File di configurazione[1]

Diversi file sono coinvolti nella specificazione completa degli indirizzi di rete e nella configurazione di TCP/IP. Usiamo per scopi illustrativi Unix. Come standard , facendo menzione di altri sistemi operativi[2] . altri sistemi operativi utilizzano differenti nomi di file[3] , ma lo scopo dei file è lo stesso.

Unix consente commenti su ogni riga di questi file di configurazione se sono preceduti dal segno #.

Symbolic Machine Names[4]: /etc/hosts

Quando un nome simbolico è usato come indirizzo obiettivo di un'applicazione, deve esservi qualche metodo per risolvere questo nome simbolico in un indirizzo di rete. Si usa comunemente un file ASCII in cui i nomi simbolici sono messi in relazione con

---

[1] Configuration files

[2] Operating systems

[3] filenames

[4] Symbolic Machine Names

gli indirizzi. Questo non si applica quando si usa Yellow Pages, Network Information Services o Domain Name Server; essi usano i propri file di configurazione.

Su sistemi Unix è usato il file /etc/hosts/ per contenere gli indirizzi di rete, così come una connessione speciale chiamata loopback che verrà descritta in seguito. L'indirizzo di connessione loopback[5] è usualmente messo in elenco come loopback del nome di macchina o host locale[6].

Il file /etc/hosts/ consiste dell'indirizzo di rete in una colonna separato dal nome simbolico in un'altra. L'indirizzo di rete può essere specificato in formato decimale[7], ottale[8], o esadecimale[9]. Può essere specificato più di un nome simbolico su una riga separando i nomi con caratteri di spazio o tabulazioni. Il file può essere lungo quanto è necessario per contenere tutti i nomi simbolici usati sulla macchina locale; essi non hanno bisogno di essere presentati in alcun ordine. Un esempio di file è il seguente:

```
# network host addresses

127.0.0.1          localhost local tpci_server

157.40.40.1        tpci_sco1

157.40.40.2        tpci_sco2
```

---

[5] loopback

loopback driver

loopback connection

[6] localhost

[7] decimal

[8] octal

[9] hexadecimal

157.40.40.3 tpci_hpws1

157.40.40.0 tpci_server tpci_main tpci

47.80.157.36 bnr.ca BNR bnr

191.13.123.4 kitty_cat

205.150.89.1 roy_maclean big_roy

210.24.47.128 bobs_machine

un amministratore di sistema o di rete può aggiornare il file in qualunque momento, e i cambiamenti hanno immediata efficacia così non si deve effettuare il riavvio della macchina per rendere i cambiamenti effettivi).

La maggior parte delle implementazioni TCP/IP[10] su altre piattaforme[11] ha un tipo simile di file per risolvere gli indirizzi IP dai nomi simbolici. NetManage ChameleonNFS che gira su macchine Windows 3.x , per esempio, usa una Host Table[12] per effettuare il collegamento fra nomi simbolici e indirizzi IP. La Host Table mostrata nella figura seguente , è un front-end grafico[13] verso un file equivalente al file visto per Unix.

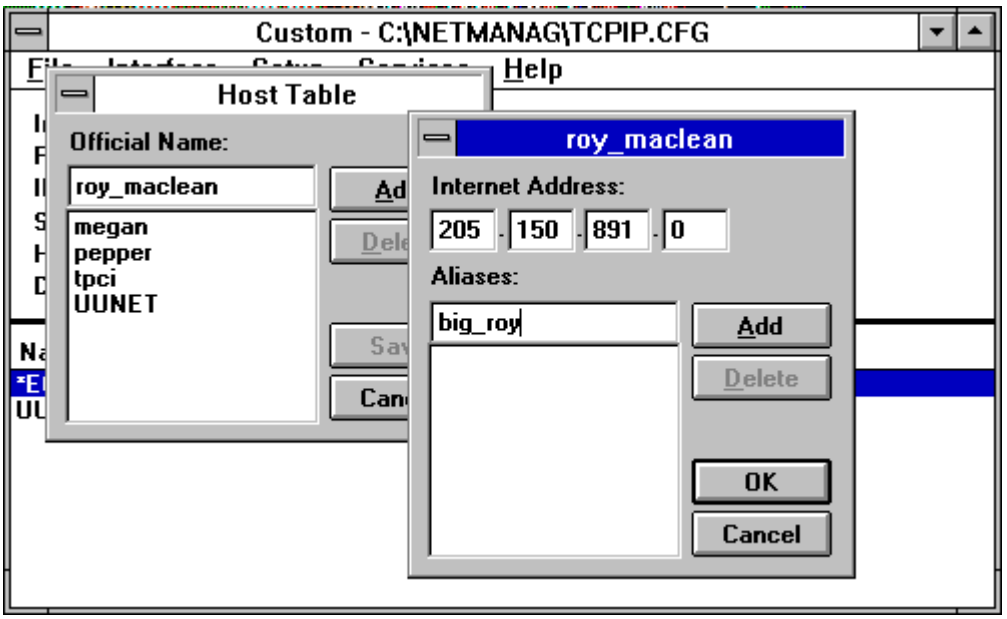
---

[10] TCP/IP implementations

[11] platforms

[12] Host Table

[13] graphical front-end



Network names[14]: /etc/networks

Le reti possono essere indirizzate mediante un nome simbolico, proprio come le singole macchine. Per risolvere i nomi di rete , è usato un altro file che contiene i corrispondenti nomi di rete. Tipicamente questo file non subisce frequenti accessi, perché pochi utenti necessitano di indirizzare un'intera rete nelle loro applicazioni. L'uso più comune del file di risoluzione del nome di rete è di specificare il nome della rete locale[15].

I sistemi Unix usualmente usano il file /etc/networks/ per specificare nomi di rete. Il formato del file fornisce un nome simbolico di rete, il suo indirizzo di rete, ed ogni alias che può essere usato. Un esempio del file è il seguente:

```
# local network names
```

[14] Network names

[15] network name risolution file

```
tpci        146.1          tpci_network  tpci_local

bnr         47.80          BNR bnr.ca

tmn         123.2.21

unique      89.123.23      UNIQUE

sco         132.147        SCO

loopback    127            localhost
```

L'ultima registrazione in questo file fornisce il nome di loopback. La prima registrazione specifica il nome della macchina locale, il suo indirizzo di rete, ed ogni variante del nome. Molte implementazioni di TCP/IP su altre piattaforme non si interessano di file come questo per la risoluzione dei nomi di rete. Parte della ragione è che molti sistemi operativi per singolo utente non richiedono il tipo di versatilità richiesta da un sistema nultiutente come Unix.

Network protocols[16]: /etc/protocols[17]

I numeri di protocollo sono usati per identificare il protocollo di trasporto nella macchina ricevente per abilitare la decodifica corretta delle informazioni contenute nei datagrammi[18]. Con TCP/IP il numero di protocollo è incastonato nel header IP. È

---

[16] Network protocols

[17] /etc/protocols

[18] datagram

usualmente utilizzato un file di configurazione per identificare tutti i protocolli di trasporto[19] disponibili e il loro rispettivo numero di protocollo.

I sistemi UNIX usano il file /etc/protocols. Usualmente questo file non è modificato dall'amministratore ma è gestito dal sistema ed aggiornato automaticamente come parte della procedura di installazione quando sono aggiunti nuovi servizi e software TCP/IP[20]. Il file contiene il nome di protocollo , il suo numero, ed ogni alias che potrebbe essere usato per quel protocollo. Un esempio è il seguente:

```
#
# Internet (IP) protocols
#
ip      0       IP      # internet protocol, pseudo protocol number
icmp    1       ICMP    # internet control message protocol
igmp    2       IGMP    # internet group management protocol
ggp     3       GGP     # gateway-gateway protocol
tcp     6       TCP     # transmission control protocol
egp     8       EGP     # Exterior-Gateway Protocol
pup     12      PUP     # PARC universal packet protocol
udp     17      UDP     # user datagram protocol
```

[19] transport protocols

[20] TCP/IP software or services

```
hello   63      HELLO   # HELLO Routing Protocol

ospf    89      OSPF    # Open Shortest Path First Routing Protocol
```

Non vi sono usualmente equivalenti di questo file su Altri sistemi che assumo che il numero di trasporto standard venga usato per ogni protocollo.

Network Services[21]: /etc/services[22]

IL file di configurazione finale usato nella maggior parte dei sistemi Unix identifica i servizi di rete esistenti. Questo file non è modificato usualmente dall'amministratore di rete ma è gestito dal software[23] nelle fasi di installazione o configurazione.

Il file per i sistemi Unix è /etc/services. Il file in formato ASCII consiste del nome del servizio, un numero di porta, e il tipo di protocollo. Il numero di porta e il tipo di protocollo sono separati da una slash. Ogni alias opzionale del servizio segue dopo il numero di porta. un esempio è il seguente:

```
# network services

echo     7/tcp

echo     7/udp

discard  9/tcp   sink  null

discard  9/udp   sink  null

ftp      21/tcp
```

---

[21] Network Services

[22] /etc/services

[23] software

```
telnet   23/tcp

smtp     25/tcp   mail mailx

tftp     69/udp

# specific services

login    513/tcp

who      513/udp   whod
```

Settaggio del Host Name[24]

TCP/IP richiede che ogni macchina sulla rete abbia un proprio indirizzo IP[25]. Usualmente ogni macchina ha anche un nome simbolico[26] univoco; altrimenti, deve essere usato l'indirizzo IP per tutte le connessioni a quella macchina. La maggior parte dei sistemi operativi ha un semplice programma che identifica il nome della macchina locale. Le macchine Unix hanno, per questo scopo, l'utility[27] hostname[28] , così come il programma uname[29]. L'utility è usualmente supportata soltanto in System V [30]e sistemi operativi compatibili.

---

[24] Setting the Host Name

[25] IP address

[26] symbolic name

[27] utility

[28] hostname

[29] uname

[30] System V

Il nome del host è qualche volta salvato in un file separato letto quando il sistema operativo parte , o può essere letto da uno dei file di configurazione menzionati precedentemente. Il hostname[31] è usato dalla maggior parte dei protocolli sul sistema e da molte applicazioni TCP/IP[32], così è importante per l'esecuzione appropriata delle operazioni di sistema. Il nome del host può talvolta essere cambiato editando[33] il file di sistema[34] che contiene il nome e poi riavviando[35] la macchina.

Su molti sistemi Unix i comandi hostname e uname effettuano l'echo[36] del nome della macchina locale, come mostra il seguente esempio

```
$ hostname

tpci_sco4.tpci.com

$ uname -n

tpci_sco4
```

nel sistema SCO Unix mostrato nell'esempio, il comando hostname restituisce il nome di dominio completamente qualificato mentre il comando uname restituisce solo il nome della macchina locale. Il comportamento esatto di questi due comandi dipende dall'implementazione.

---

[31] hostname

[32] TCP/IP applications

[33] editing

[34] system file

[35] rebooting

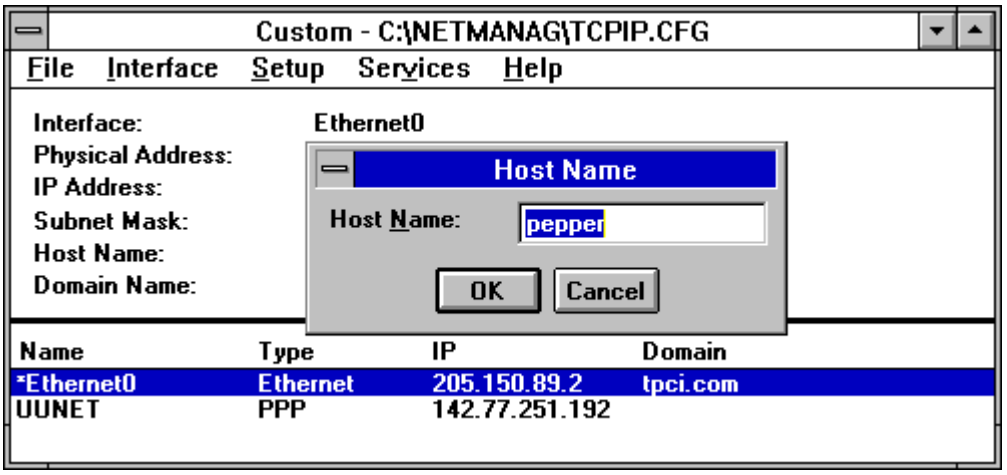
[36] echo

In un sistema Linux[37], per esempio, il comando hostname può essere usato non solo per mostrare il corretto settaggio del host name ma anche per cambiarlo quando viene usato con l'opzione –S (per Set). Per esempio il comando

hostname -S willow.tree.com

cambia il nome di dominio locale in willow.tree.com. non tutte le versioni di Linux supportano l'opzione –S nel comando hostname.

La maggior parte delle suite TCP/IP[38] per altri sistemi operativi usa un metodo più semplice per settare il nome del host. Per esempio, in Windows 3.x il pacchetto ChameleonNFS usa la seguente maschera per settare rapidamente il nome del host.



Windows NT ha servizi TCP/IP costruiti[39] nella distribuzione base. Su un sistema Windows NT il nome del host è specificato attraverso la finestra di dialogo Network[40] dal Pannello di Controllo[41] come nella figura seguente

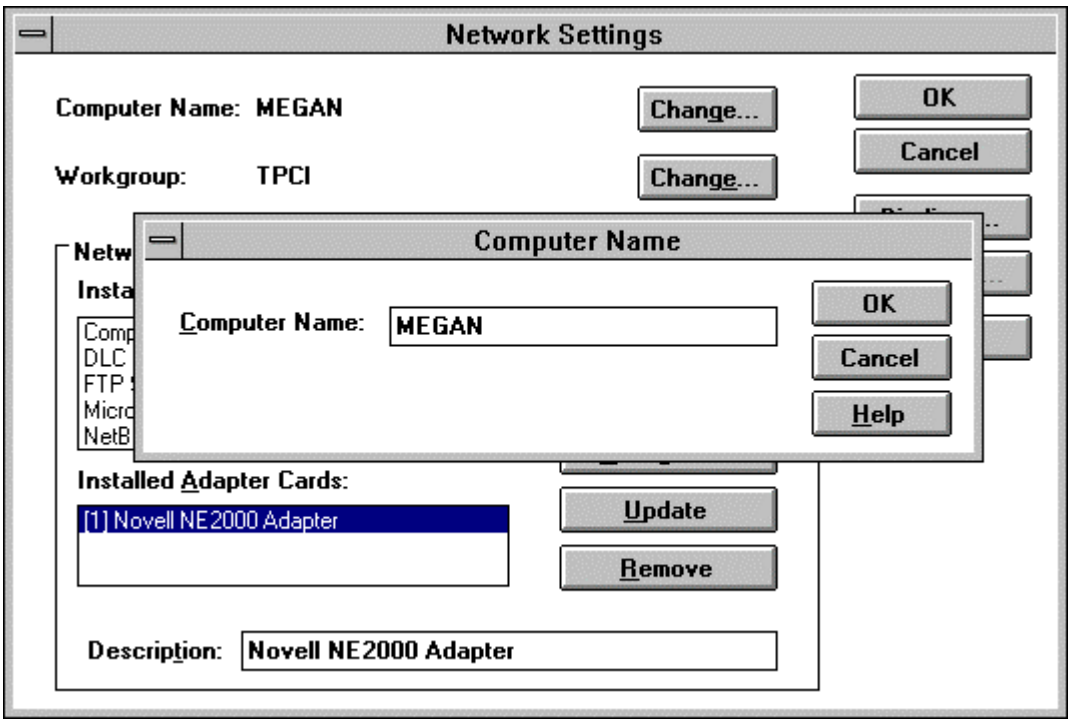
---

[37] Linux

[38] tcp/ip suites

[39] built in

[40] network



Un problema potenziale si può avere quando la macchina locale è multihomed[42] cioè basata in diverse reti con un nome ed indirizzo IP differente per ogni rete. Il singolo nome nel file di configurazione in una tale installazione potrebbe non fornire abbastanza informazioni per consentire un corretto instradamento attraverso tutte le connessioni di rete. Questo problema viene incontrato raramente, ma richiede che l'amministratore di rete setti l'hostname per ogni rete in modo attento.

Al di là del semplice problema del nome della macchina mostrato adesso, il sistema hostname[43] è un completo protocollo che abilita l'accesso alle tabelle del Network Information Center per verificare indirizzi e ottenere informazioni circa rete, gateway ed host.

---

[41] Control Panel

[42] multihomed

[43] hostname system

Il Driver Loopback

Questo driver è probabilmente il metodo diagnostico fondamentale e più spesso usato disponibile per un amministratore. Un driver loopback agisce come un circuito virtuale , abilitando informazioni in uscita ad essere immediatamente reinstradate indietro in input. Ciò permette il testing[44] dei circuiti della macchina eliminando ogni influenza esterna, come la rete stessa, gateway, o macchine remote . per convenzione ogni macchina usa l'indirizzo IP 127.0.0.1 per il driver di loopback (chiamato anche indirizzo IP localhost[45]).

Ogni sistema dovrebbe avere un driver di loopback anche se non è in rete. Ciò perché alcune applicazioni insistono nell'avere un indirizzo IP cui possono accedere per funzionare correttamente. Molti server di licenza[46] su una macchina Unix hanno questa necessità, per esempio.

Driver di loopback sono usualmente inseriti[47] come parte del kernel[48] del sistema operativo, o talvolta come un programma di utilità add-on[49]. La maggior parte dei sistemi multiutente[50] impiegano un driver interno. Unix è un buon esempio: nel

---

[44] testing

[45] localhost IP address

[46] license servers

[47] embedded

[48] kernel

[49] add-on

[50] multiuser

kernel vi è un driver[5152] disegnato specificatamente per agire come loopback driver. Il driver è quasi sempre aggiunto automaticamente quando viene installato il sistema operativo, ma alcuni sistemi operativi basati su Unix , incluse alcune versioni di Linux , non hanno questa funzione, e il driver deve essere aggiunto manualmente dall'amministratore di sistema.

Usando il driver di loopback per reindirizzare il flusso di uscita[53] , la scheda di interfaccia di rete[54] (usualmente una scheda Ethernet[55]) viene bypassata. Il driver è utile per testare installazioni di software TCP/IP, poiché esso mostra immediatamente ogni problema con la configurazione locale. Questo può esser fatto prima che la macchina sia fisicamente connessa alla rete o perfino prima che siano insellati l'hardware[56] e il software[57]. Per esempio possiamo usare il driver di loopback per testare la configurazione TCP/IP prima che sia connesso ad una rete usando il comando di ping[58] con il nome o indirizzo IP del localhost , come nell'esempio seguente

```
# ping -c5 localhost
```

---

[51] device

[52] device driver

[53] output stream

[54] network interface card

[55] Ethernet

[56] hardware

[57] software

[58] ping

PING localhost (127.0.0.1): 56 data bytes

64 bytes from localhost (127.0.0.1): icmp_seq=0 ttl=64 time=10 ms

64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0 ms

--- localhost ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 0/2/10 ms

ping -c5 127.0.0.1

PING 127.0.0.1 (127.0.0.1): 56 data bytes

64 bytes from localhost (127.0.0.1): icmp_seq=0 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0 ms

64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0 ms

--- 127.0.0.1 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 0/0/0 ms

Nel precedente esempio è stato usato il commando di ping con l'opzione –c per specificare cinque ping, il primo con il nome di localhost (che /etc/hosts risolve nell'indirizzo IP 127.0.0.1) e poi con l'indirizzo IP stesso. Se uno dei comandi fallisse , indicherebbe che vi è un problema con il file etc/hosts 8se il nome localhost non viene risolto) o con l'installazione TCP/IP (se entrambi i comandi falliscono).

La gestione di ARP[59]

Il programma arp[60] gestisce le registrazioni nelle tabelle Address Resolution Protocol[61] (ARP). Occorre ricordare che ARP fornisce il collegamento tra indirizzo IP e il sottostante indirizzo fisico[62].

Usando arp (o il suo equivalente in altri sistemi operativi) , l'amministratore può creare, modificare o cancellare registrazioni nella tabella ARP. Tipicamente ciò va fatto quando cambia l'indirizzo di rete di una macchina (sia per cambiamenti nel hardware di rete o a causa di uno spostamento fisico).

Usare ifconfig[63]

Il programma ifconfig abilita l'amministratore ad attivare e disattivare interfacce di rete[64] , e a configurarle. L'accesso al programma è in genere riservato ad un

---

[59] arp

[60] arp program

[61] Address Resolution Protocol

[62] physical address

[63] ifconfig

superutente[65] o all'amministratore di rete. Cambiamenti della configurazione[66] possono essere usualmente fatti soltanto prima che il sistema sia completamente funzionale (come nella modalità singolo utente in sistemi Unix). Quando attivato, ifconfig essenzialmente istruisce il livello di rete[67] del kernel a lavorare con l'interfaccia di rete specificata assegnando un indirizzo IP, poi inviando un comando per rendere l'interfaccia attiva nel sistema. Soltanto quando l'interfaccia è attiva il kernel del sistema operativo può inviare e ricevere dati attraverso l'interfaccia.

IL programma ifconfig abilita un amministratore di rete ad effettuare diverse funzioni utili sulla maggior parte dei sistemi operativi:

Attivare o disattivare un'interfaccia

Attivare o disattivare ARP su un'interfaccia

Attivare o disattivare la modalità debugging su un'interfaccia

Assegnare un indirizzo broadcast

Assegnare una maschera subnetwork[68]

Assegnare un metodo di instradamento[69]

Come un esempio, la versione Linux di ifconfig[70] usa il formato generale

---

[64] network interfaces

[65] superuser

[66] configuration

[67] network layer

[68] subnetwork mask

[69] routing method

[70] Linux version of the ifconfig command

ifconfig *interface_type IP_Address*

interface_type è il nome del driver dell'interfaccia (come lo per loopback, ppp per PPP[71]; e eth per Ethernet), e IP_Address è l'indirizzo IP usato da quella interfaccia. Quando usato soltanto con il nome di un'interfaccia, ifconfig usualmente restituisce informazioni sullo stato corrente dell'interfaccia come mostrato nell'esempio seguente. In questo esempio viene effettuata un'interrogazione[72] di una scheda Ethernet (chiamata ec0) e del driver loopback chiamato lo0). I flag di stato[73] dell'interfaccia sono seguiti dall'indirizzo Internet[74], l'indirizzo broadcast[75], e opzionalmente da una maschera di rete[76], che definisce l'indirizzo Internet usato per confronti di indirizzo[77] nel routing[78].

tpci_sco1-12> ifconfig ec0

ec0: flags=807<UP,BROADCAST,DEBUG,ARP>

inet 146.8.12.15 netmask fffff00 broadcast

146.8.12.15

tpci_sco1-13> ifconfig lo0

---

[71] PPP

[72] query

[73] status flags

[74] Internet address

[75] broadcast address

[76] network mask

[77] address comparison

[78] routing

```
lo0: flags=49<UP,LOOPBACK,RUNNING>
        inet 127.0.0.1 netmask ff000000
```

l'esempio precedente mostra che la connessione Ethernet ec0 è attiva (UP), capace di trasmettere in broadcast (BROADCAST), ed è in modalità debugging (DEBUG). Inoltre è attivo ARP (ARP). Ricordiamo che un messaggio broadcast è inviato a tutte le macchine sulla rete locale settando l'indirizzo IP del host con tutti i bit ad 1.

Una volta che sia stato lanciato ifconfig e l'interfaccia sia attiva , molti sistemi operativi richiedono che sia attivato il comando route per aggiungere o rimuovere percorsi nelle tabelle di instradamento del kernel. Ciò è necessario per abilitare la macchina locale a trovare altre macchine. Il formato generale del comando route in sistemi Unix o Linux è il seguente

```
route add|del IP_Address
```

I contenuti correnti delle tabelle di routing possono essere visualizzati su alcuni sistemi inserendo il solo comando route sulla linea di comando. Per esempio, in un sistema Linux settato solo con il driver di loopback si vede un risultato come questo

```
$ route
Kernel Routing Table
Destination    Gateway   Genmask   Flags  MSS  Window  Use Iface
loopback       *      255.0.0.0   U    1936  0       16  lo
```

Si può forzare la visualizzazione degli indirizzi IP in luogo dei nomi simbolici usando l'opzione –n

```
$ route -n
Kernel Routing Table
Destination    Gateway   Genmask   Flags  MSS  Window  Use Iface
127.0.0.1      *         255.0.0.0   U    1936  0       16  lo
```

Non tutte le versioni Unix o Linux mostrano questo tipo di risultati con il comando route.

L'uso di ifconfig e del programma route può essere mostrato nel setup della connessione Ethernet del sistema Slackware Linux[79]. Per rendere l'interfaccia Ethernet attiva , il comando ifconfig è dato con il nome del device Ethernet (eth0 in un sistema Slackware) e l'indirizzo locale IP. Per esempio il comando

```
ifconfig eth0 147.123.20.1
```

setta la macchina locale con l'indirizzo IP 147.123.20.1. l'interfaccia è il device Ethernet /dev/eth0. l'interfaccia può poi essere controllata con il comando ifconfig usando il nome di interfaccia

```
$ ifconfig eth0
eth0    Link encap 10Mps: Ethernet Hwaddr
   inet addr 147.123.20.1 Bcast 147.123.1.255 Mask 255.255.255.0
   UP BROADCAST RUNNING  MTU 1500 Metric 1
   RX packets:0 errors:0 dropped:0 overruns:0
```

[79] Slackware Linux

```
TX packets:0 errors:0 dropped:0 overruns:0
```

Si può vedere da questo output che l'indirizzo broadcast è stato settato sulla base dell'indirizzo IP della macchina locale. Esso è usato da TCP/IP per accedere a tutte le macchine nella rete locale in un colpo solo. L'ampiezza della Message Transfer Unit[80] (MTU[81]) è usualmente settata al valore massimo di 1500 (per reti Ethernet).

Poi viene aggiunta una registrazione alle tabelle di instradamento del kernel per premettergli di conoscere l'indirizzo di rete della macchina locale. L'indirizzo IP che viene usato con il comando route non è l'indirizzo IP della nostra macchina locale, ma quello della rete come un tutto senza l'identificatore locale. Per settare l'intera rete locale come un tutto , si usa l'opzione –net del comando route. Nel caso dell'indirizzo IP mostrato prima il comando sarebbe:

```
route add -net 147.123.20.0
```

questo aggiunge tutte le macchine della rete identificate dall'indirizzo di rete 147.123.20.0 alla lista di macchine accessibili del kernel. Un metodo alternativo è quello di usare il file /etc/networks

IL Daemon inetd[82]

Il programma inetd è un residuo dai primi tempi dello sviluppo di TCP/IP per Unix. Quando si avviava una macchina , TCP/IP si avviava e immediatamente accettava

---

[80] Message Transfer Unit

[81] MTU

[82] inetd

connessioni[83] alle sue porte, attivando[84] un processo per ciascuna. Ciò poteva risultare in molti processi identici , uno per ogni porta[85] disponibile.

Per controllare meglio i processi , il metodo inetd fu sviluppato per gestire le connessioni delle porte, togliendo quel compito dal server. La differenza primaria è che inetd crea un processo per ogni connessione che viene stabilita mentre il server crea un processo[86] per ogni porta (il che porta a molti processi inutilizzati).

Su molti sistemi alcuni dei programmi di test e utilità di informazioni di stato sono condotti attraverso inetd. Tipicamente processi come echo, discard[87] e time[88] usano inetd.

Il programma inetd usa un file di configurazione usualmente chiamato /etc/inetd.cfg, /etc/inetd/conf, o /etc/inetd.cf su sistemi Unix. Un esempio è il seguente

```
#      @(#)inetd.conf     5.2 Lachman System V STREAMS TCP  source

#

#     System V STREAMS TCP - Release 4.0

ftp       stream     tcp     nowait     NOLUID     /etc/ftpd       ftpd
```

---

[83] connections

[84] spawning

[85] ports

port

[86] process

process

[87] discard

[88] time command

```
telnet    stream    tcp    nowait    NOLUID    /etc/telnetd    telnetd

shell     stream    tcp    nowait    NOLUID    /etc/rshd       rshd

login     stream    tcp    nowait    NOLUID    /etc/rlogind    rlogind

exec      stream    tcp    nowait    NOLUID    /etc/rexecd     rexecd

finger    stream    tcp    nowait    nouser    /etc/fingerd    fingerd

comsat    dgram     udp    wait      root      /etc/comsat     comsat

ntalk     dgram     udp    wait      root      /etc/talkd      talkd

echo      stream    tcp    nowait    root      internal

discard   stream    tcp    nowait    root      internal

chargen   stream    tcp    nowait    root      internal

daytime   stream    tcp    nowait    root      internal

time      stream    tcp    nowait    root      internal

echo      dgram     udp    wait      root      internal

discard   dgram     udp    wait      root      internal

chargen   dgram     udp    wait      root      internal

daytime   dgram     udp    wait      root      internal

time      dgram     udp    wait      root      internal
```

le colonne mostrano il nome del servizio (che corrisponde ad una registrazione nel file dei servizi) il tipo di socket[89] (stream, raw o datagramma[90]), il nome del protocollo , se inetd può accettare ulteriori connessioni alla stessa porta immediatamente (nowait) o deve aspettare che il server finisca (wait), il login che possiede il servizio, il nome del programma del server, ed altri parametri opzionali necessari.

IL comando netstat[91]

Il comando netstat o una utilità similare forniscono informazioni comprensive circa il sistema locale e l'implementazione TCP/IP. È il programma più usato dagli amministratori per diagnosticare rapidamente un problema con TCP/IP. L'informazione reale e il suo formato differiscono con l'implementazione del sistema operativo, ma è usualmente fornito il seguente importante sommario:

Punti terminali della comunicazione[92]

Statistiche dell'interfaccia di rete[93]

Informazioni sui buffer dati[94]

Informazioni sulla tabella di instradamento[95]

---

[89] socket

[90] datagram

[91] netstat

[92] Communications end points

[93] Network interface statistics

[94] data buffers

Statistiche sui protocolli[96]

Su alcuni sistemi, informazioni circa le comunicazioni[97] interprocesso ed altri stack[98] di protocollo possono essere aggiunte.

Communications end points

Il comando netstat senza opzioni fornisce informazioni su tutti i terminali di comunicazione attivi. Per visualizzare tutti i terminali (attivi e passivi), netstat usa l'opzione –a.

Il risultato è formattato in colonne che mostrano il protocollo (Proto), l'ammontare di dati nelle code di ricezione ed invio (Recv-Q e Send-Q) , gli indirizzi locale e remoto, e lo stato corrente della connessione. Un esempio parziale del risultato è il seguente

```
$ netstat -a

Active Internet connections (including servers)

Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)

ip         0      0  *.*                    *.*

tcp        0   2124  tpci.login             merlin.1034            ESTABL.

tcp        0      0  tpci.1034              prudie.login           ESTABL.

tcp    11212      0  tpci.1035              treijs.1036            ESTABL.
```

---

[95] Routing table information

[96] Protocol statistics

[97] communications

[98] protocol stacks

```
tcp        0      0  tpci.1021              reboc.1024             TIME_WAIT

tcp        0      0  *.1028                 *.*                    LISTEN

tcp        0      0  *.*                    *.*                    CLOSED

tcp        0      0  *.6000                 *.*                    LISTEN

tcp        0      0  *.listen               *.*                    LISTEN

tcp        0      0  *.1024                 *.*                    LISTEN

tcp        0      0  *.sunrpc               *.*                    LISTEN

tcp        0      0  *.smtp                 *.*                    LISTEN

tcp        0      0  *.time                 *.*                    LISTEN

tcp        0      0  *.echo                 *.*                    LISTEN

tcp        0      0  *.finger               *.*                    LISTEN

tcp        0      0  *.exec                 *.*                    LISTEN

tcp        0      0  *.telnet               *.*                    LISTEN

tcp        0      0  *.ftp                  *.*                    LISTEN

tcp        0      0  *.*                    *.*                    CLOSED

udp        0      0  *.60000                *.*

udp        0      0  *.177                  *.*

udp        0      0  *.1039                 *.*

udp        0      0  *.1038                 *.*

udp        0      0  localhost.1036         localhost.syslog
```

```
udp        0      0  *.1034             *.*
udp        0      0  *.*                *.*
udp        0      0  *.1027             *.*
udp        0      0  *.1026             *.*
udp        0      0  *.sunrpc           *.*
udp        0      0  *.1025             *.*
udp        0      0  *.time             *.*
udp        0      0  *.daytime          *.*
udp        0      0  *.chargen          *.*
udp        0      0  *.route            *.*
udp        0      0  *.*                *.*
```

Nel precedente esempio vi sono tre connessioni attive, identificate dallo stato ESTABL: Uno sta inviando dati (come mostrato nella colonna Send-Q), è l'altro ha dati in ingresso nella coda. I nomi di rete e i numeri di porta delle connessioni sono mostrati quando possibile. Un asterisco significa che non vi è end point associato con quel indirizzo.

Una connessione è in attesa di essere sospesa[99] , identificata da TIME_WAIT nella colonna di stato. Dopo 30 secondi , queste sessioni vengono terminate e le

[99] hang up

connessioni liberate. Ogni riga con LISTEN come stato non ha connessioni al momento, e sta aspettando. Non vi è colonna di stato per sessioni UDP poiché esse non hanno connessione end-to-end[100].

Una registrazione CLOSED mostra che una connessione è chiusa ma non è stata portata allo stato LISTEN ancora.

Network Interface Statistics

Il comportamento dell'interfaccia di rete può esser determinato con l'opzione –i del comando netstat. Esso mostra rapidamente all'amministratore se vi sono problemi rilevanti con la connessione di rete.

Il comando netstat –i mostra il nome dell'interfaccia, il massimo numero di caratteri che può contenere un pacchetto , gli indirizzi o nomi della rete e del host , il numero di pacchetti in input, gli errori di input, i pacchetti in uscita , gli errori in uscita e il numero di collisioni che si sono avute nella corrente sessione campione. La colonna delle collisioni[101] ha senso solo con protocolli che consentono la collisione come Ethernet. Un esempio del risultato del comando è mostrato ora

```
$ netstat -i

Name  Mtu  Network     Address      Ipkts  Ierrs Opkts  Oerrs Collis

ec0   1500 tpci        merlin       34     0     125    0     0
```

---

[100] end-to-end connection

[101] collision

collisions

```
lan0   1497  47.80       tpci_hpws4  11625     0   11625     0     0

lo0    8232  loopback    localhost     206     0     206     0     0
```

Si possono ottenere informazioni più specifiche su un'interfaccia usando l'opzione –I con il nome del device ed un intervallo di tempo, specificato in secondi , ad esempio netstat –I ec0 30 da informazioni sul comportamento dell'interfaccia Ethernet ec0 negli ultimi 30 secondi.

Data buffer

Informazioni sui buffer dati possono essere ottenute con l'opzione netstat –m. monitorare il comportamento dei buffer è importante, poiché essi hanno un impatto diretto sulle prestazioni di TCP/IP. Il risultato del comando differisce in funzione della versione di Unix in uso , riflettendo le diverse implementazioni del codice TCP/IP. Qui abbiamo un esempio del risultato del comando con la versione Unix basata su System V . le registrazioni sono fornite per lo streamhead[102], coda, tabella dei descrittori del messaggio, tabella del descrittore dei dati[103] e le differenti classi di tabelle di descrittori dei dati. Le colonne mostrano il numero di blocchi configurati (config) e correntemente allocati (alloc), il numero di colonne free (free), il numero totale di blocchi in uso (total), il massimo numero di blocchi che erano in uso ad un certo istante (max), e il numero di volte che un blocco non era disponibile (fail).

---

[102] streamhead

[103] data

$ netstat -m

streams allocation:

	config	alloc	free	total	max	fail
streams	292	79	213	233	80	0
queues	1424	362	1062	516	368	0
mblks	5067	196	4871	3957	206	0
dblks	4054	196	3858	3957	206	0
class 0, 4 bytes	652	50	602	489	53	0
class 1, 16 bytes	652	2	650	408	4	0
class 2, 64 bytes	768	6	762	2720	14	0
class 3, 128 bytes	872	105	767	226	107	0
class 4, 256 bytes	548	21	527	36	22	0
class 5, 512 bytes	324	12	312	32	13	0
class 6, 1024 bytes	107	0	107	1	1	0
class 7, 2048 bytes	90	0	90	7	1	0
class 8, 4096 bytes	41	0	41	38	1	0

total configured streams memory: 1166.73KB

streams memory in use: 44.78KB

maximum streams memory used: 58.57KB

Per l'amministratore la colonna fail è importante. Essa dovrebbe sempre mostrare zero. Se appare un numero più grande, quella risorsa è stata oberata e il numero di blocchi assegnatigli dovrebbe essere incrementato.

Informazioni sulla tabella di instradamento

Le tabelle di instradamento vengono continuamente aggiornate per riflettere connessioni ad altre macchine . Per ottenere informazioni circa queste tabelle si usano le opzioni netstat –r ed –rs (l'ultima genera statistiche sulle tabelle).

I risultati di questi comandi sono mostrati nel seguente esempio. le colonne mostrano la macchina di destinazione, l'indirizzo del gateway da usare , un flag per mostrare se il percorso è attivo e se esso porta ad un gateway o ad un host , un contatore di riferimento (refs) che specifica quante connessioni possono usare simultaneamente quel percorso , il numero di pacchetti che sono stati mandati lungo quel percorso (Use), e il nome dell'interfaccia

```
$ netstat -r

Routing tables

Destination      Gateway          Flags    Refs     Use  Interface

localhost        localhost        UH         4      10  lo0

merlin           localhost        UH         2       2  ec0

treijs           hoytgate         UG         0       0  ec0

47.80            bcarh736         U         12   21029  lan0

tpci sco4-57> netstat -rs
```

routing:

0 bad routing redirects

0 dynamically created routes

0 new gateways found unreachable

2 destinations found unreachable

122 uses of a wildcard route

0 routes marked doutbful

0 routes cleared of being doubtful

0 routes deleted

Statistiche dei protocolli

Statistiche sul comportamento complessivo dei protocolli di rete possono essere ottenute con il comando netstat –s. questo usualmente fornisce sommari per IP[104], ICMP[105], TCP[106] e UDP[107]. L'output di questo comando è utile per determinare dove è stato localizzato un errore in un pacchetto ricevuto, che porta poi l'utente a isolare se l'errore è stato causato da un problema software o hardware.

Un esempio di risultato è il seguente

---

[104] IP

[105] ICMP

[106] TCP

[107] UDP

```
tpci_sco4-67> netstat -s

ip:

      183309 total packets received

      0 bad header checksums

      0 with size smaller than minimum

      0 with data size < data length

      0 with header length < data size

      0 with data length < header length

      0 with unknown protocol

      13477 fragments received

      0 fragments dropped (dup or out of space)

      0 fragments dropped after timeout

      0 packets reassembled

      0 packets forwarded

      0 packets not forwardable

      75 no routes

      0 redirects sent

      0 system errors during input

      309 packets delivered

      309 total packets sent
```

0 system errors during output

0 packets fragmented

0 packets not fragmentable

0 fragments created

icmp:

1768 calls to icmp_error

0 errors not generated because old message was icmp

Output histogram:

destination unreachable: 136

0 messages with bad code fields

0 messages < minimum length

0 bad checksums

0 messages with bad length

Input histogram:

destination unreachable: 68

0 message responses generated

68 messages received

68 messages sent

0 system errors during output

tcp:

9019 packets sent

6464 data packets (1137192 bytes)

4 data packets (4218 bytes) retransmitted

1670 ack-only packets (918 delayed)

0 URG only packets

0 window probe packets

163 window update packets

718 control packets

24 resets

9693 packets received

4927 acks (for 74637 bytes)

37 duplicate acks

0 acks for unsent data

5333 packets (1405271 bytes) received in-sequence

23 completely duplicate packets (28534 bytes)

0 packets with some dup. data (0 bytes duped)

38 out-of-order packets (5876 bytes)

0 packets (0 bytes) of data after window

0 window probes

134 window update packets

0 packets received after close

0 discarded for bad checksums

0 discarded for bad header offset fields

0 discarded because packet too short

0 system errors encountered during processing

224 connection requests

130 connection accepts

687 connections established (including accepts)

655 connections closed (including 0 drops)

24 embryonic connections dropped

0 failed connect and accept requests

0 resets received while established

5519 segments updated rtt (of 5624 attempts)

5 retransmit timeouts

0 connections dropped by rexmit timeout

0 persist timeouts

0 keepalive timeouts

0 keepalive probes sent

0 connections dropped by keepalive

0 connections lingered

0 linger timers expired

0 linger timers cancelled

0 linger timers aborted by signal

udp:

0 incomplete headers

0 bad data length fields

0 bad checksums

68 bad ports

125 input packets delivered

0 system errors during input

268 packets sent

L’utilità ping[108]

L’utilità ping (Packet[109] Internet Groper[110]) è usato per interrogare un altro sistema per assicurare che una connessione è ancora attiva. Il comando ping è disponibile sulla maggior parte dei sistemi operativi che implementano TCP/IP.

---

[108] ping

[109] packet

[110] Packet Internet Groper

Il programma ping opera inviando in uscita una richiesta echo Internet Control Message Protocol [111](ICMP[112]) . se il software IP della macchina di destinazione riceve la richiesta ICMP, manda un echo immediatamente. La macchina mittente continua a mandare una richiesta echo[113] finché il programma ping non viene terminato con una sequenza di break[114] (Ctrl+C o il tasto Delete in Unix). Dopo essere terminato , ping visualizza un set di statistiche. Un esempio è mostrato adesso

```
$ ping merlin

PING merlin: 64 data bytes

64 bytes from 142.12.130.12: icmp_seq=0.  time=20.  ms

64 bytes from 142.12.130.12: icmp_seq=1.  time=10.  ms

64 bytes from 142.12.130.12: icmp_seq=2.  time=10.  ms

64 bytes from 142.12.130.12: icmp_seq=3.  time=20.  ms

64 bytes from 142.12.130.12: icmp_seq=4.  time=10.  ms

64 bytes from 142.12.130.12: icmp_seq=5.  time=10.  ms

64 bytes from 142.12.130.12: icmp_seq=6.  time=10.  ms

--- merling PING Statistics ---

7 packets transmitted, 7 packets received, 0% packet loss
```

---

[111] Internet Control Message Protocol

[112] ICMP

[113] echo request

[114] break sequence

round-trip (ms) min/avg/max = 10/12/20

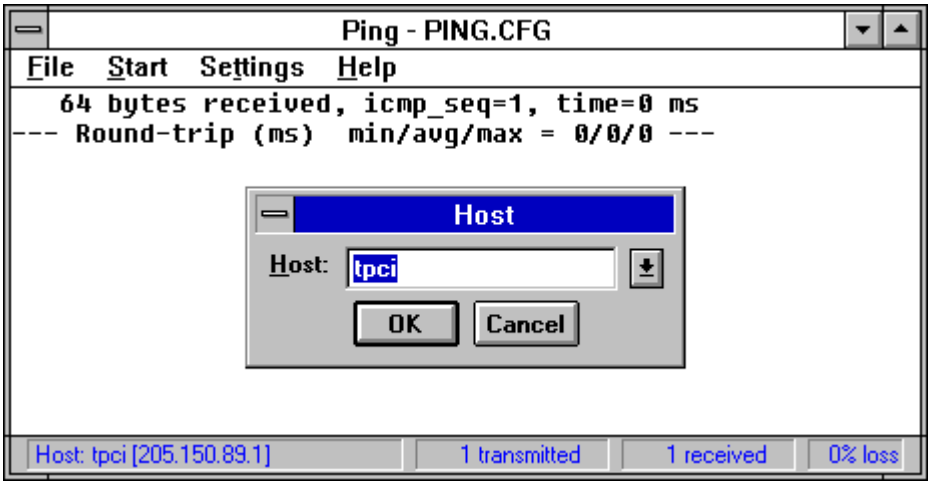
un metodo alternativo per invocare ping è di fornire il numero di volte si vuole interrogare il sistema remoto. Si può fornire anche una lunghezza di pacchetto come test. L'esempio seguente istruisce ping ad usare pacchetti da 256 byte e a tentare cinque volte. Usare ping per inviare grossi pacchetti è un metodo per determinare il comportamento della rete con pacchetti di grosse dimensioni, specialmente quando si ha la frammentazione[115]. IL programma ping è inoltre utile per monitorizzare i tempi di risposta della rete , osservando il tempo di risposta su pacchetti inviati mentre il carico della rete varia. Questa informazione può essere veramente utile nell'ottimizzazione di TCP/IP.

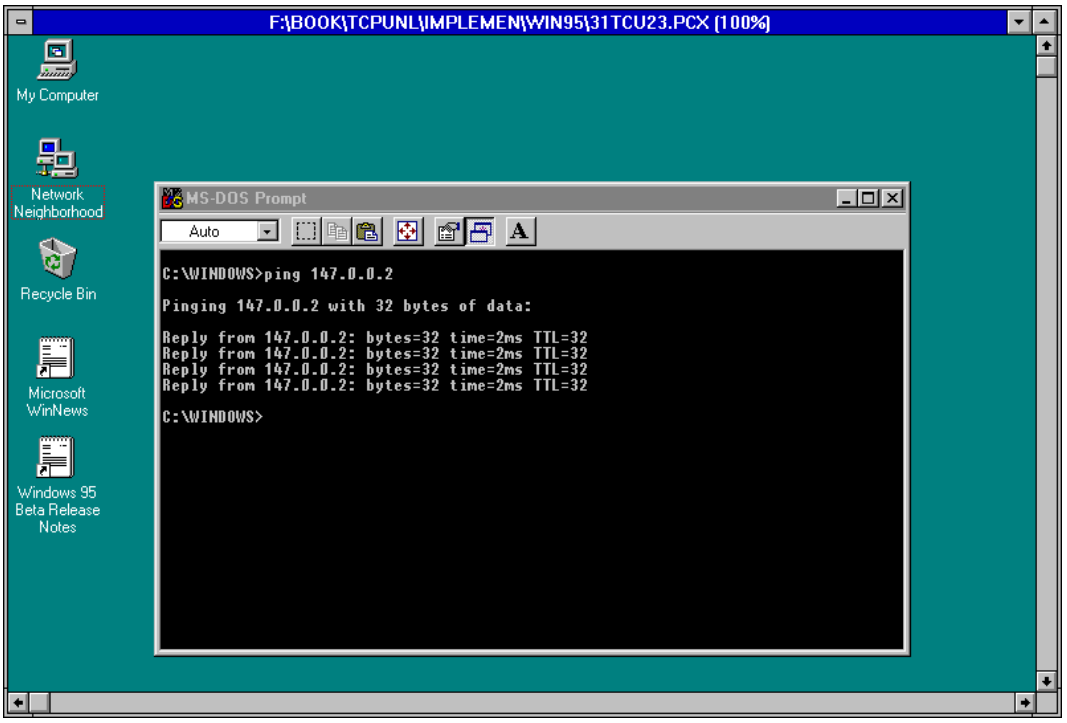
```
$ ping merlin 256 5

PING merlin: 256 data bytes

256 bytes from 142.12.130.12: icmp_seq=0.  time=20.  ms

256 bytes from 142.12.130.12: icmp_seq=1.  time=10.  ms

256 bytes from 142.12.130.12: icmp_seq=2.  time=10.  ms

256 bytes from 142.12.130.12: icmp_seq=3.  time=20.  ms

256 bytes from 142.12.130.12: icmp_seq=4.  time=10.  ms

--- merling PING Statistics ---

5 packets transmitted, 5 packets received, 0% packet loss
```

[115] fragmentation

round-trip (ms) min/avg/max = 10/13/20

La maggior parte delle implementazioni TCP non Unix forniscono un'utilità come parte della loro suite





Tracciare una connessione[116]

Vi è una opzione di tracciamento integrata in TCP/IP. Questa opzione può essere usata per tracciare un problema. Per attivare il tracciamento , una chiamata di sistema[117] è inviata al punto finale che attiva un flag. Quando il tracciamento è attivato, tutte le attività hanno un eco ad un buffer[118] o sullo schermo[119] , in dipendenza della configurazione del sistema.

L'output dell'opzione di tracciamento viene esaminato usando il programma trpt (trace report).

[116] tracing a connection

[117] system call

[118] buffer

[119] screen