UNIVERSITA' DEGLI STUDI DI PAVIA
Sede distaccata di Mantova
Laboratorio di RETI DI CALCOLATORI
(Prof. Giuseppe Federico Rossi)

Manuale di configurazione di base della rete nel sistema operativo Linux

ad uso degli Studenti del Corso di RETI DI CALCOLATORI
CdL I° livello in Ingegneria Informatica

Versione 0.5

Autore: Luca Capisani

Hanno collaborato alle prove in Laboratorio:

Luca Anselmi, Luca Capisani, Nicola Pelagatti, Veronica Menozzi,
nonché il prof. G. F. Rossi

Mantova, li 16/03/2004

1 - PREMESSA

Il presente manuale è stato pensato quale ausilio per gli Studenti del CdL in Ingegneria Informatica, impegnati nello svolgimento di esercitazioni pratiche presso il *Laboratorio di Reti di Calcolatori* attivo presso la sede distaccata di Mantova dell'Università degli Studi di Pavia.
Più precisamente verrà illustrata la configurazione di rete nel sistema operativo Linux, con particolare riferimento al protocollo TCP/IP (IPv4), ai protocolli EthernetV2 (e IEEE 802.3), al protocollo SLIP (*Serial Line Internet Protocol*) e infine al più utilizzato PPP (*Point-to-Point Protocol*).

Da sempre il sistema operativo Linux, e anche i suoi predecessori UNIX e Minix, offrono un'infinita possibilità di configurazione di rete.
E' da considerare che comandi, procedure e schemi simili a quelli di seguito descritti si trovano anche in altri sistemi operativi secondo svariate implementazioni, ma queste varianti non saranno prese in considerazione nella presente trattazione.
Tutto ciò che sarà visto è una implementazione degli standard *de facto* definiti a livello internazionale, gli RFC (*Request For Comment*), disponibili per consultazione all'URL http://www.ietf.org. Per ogni riferimento dettagliato al funzionamento dei protocolli di rete, si rimanda pertanto a tali documenti.

Chiunque può provare ciò che viene descritto, avendo a disposizione una distribuzione Linux con kernel almeno 2.2.x.x (o superiore). Da notare che nei kernel inferiori al 2.4 ci possono essere piccole varianti ai comandi.

Se si usa una 2.2.x.x (quale ad esempio Red Hat 6.2, 5.2, Mandrake 7.2, Debian), anche i PC più vecchi sono adatti allo scopo.

Sistemi operativi ancora più adatti per l'utilizzo in rete sono NetBSD e OpenBSD, nonché FreeBSD. Questi però sono versioni che derivano da UNIX, quindi i comandi e le funzionalità sono leggermente diverse da ciò che è descritto qui di seguito.

I comandi e le procedure descritte sono stati testati con successo sulla distribuzione Red Hat 9 aggiornata con kernel 2.4.20-8; tuttavia, nel kernel 2.6.x.x sono stati introdotti altri importanti tool (che non verranno trattati).

Ogni comando presentato funziona con lo standard IPv4, sulla base dei principi di funzionamento della rete Internet.
Analoghi comandi sono stati introdotti per il nuovo protocollo IPv6, per essi la sintassi rimane la stessa, se non per il fatto che terminano con "6".

Nel sistema operativo Linux, la configurazione di base della rete si compone dei seguenti punti:
- Creazione delle connessioni hardware;
- Creazione dei dispositivi (interfacce) di rete;
- Impostazione dei parametri di rete;
- Configurazione dei principali parametri di routing;
- Verifica delle impostazioni inserite;
- Verifica delle performance di rete;
- Inserimento di regole di firewalling;
- Impostazione dei nomi degli host;
- Altre categorie non trattate in questo manuale.

NOTA BENE:
Sono state istituite alcune varianti per mantenere la sicurezza nella gestione dei sistemi. In pratica, per l'esecuzione dei comandi specificati, è spesso necessario utilizzare l'utenza di amministratore (root). Se ciò fosse concesso, l'utente incauto o malintenzionato potrebbe danneggiare il sistema.
Per questo non è consentito utilizzare l'utenza **`root`**, ma sarà consentito utilizzare l'utenza **`labreti<numero pc>`** , (esempio **labreti1**); con password uguale.

Per gestire i comandi di rete, sono ammessi solamente i comandi descritti con esecuzione tramite l'utility **`sudo`**.
`sudo <nome comando> <opzioni comando>`
In pratica, ogni comando che richiede l'utenza **`root`** per poter funzionare, si digita preceduto da **`sudo.`**

Le esercitazioni sono impostate in modo tale che le configurazioni impostate non vengano memorizzate dal sistema. Quindi, se il sistema viene riavviato, le impostazioni saranno perse.

Praticamente quasi tutto ciò che troverete in questo manuale, si effettua utilizzando il "terminale", "shell" di Linux, non la sua interfaccia grafica (X-Windows, GNOME, KDE, ecc..).
Questo non per rendere le cose più difficili (a volte non lo sono affatto), ma per dare una visione più standardizzata possibile della configurazione di rete.
Infatti l'interfaccia grafica non è uguale nelle diverse versioni di Linux, e purtroppo alle volte non è pienamente collaudata (può presentare dei malfunzionamenti).
In genere, poi, quello che si può fare con l'interfaccia grafica è solamente una piccola frazione di quello che si fa a terminale. Il terminale può essere aperto anche dall'interfaccia grafica: siete liberi di scegliere!

2 - CONNESSIONI HARDWARE

I 5 PC che vi troverete di fronte sono dotati di 3 schede Ethernet 10/100 Mbps ognuno e di 2 interfacce seriali. La prima interfaccia Ethernet è chiamata eth0, la seconda eth1, la terza eth2. L'interfaccia seriale è chiamata sl0. Le interfacce Token Ring (non presenti) sono chiamate tr#, ma non saranno usate.
Le connessioni possono essere effettuate con i cavi dotati di PLUG RJ45 a 8 fili che possono essere "dritti" o "incrociati" (*cross-cable*).

GLOSSARIO:
Per poter leggere senza confondersi è necessario chiarire il concetto di **host**, **interfaccia**, **scheda**.
Host: rappresenta il singolo PC o router interconnesso nella rete. Talvolta si fa confusione perchè si chiama l'host indicando il suo indirizzo IP. Occorre fare molta attenzione perchè un host può avere molti indirizzi IP (ed è questo il nostro caso). Nella rete Internet, di solito un host ha un solo indirizzo IP pubblico, quindi dall'esterno può essere indicato con l'indirizzo IP (o anche con il nome completo di dominio). I router però hanno molti indirizzi IP pubblici, quindi fate attenzione a ciò che dite.
Interfaccia: rappresenta ciò che il sistema identifica come scheda di rete o comunque connessione di rete (cavo seriale). Un host può avere più interfacce (i nostri ne hanno 3 ciascuno, più le due seriali) ed ogni interfaccia può avere più indirizzi IP. Di solito ne ha solo uno (quello principale), ma può avere anche degli alias (come vedremo).
Scheda: è la parte "hardware" che ci interconnette realmente alla rete. Ciò che vedete dietro al PC.
È necessario avere molta familiarità con questi termini, in quanto verranno frequentemente utilizzati in tutto il manuale.
Va detto che i termini presentati vengono interpretati in vari modi dai sistemi operativi, quindi possono confondere l'utente.

Connessioni hardware per le schede Ethernet.

Ogni scheda ha una presa conforme allo standard RJ45 (simile a quella telefonica). Per poter funzionare è necessario collegare questa presa con un'altra presa che può essere un HUB (SWITCH) o un'altra scheda di rete. I cavi utilizzati sono del tipo UTP RJ45 a 8 fili, la maggior parte dei cavi è di tipo "dritto".
Se si usano i cavi dritti per interconnettere le schede è necessario utilizzare anche un HUB (concentratore), se si usano cavi incrociati è possibile interconnettere con un cavo direttamente due schede di rete tra loro.
Nelle esercitazioni si useranno spesso gli HUB (per ora si considerino HUB e SWITCH come dispositivi uguali). Un HUB (o una cascata di HUB) può interconnettere anche decine di host. Gli host connessi su un HUB si intendono come appartenenti ad una stessa sottorete.
In teoria è possibile avere più sottoreti su uno stesso HUB, ma questa possibilità non sarà presa in considerazione (nella realtà non si interconnettono più sottoreti su uno stesso HUB).

Hub & Switch

Qual'è la differenza tra HUB e SWITCH? Immaginate di aver connesso i 5 PC su un HUB. Quando un PC trasmette un pacchetto su questo HUB, l'apparecchio non fa altro che trasmettere il pacchetto

verso tutti gli altri PC. L'effetto è che nell'HUB non possono avvenire più comunicazioni indipendenti tra PC contemporaneamente. Questo fa si che la velocità massima totale di tutte le comunicazioni tra gli host connessi è pari alla velocità massima dell'HUB e delle interfacce.
In uno SWITCH ogni pacchetto viene inviato solamente al PC interessato, non agli altri (tranne all'inizio, quando lo SWITCH non conosce gli indirizzi fisici delle varie interfacce). Questo rallenta leggermente il tempo di commutazione del pacchetto, ma in compenso permette più comunicazioni indipendenti tra PC, facendo sì che la banda totale di trasmissione, ottenuta sommando le bande di trasmissione misurate su ogni connessione indipendente, risulti più grande della velocità delle interfacce.

Connessioni seriali

Le connessioni seriali si realizzano mediante cavo seriale con connettore a 9 poli (DB9). Anche qui esistono cavi dritti e incrociati; nonché dispositivi che permettono di verificare direttamente che la connessione sia stabilita.

In Linux esiste un tool chiamato **man** che contiene manuali dettagliati di tutti i comandi utilizzabili nell'ambiente. Per ogni riferimento, si consiglia di utilizzare questa guida, in quanto rappresenta una delle risorse più complete sui comandi.
Per richiamare il manuale, sarà sufficiente scrivere:

```
$ man <nome comando>
```

oppure

```
$ man <nome comando> <numero capitolo>
```

oppure

```
$ man <numero sezione> <nome comando>
```

Per uscire dal manuale sarà sufficiente premere **q**, per scorrerlo basta usare le frecce.
Se il manuale di un certo comando è troppo lungo, (es **`man bash`**) sarà sufficiente entrare nel manuale, poi digitare lo slash (“/”) e scrivere la parola cercata. Premendo INVIO, il cursore sarà posizionato alla prima corrispondenza di questa parola.
Purtroppo la pagina **`man`** di alcuni comandi non è molto completa. Capita proprio a fagiolo **`man ifconfig`** che ha diverse lacune. In questo manuale si cercherà di introdurre anche ciò che non troverete nella guida.

Su Internet si trovano miriadi di HOWTO e tutorial per ogni esigenza. Tra l'altro si trova anche la pagina **man** di **ifconfig** nella versione di UNIX FreeBSD che è molto più completa di quella che si ha a disposizione con RED HAT nel laboratorio.

3 - CREAZIONE DEI DISPOSITIVI DI RETE

In Linux, ogni dispositivo (device) è visto come un file, i files che rappresentano ogni dispositivo si trovano nella directory `/dev`.
Linux riconosce le schede di rete, generalmente durante l'installazione del sistema, quindi di solito non è necessario alcun tipo di operazione per creare i dispositivi di rete (ethernet).
I nomi utilizzati per questi dispositivi sono:
`eth<numero dispositivo>` per le schede ethernet: esempio **eth0**, **eth1**, ecc.
con eth0 in genere viene riconosciuta la prima intefaccia ethernet, con eth1 la seconda, ecc.
Per i dispositivi collegati su porte seriali, i nomi sono **sl0, sl1 ...**
Tanto per non fare nomi, la scheda di rete attualmente più diffusa è basata sul chip REALTEK RTL 8139.
Questa scheda di rete necessita di una particolare configurazione (da inserire solamente nel caso in cui non si riesca a utilizzare il riconoscimento automatico).
In pratica è necessario inserire nel file **`/etc/modules.conf`** (in altri
sistemi **`/etc/conf.modules`**), la seguente riga per ogni scheda di rete:

```
...
alias     eth# 8139too
...
```

dove # è il numero dell'interfaccia.
Questo in molti casi è sufficiente.
Se ancora non si riesce a far funzionare la scheda, il problema è sicuramente più complicato. Di aiuto possono essere **`lspci, insmod`**, il kernel 2.6, ecc.
Utile può essere il comando **`ifstatus`** che permette di verificare se il cavo di rete è connesso alla scheda (per le schede che supportano questa funzione). Altri tool da provare sono **`mii-diag`** e **`mii-tools.`**

Creazione delle interfacce *slip* su *porte seriali*

Negli esercizi è spesso richiesto di effettuare un link tramite un cavo seriale.
Questo indica che i dati passeranno su una "rete" (collegamento pointopoint) costituita da cavi seriali.
Il comando **`slattach`** permette di configurare un'interfaccia **`sl#`** dove # indica il numero dell'interfaccia (es sl0)
L'utilizzo è il seguente:

```
# sudo slattach -s 38400 -p slip ttyS0 &
```

dove:
-s 38400 è la velocità di trasmissione: sono possibili i seguenti valori:
300, 600, 1200, 2400, 4800, 9600, 14400, 28800, 33600, 38400, 56600, ecc.
-p slip è il protocollo utilizzato per effettuare la connessione: è possibile usare **slip** e **ppp** (verrà descritto l'utilizzo di slip). Slip è un vecchio protocollo di comunicazione dati per le interfacce seriali. PPP è tutt'ora molto utilizzato nelle connessioni domestiche a internet, con il modem, ma in un'altra modalità, spiegata sotto.
ttyS0 è il nome del device (/dev/ttyS0) utilizzato da Linux (Unix ...) per riconoscere la prima porta seriale (DOS: COM1), sono validi **ttyS1** (COM2) ecc.
& serve per poter ritornare immediatamente alla shell mantenendo il comando in esecuzione.
N.B.
Per ora la shell non può essere chiusa, altrimenti il comando cessa di funzionare.
Una volta lanciato questo comando, viene creato il dispositivo di rete **sl0** che rappresenta una interfaccia configurabile con gli altri comandi.

Creazione delle interfacce *ppp* su *porte seriali*

Per configurare le interfacce PPP (connessioni a internet tramite modem), bisogna specificare alcuni parametri di connessione in alcuni files. Si devono modificare i seguenti files (è meglio se si parte da files già costruiti, altrimenti la cosa si fa difficile...):
`/etc/syconfig/network-scripts/ifcfg-ppp#,` dove va inserito il nome dell'utente di connessione
`/etc/syconfig/network-scripts/chat-ppp#,` dove va inserito il numero telefonico dell'ISP
`/etc/ppp/pap-secrets,` dove va inserita la password di connessione.
Va detto che si possono configurare più ppp: ppp0, ppp1 ecc. che rappresentano più connessioni a Internet (esempio: le connessioni dell'accesso remoto di Windows).
Si trovano riferimenti molto dettagliati a riguardo nei manuali di Linux:

```
man pppd
man chat
man pppdump
man ppp-watch
```

E' anche vero che nelle distribuzioni (RedHat, Mandrake, Debian ....) l'operazione si può fare in maniera molto più conveniente e veloce utilizzando l'interfaccia grafica (anche se alcune volte non funziona a meraviglia...).
Un aiuto per verificare un possibile malfunzionamento di PPP viene dai log di sistema. Infatti, qualsiasi cosa si effettua, il sistema scrive delle righe di "log" in un file chiamato `/var/log/syslog.` Esaminando questo file, si può capire tutto ciò che può aver causato l'errore.

Terminate queste operazioni, le interfacce dovrebbero essere configurate correttamente.
Per verificare l'effettivo inserimento delle interfacce nel sistema, basta digitare il comando

```
# ifconfig -a |more
```

e verranno visualizzate tutte le interfacce configurate o da configurare con le rispettive impostazioni.

4 - IMPOSTAZIONE DEI PARAMETRI DI RETE

Ogni interfaccia di rete ha alcuni parametri che è necessario configurare per far sì che l'interfaccia funzioni. I seguenti parametri sono da configurare (alcuni prevedono una configurazione automatica):

- Indirizzo IP;
- Maschera di sottorete (subnet mask);
- MTU (Maximum Tranfer Unit);
- Indirizzo di broadcast;
- Indirizzo **`pointopoint`** (per le interfacce seriali);
- Eventuali indirizzi alias;
- Attivazione e disattivazione dell'interfaccia.

Indirizzo IP

Per le interfacce Ethernet (**`eth#`**) è necessario impostare indirizzo IP e netmask, in casi particolari anche MTU e broadcast, nonché indirizzi di alias.
Per configurare questi parametri si usa il comando **`ifconfig`**.
Per impostare l'indirizzo IP di una scheda di rete, basta scrivere:

```
$ ifconfig eth# <indirizzo IP>
```

dove **#** rappresenta il numero della scheda di rete.
Si ricorda che (salvo casi particolarissimi):

- La parte host dell'indirizzo IP di una interfaccia non può contenere tutti zeri (indirizzo di rete), tantomeno con tutti 255 (indirizzo di broadcast)
- l'indirizzo IP di una interfaccia non può essere uguale all'indirizzo di un'altra interfaccia, che sia interconnessa alla stessa sottorete (salvo casi particolari).
- l'indirizzo IP di una interfaccia non può contenere valori superiori al 255.
- il network address delle interfacce connesse ad una stessa sottorete deve essere uguale.

Maschera di sottorete

Per impostare la maschera di sotto rete di una scheda:

```
# ifconfig eth# netmask <netmask>
```

I comandi possono essere combinati:

```
# ifconfig eth# <indirizzo ip> netmask <netmask>
```

- La netmask di una interfaccia deve essere uguale per tutti i pc di quella sottorete.
- Una netmask del tipo `255.255.255.255` non è valida (ammette una sottorete con zero host).
- Se l'indirizzamento è standard, la netmask non è obbligatoria, infatti il sistema associa alle varie classi di indirizzamento la propria netmask. Nei nostri esempi non capiterà.

MTU (*Maximum Transmission Unit*)

MTU è la lunghezza massima in byte che può avere un pacchetto IP trasmesso sulla ethernet.
Per lo standard ethernet v2 (IEEE 802.3) si ha di norma `MTU = 1500`, che è il massimo valore impostabile. Il minimo MTU è impostabile a 64 byte (per le ethernet), ma il protocollo IPv4 non accetta MTU al di sotto di 68 byte. Nelle reti Token-Ring (non più utilizzate) MTU doveva essere impostato nel momento in cui si costruiva la rete). Il protocollo IPv6 ci obbliga ad utilizzare MTU

almeno di 1280 bytes. Tuttavia **ifconfig** di Linux accetta valori di MTU tra 1 e 2000.
Morale: questa impostazione viene sempre lasciata a 1500 bytes.

Per impostarlo:
 # ifconfig eth# mtu <mtu>

Indirizzo di broadcast

L'Indirizzo di broadcast è l'indirizzo che corrisponde a tutte le interfacce connesse in una sottorete. Di solito non necessita di modifiche, se fosse necessario modificarlo:
 # ifconfig eth# broadcast <broadcast ip>

Indirizzo `pointopoint` (per le interfacce seriali)

L'indirizzo **`pointopoint`** serve per impostare, nel caso di comunicazione su cavo seriale, l'indirizzo dell'altra interfaccia a cui è connesso il cavo. È necessario perché il sistema non è in grado di risolvere gli indirizzi autonomamente sulle comunicazioni seriali.
 # ifconfig sl0 <ip> netmask <netmask> pointopoint <ind p-t-p>

Alias

Impostazione degli alias di IP: è possibile impostare per una certa interfaccia di rete, più indirizzi ip. Questo significa che l'interfaccia risponde a più indirizzi contemporaneamente:
 # ifconfig eth0 inet add 192.168.0.5
In pratica ho aggiunto un alias alla scheda eth0. Adesso risponde sia sul suo indirizzo principale, sia all'indirizzo alias.
Per togliere l'alias:
 # ifconfig eth0 inet del 192.168.0.5
La parola **`inet`** può anche essere omessa.
Gli alias sono molto importanti. Infatti permettono di gestire più interfacce di rete virtuali con una sola interfaccia di rete. Questo spiega come su un server possano stare più siti web (anche se ci sono altre modalità più usate, come il virtual hosting, dove si vuole simulare un host virtuale), oppure come si possano far funzionare più sistemi operativi contemporaneamente sulla stessa macchina con funzionamento perfetto delle interfacce di rete. Per esempio, l'emulatore WMware utilizza proprio questo sistema per far funzionare la scheda di rete del sistema operativo in emulazione; un utilizzo simile si può avere nelle Virtual Private Network (VPN), per esempio si veda http://openvpn.sourceforge.net

NB:
Le impostazioni di **ifconfig** vanno perse in caso di riavvio del sistema.
Per far sì che le impostazioni IP vengano mantenute al riavvio del sistema, occorre modificare il file **`/etc/sysconfig/network-scripts/ifcfg-eth#`** dove # rappresenta il numero dell'interfaccia. Il file si può tranquillamente editare con un normale editor.
A questo scopo può essere utile il comando **`ifcfg`**.

Indirizzo di *loopback*

Va detto che in ogni host è presente una interfaccia di "`loopback`" che rappresenta l'host stesso, e di solito viene chiamata **`lo`** e ha indirizzo IP `127.0.0.1` e netmask `255.0.0.0`.
Questa interfaccia ha lo scopo di poter utilizzare lo stack di protocolli TCP/IP anche se il computer non è dotato di schede di rete. Rappresenta una interfaccia di rete "virtuale" nel senso che i dati non vengono inviati a nessun dispositivo, ma vengono ritornati all'host stesso. In questo modo è anche

possibile verificare se il sistema operativo è in grado di far funzionare correttamente il modulo TCP/IP (chiamato socket) .

Attivazione e disattivazione dell'interfaccia

Una volta impostate le interfacce, è necessario attivarle.
Per far questo si usa il comando: **`ifconfig eth# up`**

Esempi
Per attivare le interfacce:
```
ifconfig eth0 up
ifconfig sl0 up
ifconfig ppp1 up
```
Per spegnere le interfacce:
```
ifconfig eth0 down
ifconfig ppp0 down
```
Esiste anche una versione più corta del comando, visto che il suo utilizzo è frequente:
```
ifup eth0
ifdown eth0
```
ecc..

IPCALC come valido aiuto per risolvere gli esercizi proposti

In Linux esiste un tool chiamato **`ipcalc`**. Questo tool è utilissimo per poter capire se l'esercizio è stato risolto nella maniera corretta.

Ipcalc è in grado di visualizzare gli indirizzi di rete e broadcast di una sottorete, oltre ad altre informazioni importanti.

5 - CONFIGURAZIONE DEI PARAMETRI DI ROUTING

Premetto che questo argomento può essere estremamente vasto (protocolli di routing, load balancing, routing dinamico ecc.), noi ci limiteremo alle nozioni base, che nella maggior parte dei casi sono sufficienti.
Il comando principale che permette di gestire il routing tra interfacce è **route**.
Route permette di "manipolare" le tabelle di routing del sistema, inserendo o cancellando regole di routing "statiche", cioè che non cambiano mai.
Per effettuare questo passaggio, basterà implementare le tabelle di routing già predisposte durante la soluzione dell'esercizio proposto.

Seguono i comandi da utilizzare:

Per aggiungere una regola che riguarda una rete o una sottorete:
`# route add –net <dest ip> netmask <netmask> gw <gateway>`

Per aggiungere una regola riguardante un singolo host (NB: questa regola viene usata poco):
`# route add –host <dest ip> gw <gateway>`

Per aggiungere la regola di **default**:
`# route add default gw <ip gateway>`

E' da tenere presente che nella tabella di routing per ogni interfaccia vengono già inserite, come default, una o più regole.
Queste regole, che a noi non interessano, si possono contraddistinguere dal fatto che nella tabella compare "*" o 0.0.0.0 nel campo Gateway.

Per visualizzare le regole inserite, basta scrivere **`route –n`**.
Si potrebbe usare semplicemente **`route`**, ma a causa di alcuni problemi DNS, la visualizzazione potrebbe risultare molto più lenta.

6 - ABILITAZIONE DEL FORWARDING

In Linux, è possibile che un host agisca da router. In questo caso, il suo compito è di far transitare i pacchetti da una interfaccia all'altra (forwarding). Per default, e per ragioni di sicurezza, il sistema operativo disabilita la funzione di forwarding, per cui, se si intende rendere attivo il forwarding, bisogna esplicitamente abilitarlo. Nei nostri casi tale funzionalità è da considerare per i PC che funzionano da Router, per i quali verranno sfruttate quasi tutte le schede di rete. In pratica:

Per abilitare il forwarding:
 # echo 1 > /proc/sys/net/ipv4/ip_forward

Per disabilitarlo
 # echo 0 > /proc/sys/net/ipv4/ip_forward

Per controllare che il forwarding sia veramente abilitato, si può scrivere:
 # sysctl –a | grep ip_
e leggere la riga:
net.ipv4.ip_forward
se è =0 il forwarding è disabilitato se è =1 il forwarding è abilitato.

Per impostare il valore della variabile di sistema (corrisponde al comando **echo**):
 # sysctl –w net.ipv4.ip_forwarding=1

NB: Per far sì che il forwarding sia abilitato sempre all'avvio del sistema, procedere come segue:

1. Aprire il file **/etc/sysctl.conf**
2. Alla voce **net.ipv4.ip_forward** inserire il valore 1.

Ovviamente questo non è permesso in laboratorio.

Altre opzioni di `route` sono usate maggiormente per scopi raffinati o diagnostici, nonché dai programmi di instradamento automatico dei pacchetti, per esempio **routed** che funziona con la specifica RIPv2.

Una opzione che può essere interessante è `reject.` Questa opzione permette di eliminare tutti i pacchetti diretti verso una certa rete esterna. Non è una opzione utilizzata nel Firewalling, ma può servire per evitare che una certa rete esterna sia raggiunta passando da quel router.

Un'altra opzione che può avere qualche applicazione è **dev <interf.>** che permette di specificare su quale interfaccia deve essere inviato il pacchetto destinato ad una determinata rete. Nella maggior parte dei casi non è necessario specificare questo parametro, in quanto l'interfaccia da utilizzare può essere identificata immediatamente in base al proprio indirizzo IP; tuttavia permette di provare configurazioni “di test”, per esempio per testare un host che viene connesso alla stessa rete mediante due diverse interfacce. Un altro utilizzo si può trovare dei cosiddetti tunnel, ovvero quando un pacchetto non deve essere inviato direttamente ad una scheda ma deve essere prima “incapsulato” in un altro pacchetto, per poi essere immesso in un tunnel.

7 - VERIFICA DELLE CONFIGURAZIONI

Comando "Ifconfig"

Con il comando **`ifconfig -a`** si possono visualizzare tutte le configurazioni delle varie interfacce presenti, con **`ifconfig`** si possono verificare quali interfacce sono configurate e accese.

```
#ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:40:F4:6D:B4:1A
          inet addr:10.1.6.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:180 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:26816 (26.1 Kb)
          Interrupt:11 Base address:0x8000
```

Ci dice che l'interfaccia `eth0` è di tipo **Ethernet**, con MAC-address `00:40:F4:6D:B4:1A` e indirizzo IP: `10.1.6.1`. Ci dice anche che è accesa (`UP`), che ha ricevuto 0 pacchetti (bytes) e trasmesso 180 pacchetti (26816 bytes totali).

Comando "Ifstatus"

Con il comando **`ifstatus`** è possibile verificare se il sistema rileva la connessione della scheda di rete.
Può capitare infatti che, a causa di un cavo o di un dispositivo difettoso, la scheda non risulti "collegata" allo SWITCH o HUB, o ad un'altra scheda.
Digitando questo comando, ci apparirà immediatamente lo stato della connessione e le possibili risposte possono essere:

- `eth0: link beat detected`

se il cavo è connesso

- `eth0: unplugged`

se il cavo è scollegato.

Verifica della tabella di routing

Con il comando **`route -n`** si possono verificare le regole di routing impostate.
Si ricordi che le impostazioni che hanno come gateway * o 0.0.0.0 sono impostazioni inserite automaticamente dal sistema, e non ci interessano.
Un esempio di risposta a tele comando è:

```
Kernel IP routing table
Destination    Gateway   Genmask   Flags Metric Ref    Use  Iface
10.0.0.0       0.0.0.0   255.0.0.0  U     0      0      0    eth0
127.0.0.0      0.0.0.0   255.0.0.0  U     0      0      0    lo
```

Indica che non abbiamo inserito alcuna regola di routing, ma sono presenti solamente quelle "base" dell'interfaccia `eth0` e `lo`.

8 - VERIFICHE DEL FUNZIONAMENTO DELLA RETE

Siamo giunti alla prova finale, ovvero la verifica delle connessioni di rete. Si tratta di verificare se ogni PC è in grado di "comunicare" in maniera esatta con gli altri PC.

Il comando utilizzato in questo caso è il famoso **`ping.`**

`ping` ha la stessa funzione di dire "ci sei?". La risposta a questa domanda può assumere tre forme diverse:

1) Il **ping** va a buon fine e l'host interpellato risponde correttamente;
2) L'host contattato non risponde alla richiesta;
3) Il nostro host non riesce ad inviare il pacchetto di ping.

Tale comando funziona con le specifiche del protocollo ICMP; invia un messaggio ICMP "echo request", e si aspetta in risposta una ICMP "echo reply".

Un esempio di ping può essere:

```
ping 192.168.0.20
```

la cui risposta può essere:

```
64 bytes from 192.168.0.20: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=64 time=0.056 ms
.
.
.
```

Questo indica non solo che l'interfaccia interpellata sta funzionando e risponde ai "ping", ma che abbiamo anche ricevuto 64 bytes, in un tempo di andata e ritorno di 0.056 ms.
Di seguito vengono specificati casi in cui può essere utilizzato il comando ping:

NB: il comando PING esegue un test continuativo.
Per interrompere il test, è necessario premere i tasti CTRL + C.

Ping verso le interfacce locali

Una delle prime cose da verificare in seguito alla configurazione di un host, è la corretta configurazione delle interfacce locali.
Per verificare la corretta assegnazione dell'indirizzo IP all'interfaccia è possibile eseguire verso di essa un ping.

Esempio:
Supponiamo che un host abbia i seguenti indirizzi IP:

```
eth0: 192.168.0.1
eth1: 192.168.1.3
eth2: 192.168.2.5
```

Se lanciamo i comandi

```
# ping 127.0.0.1
# ping 192.168.0.1
# ping 192.168.1.3
```

```
# ping 192.168.2.5
```

Se tutte le interfacce sono funzionanti, come risposta ad ogni ping specifico avremo:

```
64 bytes from xxx.xxx.xxx.xxx: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from xxx.xxx.xxx.xxx: icmp_seq=2 ttl=64 time=0.056 ms
```

Se in questi quattro comandi ce n'è uno (o più) che non dà la risposta corretta, si è sbagliato qualcosa durante l'impostazione dei parametri di rete, oppure semplicemente non si è digitato il comando **ifconfig eth# up**.

Test di raggiungibilità degli altri host

Realizzata la rete e configurati i computer occorre verificare se il nostro host è in grado di comunicare correttamente con gli altri host.
Questo lo si fa "pingando" tutti gli altri indirizzi IP della rete.
Le possibili risposte del sistema sono:

a) Se tutto funziona correttamente, appaiono sempre righe simili a quelle sopra citate, dove xxx sono i numeri dell'indirizzo IP puntato:

```
64 bytes from xxx.xxx.xxx.xxx: icmp_seq=1 ttl=64 time=0.056 ms
```

b) Se uno dei nodi intermedi non riesce ad instradare il pacchetto di "ping" (non esiste una regola di routing per instradare quel pacchetto):

```
connect: Network is unreachable
```

c) Se la rete è raggiungibile, ma l'interfaccia non risponde:

```
From 192.168.0.1 icmp_seq=1 Destination Host Unreachable
```

Possibili cause di errore possono essere:

1. Si è sbagliato l'indirizzo IP, o comunque l'indirizzo non può essere raggiunto, pur essendo logicamente adiacente all'host in questione.
2. Alcune tabelle di routing sbagliate o incomplete sul nodo stesso o su host intermedi.
3. Non si è configurato correttamente il percorso di andata e ritorno del pacchetto, tramite il comando route sui vari host.
4. Va detto (anche se non è il nostro caso), che se tra il nostro host e l'host da pingare c'è di mezzo un firewall, talvolta i ping possono non andare a buon fine.
5. In tal caso, questo non significa che l'host non è raggiungibile, ma semplicemente che il firewall ha impedito il ping disabilitando il protocollo ICMP.

Verifica del percorso seguito da un pacchetto

Talvolta è necessario verificare quale percorso abbia seguito un pacchetto. Ci sono 2 modi.

 # ping -R <indirizzo ip>

Permette di registrare il percorso seguito sia all'andata che al ritorno. Saranno visualizzati tutti gli indirizzi IP coinvolti. Può visualizzare al massimo 9 passaggi.

 # traceroute <indirizzo ip>

Fondamentalmente esegue lo stesso compito, anche se è più adatto perchè è in grado di visualizzare un numero maggiore di host.

Verifica della tabella ARP

Ogni interfaccia di rete ha un indirizzo "fisico" di livello due (ISO-OSI) chiamato MAC-address.
Questo indirizzo viene inserito nelle PDU di livello due e identifica mittente e destinatario del pacchetto.
Il protocollo ARP serve a risolvere la corrispondenza IP-Address→MAC-Address utile per individuare gli indirizzi MAC delle schede presenti sui un link.

`arp -a` permette di visualizzare tutte le impostazioni "dinamiche" e "statiche" della tabella ARP.
È utile nel caso ci siano problemi nell'interconnessione tra pc della stessa sottorete.

Teoricamente non dovrebbero esistere al mondo due schede con uguale indirizzo. Se ciò dovesse capitare (schede di rete a basso costo), è necessario impostare da Linux (sempre che la scheda lo permetta) un indirizzo di livello 2 diverso.
In questo caso, il comando **`arp -s`** (vedere man) può essere utile per cambiare gli indirizzi.

Verifica della comunicazione

A rete configurata, una verifica molto importante consiste nell'osservare i pacchetti che transitano sul canale di comunicazione. Tale procedura è in gergo chiamata "sniffing" della rete.
In parole povere, essa consiste nel vedere "cosa passa" nei cavi.

Tcpdump

Un programma molto utile per eseguire lo sniffing è **`tcpdump`**.
Tcpdump permette di mettersi in "ascolto" su una certa interfaccia e vedere l'intestazione IP e TCP dei pacchetti che passano.
Se si lancia:
`# tcpdump -i <nome interfaccia>`
ad esempio **`sudo tcpdump -i eth0`**
Si ottengono delle righe simili:

```
17:21:17.662317 arp who-has 192.168.0.2 tell 192.168.0.1
17:21:17.662638 arp reply 192.168.0.2 is-at 0:e0:18:6:a3:e8
17:21:17.662664 192.168.0.1 > 192.168.0.2: icmp: echo request (DF)
17:21:17.662777 192.168.0.2 > 192.168.0.1: icmp: echo reply (DF)
```

In pratica significa che alle `17:21:17` il sistema **`192.168.0.1`** ha chiesto qual è il MAC-address di **`192.168.0.2`** (ARP request).
Il sistema **`192.168.0.2`** ha risposto dicendo che il proprio indirizzo MAC è: `0:e0:18:6:a3:e8` (ARP reply).
In seguito il sistema **`192.168.0.1`** ha lanciato una **`icmp echo request`** (PING) verso **`192.168.0.2`**; e il sistema **`192.168.0.2`** ha risposto (**`echo reply`**).

Dall'esempio proposto si vede come **`tcpdump`** sia in grado di fornire informazioni molto importanti sul transito dei pacchetti di rete.

Ethereal

Un altro tool, molto più potente (e un po più complicato) rispetto a tcpdump è **`ethereal`**, funziona in ambiente grafico e ha varie funzioni molto utili.
Per farlo funzionare è necessario accedere all'interfaccia grafica e scrivere a terminale grafico "**`sudo ethereal`**".
Il comando **`sudo`** è necessario poiché il sistema impedisce agli utenti normali di "sniffare la rete".
A questo punto si vada nel menù **`capture`** e si prema **`start`**, si scelga la scheda di rete da "sniffare" (sono disponibili anche le seriali sl#) e si prema OK. Se si vogliono "sniffare" tutte le schede di rete contemporaneamente, basta utilizzare la voce **any**.
A questo punto il programma raccoglierà informazioni sui pacchetti in transito, classificandoli per protocollo. Concluso l'esperimento, si prema **`stop`**.
A questo punto appaiono in modalità grafica tutti i dettagli dei pacchetti che sono transitati.
E' importante notere che con **`ethereal`** si può andare a vedere nel dettaglio quali informazioni sono transitare, mentre con **`tcpdump`** si vedono solo le intestazioni dei pacchetti.

Verifica delle comunicazioni TCP/UDP

Il comando **`netstat`** visualizza le connessioni TCP, le porte d'ascolto TCP e UDP, nonché lo stato degli stream unix.
Se si digita:

```
$ netstat –an |more
```

o meglio

```
$ netstat –an –prot inet
```

Si ottiene, nella parte iniziale della risposta, l'elenco completo delle porte TCP e UDP che attualmente il sistema sta "ascoltando".
Inoltre l'elenco contiene anche le connessioni TCP che sono instaurate (`established`); nonché l'elenco delle trasmissioni di pacchetti UDP che si sono appena verificate.

9 - IMPOSTAZIONE DI REGOLE DI FIREWALLING

Linux include una serie di tools, chiamati **`iptables`** e **`ipchains`**. Sono molto utili per impedire accessi “indiscreti” ad un host, o ad una certa rete protetta.

Senza addentrarci troppo nei particolari, nella pratica si può dire al sistema quali connessioni e indirizzi IP ritenere affidabili, mentre quali è necessario scartare.

Un'altra parte importante riguardante le regole del Firewalling rappresenta la Translation (traduzione) degli indirizzi e delle porte, nella pratica avviene una modifica dei pacchetti in transito. Questo si può fare con le funzionalità NAT , NAPT e nangle di iptables.

Per chi fosse interessato a questo argomento, un documento molto interessante da consultare si trova al sito http://www.muug.it/k gestito da Grespan Lorenzo per il Mantova Unix User Group.

10 - IMPOSTAZIONE DEI NOMI DEGLI HOST

Ogni host ha un suo nome. Tale nome è collegato ad un indirizzo IP. I nomi sono stati introdotti perchè sono molto più pratici e semplici da ricordare rispetto agli indirizzi IP.
In teoria ogni host dovrebbe conoscere sempre i nomi (e quindi gli indirizzi IP) di tutti gli altri host. Questo perché l'utente ricorda più facilmente i nomi, non gli indirizzi IP, quindi preferisce specificare i nomi.
Questo significa che ogni host deve essere in grado di "risolvere" un nome, cioè trovare l'indirizzo IP di un host specificato con un nome.
Nelle reti piccole è possibile inserire in tutti gli host i nomi degli altri host, anche se è abbastanza inutile, in quanto gli host sono pochi e avere un nome per ogni host non serve a molto.

Nelle grandi reti (Internet) ciò è fondamentale. Se non ci fosse un sistema di **risoluzione dei nomi**, sarebbero in pratica inutilizzabili la maggior parte dei servizi internet (Si noti che il servizio HTTP è basato sui nomi degli host, più che sul loro indirizzo IP).
Serve allora un sistema che in qualche modo permetta di memorizzare i nomi degli host e associarli al proprio indirizzo IP.
Tenete presente che non è possibile mantenere sulla memoria di un calcolatore, o di un gruppo ristretto di calcolatori un elenco unico e centralizzato di nomi, in quanto:

1) i nomi, gli indirizzi, le informazioni sono in continua evoluzione;
2) gli host attualmente sono milioni (si parla di quasi 200 milioni di host connessi a Internet);
3) l'operazione dovrebbe coinvolgere tutti gli host connessi;
4) L'aggiornamento e la ricerca delle voci risulterebbe difficoltoso se non straziante.
5) varie ed eventuali.

Per questo si predispongono i servizi DNS che si occupano di risolvere i nomi dell'intera rete internet.
I servizi DNS rappresentano database distribuiti su migliaia di calcolatori connessi a Internet che mantengono le informazioni per i domini registrati.
Per fare delle prove, con un PC correttamente connesso a internet (con linux) si possono usare i comandi **`nslookup`**, **`dig`**, **`host`** ... che permettono di trovare l'indirizzo IP di un host specificando il suo nome.
Ad esempio:

```
# dig www.unimn.it

; <<>> DiG 9.2.3rc2 <<>> www.unimn.it
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40200
;;  flags:  qr  aa  rd  ra;  QUERY:  1,  ANSWER:  1,  AUTHORITY:  2,
ADDITIONAL: 2

;; QUESTION SECTION:
;www.unimn.it.                  IN      A

;; ANSWER SECTION:
www.unimn.it.           86400   IN      A       217.169.102.120

;; AUTHORITY SECTION:
unimn.it.               86400   IN      NS      dns.x2000.net.
unimn.it.               86400   IN      NS      dns2.x2000.net.
```

```
;; ADDITIONAL SECTION:
dns.x2000.net.          86400   IN      A       151.99.229.225
dns2.x2000.net.         86400   IN      A       151.99.229.246

;; Query time: 190 msec
;; SERVER: 151.1.1.1#53(151.1.1.1)
;; WHEN: Sun Mar 14 12:08:25 2004
;; MSG SIZE  rcvd: 132
```

La risposta ci dice che il server www.unimn.it ha per indirizzo IP: `217.169.102.120` e che la risposta ci è stata fornita dal DNS chiamato dns.x2000.net.
Come facilmente si può intuire in una piccola rete, come la nostra del laboratorio, non serve impostare i servizi DNS. Per i computer locali, basta editare il file **`/etc/hosts`** che consente di specificare:

IP `nome completo dell'host  nome locale dell'host`
esempio:
`217.169.102.120` www.unimn.it `www`

Una volta effettuata questa operazione su tutti i PC interessati, è possibile usare i nomi locali o i nomi completi per effettuare tutte le operazioni in cui servano gli IP: sarà il sistema ad andare a trovare automaticamente gli IP.

Nell'esempio di cui sopra, sarà del tutto equivalente scrivere:
`ping` `217.169.102.120` e
`ping www.unimn.it`

Con la differenza che il nome www.unimn.it è molto più facile da ricordare!

Per far sì che un sistema Linux connesso a Internet utilizzi il servizio DNS, basta inserire l'indirizzo IP del server DNS preferito nel file **`/etc/resolv.conf`,** nella seguente modalità:

```
nameserver 151.1.1.1
```

Questo indica che il sistema utilizzerà server DNS `151.1.1.1` per trovare gli indirizzi IP degli host che non sono presenti nel file **`/etc/resolv.conf`.**
Per la cronaca, 151.1.1.1 (**`dns.it.net`**) è uno tra i principali server DNS italiani.
Nel file `resolv.conf` si possono specificare più server DNS, sempre nella forma:
`nameserver xxx.xxx.xxx.xxx`
Se si specificano più server DNS, nel caso in cui il primo non risponda, si utilizzerà il secondo, e così via.

11 - INFORMAZIONI AGGIUNTIVE SU IFCONFIG

Con l'intenzione di integrare il manuale di **ifconfig**, si riportano alcune particolarità tratte principalmente dal manuale presente nella distribuzione FreeBSD.
Un riferimento molto striminzito alle opzioni di ifconfig si ottiene digitando
ifconfig –help
Va detto anche che ciò che viene spiegato qui fa parte della documentazione di FreeBSD, quindi potrebbe essere inesatto per altre distribuzioni.

FAMIGLIE DI INDIRIZZI
Negli esempi è sempre stata omessa la voce **inet.** Altre possibilità sono: **inet6, atalk** (Apple Machintosh), **ipx** (IPX/SPX Microsoft), **link** (MAC-address di livello 2), che è sinonimo di **ether, lladdr**.

In pratica si dovrebbe scrivere:
`# ifconfig eth0 inet 10.1.5.4`
ma noi non l'abbiamo fatto perchè la parola "inet" è inserita per default.

ARP
Le opzioni **arp** e **-arp** servono per attivare o disattivare l'utilizzo di protocolli ARP per la risoluzione degli indirizzi MAC. Nelle ethernet, attivare ARP è necessario; altrimenti si devono specificare tutti gli ARP a priori con il comando **arp**.

PROMISCUOUS
Normalmente il sistema analizza solo i pacchetti che sono direttamente indirizzati al suo indirizzo IP. Come detto, se utilizziamo un HUB, ogni pacchetto è trasmesso su tutte le interfacce. Nella modalità promiscuous è possibile ricevere anche i pacchetti che non sono stati indirizzati a noi.
Si attiva nel seguente modo:
`#ifconfig eth0 promisc`
e si disattiva nel seguente modo:
`#ifconfig eth0 –promisc`
In altre implementazioni, **promisc** è simile ad **allmulti**.

DRIVER MODE
Serve per impostare la modalità di funzionamendo del driver, ad esempio, nelle lan wireless, può essere '802.11a', '802.11b', '802.11g'.

INDIRIZZI DEI TUNNEL
Servono per impostare i parametri di un **tunnel**: indirizzo di ingresso e di uscita del tunnel.
Esempio:
`#ifconfig tun+ tunnel 192.168.1.5 192.168.1.4`
Il comando **deletetunnel** serve per cancellare gli indirizzi del tunnel impostati.

STATISTICHE DI FUNZIONAMENTO
`#ifconfig –s` visualizza le statistiche di funzionamento di ciascuna interfaccia.

12 - VERIFICA DELLE PRESTAZIONI DI UNA CONNESSIONE DI RETE

Esistono programmi in grado di simulare le prestazioni di una rete, essi generano del traffico fittizio con lo scopo di calcolare la velocità di trasmissione dei dati.
Uno di questi è **iperf**. Lo si può trovare trovate sul sito http://dast.nlanr.net.
La versione presa in considerazione è la 1.7.0, l'ultima attualmente a disposizione.

Installazione

Per prima cosa è necessario scaricare il file dell'applicazione dal sito sopra citato. Ci sono due possibilità: scaricarlo per la glibc 2.1 o la glibc 2.3. Dipende dalla versione del vostro compilatore C. Se non sapete che versione avete, scaricate entrambe le versioni del programma. Quindi abbiamo scaricato il file: `iperf-1.7.0-linux-2.3.tar.gz`. Di questo programma si trova anche la versione per Windows: funziona anche tra Windows e Linux.
A questo punto è necessario installare il programma. L'operazione è relativamente semplice.

1) decomprimere il file, con il comando: `gzip -d iperf-1.7.0-linux-2.3.tar.gz`
2) Aprire l'archivio tar, con il comando: `tar xvf iperf-1.7.0-linux-2.3.tar`
3) Se tutto è andato bene, il programma è installato.

Utilizzo

Il programma **iperf** è molto semplice da utilizzare, seguono un insieme di passi da seguire

1) Predisporre la rete che si vuole testare; scegliere i due computer protagonisti del test. Il primo sarà chiamato **server**, il secondo **client** (**iperf** deve essere installato su entrambi).
2) Sul computer **server**, digitare quanto segue: `./iperf -s`
3) Sul computer **client**, digitare quanto segue: `./iperf -c <IP del server>`
4) Il programma eseguirà il test e darà lo stesso output sia sul server che sul client.

Tra le varie opzioni di questo programma, è anche possibile variare la finestra TCP (e quindi la frequenza degli ACK), è possibile variare la finestra di connessione, oppure usare il protocollo UDP.

Effettuando un **test** in locale, si è ottenuta la seguente risposta (per TCP):

```
[luca@luca iperf]$ ./iperf -c localhost
------------------------------------------------------------
Client connecting to localhost, TCP port 5001
TCP window size: 49.4 KByte (default)
------------------------------------------------------------
[   5] local 127.0.0.1 port 32771 connected with 127.0.0.1 port
5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  2.62 GBytes  2.25 Gbits/sec
```

e per UDP (dove la banda è limitata per default a 1 Mbit/sec, ma può essere cambiata):

```
[luca@luca iperf]$ ./iperf -u -c localhost
------------------------------------------------------------
Client connecting to localhost, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[   5] local 127.0.0.1 port 32769 connected with 127.0.0.1 port
```

```
5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  5] Server Report:
[  5]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.004 ms    0/
892 (0%)
[  5] Sent 892 datagrams
```

13 - ALTRI ARGOMENTI NON TRATTATI IN QUESTO MANUALE

Su internet si trovano la maggior parte dei documenti informativi riguardanti questo argomento.
Se si cercano specifiche sui protocolli di rete, i siti di riferimento sono i seguenti:

http://www.ietf.org
http://www.ietf.org/rfc
http://www.faqs.org
http://www.linux-ipv6.org
http://www.ieee.org
http://www.iana.org
http://www.6bone.net
http://www.tuttoreti.com
http://www.mrtg.org
http://dast.nlanr.net

Se si cerca documentazione su Linux o applicativi legati a linux:

http://www.gnu.org
http://www.kernel.org
http://www.linux.org
http://www.linux.it
http://www.lugman.org
http://www.google.it/linux
http://www.sourceforge.net

In molte distribuzioni di linux è presente la cartella `/usr/share/doc` che contiene varia documentazione sui tools installati nel sistema, fornendo manuali ed esempi. Molto interessante è la documentazione di `iproute`.