

Facilități TCP/IP

Versiunea Linux-ului de TCP/IP are mai multe programe utilitare care oferă informații de stare și statistici despre performanțele rețelei. Sunt disponibile mai multe facilități pentru debugging care ajută programatorul sau utilizatorul ce are cunoștințele necesare să localizeze problemele rețelei. Acest capitol examinează mulțimea de bază a acestor instrumente. Se începe cu o privire asupra fișierelor primare de configurare implicate în TCP/IP. Deși aceste fișiere au fost discutate în capitolele anterioare, merită să le analizăm din nou în detaliu. Nu toate dintre aceste instrumente și fișiere de configurare vor fi oferite cu fiecare versiune de Linux, în special din cauză că două variante (BSD și System V) ale acestor facilități sunt în distribuția generală. Vă puteți verifica pachetul software pentru a vedea ce facilități vi s-au oferit. Dacă aveți nevoie de o facilitate care nu a fost inclusă, puteți face download de pe site-urile BBS sau FTP și sperați că nu vor apărea incompatibilități! Majoritatea comenzilor și facilităților menționate în acest capitol nu sunt disponibile tuturor utilizatorilor, deși superuserii le pot accesa.

Fișiere de configurare

Mai multe fișiere sunt implicate în specificarea completă a adreselor și configurației rețelei pentru TCP/IP. Linux-ul permite comentarii pe fiecare linie a acestor fișiere de configurare, atât timp cât acestea au înaintea lor semnul diez (#). Multe sisteme Linux vor avea fișiere de configurare default, goale, cu intrări default comentate până când administratorul de sistem înlătură simbolurile de comentariu.

Denumiri simbolice de mașini: /etc/hosts

O denumire simbolică este o alternativă folosirii unei adrese IP. De exemplu, este mult mai ușor să numești o mașină vecină darkstar decât 147.23.13.32. Când un nume simbolic este folosit de aplicație ca o adresă, software-ul TCP/IP trebuie să fie capabil să transforme acel nume într-o adresă de rețea (TCP/IP folosește numai adrese IP). Fișierul ASCII /etc/hosts este folosit de obicei, cu nume simbolice atașate adreselor rețelei. (Observați că fișierul /etc/hosts nu este utilizat când sunt folosite sistemele Yellow Pages (YP), Network Information Services (NIS), sau Domain Name Server (DNS). Aceste servicii au propriile fișiere de configurare.)

Linux-ul folosește fișierul /etc/hosts pentru a păstra adresele de rețea și denumirile simbolice, dar și o conexiune numită loopback (care este examinată mai târziu în acest capitol în secțiunea "Driver Loopback"). Adresa conexiunii loopback este de obicei listată ca numele mașinii sau gazda locală.

Fișierul /etc/hosts conține adresele de rețea într-o coloană și denumirile simbolice în alta. Deși adresele de rețea pot fi specificate în format zecimal, octal sau hexazecimal, cel zecimal este cel mai des folosit (și folosirea altor formate poate crea confuzii). Puteți specifica mai multe denumiri simbolice pe linie, separându-le prin spații (spații sau tab-uri). Urmează o mostră dintr-un fișier /etc/hosts din Linux:

```
# network host addresses
```

```
127.0.0.1 localhost local merlin_server
157.40.40.12 artemis
157.40.40.2 darkstar
143.10.12.62 big_bob
153.21.63.1 tpci_server tpci_main tpci
191.13.123.4 kitty_cat
```

Când un utilizator sau o aplicație specifică un nume simbolic, Linux-ul caută în fișierul /etc/hosts un nume care să se potrivească și apoi citește adresa corespunzătoare de pe aceeași linie. Puteți schimba oricând conținutul fișierului /etc/hosts și schimbările au un efect imediat.

Denumiri de rețea: /etc/networks

Capitolul 30, "Configurarea TCP/IP", menționa fișierul /etc/networks. Acest fișier permite rețelelor să fie adresate cu un nume simbolic, la fel ca mașinile, în loc de adresa lor IP. Pentru a rezolva numele de rețea, este folosit fișierul /etc/networks care specifică denumirile simbolice de rețea. Formatul fișierului ne dă denumirea simbolică a unei rețele, adresa ei și orice alias care ar putea fi folosit. O mostră dintr-un fișier /etc/networks:

```
# local network names
tpci 146.1 tpci_network tpci_local
bnr 47.80 BNR bnr.ca
big_net 123.2.21
unique 89.12323 UNIQUE
loopback 127 localhost
```

Ultima intrare în fișier dă numele de loopback. Prima intrare specifică numele mașinii locale, adresa ei de rețea și variantele ei de denumiri care pot fi folosite în aplicații.

Protocoale de rețea: /etc/protocols

TCP/IP folosește un număr special, numit număr de protocol, pentru a identifica protocolul de transport pe care un sistem Linux îl primește. Aceasta permite ca soft-ul TCP/IP să decodeze corespunzător informațiile pe care le primește. Un fișier de configurare numit /etc/protocols identifică toate protocoalele de transport disponibile în Linux și le dă numerele de protocol respective. Toate sistemele au acest fișier, deși unele intrări pot fi comentate pentru a preveni intruziunile nedorite sau abuzurile.

De obicei fișierul /etc/protocols nu este modificat de administrator. În schimb, fișierul este menținut de soft-ul de rețea și actualizat automat

ca parte a procedurii de instalare. Fișierul conține numele protocolului, numărul lui și orice alias ce poate fi folosit pentru acest protocol. O mostră dintr-un fișier /etc/protocols:

```
# protocols

ip 0 IP # internet protocol, pseudo protocol number

icmp 1 ICMP # internet control message protocol

igmp 2 IGMP # internet group multicast protocol

ggp 3 GGP # gateway-gateway protocol

tcp 6 TCP # transmission control protocol

pup 12 PUP # PARC universal packet protocol

udp 17 UDP # user datagram protocol

idp 22 IDP # WhatsThis?

raw 255 RAW # RAW IP interface
```

Conținutul exact al fișierului /etc/protocols de pe sistemul dumneavoastră poate diferi puțin de fișierul arătat mai sus, dar numerele de protocol și numele sunt probabil foarte asemănătoare. Pot fi listate protocoale adiționale, în funcție de versiunea de Linux și de soft-ul de rețea.

Servicii de rețea: /etc/services

Ultimul fișier de configurare TCP/IP folosit pe majoritatea sistemelor Linux identifică serviciile de rețea existente. Acest fișier se numește /etc/services. Ca și fișierul /etc/protocols, acest fișier, de obicei, nu este modificat de administrator, dar este menținut de software la instalare sau configurare. Se face excepție când fișierul /etc/services are servicii lipsă, pe care soft-ul aplicației nu le-a adăugat automat. În plus, un administrator de sistem poate scurta fișierul /etc/services pentru a asigura securitatea, ca atunci când se setează un firewall pentru rețeaua locală

Fișierul /etc/services este în format ASCII și este format din numele serviciului, un număr de port și tipul protocolului. Numărul portului și tipul protocolului sunt separate prin slash. Urmează alias-urile oricărui serviciu opțional. Urmează un scurt extras dintr-o mostră de fișier /etc/services (fișierul este de obicei destul de lung):

```
# network services

echo 7/tcp

echo 7/udp

discard 9/tcp sink null

discard 9/udp sink null
```

```
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp mail mailx
tftp 69/udp
# specific services
login 513/tcp
who 513/udp whod
```

Majoritatea fișierelor /etc/services vor avea mult mai multe linii pentru că multe servicii TCP/IP sunt suportate de majoritatea versiunilor de Linux. Multe dintre sistemele Linux nu sunt folosite ca firewalls pe Internet sau între LAN-uri, așa că mulți dintre administratorii mașinilor Linux nu vor trebui să se preocupe de conținutul acestui fișier. Pe de altă parte, dacă mașina se va comporta ca un firewall sau vă preocupă foarte mult securitatea, puteți să modificați manual fișierul /etc/services.

Driverul Loopback

Driverul loopback este unul dintre cele mai des folosite și fundamentale instrumente de diagnostic disponibile unui administrator de sistem. Driverul loopback acționează ca un circuit virtual în afara și înăuntrul mașinii gazdă. Toate informațiile de ieșire sunt imediat rerutate către un input. Driverul loopback poate fi folosit pentru testarea circuitelor mașinii eliminând influențele externe (inclusiv placa de rețea, rețeaua însăși, porțile de conectare, sau mașinile de la distanță). Cu driverul loopback vă puteți asigura că mașina locală lucrează corespunzător și că orice probleme apar nu sunt aici, ci mai departe în rețea. Driverul loopback sunt încorporate ca parte a nucleului sistemului de operare Linux.

Deoarece TCP/IP cere o adresă IP destinație pentru a trimite datele, un driver loopback este setat ca o adresă de rețea specială, cu adresa IP 127.0.0.1. Intrările driverului loopback sunt totdeauna făcute în fișierul /etc/hosts, ca mai jos:

```
loopback 127 localhost
```

Driver-ul loopback este deasemenea cunoscut ca localhost, și pot fi folosite ambele nume. Dacă driver-ul loopback nu există deja pe mașină, trebuie creat cu comanda ifconfig. Pentru mai multe informații, vezi Capitolul 30, "Configurarea TCP/IP".

Comanda ifconfig

Cu programul ifconfig se pot activa și dezactiva interfețele rețelei, ca și configurarea acestora. Accesul în programul ifconfig este în general

restricționat la superuser. Cu ifconfig sunt disponibile multe opțiuni, multe dintre care administratorii nu le folosesc niciodată. De cele mai multe ori, ifconfig se folosește numai pentru a activa o interfață, cum se arată în Capitolul 30, "Configurarea TCP/IP".

Formatul comenzii ifconfig respectă de fiecare dată aceeași sintaxă. Sintaxa este:

```
ifconfig interf [adresă [param]]
```

unde interf este numele interfeței, adresă (opțional) este adresa IP sau denumirea simbolică care va fi asignată interfeței (care este verificată în /etc/hosts sau /etc/networks), și param este unul dintre argumentele opționale pentru adresă.

Când este folosită numai cu numele unei interfețe, ifconfig returnează informații despre starea curentă a interfeței, cum se arată în codul următor. În acest exemplu se realizează atât o interogare a plăcii de rețea, cât a driverului loopback. Flag-urile de stare ale interfeței sunt urmate de adresa de Internet, adresa de transmitere (broadcast), și opțional oferă o mască de rețea care definește adresa de Internet folosită pentru compararea adreselor la rutare. Output-ul dumneavoastră poate fi diferit, dar ifconfig ar trebui să arate întotdeauna informații despre interfață (doar dacă una nu a fost definită).

```
$ ifconfig eth0
```

```
eth0 Link encap: 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
```

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

```
$ ifconfig lo
```

```
lo Link encap: Local Loopback
```

```
inet addr 127.0.0.1 Bcast {NONE SET} Mask 255.0.0.0
```

```
UP BROADCAST LOOPBACK RUNNING MTU 2000 Metric 1
```

```
RX packets:0 errors:0 dropped:0 overruns:0
```

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

Output-ul comenzii ifconfig ne arată interfața, caracteristicile care îi sunt asignate, adresa de broadcast și măștile rețelei. MTU înseamnă unitatea maximă de transfer (maximum transfer unit). Dimensiunea unității maxime de transfer este de obicei setată la valoarea maximă pe care tipul interfeței o suportă (1500 pentru rețelele Ethernet). Unele sisteme de operare folosesc câmpul Metric pentru a calcula costul oricărei rute particulare, deși Linux-ul nu folosește acest câmp.

Liniiile RX și TX arată câte pachete de date au fost primite și transmise, numărul total și numărul celor cu erori, de când a început interfața în sesiunea curentă.

Așa cum am menționat anterior, `ifconfig` o listă lungă de argumente opționale pentru a modifica comportarea interfeței. Următoarele argumente sunt disponibile în majoritatea versiunilor de Linux:

`allmulti` -> Acest argument setează modul multicast. În prezent nu are suport în Linux

`-allmulti` -> Acest argument dezactivează modul multicast.

`arp` -> Folosit pentru activarea Address Resolution Protocol care detectează adresele fizice ale mașinilor din rețea. Acest argument este setat implicit.

`-arp` -> Dezactivează ARP. Setează flag-ul caracteristic NOARP.

`broadcast` -> Urmat de adresa de broadcast a rețelei, acest argument setează adresa folosită pentru adresarea tuturor mașinilor din rețea. Acest argument este folosit dacă adresa de broadcast este diferită de adresa normală calculată de TCP/IP pe baza tipului rețelei.

`down` -> Face interfața inutilizabilă pentru software-ul IP până când este folosit `up`.

`metric` -> Setează valoarea `metric` pentru interfață. Deși Linux-ul nu folosește acest argument, este inclus pentru compatibilitatea cu implementările TCP/IP mai vechi.

`mtu` -> Urmat de o valoare în bytes, acest argument setează dimensiunea unității maxime de transmitere (numărul de octeți pe care interfața îi poate accepta într-o datagramă). Valorile implicite ale sistemului sunt de obicei precise. (Ethernet 1500, SLIP 296)

`netmask` -> Urmat de o valoare a măștii, acest argument setează masca subnet.

`pointpoint` -> Este folosit pentru interfețele point-to-point IP (PLIP), conectând două mașini printr-un port paralel.

`promisc` -> Setează interfața în modul promiscuu (primește toate pachetele, chiar dacă sunt pentru adresa IP a mașinii sau nu). Folosit pentru analizarea traficului în rețea, acest argument setează flag-ul caracteristic PROMISC.

`-promisc` -> Dezactivează modul promiscuu.

`up` -> Implicat când este dată o adresă, face interfața disponibilă software-ului IP. Când este activ, interfața are caracteristicile UP și RUNNING.

Majoritatea acestor argumente pot fi folosite cu comanda `ifconfig`, deși multe nu sunt necesare pentru o rețea bine configurată.

Demonul `inetd`

Când o mașină Linux a rețelei pornește, activează TCP/IP și acceptă imediat conexiuni la porturile sale, creând un proces pentru fiecare. Pentru a controla mai bine procesele, programul `inetd` a fost dezvoltat să se ocupe singur de conexiunile la porturi, luând această sarcină de la server. O diferență importantă este aceea că `inetd` creează un proces pentru fiecare conexiune stabilă, pe când serverul creează un proces pentru fiecare port (ceea ce duce la prea multe procese nefolosite). Pe multe sisteme, unele dintre programele test și facilități pentru informațiile de stare sunt rulate cu `inetd`.

Programul `inetd` folosește fișierul de configurare `etc/inetd.conf`. Codul următor este un extras dintr-o mostră a fișierului `etc/inetd.conf`. Prima coloană arată numele serviciului (care corespunde unei intrări în fișierul de

servicii, ca /etc/services), tipul socket-ului (flux, brut sau datagramă), numele protocolului, dacă inedit mai poate accepta alte conexiuni la același port imediat (nowait) sau trebuie să aștepte să termine server-ul (wait), al cui este serviciul, numele programului serverului și parametrii opționali de care este nevoie pentru programul serverului.

```
#inetd.conf
```

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

```
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
```

Fișierul /etc/inetd.conf poate fi mult mai mare, dar extrasul de mai sus ne arată formatul general al fișierului. Fișierul /etc/inetd.conf este citit la bootarea serverului și de fiecare dată când un semnal de închidere este primit de la o aplicație. Acest lucru permite schimbări dinamice ale fișierului, orice modificare fiind citită și înregistrată la următoarea citire a fișierului.

Comanda netstat

Programul netstat oferă informații despre sistemul local și sistemul său TCP/IP. Administratorii folosesc de obicei acest program pentru a diagnostica rapid o problemă laTCP/IP. Deși formatul netstat și informațiile specifice diferă la versiunile de Linux, netstat oferă de obicei următoarele sumare, fiecare fiind discutat în detaliu mai târziu:

- * Comunicări end points
- * Statistici despre interfața rețelei
- * Informații despre tabelele de rutare
- * Statistici de protocol

La o versiune următoare vor fi probabil adăugate și informații despre comunicarea între procese și alte stive de protocol. Informația afișată poate fi modificată cu o opțiune din linia de comandă. Următoarele sunt opțiuni valide pentru majoritatea versiunilor de netstat:

- a Arată informații despre toate interfețele
- c Informațiile apar continuu, actualizate la fiecare câteva secunde
- i Arată informații despre interfețe
- n Arată adresa IP în locul denumirilor simbolice
- o Arată informații suplimentare despre starea cronometrului, timpii de expirare și timpii backoff
- r Arată informații despre tabela de rutare nucleu
- t Arată informații numai despre socket-urile TCP
- u Arată informații numai despre socket-urile UDP
- v Arată informații despre versiune
- w Arată informații numai despre socket-urile brute
- x Arată informații despre socket-uri

Output-ul unei instalări Linux tipice ce folosește comanda netstat este arătat în următoarele câteva secțiuni, care discută comanda netstat și output-ul ei mai detaliat. Cum am menționat deja, output-ul și semnificația pot fi diferite la unele versiuni, dar scopul general al instrumentului de diagnostic rămâne același.

Comunicări End Points

Comanda netstat cu nici o opțiune oferă informații despre toate comunicațiile end points active. Pentru a afișa despre un tip particular de end point (punct terminal), folosiți litera tipului din următoarea listă:

- a Toate conexiunile
- t Numai conexiunile TCP
- u Numai conexiunile UDP
- w Numai conexiunile RAW
- x Numai conexiunile socket

Pentru afișarea tuturor end point-urilor care așteaptă o conexiune (pe lângă socket-urile specificate de unul dintre flag-urile de mai sus), netstat folosește opțiunea -a. Opțiunea -a singură va afișa toate socket-urile.

Output-ul este formatat în două coloane care ne arată protocolul (Proto), cantitatea de date din cozile de primire și trimitere (Recv-Q and Send-Q), adresele locală și remote și starea curentă a conexiunii. Urmează un exemplu tăiat de output:

```

$ netstat -ta

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.

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

```

În exemplul precedent, sunt active trei conexiuni care au fost identificate de starea ESTABL. Uneia i s-au transmis date (cum s-a arătat în coloana Sent-Q), în timp ce datele trimise alteia sunt în coadă. Numele de rețea și numerele porturilor ce fac parte din conexiune sunt afișate când este posibil. Asteriscurile arată că nici un end point nu a fost încă asociat acelei adrese.

O conexiune așteaptă să fie închisă și este identificată de TIME_WAIT în coloana de stare. După treizeci de secunde, aceste sesiuni sunt terminate și conexiunile eliberate. Fiecare rând ce are LISTEN în coloana de stare nu are nici o conexiune în acest moment și așteaptă.

Poate fi folosită numai opțiunea -a pentru a afișa o listă completă a tuturor conexiunilor. Output-ul, care este destul de mare, arată la fel, dar include toate conexiunile (active și pasive):

```
$ netstat -a
```

Conexiunile Internet active(inclusiv serverele)

```
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.
```

```
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 *.* *.*

```

Output-ul este similar cu cel pentru opțiunile -ta prezentate anterior, în afară de faptul că au fost adăugate conexiunile UDP (User Datagram Protocol). Sesiunile UDP nu au coloana de stare pentru că ele nu au o conexiune end-to-end.

Statistici ale interfeței rețelei

Comportamentul interfeței de rețea (cum ar fi interfața plăcii de rețea) poate fi arătat folosind opțiunea -i a comenzii netstat. Aceste informații arată imediat dacă există o problemă majoră la conexiunile rețelei.

Comanda netstat -i afișează numele interfeței (Iface), numărul maxim de caractere pe care îl poate conține un pachet (MTU), valoarea metrică (nu este folosită de Linux) și un set de coloane pentru numărul de pachete primite fără probleme (RX-OK), primite cu erori (RX-ERR), primite dar aruncate (RX-DRP) și primite, dar pierdute datorită overrun-ului (RX-OVR). Pachetele transmise au coloane asemănătoare. Ultima coloană conține o listă de flag-uri setate pentru interfață. Urmează o mostră de output a comenzii netstat -i:

```
$ netstat -i
```

```
Kernel Interface table
```

```
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags
```

```
lo 2000 0 231 0 0 0 231 0 0 0 BLRU
```

```
eth0 1500 0 1230 2 9 12 1421 3 2 1 BRU
```

Acest extras arată că sunt folosite două interfețe: un dispozitiv Ethernet (/dev/eth0) și driver-ul loopback (lo0). În acest caz, se poate

vedea că interfața Ethernet a recepționat câteva pachete rele. Acest lucru este normal datorită naturii sistemului Ethernet, deși dacă acest număr de pachete reprezintă un procentaj prea mare din numărul total de pachete trimise, ar trebui folosite metode de diagnosticare pentru aflarea problemei.

Se pot obține mai multe informații specifice despre o interfață folosind opțiunea `-i` cu un nume de dispozitiv și un interval de timp, specificat în secunde, cum ar fi `netstat -i eth0 30`, pentru a obține informații specifice despre comportarea interfeței "eth0" (Ethernet) în ultimele 30 de secunde. De exemplu, output-ul de mai jos arată activitatea interfeței Ethernet în ultimele 30 de secunde:

```
$ netstat -i eth0 30
```

Kernel Interface table

```
Interface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags
```

```
eth0 1500 0 2341 3 5 112 2111 5 8 8 BRU
```

Coloana `Flags` din output-ul lui `netstat` evidențiază tipurile de flag-uri pe care le-am văzut cu comanda `ifconfig`. Semnificația flag-urilor este arătată în lista următoare:

- B Adresa de broadcast a fost trimisă
- L Driver-ul loopback
- M Modul promiscuu
- N Trailer-urile sunt evitate
- O ARP dezactivat
- P Conexiune point-to-point
- R Running
- U Interfața e activă

Cum se vede din extrasul de mai sus, mai multe flag-uri pot fi combinate într-un singur bloc.

Buffere de date

Versiunile de `netstat` care sunt bazate pe System V UNIX (în loc de BSD UNIX) permit afișarea statisticilor buffer-elor de date. Informațiile despre buffer-ele de date TCP/IP pot fi obținute cu opțiunea `-m` a comenzii `netstat`. Monitorizarea comportamentului buffer-elor este importantă pentru că ei au un impact direct asupra performanțelor TCP/IP. Output-ul comenzii `netstat -m` diferă în funcție de versiunea soft-ului de rețea Linux care este folosit, reflectând implementarea diferită a codului TCP/IP.

Output-ul comenzii `netstat -m` este prezentat mai jos. În această versiune, sunt oferite intrări pentru capul fluxului, coadă, tabela descriptorilor de mesaje (`mblks`), tabela descriptorilor de date (`dblks`), și diferite clase de tabele de descriptori de date. Coloanele arată numărul de blocuri alocate curent (`alloc`), numărul de coloane libere (`free`), numărul total de blocuri care sunt folosite (`total`), numărul maxim de blocuri care sunt folosite în același timp (`max`), și de câte ori un bloc a fost nedisponibil (`fail`).

```

$ 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

```

Coloana fail este importantă și ar trebui să arate mereu zero. Dacă aici apare un număr mare, înseamnă că resursa respectivă a fost suprataxată și numărul de blocuri pentru această resursă trebuie crescut (acțiune urmată de recompilarea nucleului și rebootarea sistemului pentru ca schimbările să aibă efect).

Informații despre tabela de rutare

Tabelele de rutare sunt actualizate continuu pentru a reflecta conexiunile la alte mașini. Pentru a obține informații despre tabela de rutare, sunt folosite opțiunile -r și -rs ale comenzii netstat (ultima generează statistici despre tabelele de rutare).

Output-urile comenzilor netstat -r și netstat -rs sunt arătate mai jos. Coloanele arată mașina destinație, adresa porții care va fi folosite (un asterisc înseamnă că nu va fi folosită nici o poartă), coloana Genmask specifică generalitatea rutei (ce adresă IP se potrivește cu ea), un set de flag-uri, o valoare metrică (nu este folosită), un contor de referințe (Refs)

care specifică câte conexiuni active pot folosi simultan această rută, numărul de pachete care au fost trimise pe această rută (Use), și numele interfeței.

```
$ netstat -r
```

Tabela de rutare nucleu

```
Destination Gateway Genmask Flags Metric Ref Use Iface
```

```
loopback * 255.0.0.0 U 0 0 21 lo
```

```
big_system * 123.23.1.0 UGN 1 0 321 eth0
```

```
small_system * 165.213.14.0 UN 1 0 1213 eth0
```

Flag-urile sunt folosite pentru a arăta diferite caracteristici ale rutei. Flag-uri valide:

- D Generată de ICMP
- G Folosește o poartă
- H Doar o singură gazdă poate fi găsită pe această cale (cum ar fi loopback)
- M Modificată de ICMP
- U Interfața e activă

Opțiunile `-r` și `-rs` pot fi combinate cu opțiunea `-n` pentru a afișa adresele IP ale intrărilor în tabela de rutare, în loc de denumirile simbolice (cum s-a arătat mai sus). Așezarea pe ecran și informațiile afișate de comanda `netstat` pot varia în funcție de implementările Linux, ca în exemplul următor:

```
$ netstat -nr
```

Kernel routing table

```
Destination Gateway Genmask Flags Metric Ref Use Iface
```

```
127.0.0.1 * 255.0.0.0 U 0 0 21 lo
```

```
123.23.1.2 * 123.23.1.0 UGN 1 0 321 eth0
```

```
165.213.14.1m * 165.213.14.0 UN 1 0 1213 eth0
```

Folosind acest flag nu trebuie să mai faceți traducerea adreselor IP.

Statistici de protocol

Versiunile de `netstat` bazate pe System V (opuse majorității versiunilor Linux bazate pe BSD) vă permit să afișați statistici de protocol. Statisticile despre comportarea protocoalelor de rețea pot fi obținute cu ajutorul comenzii `netstat -s`. Aceasta oferă de obicei sumare pentru IP (Internet Protocol), ICMP (Internet Control Message Protocol), TCP (Transmission Control Protocol), și UDP (User Datagram Protocol). Output-ul acestei comenzi este util pentru a determina unde a fost localizată o eroare

într-un pachet primit, și apoi a ajuta utilizatorul să afle dacă eroarea s-a datorat unei probleme de software sau de rețea.

Comanda netstat -s oferă un output interactiv, cum se arată mai jos:

```
$ 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 unsend 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
```

Din nou, așezarea exactă a output-ului se schimbă în funcție de versiunea codului de rețea. Oricum, se pot folosi informațiile de bază în orice format.

Comanda ping

Capitolul 30, "Configurarea TCP/IP", ne arăta cum să folosim comanda ping pentru a verifica dacă interfețele funcționau corespunzător. Putem folosi programul ping (Packet Internet Groper) pentru a verifica dacă conexiunea este încă activă.

Programul ping lucrează trimițând o cerere echo Internet Control Message Protocol (ICMP). Dacă software-ul IP al mașinii destinație primește cererea ICMP, va trimite imediat înapoi un răspuns echo. Mașina inițială va continua să trimită cereri echo până când programul ping se termină cu o secvență break (Ctrl-c sau DEL în UNIX). După terminare, ping afișează un set de statistici. Urmează o mostră a unei sesiuni ping:

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

O metodă alternativă de a folosi ping este de a da numărul de câte ori doriți să interogați mașina de la distanță (remote). Deasemenea, puteți da lungimea pachetului ca test. Comanda următoare instruește ping-ul să folosească pachete de date de 256 bytes și să încerce de cinci ori:

```
$ 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
```

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

Folosirea comenzii ping pentru a trimite pachete mari este o metodă de a determina comportamentul rețelei cu pachetele de dimensiuni mari, mai ales când trebuie să apară fragmentarea. Deasemenea, programul ping este folositor pentru monitorizarea timpului de răspuns al rețelei, observând timpul de răspuns la pachetele trimise când rețeaua (sau mașina) încarcă schimbările. Această informație poate fi foarte folositoare în optimizarea TCP/IP. Câteva implementări mai vechi ale ping-ului pur și simplu răspund cu un mesaj că sistemul de la celălalt capăt este activ (mesajul este de forma "X is alive"). Pentru a rula programul interactiv, trebuie folosită opțiunea -s.

Programul ping este util pentru diagnosticări deoarece ne spune dacă software-ul TCP/IP funcționează corect, dacă un dispozitiv local de rețea poate fi adresat (validându-I adresa), și dacă mașina de la distanță poate fi accesată (din nou validând adresa și testând rutarea). Deasemenea verifică software-ul mașinii de la distanță.

Comanda arp

Programul arp manipulează intrările în tabelele ARP (Address Resolution Protocol) ale sistemului. ARP oferă legătura dintre adresa IP și adresa fizică. Cu arp putem crea, modifica, sau șterge intrările din tabela ARP. Aceste operațiuni trebuiesc efectuate când adresa unei mașini din rețea se schimbă (fie din cauza unei schimbări în hardware-ul rețelei, fie din cauza unei schimbări fizice).

Pentru a folosi programul arp, trebuie respectat unul dintre formatele următoare:

```
arp [-v] [-t tip] -a [numele_gazdei]
```

```
arp [-v] [-t tip] -s numele_gazdei adresa_fizică
```

```
arp [-v] -d numele_gazdei [numele_gazdei...]
```

Când specificăm numele gazdei putem folosi numele simbolic sau adresa IP.

Pentru a afișa intrarea pentru o gazdă sau o adresă IP, folosim primul format arătat mai sus. Dacă nu dăm numele gazdei, sunt arătate toate gazdele. De exemplu, pentru a verifica intrarea ARP pentru mașina de la distanță darkstar, folosim comanda următoare:

```
$ arp -a darkstar
```

```
IP address HW type HW address
```

```
147.12.32.1 10Mbps Ethernet 00:00:C0:5A:3F:C2
```

Această comandă arată că mașina darkstar are adresa IP 147.12.32.1 și este găsită printr-o conexiune Ethernet cu 10Mbps. Putem modifica puțin output-ul folosind opțiunea -t cu un tip specific de interfață. Valorile valide sunt ax25 (AMPR AX .25 networks), ether (10Mbps Ethernet), și pronet (IEEE 802.5 Token Ring). De exemplu, pentru a arăta numai toate conexiunile Ethernet, folosim următoarea comandă:

```
arp -t ether -a
```

Pentru a adăuga o intrare în tabele ARP, folosim al doilea format al comenzii, utilizând opțiunea -s. Când adăugăm o intrare, adresa fizică face referire la adresa fizică a adapter-ului (de obicei șase seturi de cifre hexazecimale separate pe coloane). De exemplu, pentru a adăuga o intrare pentru mașina de la distanță big_cat, folosim comanda

```
arp -s big_cat 00:00:c0:10:A1
```

unde se specifică adresa fizică a rețelei (placa de rețea).

În cele din urmă, ultimul format al comenzii arp arătat mai sus este folosit pentru a șterge intrări din tabela ARP. Acest format poate fi necesar când am adăugat incorect o intrare în tabelă sau rețeaua s-a modificat. Pentru a șterge o intrare pentru mașina x-wing, folosim această comandă:

```
arp -d x-wing
```

Multe alte opțiuni sunt valide în multe versiuni de arp, dar probabil nu va trebui să folosim deloc comanda arp (cu atât mai puțin aceste opțiuni). Dacă aveți nevoie de mai multe informații, paginile man conțin o listă de opțiuni valide și funcțiile lor.

Comanda traceroute

Majoritatea sistemelor Linux au o facilitate numită traceroute, care trimite o serie de datagrame UDP (User Datagram Protocol) spre o mașină destinație. Datagramele sunt construite puțin diferit în funcție de locația lor în fluxul trimis mașinii de la distanță. Primele trei datagrame au un câmp numit Time to Live (TTL) setat pe valoarea unu, cu semnificația că prima dată când un router întâlnește mesajul îl trimite înapoi cu un mesaj de expirare (datagrama a fost ignorată). Următoarele trei mesaje au câmpul TTL setat pe valorile doi, trei, patru și așa mai departe, astfel încât fiecare

router prin care trec mesajele să returneze un mesaj de expirare până când mașina destinație este găsită cu succes.

Output-ul comenzii traceroute arată timpul rotunjit de călătorie a fiecărui mesaj (care este util pentru identificarea gâtuiturilor din rețea) și eficiența algoritmilor de rutare (printr-un număr de routere care pot să nu constituie ruta cea mai bună). Urmează o mostră dintr-un output al comenzii traceroute:

```
$ traceroute black.cat.com
```

```
1 TPCI.COM (127.01.13.12) 51ms 3ms 4ms
```

```
2 BEAST.COM (143.23.1.23) 60ms 5ms 7ms
```

```
3 bills_machine.com (121.22.56.1) 121ms 12ms 12ms
```

```
4 SuperGateway.com (130.12.14.2) 75ms 13ms 10ms
```

```
5 black.cat.com (122.13.2.12) 45ms 4ms 6ms
```

Acest output arată fiecare router care a primit mesajele până când a fost găsită mașina destinație. Comanda traceroute are multe opțiuni ce îi modelează comportarea, acestea fiind explicate în paginile din man. Comanda traceroute este folosită de obicei de către administratorii de sistem sau de rețea când se ivesc probleme de livrare a mesajelor sau când rețeaua pare să se comporte foarte încet. Deoarece majoritatea sistemelor Linux sunt pe rețele locale mici sau independente, este posibil să nu folosim niciodată comanda traceroute.

Comanda rpcinfo

Pentru serviciile RPC (Remote Procedure Call), o facilitate numită rpcinfo poate determina ce servicii RPC sunt active pe mașina locală sau orice sistem de la distanță ce oferă suport pentru RPC. Opțiunile comenzii rpcinfo variază în funcție de implementare, dar toate implementările permit flag-uri pentru a decide ce tip de serviciu să fie activat.

De exemplu, opțiunea -p afișează portmapper-ul local. Următorul exemplu ne arată opțiunile comenzii rpcinfo în versiunea din Slackware Linux, precum și output-ul portmapper-ului.

```
merlin:~# rpcinfo
```

```
Usage: rpcinfo [ -n portnum ] -u host prognum [ versnum ]
```

```
rpcinfo [ -n portnum ] -t host prognum [ versnum ]
```

```
rpcinfo -p [ host ]
```

```
rpcinfo -b prognum versnum
```

```
rpcinfo -d prognum versnum
```

```
merlin:~# rpcinfo -p
```

```
program vers proto port
```

100000 2 tcp 111 portmapper

100000 2 udp 111 portmapper

100005 1 udp 650 mountd

100005 1 tcp 652 mountd

100003 2 udp 2049 nfs

100003 2 tcp 2049 nfs

La fel ca și în cazul comenzii traceroute, cei mai mulți administratori de sistem nu vor trebui niciodată să folosească rpcinfo. Dacă sunteți programator sau administrator de rețea, acestea sunt totuși facilități despre care ar trebui să știți.