 5 hosts_access`.

Il primo file, come suggerisce il nome, elenca le regole che negano l'accesso, il secondo quelle che lo consentono. Si tenga presente che il funzionamento dei *tcp wrappers* è tale che prima viene controllato `hosts.allow`, e se una regola corrisponde l'accesso è garantito ed il controllo è finito, altrimenti viene controllato `hosts.deny` e se una regola corrisponde l'accesso è negato; di default l'accesso è garantito.

In genere allora quello che si fa è negare tutti gli accessi in `hosts.deny` e consentire solo quelli voluti in `hosts.allow`. Per questo l'esempio tipico di `hosts.deny` è il seguente:

```
ALL: ALL
```

La sintassi delle regole si intravede già in questo esempio; al solito righe vuote e tutto quello che segue un `#` viene ignorato, ogni riga poi ha la forma:

```
lista dei server: lista dei client : comando shell
```

dove la terza parte (`: comando shell`) è opzionale e può essere omessa.

Un ulteriore esempio, più significativo del precedente, che consente l'accesso ad `ssh`, ma blocca tutto il resto eccetto in locale, è quello del contenuto di `hosts.allow` riportato di seguito:

```
sshd:     ALL
portmap:  127.0.0.1
leafnode: 127.0.0.1
```

In generale per lista dei server si intende il nome (o i nomi, se se ne vuole indicare più di uno) del programma che gestisce lo specifico servizio. Occorre fare attenzione, perché il nome è quello del programma che fornisce il servizio (ad esempio `in.telnetd`), non quello del servizio.

Nell'esempio indicato si sono indicati tre servizi, il primo è la *secure shell*, che permetti un collegamento sicuro da remoto, che è gestita come server dal programma `sshd`; il secondo è un servizio di sistema, il terzo è un programma che permette di tenere un servere di news in locale (a scopo di caching).

Nella lista dei client si indicano invece gli IP o le reti a cui si vuole consentire l'accesso. Per le reti in genere si usa la notazione sia numerica che alfabetica, e si può usare una `*` come wildcard per raggruppare indirizzi; si può anche specificare una netmask con un indirizzo de tipo:

```
131.155.72.0/255.255.254.0
```

e specificare la lista degli indirizzi usando un file dando il pathname completo.

5 Il collegamento da casa

In questa ultima parte cercheremo di prendere in esame le modalità con cui si può effettuare il collegamento ad internet dal computer di casa, approfondendo la modalità più comune, cioè quella dell'uso del modem analogico.

In generale, come accennato in sez. 2.2, tutte le distribuzioni provvedono dei tool opportuni per configurare la vostra connessione; ad esempio potete usare `wvdial`, `kppp`, `pppconfig`[20], `RP3` (questo è di Redhat), `isdnconfig` (se avete una ISDN) o il programma che la vostra distribuzione vi ha fornito all'oupo.

In questo caso tutto il lavoro consisterà nel provvedere i dati richiesti: di solito si tratta del numero di telefono del provider, nome dell'account e relativa password, che vi dovrete essere procurati al momento in cui avete richiesto l'abbonamento. Il programma si occuperà di settare opportunamente tutto il dovuto.

Nel resto di questa sezione tratteremo invece i concetti base per poter effettuare a mano questa configurazione, per chi è intenzionato a capire un po' più nel dettaglio il funzionamento della questione.

5.1 Il settaggio del modem

In generale il settaggio del modem è il passo più complicato della faccenda, superato il quale tutto il resto è abbastanza semplice. Anzitutto occorre distinguere fra i vari tipi di modem, che possono essere analogici, ISDN, o anche ADSL.

Fare una disamina accurata di tutte queste possibilità non è possibile, dato che la problematica è estremamente vasta, e anche nella documentazione che di solito trovate allegata alla vostra distribuzione è coperta da vari HOWTO, a cui vi rimando. Mi limiterò a trattare, e comunque in maniera superficiale, solo il caso dei modem analogici, che sono poi i modem propriamente detti.

[20] questo è specifico di Debian.

Il modem, il cui nome è un'abbreviazione di *modulatore-demodulatore* è quell'apparecchio che vi permette di far passare dei dati attraverso una linea telefonica. Lo scopo del modem è appunto quello di trasformare dei segnali elettrici in segnali acustici (modulatore) e viceversa (demodulatore), così che l'informazione digitale trasmessa dal computer (che è sempre in forma di numeri) possa passare per una linea analogica come quella telefonica.

Si tenga presente che, anche se questo è lo scopo per cui li usa la maggior parte della gente, non è detto che un modem serva solo per andare su internet; alcuni consentono di ricevere fax, o anche di telefonare. In generale comunque possono essere anche usati semplicemente per collegare due computer attraverso una linea telefonica, senza passare attraverso Internet.

Usare un modem con GNU/Linux è tutto sommato abbastanza semplice, purché si tratti effettivamente di un modem. Infatti alcuni produttori han pensato bene, per abbassarne il prezzo, di far fare al processore del computer buona parte del lavoro che di norma viene eseguito dai componenti di un vero modem. Sono nati così i "winmodem", cioè modem che non sono solo delle schede sonore di bassa qualità, che vengono programmate opportunamente da un "driver" per fare quei "fischi" caratteristici che il modem dovrebbe fare da solo.

Così per risparmiare quando va bene una decina di euro, si viene a perdere un buon 10–20% delle prestazioni del processore, (e se fate i conti vedete che con 10 euro non avreste perso tutte le volte che navigate in internet i 200 euro di differenza che ci sono fra il vostro processore a 2GHz e uno a 1600).

La gran parte dei modem "interni" sono winmodem; in questo caso, a parte un paio di eccezioni, se la ditta non produce un driver per Linux non c'è niente da fare. Ci sono alcuni progetti per produrre dei driver per questi apparecchi, ma funzionano solo con alcuni modelli.

Potete comunque controllare sul sito http://www.idir.net/ gromitkc/winmodem.html se il vostro modem interno è uno di quelli supportati, lì trovate anche le indicazione su dove recuperare i driver. In genere questi sono dei moduli, e basta[21] caricarli con il comando `modprobe` per poter far funzionare il vostro "linmodem". In genere comunque, specie per i vecchi modem ISA, farli riconoscere può essere complicato, nel qual caso vi consiglio una lettura del Modem-HOWTO che trovate sicuramente allegato alla vostra documentazione, [22] e che contiene un sacco di informazioni utili.

Tutti i modem esterni sono invece modem veri, e basta attaccarli ad una porta seriale (dovete dunque avere una seriale libera, ed avere attivato il supporto per la seriale nel kernel). In genere tutto il settaggio che c'è da fare è cercare di capire su quale delle porte seriali lo si è attaccato; ma molti dei programmi di configurazione automatici per l'accesso ad internet (come `vwdial` e `pppconfig`) sono in grado di rilevarlo da soli.

In generale per verificare il funzionamento di un modem dopo averlo acceso si può usare il programma `minicom`, che permette di inviare comandi allo stesso da una finestra di terminale. Il programma è dotato di help in in linea, e vi permette di configurare tutte le opzioni con una interfaccia a finestre testuale.

Le opzioni si settano invocando il comando con lo switch `-s` o richiamando il menù di configurazione dall'interno con `C-a o`. In questo caso il primo settaggio da fare è quello di indicare su quale porta seriale sta il modem (andando su `serial port setup` e settando il device premendo `a`), specificando il nome del device; questo sarà nella forma `/dev/ttySn` dove `n` è un numero, che di solito o è 0 o è 1 (se, come nella maggior parte dei casi, avete solo due porte seriali).

Se avete azzeccato la porta giusta ed il modem vi risponde dovreste ottenere una schermata del tipo:

[21] sempre che vi fidiate, nel caso si tratti di un driver proprietario, a infilare nel vostro kernel del codice che nessuno ha mai potuto controllare.

[22] con molta probabiltà nella directory `/usr/share/doc/HOWTO`.

```
Welcome to minicom 1.83.1

OPTIONS: History Buffer, F-key Macros, Search History Buffer, I18n
Compiled on Nov 21 2001, 00:35:58.

Press CTRL-A Z for help on special keys

ATZ
OK
```

e a questo punto potete provare a vedere se il collegamento al telefono funziona componendo un numero con un comando del tipo:

```
ATDT055412737
```

e sentire se compone il numero. Se va potete concludere che il modem è a posto.

5.2 Il programma chat

Avere un modem funzionante ovviamente non basta, occorre anche

5.3 Il protocollo PPP

Il secondo ed ultimo passo per effettuare la connessione è quella di stabilire un connessione usando il PPP. Fino ad ora abbiamo sempre parlato di TCP/IP ed ecco all'improvviso che salta fuori un nuovo protocollo, il PPP, che sta per *Point to Point Protocol*; viene naturale chiedersi in che rapporti stiano.

Il PPP nasce, come dice il nome, per gestire connessioni punto–punto. Serve cioè non a fare una rete, ma a connettere fra loro due host. È un protocollo generico che può essere usato per *incapsulare* un protocollo di rete e effettuare il suo trasporto attraverso una connessione diretta (nel caso del modem un link seriale) fra due computer. In questo modo, attraverso la connessione PPP, potrete mettere in collegamento, passando per il collegamento diretto fra due stazione di ognuna di esse, due reti diverse.

Normalmente quello che avviene è che il vostro computer di casa si collega a quello del vostro provider, ed è quest'ultimo che è attaccato ad Internet, che così anche voi potete raggiungere. Per effettuare questo collegamento si usa appunto PPP, il cui supporto deve essere stato compilato (direttamente nel kernel o come modulo), il quale creerà una nuova interfaccia di rete (normalmente sarà `ppp0`, dato che di solito non si ha più di un modem), sulla quale passeranno i pacchetti TCP/IP da e verso Internet.

Questo viene fatto attraverso un opportuno programma, `pppd`, che si occupa di effettuare tutte le operazioni