

8. CONFIGURAREA, DIAGNOSTICAREA SI MONITORIZAREA RETELEI

8.1. Configurarea retelei.....	2
8.1.1. Placa de retea sau interfata de retea?.....	2
8.1.2. Ce inseamna configurarea retelei.....	2
8.1.3. Configurarea interfetelor de retea.....	3
8.1.3.1. Denumirile interfetelor de retea in Linux.....	3
8.1.3.2. Utilitare si modalitati de configurare a interfetelor de retea.....	3
8.1.3.3. Vizualizarea configurarii interfetelor de retea.....	4
8.1.3.4. Activarea si dezactivarea interfetelor de retea.....	4
8.1.3.5. Configurare manuala temporara.....	5
8.1.3.5.1. Folosind comanda ifconfig.....	5
8.1.3.5.2. Folosind comanda ip.....	5
8.1.3.5.3. Configurarea de adrese multiple pe aceiasi interfata.....	5
8.1.4. Configurarea parametrilor globali.....	6
8.1.4.1. Configurarea rutei default.....	6
8.1.4.2. Configurarea serverelor DNS.....	7
8.1.5. Configurare permanenta - exemple pentru diverse distributii.....	7
8.1.5.1. Fedora.....	7
8.1.5.2. Debian si distributiile derivate.....	8
8.1.5.3. Slackware.....	8
8.1.5.4. Gentoo.....	8
8.1.6. Network Manager.....	9
8.1.6.1. Prezentare, capabilitati, terminologie.....	9
8.1.6.2. Relatia cu alte utilitare/modalitati de configurare.....	9
8.1.6.3. Configurarea folosind nmcli.....	10
8.2. Utilitare de diagnosticare a retelei.....	11
8.2.1. Comenzile mii-tool, ethtool si iwconfig.....	11
8.2.2. Comanda ping.....	12
8.2.3. Comanda traceroute.....	12
8.2.4. Comanda netstat.....	13
8.2.5. Comanda route.....	14
8.2.6. Analizoare de protocol: tcpdump, tshark/wireshark.....	14
8.3. BIBLIOGRAFIE.....	14

8.1. Configurarea rețelei

8.1.1. Placa de rețea sau interfața de rețea?

Pentru început, trebuie făcută distincția între două noțiuni:

- **placa de rețea** este dispozitivul hardware ce permite stației conectarea fizică la o rețea. Ea poate fi o placă de sine statatoare introdusă într-un slot al plăcii de bază, un dispozitiv USB sau un chip inclus în placă de bază a stației
- **interfața de rețea** reprezintă o noțiune abstractă – este o construcție software prin intermediul căreia sistemul de operare poate trimite și primi date

Placa de rețea nu are neapărat o interfață corespunzătoare, și invers – iată exemple:

- placa de rețea fără interfață: atunci când kernel-ul Linux nu are driverele necesare pentru a controla o placă de rețea, nu va crea o interfață corespunzătoare ei
- interfața de rețea fără placă: interfața de loopback, care există în orice sistem de operare al unei stații conectate într-o rețea TCP/IP. Toate datele trimise prin această interfață se transmit înapoi aceleiași mașini, fără a părăsi stația în cauză (o “buclă” locală, în kernel)

8.1.2. Ce înseamnă configurarea rețelei

Configurarea rețelei pe o stație Linux presupune stabilirea valorilor pentru un set de parametri, cei mai des întâlniți fiind:

- **adresa IP a stației** – identifică în mod unic stația în rețeaua din care face parte. Adresa IP este un caz particular de adresă de nivel logic, prezintă în rețelele TCP/IP
- **masca de rețea** (network mask) – stabilește subrețeaua din care face parte stația. Pe baza combinației IP/netmask se pot deduce o întreagă serie de parametri (adresa subrețelei, adresa de broadcast, numărul maxim de adrese din acea rețea etc)
- **adresa routerului implicit** (numită în general *default gateway* sau *default route*) – este adresa către care stația trimite pachetele pentru care nu are rute directe. Reprezintă în general adresa stației/dispozitivului prin care rețeaua are acces către alte rețele sau internet
- **servere DNS** – lista de servere pe care stația Linux le va interoga atunci când încearcă să determine adresele IP corespunzătoare diferitelor nume cu care operează (ex: nume de site-uri, porțiuni de adrese email etc)

Parametrii amintiți mai sus se încadrează în două categorii:

- parametri per-interfață – sunt parametri care au câte un set de valori pentru fiecare interfață de rețea. Aici se încadrează adresa IP și netmask-ul
- parametri globali – sunt parametri de sistem, care au la un moment dat o singură valoare, indiferent de numărul de interfețe de rețea din sistem. Aici intră ruta default și serverele DNS

Deși afirmația de mai sus în privința valorilor instantanee ale parametrilor globali este adevărată, există scenarii în care ei pot să depindă de interfață. Spre exemplu, să presupunem că un utilizator are un laptop care dispune de două interfețe de rețea – una cu fir și alta fără fir – pe care îl folosește la serviciu într-o rețea cu fir și acasă wireless. În aceste condiții, sistemul de operare va fi configurat nu numai cu adrese diferite pentru fiecare interfață în parte, ci și cu rute default și servere DNS diferite, în funcție de interfața folosită în acel moment. Dacă adresa IP și netmask-ul pot fi nativ configurate pentru fiecare interfață în parte, în schimb ruta default și serverele DNS ar trebui modificate de către utilizator de fiecare dată când trece dintr-o rețea în alta. De aceea, sistemele de operare Linux moderne pun la dispoziția utilizatorului posibilitatea de a include și default gateway/servere DNS în configurarea per-interfață, și vor folosi setările în cauză numai atunci când interfața este activă.

8.1.3. Configurarea interfetelor de retea

8.1.3.1. Denumirile interfetelor de retea in Linux

Traditional, in Linux interfetele de retea sunt denumite in functie de tehnologia in care functioneaza. Numele fiecărei interfete include de asemenea un numar, care permite distingerea între mai multe interfete ce lucreaza in aceeasi tehnologie, numerotarea facandu-se de la 0:

- interfetele Ethernet cu fir vor fi denumite in general eth0, eth1, ...
- interfetele wireless sunt in general denumite wlan0, wlan1, ..., dar exista si cazuri in care astfel de interfete sunt denumite eth0, eth1... (in functie de producator si driver)
- interfetele PPP (create atunci cand utilizatorul foloseste protocolul PPP – ex: PPPoE sau modem dial-up) vor fi denumite ppp0, ppp1,...

Nota: numele unei interfete de retea poate fi schimbat fie cu utilitarul `nameif`, fie cu comanda `ip link` (vezi mai jos `iproute2`), fie editand regulile `udev` (`udev` fiind programul care administreaza fisierele dispozitiv).

Acest mod de denumire (impreuna cu intregul sistem din spatele sau) a dovedit insa niste neajunsuri de-a lungul timpului. Interfetele nu primeau intotdeauna acelasi nume - spre exemplu, o interfata numita `eth0` era posibil sa capete numele `eth1` dupa reboot, dupa introducerea unei alte interfete sau dupa actualizarea driverului, ceea ce ar fi reprezentat o problema serioasa, de pilda, pentru un firewall ale carui reguli includeau si nume de interfete! De aceea, in ultima vreme exista un curent ce militeaza pentru denumirea interfetelor intr-un mod stabil, desi - e drept - nu foarte prietenos. Asa se face ca in distributii precum Fedora sau CentOS intalnim nume de interfete precum `enp12s0` sau `p2p1`. Consultati bibliografia pentru link-uri utile in privinta noii strategii de denumire a interfetelor.

8.1.3.2. Utilitare si modalitati de configurare a interfetelor de retea

Configurarea unei interfete de retea presupune in general doua aspecte:

- **configurarea legaturii de date** – conexiunea între placa de retea si restul retelei
 - in cazul unei interfete cu fir, aceasta ar putea insemna viteza la care functioneaza legatura de date (10/100/1000Mbps) si modul de lucru (half-duplex sau full-duplex). Se pot folosi in acest scop utilitare precum **mii-tool** sau **ethtool**
 - in cazul interfetelor wireless, pentru ca placa de retea sa poata comunica cu restul retelei, este necesar ca ea sa se asocieze la un access point, iar pentru aceasta trebuie stabiliti parametrii canalului de comunicatie dintre ele (SSID-ul retelei, eventuali parametri de autentificare, de criptare a canalului de comunicatie etc). Utilitarul folosit in acest scop este **iwconfig**.
- **configurarea de nivel logic** – presupune stabilirea parametrilor de nivel retea ai interfetei (adresa IP, netmask etc). Exista doua seturi de utilitare ce pot fi folosite:
 - utilitarele clasice **ifconfig** si **route**. Mult timp, acestea au reprezentat singura modalitate de a configura interfete de retea in Linux si Unix. Desi in ziua de astazi ele nu pot profita de toate capabilitatile kernelului Linux, sunt inca preferate de multi administratori datorita faptului ca se regasesc si in multe dintre sistemele de operare Unix, unde opereaza cu o sintaxa asemanatoare
 - utilitarele din pachetul `iproute2` (ex: **ip** si **tc**). Pachetul `iproute2` contine utilitare ce valorifica facilitatile moderne ale kernelului Linux, permitand multe operatii suplimentare fata de utilitarele clasice. Distributiile Linux ale zilelor noastre migreaza lent catre aceasta modalitate de configurare, pastrand in acelasi timp compatibilitatea inversa cu vechiul set de utilitare

Toate utilitarele enumerate mai sus realizeaza o configurare volatila – ea se pierde in cazul repornirii sistemului de operare. Pentru o configurare permanenta este necesara editarea unuia sau mai multor fisiere de configurare (care pot insa diferi mult de la o distributie la alta); parametrii salvati in aceste fisiere vor fi insa folositi in final tot pentru apelarea utilitarelor amintite mai sus. Administratorul de sistem Linux trebuie sa stapaneasca in

primul rand utilitarele de configurare manuala (care ii asigura cvasi-independenta de distributie) si abia apoi locatia si formatul fisierelor de configurare a parametrilor de retea.

8.1.3.3. Vizualizarea configurarii interfetelor de retea

Lista interfetelor de retea împreuna cu parametrii lor de configurare poate fi obtinuta cu comanda **ifconfig**. Rulata fara argumente, comanda afiseaza doar interfetele active; optiunea **-a** determina afisarea tuturor interfetelor detectate de catre kernel:

```
eth0      Link encap:Ethernet  HWaddr 00:0f:b0:c6:46:e6
          inet addr:10.0.0.4  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20f:b0ff:fec6:46e6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
          RX packets:844555 errors:0 dropped:0 overruns:0 frame:0
          TX packets:399442 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1274629391 (1.1 GiB)  TX bytes:28194024 (26.8 MiB)
          Interrupt:23

wlan0    Link encap:Ethernet  HWaddr 00:14:a5:73:50:93
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::214:a5ff:fe73:5093/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:3748962 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2871527 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3350557988 (3.1 GiB)  TX bytes:1563858582 (1.4 GiB)
```

Pentru fiecare interfata, pe prima linie este afisata incapsularea folosita pentru legatura de date si adresa fizica a interfetei (in cazul nostru, tehnologia folosita este Ethernet, adresa fizica corespunzatoare fiind cea MAC). A doua linie contine configurarea de nivel logic pentru IPv4 (adresa IP, netmask, adresa de broadcast). Daca stiva IPv6 este instalata, fiecare interfata va dispune si de un set de parametri IPv6, prezenti pe linia imediat urmatoare. Urmeaza setul de flag-uri asociate interfetei (ex: daca este sau nu activa, daca lucreaza in mod broadcast sau point-to-point etc). Output-ul lui *ifconfig* prezinta de asemenea si cantitatea de trafic a interfetei (transmisie si receptie).

Alternativ, configurarea de nivel logic poate fi afisata si cu comanda **ip address list**:

```
1: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1492 qdisc pfifo_fast state UP qlen 1000
   link/ether 00:0f:b0:c6:46:e6 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.4/24 brd 10.0.0.255 scope global eth0
   inet6 fe80::20f:b0ff:fec6:46e6/64 scope link
       valid_lft forever preferred_lft forever
2: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
   link/ether 00:14:a5:73:50:93 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.3/24 brd 192.168.1.255 scope global wlan0
   inet6 fe80::214:a5ff:fe73:5093/64 scope link
       valid_lft forever preferred_lft forever
```

8.1.3.4. Activarea si dezactivarea interfetelor de retea

Odata detectata de kernel, o interfata de retea se poate afla in starea up (activa) sau down (inactiva). O interfata down nu poate fi folosita pentru a transmite date. Comutarea intre cele doua stari se poate realiza fie cu comanda **ifconfig**, fie cu comanda **ip link set**, dupa cum urmeaza:

```
# activarea unei interfete inactive
ifconfig eth0 up
ip link set dev eth0 up

# dezactivarea unei interfete active
ifconfig eth0 down
ip link set dev eth0 down
```

8.1.3.5. Configurare manuala temporara

8.1.3.5.1. Folosind comanda ifconfig

ifconfig este utilitarul traditional Linux/Unix pentru configurarea interfetelor de retea. El poate fi folosit pentru a configura interfete de retea lucrând in diverse tehnologii (nu numai Ethernet) si cu diverse stive de protocoale (nu numai TCP/IP).

***Nota:** a nu se confunda ifconfig din Unix/Linux cu ipconfig din Windows. Ultimul este folosit numai pentru vizualizarea configurarii IP si pentru configurarea dinamica (prin DHCP), pe cand primul poate configura diverse adrese de nivel logic.*

Sintaxa generala a comenzii *ifconfig* pentru configurarea parametrilor unei interfete de retea este:

```
ifconfig nume_interfata adresa_IP netmask masca_retea broadcast adresa_broadcast
# ...sau, alternativ:
ifconfig nume_interfata adresa_IP/nr_bit_i_de_1_din_netmask
```

***Nota:** cea de-a doua sintaxa folosește așa-numita sintaxa CIDR, în care porțiunea de după caracterul / indica numărul de biti de 1 din netmask. Astfel, netmask-ul 255.255.255.0 conține 3x8=24 de biti de 1 și deci s-ar putea scrie ca /24.*

Daca sunt omise netmask-ul si adresa de broadcast, ifconfig le va alege automat in functie de clasa din care face parte adresa IP:

```
ifconfig eth0 192.168.0.34 # clasa C, netmask implicit 255.255.255.0 (/24), broadcast 192.168.0.255
ifconfig eth0 10.0.0.34 # clasa A, netmask implicit 255.0.0.0 (/8), broadcast 10.255.255.255
```

***Nota:** la configurarea unei interfete in Linux, ea este automat activata. Acest lucru nu este neaparat valabil si in Unix – spre exemplu, in Solaris este necesara activarea explicita a interfetei dupa configurarea sa.*

8.1.3.5.2. Folosind comanda ip

Comanda **ip** poate fi si ea folosita pentru configurarea de nivel logic a unei interfete, folosind **ip address add** urmat de adresa/netmask si de numele interfetei. Netmask-ul poate fi specificat atât în format clasic, cât și CIDR. *Atenție!* Dacă uităm sa specificăm netmask-ul, valoarea sa implicita este 255.255.255.255, adică fiecare adresa IP formeaza propria-i rețea!

```
ip address add 192.168.0.46/24 dev eth0 # varianta CIDR
ip address add 192.168.0.46/255.255.255.0 dev eth0 # varianta clasica
ip address add 192.168.0.46 dev eth0 # adaugare adresa cu netmask 255.255.255.255!
```

8.1.3.5.3. Configurarea de adrese multiple pe aceeasi interfata

Cu setul clasic de utilitare, a configura mai multe adrese pe aceeasi interfata presupune crearea de așa-numite "alias"-uri ale interfetei – un fel de subinterfete care pot avea adresa proprie, dar care in restul de privinte nu

functioneaza ca o interfata in toata regula (spre exemplu, in firewallul Linux nu putem filtra dupa subinterfete). Alias-urile sunt de forma **nume_interfata:numar**, unde *nume_interfata* este denumirea interfetei originale, iar numarul incepe de la zero si numeroteaza subinterfetele din cadrul acelei interfete. Alias-urile se creeaza prin simpla lor configurare; spre exemplu, daca interfata de baza este *eth0*, atunci pentru a-i atribui 3 adrese IP (creand astfel automat doua alias-uri) vom scrie:

```
ifconfig eth0 192.168.0.15
ifconfig eth0:0 10.0.0.15/16
ifconfig eth0:1 172.16.0.15
```

Stergerea uneia dintre adrese presupune desfiintarea alias-ului, care se realizeaza prin dezactivarea subinterfetei corespunzatoare:

```
ifconfig eth0:1 down
```

Prin contrast, comanda **ip** permite simpla adaugare de adrese secundare cu aceeasi sintaxa ca pentru setarea adresei initiale, iar stergerea se realizeaza cu **ip address del**:

```
# adaugarea acelorasi 3 adrese ca in exemplul de mai sus
ip address add 192.168.0.15/24 dev eth0
ip address add 10.0.0.15/16 dev eth0
ip address add 172.16.0.15/16 dev eth0

# ...si stergerea ultimei
ip address del 172.16.0.15/24 dev eth0
```

Adresele adaugate cu **ip address add** nu se vad implicit in output-ul lui **ifconfig**! Pentru a pastra compatibilitatea cu acest utilitar, la adaugarea unei adrese cu comanda **ip** se poate specifica suplimentar parametrul **label** care da numele subinterfetei create automat pentru noua adresa:

```
ip address add 193.8.67.29/24 dev eth0 label eth0:3
```

8.1.4. Configurarea parametrilor globali

8.1.4.1. Configurarea rutei default

Fiecare statie conectata intr-o retea TCP/IP mentine o asa-numita tabela de rutare – o succesiune de inregistrari care specifica rute pentru diferite retele destinatie. O ruta are ca elemente de baza:

- retea destinatie (ex: 10.0.0.0/24)
- interfata pe care trebuie expediat pachetul pentru a ajunge la acea destinatie
- masina intermediara (routerul) catre care trebuie trimis pachetul, in cazul in care retea destinatie nu este direct conectata (adresa IP destinatie nu se afla in aceeasi retea cu expeditorul)

Adresa IP destinatie a fiecarui pachet expediat de o statie conectata intr-o retea TCP/IP este confruntata pe rand cu fiecare retea destinatie din tabela de rutare. Prima inregistrare care corespunde (in sensul ca adresa IP destinatie a pachetului face parte din retea destinatie a inregistrarii in cauza) va determina expedierea pachetului pe interfata specificata in cadrul inregistrarii. Exista insa si situatia in care destinatia unui pachet nu corespunde cu nici una dintre rutele tabelului, iar in acest caz va fi folosita pentru el ruta default, daca exista; daca aceasta nu este prezenta, pachetul este anulat.

Ruta default specifica statia careia i se vor transmite pachetele a caror destinatie nu are alta intrare corespondenta in tabela de rutare (statie numita si “default gateway”). Ea are ca adresa destinatie 0.0.0.0 (orice retea). Ruta default asigura conectivitatea cu alte retele decât cele direct conectate sau cu internetul.

Pentru manipularea rutei default se poate folosi fie comanda **route**, fie **ip route**, ca in exemplele urmatoare:

```
route add default gw 10.0.0.1      # adaugare ruta default cu adresa 10.0.0.1
route del default                  # stergere ruta default
ip route add default via 10.0.0.1 # ...si aceleasi operatii efectuate cu comanda ip
ip route del default
```

Ruta default stabilita cu *route* sau *ip* este de asemenea volatila, ea pierzandu-se la restart sau la dezactivarea interfetei corespunzatoare. Pentru permanentizare este necesara salvarea sa intr-unul dintre fisierele de configurare (vezi mai jos).

8.1.4.2. Configurarea serverelor DNS

Serverele DNS beneficiaza din start de configurare permanenta, ele fiind specificate in fisierul **/etc/resolv.conf**. Pentru fiecare server DNS se va introduce in fisier o linie ce contine directiva **nameserver** urmata de adresa serverului dorit:

```
nameserver 10.0.0.1
nameserver 8.8.8.8
```

Atentie! Având în vedere ca o stație se poate conecta succesiv în mai multe rețele, așteptați-va ca fișierul *resolv.conf* să fie rescris automat de către utilitarele de configurare automata de rețea (ex: network manager). Acesta este și motivul pentru care, în destule distributii, suntem avertizați sa nu-l editam direct - nu pentru ca am strica ceva, ci fiindcă setarile efectuate manual s-ar putea sa nu supravietuiasca mult.

8.1.5. Configurare permanenta - exemple pentru diverse distributii

8.1.5.1. Fedora

In Fedora, setarile de retea se gasesc in fisiere text ce contin un ansamblu de atribuirii de variabile, de forma PARAMETRU=valoare. Iata-le pe cele mai importante:

- **/etc/sysconfig/network** – contine setari de sistem precum numele statiei, activarea sau dezactivarea generala a stivelor IPv4 si IPv6, ruta default etc

```
NETWORKING=yes
NETWORKING_IPV6=no
GATEWAY=10.0.0.1
```

- **/etc/sysconfig/network-scripts/ifcfg-numeInterfata** – aceste fisiere contin parametrii fiecărei interfete (adresa IP, netmask etc) – iata cateva exemple:

```
DEVICE=eth0
IPADDR=10.0.0.4
NETMASK=255.255.255.0
BOOTPROTO=static
ONBOOT=yes
[...diferiti alti parametri de configurare...]
```

Pentru setul complet de parametri configurabili in aceste fisiere consultati link-ul către Fedora Networking Guide din bibliografie.

8.1.5.2. Debian si distributiile derivate

In Debian, fisierul folosit pentru configurarea interfetelor de retea este **/etc/network/interfaces**.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

auto wlan0
iface wlan0 inet static
address 192.168.1.3
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

In exemplul de mai sus vedem 3 interfete:

- interfata de loopback numita **lo**, activata automat la bootare (efectul directivei *auto*)
- interfata ethernet **eth0**, activata in momentul detectarii de catre kernel (*allow-hotplug*) si configurata automat prin DHCP
- interfata wireless **wlan0**, configurata manual si activata automat la bootare

8.1.5.3. Slackware

In Slackware configurarea retelei rezida in **/etc/rc.d/rc.inet1.conf**, care consta din atribuirii de valori pentru diversii parametri – atat cei per-interfata cat si cei globali:

```
# Config information for eth0:
IPADDR[0]="10.0.0.15"
NETMASK[0]="255.255.255.0"

# Default gateway IP address:
GATEWAY="10.0.0.1"
```

8.1.5.4. Gentoo

In Gentoo, fiecare interfata beneficiaza de un fisier de configurare: **/etc/conf.d/net.numeInterfata**. Continutul sau este asemanator cu urmatorul:

```
# interfata configurata prin DHCP
config_eth0=( "dhcp" )

# interfata configurata static
config_eth1=( "192.168.0.7/24" )
routes_eth1=( "default via 192.168.0.1" )
```

8.1.6. Network Manager

8.1.6.1. Prezentare, capabilitati, terminologie

Network manager este un set de utilitare ce are ca scop simplificarea configurarii rețelei. El este utilizat în general în distribuțiile de tip desktop, unde configurarea de rețea trebuie să fie cât mai bine automatizată, presupunând un efort minim din partea utilizatorului.

Network Manager permite configurarea rețelei în două moduri:

- dacă sistemul de operare dispune de interfața grafică, există un applet GNOME care integrează toate operațiile legate de rețea (vezi imaginea alăturată)
- suplimentar, există și un utilitar de linie de comandă numit **nmcli**, foarte util în configurarea manuală

Nota: în Fedora/RHEL/CentOS există și un utilitar numit **nmtui** ce permite configurarea lui *Network Manager* prin intermediul unei interfețe text.

Network manager operează cu următoarele noțiuni:

- **device** - reprezintă o interfață din sistem utilizabilă pentru conectarea la rețea.
- **conexiune** - reprezintă un ansamblu de setări atasate unui device și care îi asigură acestuia conectivitatea - practic, un profil de conectare. Un device poate avea mai multe conexiuni definite în sistem atasate lui, din care una singură poate fi activă la un moment dat; exemplul tipic este cel al interfețelor wireless, unde avem câte un astfel de profil pentru fiecare rețea wireless la care ne conectăm. Applet-ul **nm** arată în general conexiunile, nu interfețele.

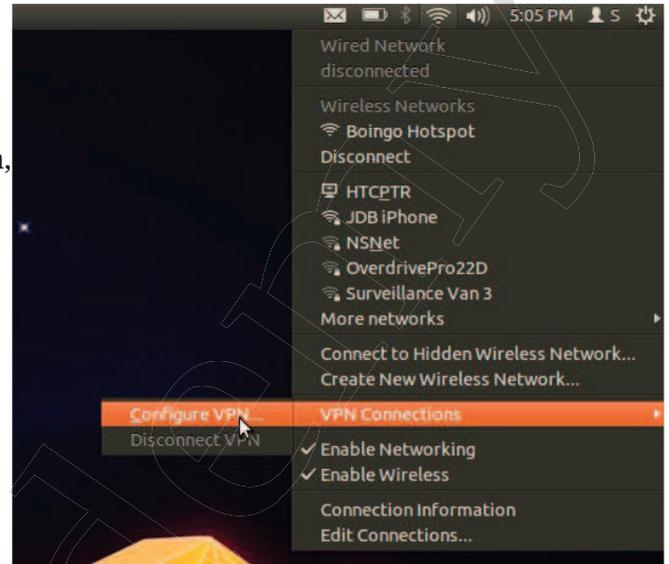
Asadar, când *network manager* spune “device” gândim “interfață”, iar când el spune “conexiune” gândim “profil de conectare”.

Nota: pe o interfață poate fi activ un singur profil la un moment dat; putem avea mai multe profile active, însă pe interfețe diferite.

8.1.6.2. Relația cu alte utilitare/modalități de configurare

Când este pornit, *Network Manager* preia controlul asupra întregii configurări a rețelei, acționând în mod automat în background și configurând interfețele definite în sistem. De aceea este bine să nu se folosească în paralel cu el alte utilitare de configurare de rețea deoarece vor exista conflicte. Chiar și în cazul configurării manuale a unei interfețe (folosind `ifconfig/ip`), aceasta va fi deseori suprascrisă automat de către *Network Manager*. Dacă se dorește configurarea manuală sau cu utilitare alternative, este necesară oprirea prealabilă a serviciului *network manager* sau scoaterea interfeței în cauză de sub controlul acestuia.

Pe de altă parte, *network manager* onorează configurarea persistentă efectuată în fișierele distribuției în cauză; spre exemplu, în Ubuntu, dacă o interfață este configurată în `/etc/network/interfaces`, ea nu va mai fi controlată de către *network manager*, iar applet-ul va indica acest lucru cu formularea “device not managed”. Dacă dorim ca o anumită interfață configurată manual să re-între în “patrimoniul” lui *network manager*, este necesar să eliminăm orice mențiune a ei din fișierul de configurare `interfaces`.



8.1.6.3. Configurarea folosind nmcli

Sintaxa generala a **nmcli** este:

```
nmcli <optiuni> element comanda
```

Elementul reprezinta categoria de operatii vizata si poate fi:

- **general** (prescurtat **g**) - pentru a manipula/vizualiza setari globale ale lui network manager
- **networking** (prescurtat **n**) - pentru a activa/dezactiva reseaua si a vizualiza starea sa curenta
- **radio** (prescurtat **r**) - pentru a activa/dezactiva interfete fara fir si a vizualiza starea lor curenta
- **connection** (prescurtat **c**) - pentru a manipula (activa/dezactiva/etc) conexiunile definite in sistem
- **device** (prescurtat **d**) - pentru a manipula lista de dispozitive administrate de network manager

In toate cazurile, comanda poate fi *help*, ceea ce afiseaza informatii utile despre comenzile posibile asociate elementului in cauza:

```
student@server:~$ nmcli connection help
Usage: nmcli connection { COMMAND | help }
COMMAND := { list | status | up | down | delete }

list [id <id> | uuid <id>]
status [id <id> | uuid <id> | path <path>]
up id <id> | uuid <id> [iface <iface>] [ap <BSSID>] [--nowait] [--timeout <timeout>]
down id <id> | uuid <id>
delete id <id> | uuid <id>
```

Prezentam in continuare cateva operatii utile realizabile prin intermediul *nmcli*:

- vizualizare stare curenta: **nmcli general**

```
student@server:~$ nmcli general
STATE      CONNECTIVITY  WIFI-HW  WIFI      WWAN-HW  WWAN
connected  full          enabled  enabled   enabled   disabled
```

- activare/dezactivare globala networking: **nmcli networking on/off**. *Atentie! La dezactivare vor fi deconfigurate toate interfețele administrate de catre network manager!*
- activare/dezactivare globala wifi: **nmcli radio wifi on/off** (atrage dupa sine si dezactivarea interfetelor wireless administrate de network manager)
- manipularea conexiunilor
 - listare conexiuni definite: **nmcli con**. In acest fel aflam numele conexiunilor definite in sistem si uuid-ul acestora, utilizabile in alte comenzi nmcli

```
student@server:~$ nmcli con
NAME                UUID                                  TYPE                TIMESTAMP-REAL
Wired connection 1  4cd7bb84-a2c2-45fd-87c8-bdb780bb7195  802-3-ethernet     Fri 30 Oct 2015 12:21:14 PM EET
test                8dd01e82-38fb-4e89-a3b0-14fb11accf69  802-3-ethernet     Fri 30 Oct 2015 11:16:36 AM EET
p2pl               91ef09ef-9a2b-441c-a642-110775b4bd0a  802-3-ethernet     Fri 30 Oct 2015 12:21:14 PM EET
```

- listare conexiuni active: **nmcli con show active**. In acest fel aflam care dintre conexiunile raportate de *nmcli con* este activa pe fiecare interfata
- activare/dezactivare conexiune: **nmcli con up/down nume**. *Atentie! A pune o conexiune in starea down doar dezactiveaza acea conexiune, ceea ce inseamna ca network manager va incerca apoi din nou sa configureze interfata in cauza folosind una dintre celelalte conexiuni atasate acelei interfete! Daca se doreste dezactivarea temporara a unei interfete - in sensul ca network manager sa nu mai*

incerce sa o configureze automat - este recomandabila folosirea lui `nmcli dev disconnect` (vezi mai jos)

- manipulare device-uri de retea administrate de catre network manager
 - listare device-uri: **nmcli dev**. De aici aflam numele device-urilor, utilizabile in comenzile urmatoare

```
student@server:~$ nmcli dev
DEVICE  TYPE      STATE
p2p1    ethernet  connected
p7p1    ethernet  connected
lo      loopback  unmanaged
```

- dezactivare temporara device, astfel incat acesta sa nu mai fie configurat automat de catre network manager: **nmcli dev disconnect nume_device**. Reactivarea se poate realiza cu `nmcli con up nume` sau cu `nmcli dev connect iface nume`, in functie de versiunea de nmcli utilizata
- management wifi
 - listare retele wifi detectate in zona: **nmcli dev wifi list**. De aici aflam SSID-urile (numele) retelelor si daca folosesc sau nu criptare

```
student@server:~$ nmcli dev wifi list
SSID          BSSID          MODE          FREQ          RATE          SIGNAL  SECURITY  ACTIVE
'office'      00:25:9C:DF:08:8D  Infrastructure  2437 MHz      54 MB/s      59      WPA WPA2  no
'hr'          C4:6E:1F:B2:FB:CC  Infrastructure  2422 MHz      54 MB/s      55      WPA2     no
```

- conectare la o retea wifi: **nmcli dev wifi connect SSID password parola**

Nota: in unele distributii ce contin versiuni mai vechi de network manager vom constata diferente precum:

- comenzile **general** si **networking** lipsesc; o parte din facilitatile lor se regaseste in comanda **nm**
- comanda **show** are alte doua variante, **status** sau **list**

8.2. Utilitare de diagnosticare a retelei

8.2.1. Comenzile mii-tool, ethtool si iwconfig

Aceste utilitare pot seta sau afisa configurarea de nivel fizic si legatura de date:

- in cazul interfetelor cu fir, **mii-tool** sau **ethtool** afiseaza daca exista o conexiune, viteza la care s-a realizat ea si modul de lucru (half sau full duplex). Comanda `mii-tool` poate fi folosita, spre exemplu, pentru a afla in care placa de retea a fost introdus cablul in cazul unei statii cu mai multe interfete
- in cazul celor fara fir, comanda **iwconfig** afiseaza parametrii conexiunii intre placa de retea si access point

```
root@Desktop # mii-tool eth0
eth0: 100 Mbit, full duplex, link ok
root@Desktop # ethtool eth0
Settings for eth0:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Half 1000baseT/Full
    Advertised auto-negotiation: Yes
    Speed: 100Mb/s
    Duplex: Full
```

```
[...restul output-ului a fost omis...]  
root@Desktop # iwconfig wlan0  
wlan0 IEEE 802.11 ESSID:"home"  
Mode:Managed Frequency:2.412 GHz Access Point: 00:14:BF:6B:AC:1E  
Bit Rate=54 Mb/s Tx-Power=20 dBm  
Retry min limit:7 RTS thr:off Fragment thr=2352 B  
Encryption key:25F7-B330-78E9-D4B6-D933-4856-EEBD-0B67 [2]  
Link Quality=70/100 Signal level=-42 dBm Noise level=-72 dBm  
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0  
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Nota: comanda `iwconfig` rulata fara argumente este o modalitate de a afla care dintre interfetele detectate de kernel sunt wireless si care de alta natura.

8.2.2. Comanda ping

Comanda **ping** este folosita pentru verificarea conectivitatii si a timpului dus-intors intre doua statii aflate in aceeasi retea sau retele diferite (inclusiv internet). Ea primeste ca argument adresa IP sau numele DNS al statiei cu care se doreste verificarea conectivitatii. Comanda trimite o succesiune de pachete (numite *echo request*) pentru care asteapta raspuns (*echo reply*). Se va afisa pe ecran cate o linie pentru fiecare raspuns primit, care include printre altele timpul dus-intors al pachetelor intre cele doua statii pentru acel raspuns. Rulata fara optiuni, comanda va continua sa trimita pachete pana cand este oprita manual de catre utilizator folosind combinatia control-c, moment in care va afisa statisticile rezultate:

```
root@Desktop # ping yahoo.com  
PING yahoo.com (209.131.36.159) 56(84) bytes of data.  
64 bytes from bl.www.vip.spl.yahoo.com (209.131.36.159): icmp_seq=1 ttl=41 time=210 ms  
64 bytes from bl.www.vip.spl.yahoo.com (209.131.36.159): icmp_seq=2 ttl=41 time=210 ms  
[...dupa mai multe raspunsuri primite, utilizatorul apasa control-c...]  
--- yahoo.com ping statistics ---  
10 packets transmitted, 10 received, 0% packet loss, time 9042ms  
rtt min/avg/max/mdev = 209.848/210.938/212.732/1.144 ms
```

Atunci cand cealalta statie raspunde cu un procent de pierderi mic sau nul, tragem concluzia ca statia in cauza este pornita, configurata corect si in plus pachetele nu sunt filtrate niciunde pe parcurs. Cand statia tinta nu raspunde sau ping evidentiaza pierderi mari, nu putem trage o concluzie unica ci trebuie sa investigam mai departe: spre exemplu, statia poate fi oprita, poate rula un firewall care opreste pachetele pe care le trimitem, sau poate exista o problema pe traseu (firewall, problema de rutare, congestie etc) care impiedica pachetele sa ajunga sau sa se intoarca de la destinatie.

Nota: comparativ cu Windows, ping rulat fara argumente prezinta in Linux urmatoarele deosebiri:

- nu se opreste automat dupa 4 pachete trimise, ci continua pana cand utilizatorul apasa CTRL-c
- nu afiseaza mesajele "request timed out" pentru pachetele ce nu primesc raspuns; exista insa optiunea -O care determina afisarea unui mesaj cu acelasi scop ("no answer yet for icmp_seq=1")

8.2.3. Comanda traceroute

Aceasta comanda afiseaza statiile/routerele intermediare prin care trec pachetele trimise de noi catre o anumita destinatie. Comanda primeste ca argument adresa IP sau numele statiei tinta si afiseaza, pentru fiecare router, timpul dus-intors, adresa si eventual numele DNS (daca exista – el fiind obtinut prin rezolutie DNS inversa).

```

root@Desktop # traceroute yahoo.com
traceroute to yahoo.com (209.131.36.159), 30 hops max, 40 byte packets
 1  36  13  16          72.249.128.105  -
 2  33  33  14          216.52.191.33   core2.tge5-1-bbnet1.ext1.dal.pnap.net
 3  26  28  32          4.59.36.65      xe-4-3-0.edge2.dallas3.level3.net
 4  57  44  47          4.68.17.67      ae-21-79.car1.washington1.level3.net
 5  *   *   *
 6  62  59  48          69.147.114.224  b1.www.vip.re3.yahoo.com
    
```

Ca optiuni utile ale comenzii *traceroute* amintim **-n** (numeric), care are ca efect afisarea simplei adrese a fiecarui router in parte, fara a se mai incerca determinarea numelui acestuia (operatie deseori consumatoare de timp), si optiunea **-I** care face ca traceroute sa foloseasca pachete ICMP (caci implicit este folosit UDP).

Comanda *traceroute* poate fi folosita pentru a determina eventualul segment in care conexiunea intre doua statii inceteaza sa functioneze. Spre exemplu, daca in lista afisata routerele inceteaza sa mai raspunda imediat dupa gateway-ul retelei noastre, vom banui fie o problema cu routerul, fie una de conexiune la internet (caz in care ne vom adresa providerului).

8.2.4. Comanda netstat

Aceasta comanda este folosita de obicei pentru afisarea conexiunilor active si a porturilor deschise. Rulata fara argumente ea va afisa conexiunile TCP active ale statiei si conexiunile interne inter-proces. Este necesara in general filtrarea sau imbogatirea acestui output, lucru realizat prin optiuni:

- **-t** – va afisa numai conexiunile TCP
- **-u** – va afisa numai “dialogurile” UDP (acesta nefiind un protocol orientat pe conexiune). Poate fi combinata cu optiunea **-t** pentru a obtine informatiile corespunzatoare ambelor protocoale
- **-n** – va afisa adresele si porturile in forma numerica, fara a mai incerca sa determine numele corespunzatoare adreselor si numele serviciilor care ruleaza traditional pe acele porturi
- **-a** – afiseaza nu numai conexiunile active, ci si porturile deschise (aplicatii server aflate in asteptarea cererilor de conexiune). Este foarte utila in determinarea listei de porturi deschise ale statiei
- **-p** – afiseaza suplimentar programul (PID si nume de executabil) care asculta pe fiecare port
- **-l** – afiseaza numai porturile TCP deschise (aflate in starea “listen”)

```

root@Desktop # netstat -tupan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*                LISTEN      2915/mysqld
tcp        0      0 0.0.0.0:22             0.0.0.0:*                LISTEN      5509/sshd
tcp        0      0 192.168.1.3:43622      76.13.15.38:5050        ESTABLISHED 5847/pidgin
tcp        0      0 192.168.1.3:35568      216.239.59.109:993      ESTABLISHED 4395/icedove-bin
tcp        0      0 192.168.1.3:48038      93.115.117.10:143      ESTABLISHED 4395/icedove-bin
tcp6       0      0 :::80                  :::*                    LISTEN      3516/apache2
tcp6       0      0 :::22                  :::*                    LISTEN      5509/sshd
    
```

Atunci cand, pentru o conexiune in starea LISTEN (port deschis), gasim pe coloana “Local address” adresa `0.0.0.0:numarport`, aceasta inseamna ca serverul in cauza asculta pe toate adresele IP definite in sistem. Aceeasi semnificatie o are combinatia `:::numarport`, dar de data aceasta este vorba despre adrese IPv6. După cum se va vedea în capitolele dedicate serverelor, un server poate fi configurat sa asculte numai pe o parte dintre adresele disponibile pe statia pe care ruleaza.

8.2.5. Comanda route

Aceasta comanda este in general folosita pentru manipularea rutelor, inasa poate fi considerata in mod egal o unealta de diagnostic deoarece prin intermediul sau se poate vizualiza tabela de rutare:

```
root@Desktop # route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.1.0      0.0.0.0         255.255.255.0  U         0      0      0 wlan0
10.0.0.0         0.0.0.0         255.255.255.0  U        1000    0      0 eth0
0.0.0.0          192.168.1.1    0.0.0.0        UG         0      0      0 wlan0
```

In acest fel putem determina probleme simple cu cauze majore – spre exemplu, lipsa rutei default, introducerea ei incorecta, dublarea etc.

Nota: in multe dintre Unix-uri comanda pentru afisarea tabelii de rutare este **netstat -rn**. Ea reprezinta acolo unica solutie, in timp de in Linux este o simpla alternativa la **route -n**.

8.2.6. Analizoare de protocol: tcpdump, tshark/wireshark

Aplicatiile de acest gen captureaza pachetele primite de interfata de retea specificata si le “traduc” pentru ochii utilizatorului, afisand intr-un format lizibil diferitele informatii ale protocoalelor incapsulate in pachete.

tcpdump este folosit in Linux si multe dintre Unix-uri, pe cand **tshark** si **wireshark** sunt aparitii mai recente dar cu mare popularitate. **tcpdump** si **tshark** actioneaza din linia de comanda, oferind o interfata text, pe cand **wireshark** dispune de interfata grafica si nenumarate facilitati.

Folosirea acestui tip de aplicatii este indicata numai atunci cand utilizatorul este familiarizat cu protocoalele stivei TCP/IP si poate diagnostica o problema analizand schimbul de pachete dintre statii. Analizoarele de protocol ofera cea mai “joasa” viziune a functionarii retelei pe o statie, inasa in acelasi timp pot diagnostica probleme care nu sunt vizibile in logurile sistemului sau ale aplicatiilor.

8.3. BIBLIOGRAFIE

- Lista rapida de comenzi ip: <http://lists.gslug.org/pipermail/gslug-general/2008-January/000318.html>
- Linux Advanced Routing&Traffic Control HOWTO: <http://lartc.org/howto>
- Documentatie iproute2: <http://www.policyrouting.org/iproute2.doc.html>
- Configurare retea in Fedora: https://docs.fedoraproject.org/en-US/Fedora/22/html/Networking_Guide/index.html
- Configurare retea in Debian: <http://people.debian.org/~osamu/pub/po4a/html/ch05.en.html>
- Noua strategie de denumire a interfetelor:
 - <http://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/>
 - http://cgkit.freedesktop.org/systemd/systemd/tree/src/udev/udev-builtin-net_id.c#n20