Lucrarea 11.

ADMINISTRARE ŞI CONFIGURARE TCP/IP.

Cuprins:

Cu toate că funcţionarea **TCP/IP** este transparentă pentru utilizator, apar uneori probleme de comunicare. Un utilizator care "se respectă" doreşte să poată verifica starea conexiunii, iar dacă este cazul, să poată interveni asupra parametrilor în scopul îmbunătăţirii ei. De multe ori este important să se şie dacă o anumită defecţiune se datorează unei componente din reţea, sau maşinii locale. În orice caz, o bună instalare şi configurare, a aplicaţiilor şi protocoalelor de reţea, poate duce la rezultatele aşteptate. Nici într-un caz, o maşină rău configurată sau administrată, nu va putea fi exploatată în scopul pentru care a fost concepută.

TCP/IP dispune de o suită de programe utilitare şi fişiere de configurare, care pot să furnizeze informaţii de stare şi să verifice performanţele unei reţele construite pe baza acestui protocol.

În continuare vor fi prezentate câteva dintre programele utilitare de bază, întâlnite la aproape toate versiunile de implementare ale **TCP/IP**. Cu toate că această suită de protocoale este standardizată, pot exista diferenţe între versiuni, în general doar de formă nu şi de fond.

Nu toate comenzile prezentate în această secţiune sunt accesibile utilizatorului obişnuit, unele dintre acestea fiind "unelte" de administrare, vor putea fi lansate doar de utilizatori privilegiaţi, cu drepturi de administrare. Comenzile prezentate în continuare vor fi detaliate doar atât cât este nevoie pentru a oferi o imagine de ansamblu asupra **TCP/IP** şi a modului de funcţionare [6, 7].

Fişiere de configurare.

Deoarece **TCP/IP** a fost proiectat iniţial să servească sistemul de operare **UNIX**, a păstrat, după implementarea lui pe alte sisteme, majoritatea fişierelor şi comenzilor de configurare iniţiale. Există totuşi şansa ca o parte dintre acestea să se regăsească sub nume diferite, dar semnificaţia lor este de cele mai multe ori aceeaşi.

În fişierele de configurare **UNIX** este permisă prezenţa comentariilor în orice linie, dacă aceasta începe cu semnul # (diez), spre deosebire de MS Windows unde comentariile încep cu semnul ; (punct şi virgulă). Dacă ne uităm în directorul unde sunt păstrate fişierele de configurare ale sistemului de operare de pe maşina proprie, vom găsi fişiere ca `hosts`, `services` sau `protocols`, care arată la fel. Acestea se găsesc în diferite directoare, în funcţie de sistemul de operare folosit, aşa cum se poate vedea mai jos:

Windows 95/98	`C:\WINDOWS\`,
Windows NT 4 sau 2000	`C:\WINNT\ETC\`,
Windows XP	`C:\WINDOWS\system32\drivers\etc`,
Linux	`/etc/`

De obicei aceste fişiere conţin mai multe linii decât trebuie pentru o configuraţie standard. Multe dintre ele încep cu semnul "#" ceea ce înseamnă că acea linie (de fapt conţinutul ei) nu se ia în seamă de către sistemul de operare. În timpul configurării, dacă este cazul, administratorul de sistem va şterge, sau dimpotrivă va adăuga acolo unde este nevoie, semnul "#" din faţa unor linii.

Un utilizator obişnuit nu va putea, din motive de securitate, să editeze acele fişiere de configurare, în cel mai bun caz le va putea doar citi. Dacă se operează cu drepturi de administrator, se recomandă salvarea unor copii ale fişierelor de configurare iniţiale, pentru cazul în care se fac modificări eronate în ele. Strecurarea unor greşeli în astfel de fişiere poate duce la rezultate imprevizibile, de cele mai multe ori grave, în funcţionarea unui sistem!

Numele simbolice ale maşinilor. Fişierul `/etc/hosts`.

Adresarea şi rutarea pachetelor **IP** în reţea se face pe baza adresei **IP** scrisă, aşa cum s-a putut vedea mai înainte cu ajutorul a 16 biţi. Chiar dacă se foloseşte formatul zecimal de scriere a fiecărui octet (Doted Decima Format - format zecimal cu puncte pentru delimitare), memorarea şi operarea cu aceste adrese se poate face doar în reţele mici, de câteva maşini. Mult mai comodă este utilizarea unor nume simbolice (numite impropriu adresa literală) organizate în structuri ierarhice.

Ori de câte ori o aplicaţie foloseşte un nume (adresă) simbolică, trebuie să se folosească o metodă prin care acestea să fie transformate în adresa **IP** (numerică) corespunzătoare.

Prima şi cea mai simplă metodă de asociere, a celor două tipuri de adrese, este utilizarea unui fişier text (`ASCII`) (`/etc/hosts` la sistemele de operare UNIX) în care sunt trecute într-un tablou formal, adresa **IP** şi numele simbolic.

Acest fişier are utilitate limitată, atunci când sistemul foloseşte alte mecanisme mai evoluate de conversie, cum sunt **YP** (Yellow Pages - pagini aurii), **NIS** (Network Information Services - serviciu de informaţii de reţea) sau **DNS** (Domain Name System - sistemul numelor de domenii). Un astfel de fişier arată ca în exemplul de mai jos:

```
# Acest fişier conţine asocierea adreselor IP cu numele maşinilor.
# Fiecare articoltrebuie să fie trecut într-o singură linie.
# Adresa IP trebuie aşezată pe prima coloană, urmată de numele
# simbolic
# corespunzător şi de numele scurt, dacă este nevoie. Adresa IP şi
# numele simbolic trebuie separate de cel puţin un caracter spaţiu
# sau tab.
# Adresa IP se scrie în format zecimal, sau hexazecimal, cu puncte de separare.
# Se pot introduce linii goale, sau comentarii (ca acesta), în linii
# individuale, sau în continuarea numelui maşinii, dacă încep cu semnul "#".
#
# Exemple:
#
#      102.54.94.97     rhino.acme.com          # server
#       38.25.63.10     x.acme.com              # maşina client
127.0.0.1       localhost.localdomain  localhost # bucla locală
193.226.5.178   masserv.utcluj.ro       masserv
```

```
172.27.192.11   mas2.b25.big.utcluj.ro  mas2
192.168.120.4   mas04.mas.utcluj.ro     mas04
192.168.120.1   Masurari_1.mas.utcluj.ro
192.168.120.1   dns-ftp.mas.utcluj.ro   dns-ftp  # server proxi
172.27.192.1    gate-b25.big.utcluj.ro  gateway

192.168.120.28   TeleTx.mas.utcluj.ro   # Calculatorul meu
#
# Pot exista mai multe nume simbolice asociate la aceeaşi adresă IP,
# dar nu şi invers!
# Se pot introduce oricâte linii sunt necesare şi
# nu contează ordinea în care sunt trecute articolele.
#
# Atenţie, aceste numere sunt doar exemple!
# Utilizarea lor în alte fişiere de configurare
# nu poate decât să încurce puţin lucrurile!
```

Se observă că în acest fişier există o linie care se referă la adresa `127.0.0.1` şi care are numele simbolic `localhost`. Acesta este o interfaţă virtuală de reţea, numită buclă internă (`loopback`) sau maşina locală. Aceasta va fi analizată în cursul 12.

Administratorul de sistem poate să actualizeze fişierul `/etc/hosts` în orice moment deoarece modificarea are efect imediat (nu este necesară re-pornirea sistemului). Ori de câte ori a aplicaţie foloseşte pentru adresare numele simbolic, acest fişier este citit de către componente ale nucleului sistemului de operare şi numele simbolic va fi înlocuit cu adresa IP.

Alte sisteme de operare pot folosi nume şi formate diferite, în locul celui de `hosts`, dar rolul acestora este acelaşi.

Numele reţelelor `/etc/networks`.

Şi reţelele pot fi adresate cu ajutorul numelor simbolice, la fel ca şi maşinile. Pentru a asocia numelui unei reţele adrese numerice, unele implementări TCP/IP folosesc fişierul `/etc/networks`. Aceste fişiere nu sunt apelate des, deoarece sunt rare situaţiile în care este nevoie de adresarea unei reţele întregi. Fişierul conţine numele simbolic, partea din adresa IP care se referă la reţea şi un eventual alias al acesteia, formatul fişierului `networks` fiind asemănător cu cel al fişierului `hosts`. Un exemplu de fişier `/etc/networks` se poate vedea mai jos.

```
# Acest fişier conţine nume/număr de reţea asociate reţelelor locale.
# Adresele de reţea sunt exprimate în format zecimal cu puncte.
#
# Format:
#
# <numele reţelei>  <numărul reţelei>     [alias...]  [#<comentariu>]
#
# Example:
#
#    loopback     127
#    campus       284.122.107
#    london       284.122.108
loopback          127                   localhost
b25.big           172.27.192             retea_locala
# reţea locală
```

Se observă că fişierul `/etc/networks` este diferit de `/etc/hosts` prin faptul că ordinea, dintre numele simbolic şi adresa numerică, este inversată.

Ultimul articol din acest exemplu este cel al buclei locale.

DNS (Domain Name System). Fişierul `/etc/resolv.conf`.

Deoarece numărul terminalelor dintr-o reţea este în continuă creştere, iar asocierea dintre numele simbolice şi adresele **IP**, într-o continuă schimbare, evidenţa ţinută de administratorii de reţea, prin fişierele `hosts` şi `networks`, este imposibilă. De aceea, pentru administrarea listelor se foloseşte o soluţie de evidenţă distribuită.

DNS este un sistem, cu o structură ierarhică, prin care se ţine o evidenţă a numelor simbolice, ale terminalelor şi reţelelor, prin intermediul unor aplicaţii de baze de date, interconectate şi distribuite în reţea.

Pentru aflarea adresei de reţea, în cazul în care se cunoaşte numele simbolic al terminalului destinaţie, componenta nivelului **OSI 3**, care îndeplineşte această funcţie, va trimite un mesaj de interogare unei aplicaţii **DNS** (Server **DNS**, sau de nume). Dacă informaţia se găseşte în listele proprii, acesta va răspunde cererii, iar dacă nu, va interoga, la rândul lui, un server de nume ierarhic superior.

Fiecare sistem de operare, care utilizează suita **TCP/IP**, utilizează opţiuni de configurare pentru server de nume. În sistemele de operare **Linux**, specificarea adresei server-ului de nume, ierarhic superior, se face în fişierul `/etc/resolv.conf` . Conţinutul acestuia este:

```
search utcluj.ro
nameserver 193.226.5.35
nameserver 193.226.5.33
```

Pentru a afla adresa **IP** a unui terminal, când se cunoaşte numele simbolic, se pot folosi comenzi ca `dig` sau `nslookup`. De ecemplu, pentru a afla adresa **IP** a terminalului `www.utcluj.ro` aplicând comanda `nslookup` se obţine, în urma interogării server-ului **DNS** `193.226.5.35`, următorul răspuns al sistemului:

```
root@masserv:~# nslookup www.utcluj.ro
Server:         193.226.5.35
Address:        193.226.5.35#53

Non-authoritative answer:
www.utcluj.ro   canonical name = webhosting.utcluj.ro.
Name:   webhosting.utcluj.ro
Address: 193.226.5.150

root@masserv:~#
```

Ordinea în care se face căutarea depinde de conţinutul fişierului `/etc/bind` al cărui conţinut se poate vedea în liniile următoare:

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

Cuvândul `order` defineşte ordinea în care se face căutarea, în acest caz mai întâi în fişierul `/etc/hosts` , iar dacă răspunsul nu se găseşte acolo, se va interoga server-ele **DNS** (`bind` – ataşat) trecute în fişierul `/etc/resolv.conf`.

Expresia `multi on` indică faptul că în cazul în care sunt asociate mai multe nume simbolice aceleiaşi adrese **IP**, să le accepte pe toate.

Protocoalele încapsulate în pachete IP (de reţea) `/etc/protocols`.

Numerele de protocoale se folosesc pentru identificarea protocoalelor încapsulate în pachete **IP**, pentru ca maşina care recepţionează o datagramă să ştie cum să o decodifice. Încazul suitei **TCP/IP** numărul de identificare a protocolului este trecut în antetul **IP**.

Pentru identificarea protocoalelor, folosite într-o reţea, se foloseşte de obicei un fişier care, la sistemele de operare **UNIX** şi **MS Windows**, se numeşte `protocols (/etc/protocols)`. Acest fişier nu trebuie modificat, de către administratorul de sistem, decât în cazul adăugării de procese de sistem care necesită alte tipuri de protocoale, decât cele existente pe maşină, la un moment dat. Acest lucru se face, de obicei, automat în timpul procesului de instalare a unui nou pachet software. Fişierul `/etc/protocols` conţine numele protocolului, numărul lui de

identificare, şi un alias, dacă este cazul. Mai jos, se poate vedea, ca exemplu, conținutul unui astfel de fişier.

```
# /etc/protocols:
#
# Internet (IP) protocols
#
#       from: @(#)protocols     5.1 (Berkeley) 4/17/89
#
# Actualizate pentru NetBSD bazat pe RFC 1340, Assigned Numbers (Iulie 1992).
#
# Format:
#
# <nume protocol>  <numărul alocat>  [alias...]   [#<comentariu>]

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
hello    63       HELLO   # HELLO Routing Protocol
ospf     89       OSPF    # Open Shortest Path First Routing Protocol
```

Se observă din acest exemplu că **IP** are alocat numărul 0 şi **TCP 6**. Valorile trecute în acest tabel trebuie respectate şi nu se acceptă schimbarea lor, decât în anumite situații bine justificate.

Servicii de rețea `/etc/services`.

Ultimul dintre fişierele comune de configurare, folosit de majoritatea sistemelor de operare, identifică serviciile, sau aplicațiile deservite de protocoalele de transport **TCP** sau **UDP**. Acest fişier nu se editează de către administratorul de sistem, decât în situații foarte rare. În cazul în care se adaugă un nou serviciu, fişierul `/etc/services` va fi editat automat în timpul procesului de instalare, când se vor adăuga liniile care conțin numerele de identificare ale noilor servicii.

Fişierul este în format text `ASCII` şi conține numele serviciului, numărul portului la care acel serviciu aşteaptă datagrame şi tipul protocolului de transport folosit. Numărul portului şi tipul protocolului sunt separate prin semnul "/".

Un exemplu de astfel de fişier se poate vedea într-un paragraf anterior. Fişierul `/etc/services` este în general destul de lung, deoarece trebuie să conțină toate numerele de identificare a serviciilor care se folosesc în acea rețea.

```
# /etc/services:
#
# Servicii de rețea, în manieră Internet
#
# De notat că în atribuțiile IANA intră alocarea numerelor "bine cunoscute"
# de porturi, atât pentru TCP cât şi pentru UDP. De acum încolo, multe numere
# vor avea două articole, chiar dacă protocolul nu suportă transfer prin UDP.
# Actualizat de RFC 1700, "Assigned Numbers" (Octombrie 1994).  Nu sunt incluse
# toate porturile, ci doar cele mai uzuale dintre ele.
#
# Format:
#
# <numele serviciului>  <numărul portului>/<protocol>  [alias...]   [#<comentariu>]
#

tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
```

```
qotd            17/tcp          quote
ftp-data        20/tcp
ftp             21/tcp
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
```

Fişierul `/etc/protocols` ca şi fişierul `/etc/services` sunt folosite de anumite aplicații de administrare pentru a prezenta numele protocolul în forma sa literală, în locul celei nomerice.

Numele maşinii `hostname`

În concepția **TCP/IP** trebuie ca fiecare maşină dintr-o rețea să aibă, pe lângă adresa **IP**, un nume simbolic. Cele mai multe sisteme de operare de rețea folosesc nume (identificatori alfanumerici) pentru a putea deosebi o maşină de alta. Pentru modificarea şi verificarea numelui maşinii se folosesc programe specializate (la sistemele **UNIX**, hostname sau uname, iar în **MSWindows** module pentru rețea din Control Panel).

În cazul sistemelor **UNIX**, numele maşinii se salvează într-un fişier `/etc/hostname`, sau într-unul dintre fişierele prezentate mai sus, care este citit la pornire. Multe dintre aplicații folosesc, în timpul tranzacțiilor, numele maşinii şi de aceea acesta trebuie stabilit corect de la început. Modificarea numelui se poate face şi prin editarea fişierului şi re-pornirea maşinii, sau cu ajutorul utilitarelor specifice care au avantajul că asigură editarea corectă şi schimbarea acestuia fără re-start.

În majoritatea sistemelor de operare **UNIX** şi a versiunilor sale (**Linux, SCO, BSD**) comanda `hostname` şi `uname -n`, sunt cele care au ca răspuns numele maşinii. O astfel de sesiune de comenzi se poate vedea în secvența de mai jos.

```
[root@masserv /root]# hostname
masserv.utcluj.ro
[root@masserv /root]# uname -n
masserv.utcluj.ro
[root@masserv /root]#
```

La majoritatea sistemelor UNIX comanda `hostname` se foloseşte şi pentru a seta numele maşinii. Trebuie însă, ca la majoritatea comenzilor, să fie consultat manualul comenzilor deoarece unele opțiuni pot să fie diferite, de la o versiune la alta.

```
[root@masserv /root]# hostname numeschimbat.utcluj.ro
[root@masserv /root]# hostname
numeschimbat.utcluj.ro
[root@masserv /root]#
```

Pentru sistemele de operare din familia MSWindows, numele maşini se poate seta din panoul de control (Control Panel >Network... Properties >TCP/IP >DNS Configuration), sau din opțiunile My Computer > Properties > Computer Name, aşa cum se poate vedea în **figura 11.1.**

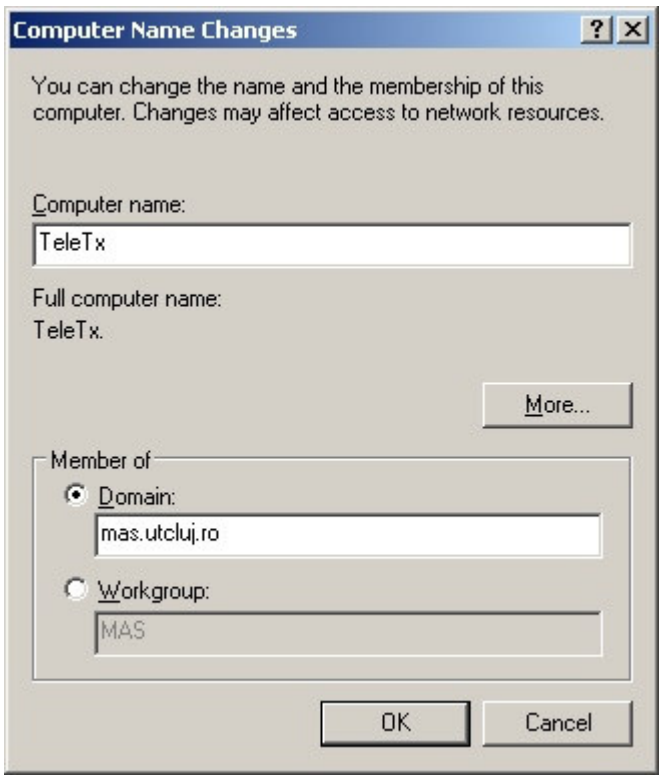

Fig. 11.1. Configurarea numelui simbolic şi al celui de domeniu, la Windows XP.

La sistemul de operare **Windows NT** numele maşinii este specificat cu ajutorul unei casete de dialog din componenţe modulului **Network**. Cu ajutorul acestei casete se poate schimba numele şi noua opţiune devine activă imediat. Cu toate acestea, se recomandă, atât la sistemul **UNIX** cât şi la **Windows**, repornirea calculatorului după efectuarea schimbărilor. Chiar dacă fişierele de configurare sunt actualizate, rămân încă destule locaţii de memorie în care aceste opţiuni rămân neschimbate (zone de memorie cache) şi care se şterg doar după o nouă pornire.

Există situaţii când pe aceeaşi maşină se definesc mai multe nume (multihomed) în funcţie de tipul serviciilor pe care le serveşte, sau în cazul în care aceasta aparţine mai multor reţele. De obicei această problemă se rezolvă prin adăugarea opţiunii de nume la nivelul aplicaţiilor. Se observă din fişierul `hosts` că o maşină (adresă **IP**) poate avea asociate mai multe nume, dar nu şi invers.

Bucla locală `loopback`.

Dispozitivul virtual numit bucla locală (*loopback driver*) este una dintre componentele fundamentale ale unui sistem de operare care foloseşte **TCP/IP**. Bucla locală funcţionează ca un dispozitiv virtual, cu ajutorul căruia se poate construi un circuit prin care se trimit informaţii care sunt imediat rutate spre o intrare (de asemenea virtuală). Aceasta permite testarea proceselor de reţea chiar în absenţa unei interfeţe de reţea (**NIC** - Network Interface Card) reale. *Loopback driver*, fiind o bucla internă, permite verificarea funcţionării componentelor, corespunzătoare nivelelor **OSI** de la 3 până la 7, deoarece bucla se închide deasupra nivelului legăturilor de date. Convenţional, adresa **IP** asociată acestui „dispozitiv virtual" este `127.0.0.1`, iar numele simbolic `localhost` (maşina locală).

Suita **TCP/IP** instalează întotdeauna această componentă, chisr dacă maşina respectivă nu va fi folosită vreo dată într-o reţea. Un administrator poate folosi bucla locală pentru a verifica funcţionarea aplicaţiilor, a componentelor de transport şi de adresare **IP** ale sistemului pe care îl administrează. Nu poate verifica însă, componentele nivelului fizic, deoarece bucla locală se închide la un nivel deasupra acestuia.

Pentru a verifica dacă cele spuse mai sus sunt adevărate, trebuie să apelăm la câteva comenzi specifice de configurare. La sistemele de operare **UNIX** şi la variante ale acestuia (**Linux, SCO** etc.) se poate folosi comanda `ifconfig`, aşa cum se poate vedea în exemplul de mai jos.

```
[root@masserv /root]# ifconfig lo
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
[root@masserv /root]#
```

Se poate vedea cum acestei "interfețe" i s-a atribuit adresa **IP** 127.0.0.1, cu masca de rețea `255.0.0.0`, alături de alte caracteristici care simulează componentele nivelului legăturilor de date (Link encapsulation - încapsularea pachetelor **IP** în cadre de format "`local loopback`"). O altă modalitate de a verifica dacă această interfață virtuală este instalată pe o mașină, este de a verifica dacă adresa acesteia se află în tabela de rute. Dacă există o linie care se referă la aceasta, se trece la utilizarea comenzii `ping`, aşa cum se poate vedea în exemplul de mai jos.

```
[root@masserv /root]# route
Kernel IP routing table
Destination   Gateway    Genmask         Flags Metric Ref  Use Iface
192.168.120.1   *        255.255.255.255 UH    0      0     0  eth1
192.168.120.0   *        255.255.255.0   U     0      0     0  eth1
193.226.5.178   *        255.255.255.255 UH    0      0     0  eth0
193.226.5.176   *        255.255.255.248 U     0      0     0  eth0
127.0.0.0       *        255.0.0.0       U     0      0     0  lo
default         193.226.5.177   0.0.0.0  UG    0      0     0  eth0
[root@masserv /root]#ping localhost -c 3
PING localhost.localdomain (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=0.1 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.3 ms
[root@masserv /root]#
```

Dacă ne apare un răspuns asemănător cu acesta, înseamnă că pe sistem bucla internă este instalată, configurată corect şi, ceea ce este cel mai important, funcționează. Utilizarea comenzii `ping` cu argumentul `localhost` are ca scop verificarea funcționării serviciului *resolver* de asociere a numelui simbolic la o adresă **IP** (listă aflată în fişierul `/etc/hosts`).

Gestiunea ARP.

ARP (Address Resolution Protocol) este o componentă importantă a suitei **TCP/IP**, pentru că permite determinarea adresei de la nivelul legăturii de date (uneori impropriu denumită adresă fizică), când se cunoaşte adresa de rețea (**IP**). Gestiunea acestor asocieri se face, de cele mai multe ori automat, cu ajutorul tabelei **ARP**.

În cazuri speciale, administratorul unui sistem poate să invoce comanda **ARP** pentru a adăuga, şterge, sau modifica anumite poziții din tabelă. O situație în care actualizarea tabelelor trebuie să fie făcută manual este aceea când o maşină, din diferite motive, şi-a schimbat adresa fizică fără să o schimbe pe cea **IP**, iar asocierea anterioară nu a fost încă actualizată. Deşi programele care gestionează acest protocol pot să difere de la o implementare la alta, rolul acestora este acelaşi. Un exemplu de aplicare a comenzii **ARP** în scopul afişării tabelei sau ştergerii unei linii de asociere, se poate vedea mai jos. Mai întâi pentru o maşină pe care este instalat un sistem de operare **Linux**, iar apoi pentru **Windows NT**.

```
[root@masserv /root]# arp -a
? (193.226.5.177) at <incomplete> on eth0
? (192.168.120.111) at 00:00:B4:36:0B:18 [ether] on eth1
[root@masserv /root]#arp -d 192.168.120.111
[root@masserv /root]#arp -a
? (193.226.5.177) at <incomplete> on eth0
[root@masserv /root]#
```

Utilitarul `ifconfig`

Programul `ifconfig` (sau variante ale sale) permite unui administrator să configureze o interfață de rețea şi să o activeze sau dezactiveze. Acesta instruieşte componentele de rețea, ale nucleul sistemului de operare, cum să folosească interfața de rețea specificată (asocierea unei adrese IP, parametri de încapsulare, activare sau dezactivare etc.). Este de remarcat faptul că nucleul poate să folosească o anumită interfață de rețea, să trimită şi să primească date, numai dacă aceasta este activată înainte, de aceea în timpul procesului de inițializare a unui sistem, trebuie prevăzute scripturi de configurare şi activare a interfețelor. Accesul la programul `ifconfig` este restricționat, în general, doar utilizatorii privilegiați (administratori) au drepturi care le permit să modifice componente ale nucleului sistemului de operare. Cele mai des folosite funcții ale acestui program, sunt următoarele:

- activarea sau dezactivarea unei interfețe,
- activarea sau dezactivarea componentei ARP la o interfață,
- activarea sau dezactivarea modului de depanare software a unei interfețe,
- asocierea adresei IP, de broadcast şi a măştii de rețea,
- alegerea metodei de rutare aplicată unei interfețe.

Deoarece formatul, sau opțiunile comenzii, pot să fie diferite în funcție de sistemul de operare pe care este implementat acesta, se recomandă utilizarea documentației specifice. În cazul sistemelor de operare **UNIX** acesta se găseşte sub forma unor fişiere care se pot apela cu comanda:

```
[root@masserv /root]# man <nume_comandă>
```

Pentru sistemul Linux, sintaxa comenzii se poate vedea în exemplul de mai jos:

```
[root@masserv /root]# ifconfig interfață adresa_IP mask masca_de_rețea up
```

unde: *`interfață`* reprezintă tipul interfeței urmat imediat de numărul de identificare al acesteia, *`adresa_IP`* reprezintă adresa **IP** asociată interfeței, *`masca_de_rețea`* reprezintă masca de rețea în formatul zecimal cu punct, iar "`up`" este opțiunea de activare a interfeței. În locul măştii de rețea (`mask` *`masca_de_rețea`*) se poate folosi */prefix,* unde prefixul reprezintă numărul de biți egali cu 1 din masca de rețea.

```
[root@masserv /root]# ifconfig eth1 192.168.122.1 netmask 255.255.255.0 up
sau
[root@masserv /root]# ifconfig eth1 192.168.122.1/24 up
```

Pentru a dezactiva o interfață, se foloseşte comanda cu opțiunea `down` şi fără a defini masca de rețea.

```
[root@masserv /root]# ifconfig eth1 192.168.122.1 down
```

Dacă dorim să vedem cum este configurată o anumită interfață, se foloseşte comanda urmată de numele interfeței.

```
[root@masserv /root]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:00:21:E7:F4:E2
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0xef40
[root@masserv /root]#
```

Se pot observa parametri importanți de configurare a interfeței Ethernet, cum sunt modul de încapsulare a cadrelor, adresa **MAC**, adresa **IP** şi masca de rețea, dacă interfața este activă (**UP**) sau

nu, **MTU** (Maximum Transfer Units) şi câteva informații statistice din care se poate determina starea legăturii.

Metrica este un parametru folosit la rutare pentru a caracteriza interfața din punctul de vedere al vitezei de transfer. Pentru identificarea corectă a interfeței, în acest răspuns al comenzii, apar numărul cererii de întrerupere şi adresa de bază a interfeței.

Pentru a vedea configurația tuturor interfețelor instalate pe o maşină, se poate folosi utilitarul ifconfig fără opțiuni.

```
[root@masserv /root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:B4:9D:F2:67
          inet addr:193.226.5.178  Bcast:193.226.5.183
          Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0xef80
eth0:0    Link encap:Ethernet  HWaddr 00:00:B4:9D:F2:67
          inet addr:192.168.120.1  Bcast:192.168.120.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:11 Base address:0xef80
eth1      Link encap:Ethernet  HWaddr 00:00:21:E7:F4:E2
          inet addr:192.168.122.1  Bcast:192.168.122.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0xef40
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
[root@masserv /root]#
```

Se poate observa, în exemplul de mai sus, că interfeței `eth0` i s-au asociat două adrese **IP** din subrețele diferite (subinterfețe).

Acest lucru permite crearea de rețele logice diferite utilizând aceeaşi rețea fizică. Artificiul are multiple utilizări practice legate de faptul că o poartă de acces , în general, nu transmite pachete cu adresă de *broadcast*. Din analiza răspunsului oferit de utilitarul `ifconfig` putem observa adresa de *broadcast* a fiecărei subrețele, alături de valoarea aleasă pentru **MTU**, proprie fiecărei interfețe.

Utilitarul `route`**.**

După configurarea unei interfețe, este necesară adăugarea adresei de rețea şi de maşină în tabela de rutare, în cazul în care acest lucru nu se face automat. Pentru aceasta se foloseşte utilitarul route, care permite afişarea tabelei de rutare, modificarea câmpurilor şi adăugarea sau ştergerea unei linii. Tabela de rutare, pe care o foloseşte nucleul sistemului de operare, are ca scop informarea acestuia despre primul pas (next hop), pe care trebuie trimis un pachet pentru a ajunge spre o anumită rețea sau maşină. Pentru un sistem **UNIX**, sau **Linux,** o astfel de comandă poate să arate ca în exemplul de mai jos. În cazul în care adresele **IP** ale porților de acces au asociate nume simbolice, acestea apar în tabele de rutare, dacă care acestea se pot afla de la un server de nume (**DNS**).

```
[root@masserv /root]# route
Kernel IP routing table
Destination    Gateway      Genmask     Flags Metric Ref Use Iface
192.168.122.1   *          255.255.255.255 UH   0    0     0 eth1
193.226.5.178   *          255.255.255.255 UH   0    0     0 eth0
193.226.5.176   *          255.255.255.248 U    0    0     0 eth0
192.168.120.0   *          255.255.255.0   U    0    0     0 eth0
```

```
192.168.122.0   *          255.255.255.0   U    0    0     0 eth1
127.0.0.0       *          255.0.0.0       U    0    0     0 lo
default      gate-b26.b25. 0.0.0.0         UG   0    0     0 eth0
[root@masserv /root]#
```

Dacă nu se doreşte rezolvarea şi afişarea numelor simbolice, se va folosi comanda `route` cu opţiunea `-n` (numeric, adresa **IP**).

```
[root@masserv /root]# route -n
Kernel IP routing table
Destination    Gateway      Genmask      Flags Metric Ref Use Iface
192.168.122.1  0.0.0.0      255.255.255.255 UH   0    0     0 eth1
193.226.5.178  0.0.0.0      255.255.255.255 UH   0    0     0 eth0
193.226.5.176  0.0.0.0      255.255.255.248 U    0    0     0 eth0
192.168.120.0  0.0.0.0      255.255.255.0   U    0    0     0 eth0
192.168.122.0  0.0.0.0      255.255.255.0   U    0    0     0 eth1
127.0.0.0      0.0.0.0      255.0.0.0       U    0    0     0 lo
0.0.0.0        193.226.5.177 0.0.0.0        UG   0    0     0 eth0
[root@masserv /root]#
```

În cazul în care adresa reţelei din care face parte interfaţa, care tocmai a fost instalată cu utilitarul `ifconfig`, nu se află între liniile de mai sus, va folosi utilitarul `route` pentru a defini ruta spre această destinaţie, care pentru un sistem de operare Linux, arată ca în exempul de mai jos:

```
[root@masserv /root]# route add -net 192.168.121.0 netmask 255.255.255.0 eth1
[root@masserv /root]#
```

sau mai pe scurt,

```
[root@masserv /root]# route add -net 192.168.121.0/24 eth1
[root@masserv /root]#
```

Se observă că în locul măştii de reţea definită în formatul zecimal, se poate folosi semnul "/" urmat de numărul biţilor, din mască, egali cu 1 (în acest caz 24). Opţiunea `-net` indică faptul că adresa este de reţea şi nu de maşină. În cazul în care trebuie adăugată o rută spre o singură adresă **IP** se va trece în locul opţiunii `-net`, opţiunea `-host` care este echivalentă implicit cu masca de reţea cu toţi biţii egali cu 1 (`255.255.255.255` sau `/32`), motiv pentru care aceasta nu se mai menţionează.

```
[root@masserv /root]# route add -host 192.168.121.2 eth1
[root@masserv /root]#route -n
Kernel IP routing table
Destination   Gateway   Genmask       Flags Metric Ref  Use Iface
192.168.122.2 0.0.0.0   255.255.255.255 UH    0    0      0 eth1
```

Dacă însă în tabela de rutare se găseşte o rută greşită, sau din motive de reconfigurare aceasta trebuie ştearsă, acest lucru se poate face în felul următor:

```
[root@masserv /root]# route del -net 192.168.121.0/24 eth1
```

sau mai simplu,

```
[root@masserv /root]# route del -net 192.168.121.0/24
[root@masserv /root]#
```

La versiunile actuale ale sistemelor de operare, o dată cu configurarea interfeţelor, tabela de rutare este actualizată cu rute către reţelele din care fac parte interfeţele respective (interfeţei `eth1` i s-a atribuit adresa **IP** `192.168.122.1` cu masca `255.255.255.0`, iar ruta spre reţeaua `192.168.122.0` a fost adăugată automat prin această interfaţă). Rămâne să fie adăugate manual doar rutele "speciale".

"Superserverul" `inetd`

Programul `inetd` a fost folosit încă de la început, în sistemele de operare **UNIX**, pentru lansarea la pornire a aplicațiilor (demonilor), care sunt necesari componentelor **TCP/IP** la stabilirea conexiunilor (a porturilor), precum şi a proceselor însoțitoare. Se pot astfel lansa în execuție mai multe procese, uneori identice, câte unul pentru fiecare port.

În acest fel, acest program preia de la server sarcina acestei gestiuni, deschizând câte un proces pentru fiecare conexiune stabilită, spre deosebire de server care deschide câte un proces pentru fiecare port (soluție mai puțin eficientă). Multe dintre serviciile oferite de un sistem **UNIX** sunt lansate la pornire prin intermediul programului `inetd`, în funcție de conținutul fişierului de configurare `/etc/inetd.conf`. Conținutul unui astfel de fişier se poate vedea în exemplu de mai jos:

```
#
# inetd.conf    Acest fişier descrie serviciile care sunt
# disponibile   prin intermediul
# super serverului TCP/IP INETD. Pentru configurarea # proceselor deschise
# de INETD, se editează acest fişier, după care
# trebuie trimis un semnal
# SIGHUP, cu comanda:
#'killall -HUP inetd'
# Această comandă are ca efect re-citirea de către inetd a
# fişierului /etc/inetd.conf.
#
# Versiune:     @(#)/etc/inetd.conf  3.10  05/27/93
#
# Autori:  Originalul luat de la BSD UNIX 4.3/TAHOE.
# Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#
# Modificat pentru Debian Linux, de către Ian A. Murdock
# <imurdock@shell.portal.com>
#
# Modificat pentru RHS Linux de către Marc Ewing <marc@redhat.com>
#
# Formatul fişierului este:
#
# <nume_serviciu> <tip_de_socket> <protocol> <atribute> <utilizator>
# <cale_server> <argumente>
#
# Echo, discard, daytime, şi chargen sunt utilizate pentru testare.
#
#
#echo stream     tcp  nowait     root internal
#echo dgram udp  wait root internal
#discard   stream     tcp  nowait     root internal
#discard   dgram udp  wait root internal
#daytime   stream     tcp  nowait     root internal
#daytime   dgram udp  wait root internal
#chargen   stream     tcp  nowait     root internal
#chargen   dgram udp  wait root internal
#time stream     tcp  nowait     root internal
#time dgram udp  wait root internal
#
# Acestea sunt servicii standard.
#
# Modificare Florin
#ident     stream     tcp  nowait     sys  /usr/sbin/tcpd  in.identd
-i -l
#
ftp  stream     tcp  nowait     root /usr/sbin/tcpd  in.ftpd
-L -i -o
telnet     stream     tcp  nowait     root /usr/sbin/tcpd  in.telnetd
#
# Shell, login, exec, comsat şi talk sunt protocoale BSD.
#
#shell     stream     tcp  nowait     root /usr/sbin/tcpd  in.rshd
#login     stream     tcp  nowait     root /usr/sbin/tcpd  in.rlogind
#exec stream     tcp  nowait     root /usr/sbin/tcpd  in.rexecd
#comsat    dgram udp  wait root /usr/sbin/tcpd  in.comsat
#talk dgram udp  wait root /usr/sbin/tcpd  in.talkd
```

```
#ntalk     dgram udp  wait root /usr/sbin/tcpd  in.ntalkd
#dtalk     stream     tcp  waut nobody     /usr/sbin/tcpd  in.dtalkd
#
# Pop şi imap sunt servicii de mail.
#
#pop-2   stream  tcp     nowait  root    /usr/sbin/tcpd   ipop2d
#pop-3   stream  tcp     nowait  root    /usr/sbin/tcpd   ipop3d
#imap    stream  tcp     nowait  root    /usr/sbin/tcpd   imapd
#
# Servicii Internet UUCP.
#
#uucp stream     tcp  nowait     uucp /usr/sbin/tcpd /usr/lib/uucp/uucico  -l
#
# Serviciul Tftp este folosit în primul rând pentru încărcarea de la distanşă a
sistemelor
# de operare.  Cele mai multe maşini rulează acest serviciu doar în cazul în care sunt
# "servere de boot"
# Nu înlăturaţi semnul de comentariu decât atunci când sunteţi siguri că aveţi nevoie de
# acest serviciu.
#
#tftp dgram udp  wait root /usr/sbin/tcpd  in.tftpd /manevra/tftp
#bootps   dgram udp  wait root /usr/sbin/tcpd  bootpd
#
# Finger, systat şi netstat oferă în exterior informaţii despre utilizatori, care pot fi
# "foarte preţioase" pentru cei care vor să atace sistemul (system crackers). Cei mai
mulţi
# administratori de sistem aleg să dezactiveze aceste servicii pentru a mări securitatea
# sistemului lor.
#
#finger    stream     tcp  nowait     root /usr/sbin/tcpd  in.fingerd
#cfinger stream tcp  nowait     root /usr/sbin/tcpd  in.cfingerd
#systat    stream     tcp  nowait     guest /usr/sbin/tcpd  /bin/ps -auwwx
#netstat   stream     tcp  nowait     guest /usr/sbin/tcpd  /bin/netstat
-f inet
#
# Servicii de autentificare
#
#auth   stream  tcp     nowait    nobody    /usr/sbin/in.identd
in.identd -l -e -o
#
# Sfârşitul fişierului inetd.conf
```

Coloanele din cuprinsul fişierului `inetd.conf` au următoarele semnificaţii:

1. reprezintă numele serviciului, care corespunde unei linii din fişierul `/etc/services`,

2. tipul de socket (flux continuu, caracter de caracter sau datagrame),

3. numele protocolului,

4. atribute care definesc dacă `inetd` poate accepta imediat alte conexiuni pe acelaşi port (`nowait`), sau trebuie să aştepte (`wait`) ca serverul respectiv să închidă o conexiune anterioară,

5. numele utilizatorului care este "proprietarul" acelui serviciu (nivelul drepturilor la care rulează acel serviciu),

6. calea începând de la rădăcina sistemului de fişiere şi numele programului server care oferă acel serviciu,

7. parametrii opţionali, necesari pentru configurarea programului server.

Fişierul de configurare este citit, de `inetd`, la pornirea sistemului şi de fiecare dată când se primeşte de la aplicaţie un semnal de suspendare a serviciului. Aceasta permite ca modificările aduse fişierului de configurare să intre în vigoare la fiecare pornire a unui serviciu (deschidere a unei conexiuni în urma unei cereri), ceea ce oferă un caracter dinamic al actualizării schimbărilor.

Serviciile lansate prin intermediul programului `inetd` beneficiază, între altele, de liste de control al accesului (ACL - Access Control List) prin intermediul fişierelor `/etc/hosts.allow` şi `/etc/hosts.deny` cu ajutorul cărora se poate limita accesul unor maşini (pe baza adresei **IP**) la anumite servicii. Structura unui astfel de fişier este dată în exemplul de mai jos:

```
#
# hosts.deny    Acest fişier reprezintă lista maşinilor cărora "nu
# l-i se permite" să folosească servicii locale gestionate de INET,
# şi hotărâte de serverul '/usr/sbin/tcpd'.
#
# Formatul fişierului:
#
# <Nume_server>: <adresa_IP sau numele_simbolic>
#
# al maşinii căreia i se refuză accesul la serviciu. În cazul unor
# liste, adresele vor fi separate prin "," şi spaţiu.
#
ALL: ALL
# Primul ALL: înseamnă toate serviciile (serverele), iar al doilea
# toate adresele.
```

Pentru siguranţă, se blochează, cu ajutorul fişierului `/etc/hosts.deny`, accesul la servicii a tuturor adreselor (sau numelor simbolice), după care se permite selectiv, prin înscriere în fişierul `/etc/hosts.allow`, accesul la servicii a maşinilor dorite.

```
#
# hosts.allow
# Acest fişier reprezintă lista maşinilor cărora "l-i se permite"
# accesul la serviciile locale gestionate de INET, şi hotărâte de
# serverul '/usr/sbin/tcpd'.
#
# <Nume_server>: <adresa_IP sau numele_simbolic> al maşinii căreia i
# se permite accesul la serviciu. În cazul unor liste, adresele vor
# fi separate prin "," şi spaţiu.
#
#in.ftpd: 212.135.190.91 , 193.231.107.34
ALL: 172.27.192. , 172.27.208.230. , 192.168.120. , 192.168.121
ALL: 193.226.5. , 192.168.122.10
in.ftpd: 194.176.188.93 192.168.120.
#
```

Aceste liste de control operează la nivelul de transport (**OSI 4**) al nucleului sistemului de operare şi afectează doar serviciile gestionate de `inetd`. Servicii ca, de exemplu, un server de web vor avea propriile liste de control. De asemenea, în cazul "zidurilor de foc" filtrarea accesului se face la nivelul de aplicaţie (**OSI 7**) prin procedee specifice.

Comanda `netstat`**.**

Comanda (programul) `netstat` analizează variabilele de proces şi furnizează informaţii despre parametrii **TCP/IP** de pe maşina locală. Acest program este prezent la majoritatea sistemelor de operare care folosesc acest protocol şi este folosit de administratorii de reţea pentru a obţine informaţii utile în activităţile de diagnosticare. Formatul şi parametrii de afişare a informaţiilor poate să difere de la un sistem de operare la altul, dar în general acestea se asemănă. Informaţiile care se pot obţine, cu ajutorul acestei comenzi, sunt următoarele:

- adresele maşinilor conectate şi starea legăturii,
- porturile care se folosesc pentru conectare,
- informaţii statistice despre protocolul de transport folosit,
- informaţii despre tampoanele de date,
- parametri de stare a interfeţelor de reţea,
- informaţiile conţinute în tabela de rutare.

În funcţie de versiunile sistemelor de operare, comanda `netstat` poate să furnizeze şi alte informaţii despre comunicaţia dintre procese ale aceleiaşi maşini.

Comunicaţia dintre maşinile aflate la capetele unui circuit virtual.

Dacă se foloseşte comanda `netstat` fără opţiuni, aceasta va furniza informaţii despre conexiunile active. Pentru a fi afişate toate conexiunile, atât cele active cât şi cele pasive, va trebui adăugată opţiunea "`-a`". Formatul de afişare a informaţiilor este pe coloane, indicând protocolul (`Proto - tcp, udp, raw`), volumul datelor înmagazinate în coada (tamponul) de recepţie

(Recv-Q), dar care nu au fost încă copiate de către aplicația de la nivelul superior şi volumul datelor înmagazinate în coada (tamponul) de transmisie (Send-Q), dar care nu au primit confirmarea (ACK). Răspunsul comenzii mai conține adresele **IP** (cea locală şi cea de la distanță) împreună cu porturile protocolului de transport şi starea curentă a conexiunilor virtuale. Aceasta din urmă este indicată doar pentru protocoalele de transport bazate pe conexiune (de exemplu **TCP**), iar pentru celelalte, acest câmp este lăsat gol. Stările posibile sunt următoarele:

- ESTABLISHED, sochetul are o conexiune stabilită,
- SYN_SENT. sochetul încearcă să stabilească o conexiune,
- SYN_RECV, sochetul local a primit, din partea rețelei, o cerere de deschidere a unei conexiuni,
- FIN_WAIT1, sochet închis, conexiune în curs de închidere,
-FIN_WAIT2, conexiune închisă, sochetul aşteptă închiderea conexiunii de către maşina de la distanță,
-TIME_WAIT, conexiune închisă, sochetul aşteaptă pachetele expirate, dar "rătăcite" prin rețea,
- CLOSED, sochetul nu este în stare de utilizare,
- CLOSE_WAIT, maşina de la distanță a închis legătura, sochet în curs de închidere,
- LAST_ACK, maşina de la distanță a închis legătura, sochetul este închis, dar se aşteaptă ultimul mesaj de confirmare,
- LISTEN, sochetul local aşteaptă cereri de deschidere a conexiunilor (doar pentru servicii care folosesc protocoale de transport bazate pe conexiune),
- CLOSING, ambele socheturi sunt închise, dar încă nu au fost trimise toate datele conținute în cozi,
- UNKNOWN, stare necunoscută a sochetului. Este de preferat ca această stare să nu apară!

Un exemplu de afişare a informațiilor furnizate de sistem, în urma lansării comenzii netstat –a, se poate vedea mai jos:

```
[root@masserv /root]# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address     Foreign Address       State
tcp  0      0 masserv.utcluj.ro:ftp bavaria.utcluj.ro:1027  ESTABLISHED
tcp  0      0 masserv.utcluj.ro:1026 bavaria.utcluj.r:telne ESTABLISHED
tcp  0      0 masserv.utcluj.ro:ftp masserv.utcluj.ro:1025  ESTABLISHED
tcp  0      0 masserv.utcluj.ro:1025 masserv.utcluj.ro:ftp  ESTABLISHED
tcp  0      0 masserv.utcluj.r:telne masserv.utcluj.ro:1024 TIME_WAIT
tcp  0      0 *:www                 *:*                     LISTEN
tcp  0      0 *:telnet              *:*                     LISTEN
tcp  0      0 *:ftp                 *:*                     LISTEN
udp        0      0 *:tftp          *:*
udp        0      0 *:syslog        *:*
raw        0      0 *:icmp          *:*                     7
raw        0      0 *:tcp           *:*                     7

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags     Type       State         I-Node Path
unix  0      [ ]       STREAM     CONNECTED     226    @0000003d
unix  1      [ ]       STREAM     CONNECTED     848    @0000005c
unix  1      [ ]       STREAM     CONNECTED     626    @00000049
unix  0      [ ACC ]   STREAM     LISTENING     609    /dev/log
unix  0      [ ACC ]   STREAM     LISTENING     779    /dev/gpmctl
unix  1      [ ]       STREAM     CONNECTED     711    @00000050
unix  1      [ ]       STREAM     CONNECTED     849    /dev/log
unix  1      [ ]       STREAM     CONNECTED     712    /dev/log
unix  1      [ ]       STREAM     CONNECTED     627    /dev/log
[root@masserv /root]#
```

Se observă că mesajul este structurat în două mari componente. Prima (Active Internet connections) se referă la legături care se stabilesc între componentele de nivel **OSI 3**, (sau mai jos 1,2), conexiuni între maşini diferite, sau aplicații client server de pe aceeaşi maşină, dar care

folosesc pentru comunicare nivelul de rețea (servere de telnet, ftp, web etc.). A doua (`Active UNIX domain sockets`) se referă la procesele locale care comunică între ele prin intermediul nivelului de transport. Protocolul (`Proto` - în mod uzual `unix`), are ataşate la un anumit sochet un număr de procese, specificate prin `RefCnt` (Reference Count), caracterizate de atribute de acceptare (Flags, ACC - sochetul acceptă date, W - "wait", N - "no space", etc.) şi bazate sau nu pe conexiune (de tip stream, datagram, raw etc.). Starea de interacțiune a proceselor (State) poate să aibă una dintre următoarele variante:
- `FREE`, nu este alocat sochet,
- `LISTENING`, sochetul aşteaptă o cerere de conexiune,
- `CONNECTING`, sochetul este pe cale să stabilească o legătură.
- `CONNECTED`, sochet conectat, conexiune stabilită,
- `DISCONNECTING`, sochetul este în curs de deconectare,
- (`spațiu gol`), sochet neconectat,
- `UNKNOWN`, stare necunoscută. Este de preferat să nu apară astfel de situații.

Ultimele două coloane se referă la identificatorul şi numele procesului (`I-Node, PID/Program name`) care foloseşte sochetul respectiv şi calea spre acest proces.

În exemplul de mai sus se pot vedea câteva conexiuni deschise între două maşini (sau de pe aceeaşi maşină) altele în aşteptare şi procese locale care utilizează sochet pentru comunicare. Astfel, se poate vedea, în linia întâi din listă, o conexiune cerută de către un client (probabil de FTP) de pe maşina "`bavaria.utcluj.ro`", având portul de inițiere `1027`, spre serverul de `FTP` de pe maşina locală "`masserv.utcluj.ro`". În mod asemănător se poate vedea, în linia a doua, conexiunea inițiată de clientul de `Telnet`, pe portul `1026` de pe maşina locală "`masserv.utcluj.ro`" şi stabilită cu serverul corespunzător de pe maşina de la distanță "`bavaria.utcluj.ro`".

În afara acestor conexiuni se pot vedea, în liniile 3-4 conexiunile stabilite pe maşina locală între un client de **`FTP`** cu serverul corespunzător de pe aceeaşi maşină. În linia 5 avem informații despre o conexiune, stabilită cu serverul de Telnet, dar care a fost încheiată. Cu toate acestea, sochetul deschis cu această ocazie, mai aşteaptă încă 30 secunde (`TIME_WAIT`) până să închidă definitiv portul. Pe lângă conexiunile stabilite se pot vedea în liniile 6-12 porturile deschise (în aşteptarea unor cereri de stabilire a conexiunilor) de câteva servere care rulează pe această maşină (`www, Telnet, FTP, syslog` etc). Porturile în aşteptare de cereri de stabilire a unei conexiuni nu pot fi afişate în listă decât dacă se foloseşte comanda `netstat` cu opțiunea `-a`. În cazul lansării comenzii `netstat` cu opțiunea `-n` se vor afişa valorile numerice ale porturilor şi numele simbolice ale maşinilor implicate în conexiuni. Prezența asterixului (*) în lista afişată, ne informează că nu există conexiuni stabilite, asociate portului respectiv.

În cea de a doua parte a mesajului afişat de comanda `netstat` se pot observa conexiunile dintre procesele locale care utilizează pentru comunicarea dintre ele protocoale de transport.

Informații statistice despre interfețele de rețea.

În afară de comanda `ifconfig`, informații despre starea interfețelor de rețea se pot obține şi cu ajutorul comenzii `netstat` utilizând parametrul `-i`. Această opțiune oferă doar informații legate de componentele de nivel OSI 3 şi 4 (Rețea şi Transport). În cazul în care aceeaşi interfață fizică are alocate mai multe adrese IP, informațiile se referă la toate luate împreună. Această comandă se foloseşte pentru verificarea funcționării punctelor de acces la servicii, corespunzătoare nivelelor OSI 2, 3 şi 4 (Legătură, Rețea şi Transport). Răspunsul sistemului la această comandă se poate vedea în exemplul prezentat mai jos:

```
[root@masserv /root]# netstat -i
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500 0    0     0      0      0       0    0      0      0 BRU
eth0:1500 0   - no statistics available -                        BRU
eth0:1500 0     no statistics available -                        BRU
eth1 1500 0    0     0      0      0       0    0      0      0 BRU
```

```
lo   3924 0    28    0      0      0       28   0      0      0 LRU
[root@masserv /root]#
```

Se observă că prima coloană se referă la numele interfeței (`iface`), a doua reprezintă dimensiunea maximă a segmentului de date care poate să fie transmis (`MTU`) de pe interfața respectivă, iar a treia, metrica administrativă (`Met`) asociată acelei interfețe.

Următoarele coloane oferă informații asupra stării tampoanelor de date (Rx - recepție, Tx - transmisie), respectiv numărul segmentelor recepționate (sau transmise) corect (`Rx-OK, Tx-OK`), a celor ratate (`Rx-ERR, Tx-ERR`), descărcate forțat (`Rx-DRP, Tx-DRP`), sau a celor care au depăşit spațiul de stocare temporară (`Rx-OVR, Tx-OVR`).

Pentru a obține informații asupra stării de congestie a segmentului de rețea din care face parte maşina locală, va trebui folosită comanda `ifconfig -i <nume_interfață>`.

Tabela informațiilor de rutare.

Tabela de rutare este actualizată periodic pentru ca sistemul să ştie calea cea mai potrivită spre o anumită destinație. Tabela de rutare are componente fixe (rute statice definite de administrator) şi componente variabile (rute dinamice definite cu ajutorul protocoalelor de rutare).

Pentru a afla conținutul tabelei de rutare, în afara comenzii `route`, se poate folosi comanda `netstat` cu opțiunea `-r`.

Pentru a obține informații care conțin doar adresa **IP** (nu şi numele simbolic al maşinilor sau porților de acces) se va folosi şi opțiunea `n`. Răspunsul sistemului se poate vedea în exemplul de mai jos.

```
[root@masserv /root]# netstat -rn
Kernel IP routing table
Destination   Gateway    Genmask       Flags  MSS Window irtt Iface
192.168.122.1 0.0.0.0    255.255.255.255 UH     0 0          0 eth1
193.226.5.178 0.0.0.0    255.255.255.255 UH     0 0          0 eth0
193.226.5.176 0.0.0.0    255.255.255.248 U      0 0          0 eth0
172.27.192.0  0.0.0.0    255.255.255.0   U      0 0          0 eth0
192.168.120.0 0.0.0.0    255.255.255.0   U      0 0          0 eth0
192.168.122.0 0.0.0.0    255.255.255.0   U      0 0          0 eth1
127.0.0.0     0.0.0.0    255.0.0.0       U      0 0          0 lo
0.0.0.0   193.226.5.177  0.0.0.0         UG     0 0          0 eth0
[root@masserv /root]#
```

Se poate observa asemănarea dintre răspunsul comenzii `netstat -rn` şi cel al comenzii `route -n`.

Informații statistice despre protocoalele de rețea şi transport.

Pentru a obține informații despre protocoalele de rețea şi transport se poate folosi comanda `netstat -s`. Aceasta furnizează o listă a caracteristicilor mai importante ale protocoalelor **`IP`, `ICMP`, `TCP`** şi **`UDP`**. Informațiile sunt utile pentru determinarea eventualelor erori care pot să apară la aceste nivele. Localizarea eventualelor defecte nu se poate face fără izolarea lor pe nivele. De obicei, cele mai greu de depistat sunt defectele specifice nivelelor **OSI 1** şi **2**.

Informațiile obținute cu ajutorul acestei comenzi pot duce la eliminarea celor de la nivelul **3** şi **4** şi arată ca în exemplul de mai jos.

```
[root@masserv /root]# netstat -s
Ip:
    361 total packets received
    0 forwarded
    0 incoming packets discarded
    12 incoming packets delivered
    384 requests sent out
Icmp:
    35 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 3
    35 ICMP messages sent
```

```
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 3
Tcp:
    2 active connections openings
    0 passive connection openings
    0 failed connection attempts
    0 connection resets received
    0 connections established
    314 segments received
    314 segments send out
    0 segments retransmited
    0 bad segments received.
    0 resets sent
Udp:
    0 packets received
    12 packets to unknown port received.
    0 packet receive errors
    35 packets sent
TcpExt:
[root@masserv /root]#
```

Formatul clar al mesajului de răspuns nu necesită comentarii suplimentare!

Utilitarul `ping`**.**

Utilitarul `ping` (Packet Internet Groper) este folosit pentru a verifica starea unei legături de rețea. Acesta utilizează **ICMP** (Internet Control Message Protocol), transmițând pachete de date cu cereri de răspuns (în ecou), maşinii de la distanță.

Starea legăturii se poate determina prin numărul de pachete recepționate corect şi prin timpul necesar pentru străbaterea rețelei *(roud trip)*. Aproape toate implementările de **TCP/IP** conțin componentele necesare acestui utilitar, dar nu toate lansează implicit aceste componente (uneori din motive de securitate).

În urma lansării în execuție a utilitarului (comenzii) `ping`, pe ecran apare câte un mesaj pentru fiecare pachet de răspuns recepționat corect. Aceasta poate să dureze un timp nedefinit (până se apasă tastele `Ctrl+C` în cazul sistemelor de operare Unix sau Linux).

După oprirea procesului de transmitere a pachetelor cu cereri, utilitarul `ping` calculează câțiva indicatori statistici asociați (numărul de pachete transmise, recepționate, fracția pierdută şi valoarea minimă, maximă şi medie a timpului scurs între transmisia şi recepția unui pachet - *round trip time*). Un astfel de răspuns se poate vedea în exemplu de mai jos.

```
[root@masserv /root]# ping 192.168.120.11
PING 192.168.120.11 (192.168.120.11): 56 data bytes
64 bytes from 192.168.120.11: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 192.168.120.11: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 192.168.120.11: icmp_seq=2 ttl=255 time=0.1 ms
64 bytes from 192.168.120.11: icmp_seq=3 ttl=255 time=0.1 ms
64 bytes from 192.168.120.11: icmp_seq=4 ttl=255 time=0.1 ms
64 bytes from 192.168.120.11: icmp_seq=5 ttl=255 time=0.1 ms
64 bytes from 192.168.120.11: icmp_seq=6 ttl=255 time=0.1 ms
--- 192.168.120.11 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.4 ms
[root@masserv /root]#
```

În acest exemplu utilitarul `ping` transmite pachete care conțin 64 octeți (56 în câmpul datelor utile plus 8 în antet), care au o durată de viață de 255 secunde, suficient de mare pentru a nu se autodistruge până la o destinație normal accesibilă.

În cazul în care se doreşte modificarea dimensiunii pachetului de date, a numărului de pachete sau a altor parametri, se va folosi comanda `ping` însoțită de opțiuni. De exemplu, pentru a transmite 4 pachete cu dimensiunea de 264 octeți (156 în câmpul util plus 8 în antet), transmise la interval de 1 secundă, se va folosi comanda ca în exemplul următor:

```
[root@masserv /root]# ping 192.168.120.11 -c 4 -s 256 -i 1
```

```
PING 192.168.120.11 (192.168.120.11): 256 data bytes
264 bytes from 192.168.120.11: icmp_seq=0 ttl=255 time=0.2 ms
264 bytes from 192.168.120.11: icmp_seq=1 ttl=255 time=0.1 ms
264 bytes from 192.168.120.11: icmp_seq=2 ttl=255 time=0.1 ms
264 bytes from 192.168.120.11: icmp_seq=3 ttl=255 time=0.1 ms
--- 192.168.120.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.2 ms
[root@masserv /root]#
```

În cazul utilizării numelui simbolic în locul adresei **IP**, utilitarul ping face o cerere unui server de nume, pentru a afla adresa **IP**, după care o va utiliza pe aceasta. În exemplul de mai jos se poate vedea un astfel de caz.

```
[root@masserv /root]# ping bavaria.utcluj.ro -c 5
PING bavaria.utcluj.ro (193.226.5.35): 56 data bytes
64 bytes from 193.226.5.35: icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 193.226.5.35: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 193.226.5.35: icmp_seq=2 ttl=255 time=0.1 ms
64 bytes from 193.226.5.35: icmp_seq=3 ttl=255 time=0.1 ms
64 bytes from 193.226.5.35: icmp_seq=4 ttl=255 time=0.1 ms
--- bavaria.utcluj.ro ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.3 ms
[root@masserv /root]#
```

Acest utilitar este folosit foarte des pentru verificare stării unei legături fizice. Administratorii de rețea apelează la acesta de fiecare dată când sunt dubii legate de funcționarea rețelei. Se poate spune că aceasta este cea mai frecvent utilizată comandă într-un sistem de operare de rețea!

Primul lucru pe care îl face un administrator de rețea, este să apeleze comanda ping pentru a verifica legătura dintre maşina sa şi prima poartă de acces spre Internet! În afară de aceasta, comanda se mai foloseşte şi pentru a verifica parametrii protocoalelor utilizate, iar atunci când este cazul, să poată să le modifice corespunzător.

În locul adresei **IP** comanda ping acceptă şi numele simbolic. Înainte de a trimite pachete spre o maşină de la distanță, utilitarul ping face o cerere unui server de nume (**DNS**) pentru a afla adresa **IP** corespunzătoare. După aceasta va trimite pachetele de verificare acestei adrese.

În cazul în care serverul de nume nu este accesibil, sau serviciul nu funcționează corect, comanda ping ratează acțiunea.

În concluzie, se poate astfel determina cu ajutorul acestei comenzi şi modul în care funcționează serverul de nume. Mergând din aproape în aproape, cu ajutorul comenzii se poate verifica şi starea serviciului de rutare în rețea. Dacă există răspuns de la prima poartă de acces, dar nu şi în afara ei, se poate presupune că rutarea nu este corespunzătoare, sau există probleme de acolo mai departe.

Majoritatea versiunilor **MS Windows** oferă varianta acestui utilitar sub forma unei comenzi **DOS**, foarte asemănătoare cu variantele sale **Unix**.

„Desenarea" traseului unei conexiuni cu utilitarul `traceroute`**.**

Pentru a verifica traseul unei conexiuni, **TCP/IP** a fost "echipat" corespunzător. Pentru a iniția înregistrarea unei rute, se face o cerere către maşina de la distanță care activează atributul corespunzător din antetul pachetului **IP**, care declanşează ecouri trimise spre maşina sursă.

Când este activată înregistrarea rutei, activitățile sunt însoțite de ecouri corespunzătoare care pot să fie înscrise în tampoane de memorie, în fişiere sau afişate pe ecran. O soluție mai convenabilă de determinare a traseului unei conexiuni este aceea care foloseşte pachete **`ICMP`**, asemănătoare utilitarului `ping`.

Un astfel de utilitar, folosit pentru verificarea traseului spre o destinație este `traceroute`. Acesta foloseşte câmpul dedicat timpului de viață (time to live) al unui pachet **IP** încercând să obțină o informație referitoare la depăşirea timpului de răspuns al fiecărei porți de acces

(`TIME_EXCEEDED`) de pe acel traseu. Singurul parametru cert este adresa maşinii destinație şi numărul de paşi supuşi verificării.

Utilitarul traceroute trimite pachete cu lungimea implicită de 40 octeți, dar dimensiunile acestora se pot alege în funcție de opțiunile adăugate în linia de comandă, după adresa maşinii destinație. Un exemplu de determinare a traseului spre o anumită destinație este prezentat mai jos:

```
[root@masserv /root]# traceroute masserv
traceroute to masserv.utcluj.ro (193.226.5.178), 30 hops max, 40 byte packets
1  masserv (193.226.5.178)  0.337 ms  0.178 ms  0.133 ms
[root@masserv /root]#
```

Se observă că ruta conține doar un singur tronson, deoarece terminalul destinație coincide cu cel sursă. Chiar şi în această situație utilitarul `traceroute` „desenează" această „rută".

În cazul unei rute mai lungi, traseul va fi reprezentat prin adresa şi numele simbolic al nodurilor dintre tronsoane, alături de timpul necesar pachetelor să le străbată, aşa cum se poate vedea în exemplul de mai jos.

```
root@masserv:~# traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 209.85.135.99
traceroute to www.l.google.com (209.85.135.99), 30 hops max, 38 byte packets
 1  ge-4-5.r-core.utcluj.ro (193.226.5.25)  1.909 ms  0.929 ms  0.956 ms
 2  gig-1-1.acc1.clu.roedu2.net (217.73.171.73)  0.829 ms  0.795 ms  1.275 ms
 3  ten-2-1.core1.clu.roedu2.net (89.37.2.129)  6.571 ms  6.490 ms  6.576 ms
 4  gig-3-5.core1.buc3.roedu2.net (89.37.0.29)  6.549 ms  6.631 ms *
 5  ten-1-4.core2.buc1.roedu2.net (89.37.0.2)  6.656 ms  7.932 ms  7.314 ms
 6  89.37.0.174 (89.37.0.174)  11.700 ms  11.611 ms  10.892 ms
 7  roedunet1.rt1.bud.hu.geant2.net (62.40.124.105)  41.109 ms  42.560 ms  40.38
6 ms
 8  bpt-b2-link.telia.net (213.248.76.29)  38.930 ms  30.523 ms  33.548 ms
 9  hbg-bb2-link.telia.net (80.91.250.134)  58.940 ms  53.881 ms  54.347 ms
10  hbg-b2-link.telia.net (80.91.249.201)  54.386 ms  51.112 ms  53.098 ms
11  google-ic-122934-hbg-b2.c.telia.net (213.248.100.170)  53.835 ms  49.055 ms
 47.195 ms
12  66.249.95.132 (66.249.95.132)  52.167 ms  75.071 ms  63.843 ms
13  72.14.232.140 (72.14.232.140)  63.699 ms  62.048 ms  62.687 ms
14  72.14.232.209 (72.14.232.209)  70.947 ms  62.485 ms  63.531 ms
15  209.85.248.248 (209.85.248.248)  65.045 ms  71.417 ms  72.679 ms
16  66.249.94.85 (66.249.94.85)  86.861 ms  75.155 ms  76.345 ms
17  72.14.239.58 (72.14.239.58)  64.385 ms  71.212 ms 209.85.253.22 (209.85.253.
22)  75.813 ms
18  mu-in-f99.google.com (209.85.135.99)  73.700 ms  78.997 ms  78.082 ms
root@masserv:~# exit
logout
```