

O adresă de legătură locală IPv6 permite unui dispozitiv să comunice cu alte dispozitive de pe aceeași legătură și numai în respectiva legătură (subrețea). Pachetele cu o sursă sau destinație de legătură locală nu pot fi rutate în afara legăturii de unde provin.

Spre deosebire de adreselor de legătură locală IPv4, adrese de legătură locală IPv6 au un rol important în diferite aspecte ale rețelei. Adresa unicast globală nu este o necesitate; însă, orice interfață de rețea trebuie să aibă o adresă de legătură locală.

Dacă o adresă de legătură locală nu este configată manual pe o interfață, dispozitivul va crea automat propria adresă fără comunicarea cu un server DHCP. Hosturile ce permit IPv6 crează o adresă de legătură locală IPv6 chiar dacă dispozitivul nu a fost asignat cu o adresă IPv6 unic平 globală. Acest lucru permite ca dispozitivele IPv6 să comunice cu alte dispozitive IPv6 din aceeași subrețea. Acest lucru include comunicare cu default gateway (router).

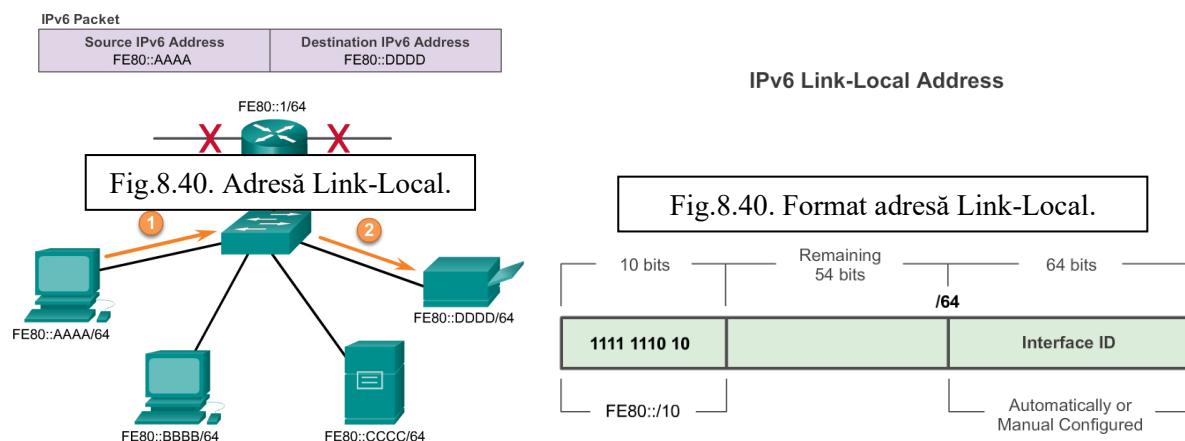
Adresele de legătură locală IPv6 sunt în spațiul FE80::/10. /10 ce indică faptul că primii 10 biți sunt 1111 1110 10xx xxxx. Primul hextet are spațiul de la 1111 1110 1000 0000 (FE80) la 1111 1110 1011 1111 (FEBF).

Fig.8.40 arată un exemplu de comunicare prin utilizarea adreselor de legătură locală IPv6. Fig.8.41 arată formatul unei adrese de legătură locală IPv6.

Adresele de legătură locală IPv6 sunt folosite și de protocoalele de rutare IPv6 în schimbul de mesaje ca adrese next-hop din tabela de rutare IPv6. Adresele de legătură locală IPv6 sunt discutate mai în detaliu în acest curs.

**Notă:** În mod normal, adresa de legătură locală a routerului și nu cea unicast globală este folosită ca default gateway pentru alte dispozitive din legătură (subrețea).

#### IPv6 Link-Local Communications



#### 8.14.4.1 Adresele IPv6 Unicast

Adresele unicast globale IPv6 sunt unice global și rutabile pe Internet IPv6. Aceste adrese sunt echivalente adreselor publice IPv4. The Internet Committee for Assigned Names and Numbers (ICANN), operatorul pentru Internet Assigned Numbers Authority (IANA), alocă blocurile de adresă IPv6 către cinci RIRs. Actual, numai adresele unicast globale cu primii trei biți de 001 sau 2000::/3 au fost atribuite. Reprezintă numai 1/8 din totalul spațiului de adrese IPv6 disponibile, excludând doar o foarte mică parte de alte tipuri de adrese unicast și multicast.

**Notă:** Adresa 2001:0DB8::/32 a fost rezervată pentru scopuri de documentare, inclusiv pentru utilizarea în exemple.

Fig.8.42 arată structura și spațiul unei adrese unicast globale.

O adresă unicast globală are trei părți:

- *Prefix de rutare global.*
- *ID-ul de subrețea.*
- *ID-ul de interfață.*

#### 8.14.4.2 Prefixul Global de Rutare

Prefixul de rutare global este prefixul, sau rețeaua, partea de adresă ce este atribuită de către furnizor, cum ar fi un ISP, clientului sau unei locații. Actual, RIRs a atribuit un prefix de rutare global /48 clienților. Acesta include pe oricine de la rețelele de întreprindere la gospodării individuale. Există mai mult decât necesar spațiu de adrese pentru alți clienți.

Fig.8.43 arată structura unei adrese unicast globale folosind un prefix de rutare global /48. Prefixele /48 sunt cele mai cunoscute prefixe de rutare globale și vor fi discutate mai mult pe exemplele din acest curs.

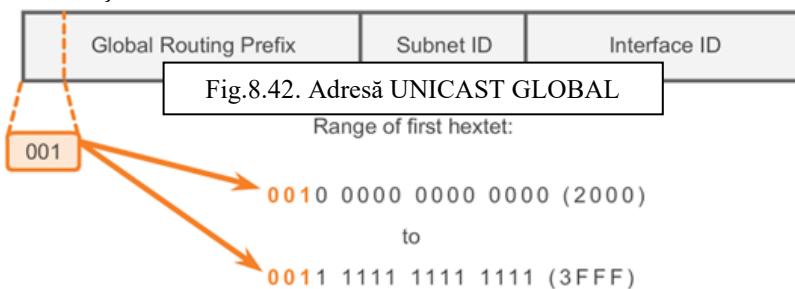
De exemplu, adresa IPv6 2001:0DB8:ACAD::/48 are un prefix ce indică faptul că primii 48 de biți (3 hexteți) (2001:0DB8:ACAD) este prefixul sau partea de rețea a adresei. “::” înseamnă că restul adresei conține zerouri.

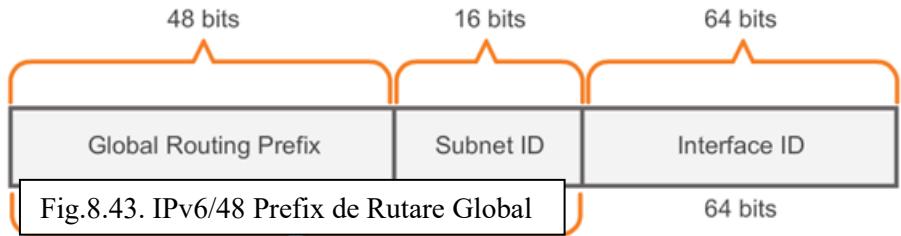
**IDul Subrețelei** - ID-ul de subrețea este folosit de o organizație pentru identificare subrețelelor din cadrul locației.

**IDul Interfeței** – Id-ul de interfață IPv6 este echivalent cu partea de host a unei adrese IPv4. Termenul de ID de interfață este folosit deoarece un singur host poate avea mai multe interfețe, fiecare având una sau mai multe adrese IPv6.

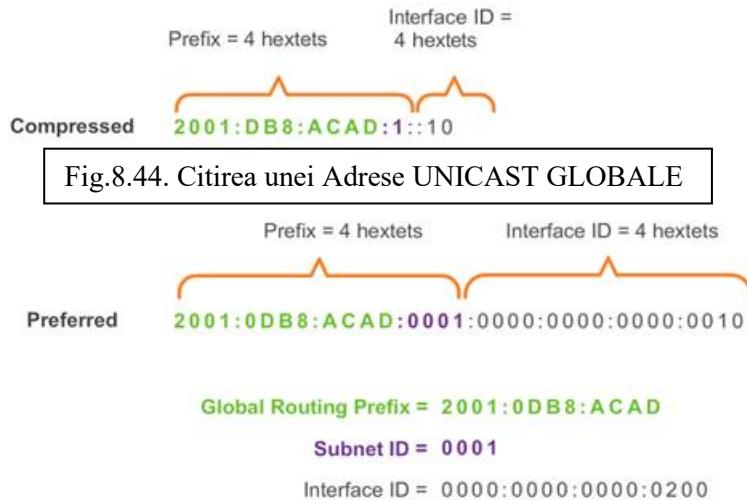
**Notă:** Spre deosebire de IPv4, în IPv6, adresa cu toți biții de 0 poate fi atribuită unui dispozitiv deoarece nu există adrese de broadcast în IPv6. Însă, adresa cu toți biții de 0 este rezervată ca adresă anycast Subnet-Router și ar trebui să fie atribuită numai routerelor.

O modalitate simplă de citire a celor mai multe adrese IPv6 este numărarea numărului de hexteți. Așa cum este arătat și în Fig. 3, într-o adresă unicast globală /64 primii patru hexteți sunt partea de rețea a adresei, cu patru hexteți indicând IDul subrețelei. Ceilalți patru hexteți reprezintă IDul de interfață.





A /48 routing prefix + 16 bit Subnet ID = /64 prefix.



## 8.15 ConFig.rea Routerului

Cele mai multe comenzi de conFig.re și verificare din IPv6 din IOS sunt similare cu cele din IPv4. În multe cazuri, singura diferență este utilizarea **ipv6** în locul **ip** din interiorul comenziilor.

Comanda **interface** de conFig.re a unei adrese unicast globale IPv6 pe o interfață este **ipv6 address ipv6-address/prefix-length**.

De reținut faptul că nu există spațiu dintre *ipv6-address* și *prefix-length*.

Exemplul de ConFig.re va folosi topologia din Fig. 1 și următoarele subrețele IPv6:

- 2001:0DB8:ACAD:0001:/64 (or 2001:DB8:ACAD:1::/64).
- 2001:0DB8:ACAD:0002:/64 (or 2001:DB8:ACAD:2::/64).
- 2001:0DB8:ACAD:0003:/64 (or 2001:DB8:ACAD:3::/64).

Așa cum se observă și în Fig. 2, comenziile necesare pentru conFig.rea unei adreselor unicast globale IPv6 pe o interfață GigabitEthernet 0/0 de pe R1 sunt:

**Router(config)#interface GigabitEthernet 0/0**

**Router(config-if)#description Legatura la Rețea 2001:db8:acad:1::0/64**

**Router(config-if)#ipv6 address 2001:db8:acad:1::1/64**

**Router(config-if)#no shutdown**

## 8.15 ConFig.rea Hostului

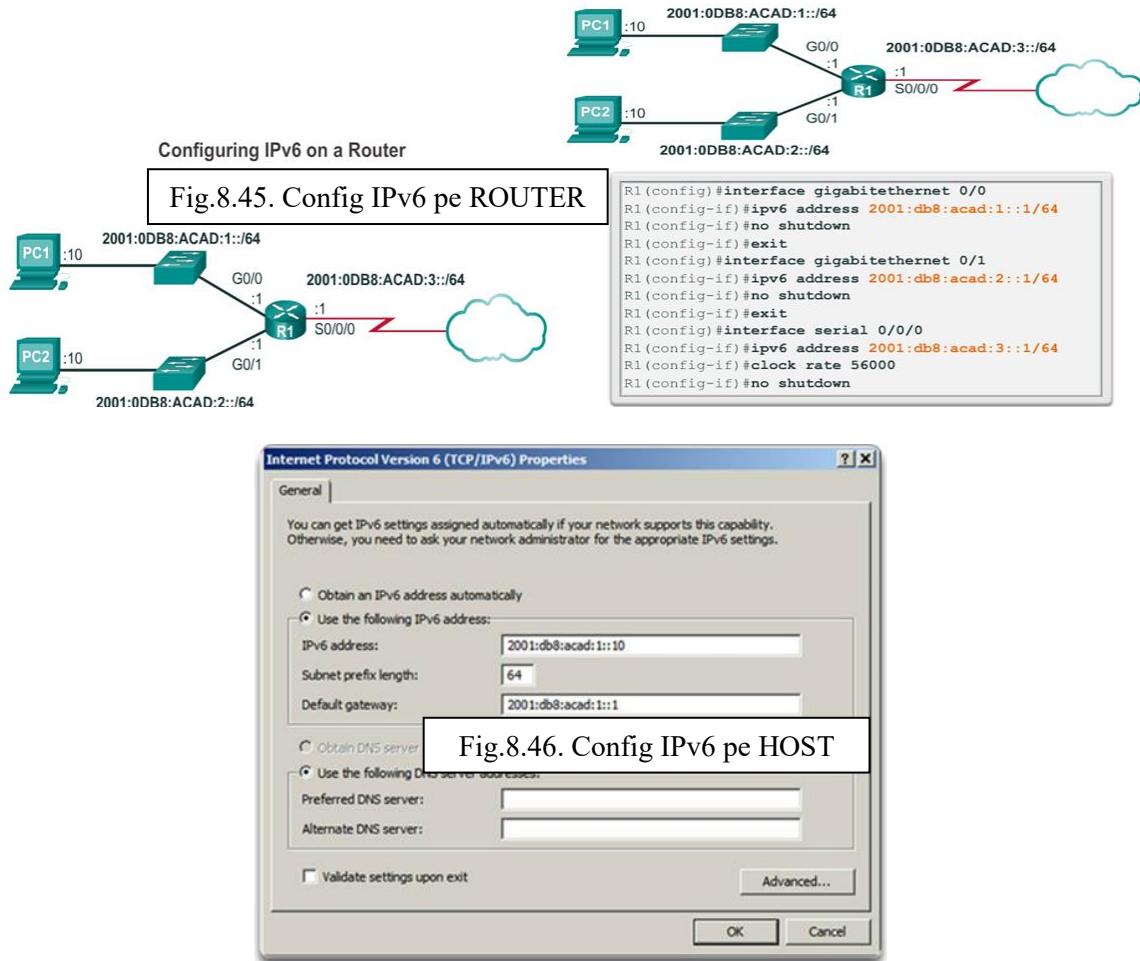
ConFig.rea manuală a unei adrese IPv6 pe un host este similară cu cea din IPv4.

Așa cum se observă și în Fig. 3, adresa de default gateway conFig.tă pentru PC1 este 2001:DB8:ACAD:1::1, adresa unicast globală a interfeței R1 GigabitEthernet este din aceeași rețea.

Ca și în IPv4, configurația adreselor în mod static pe clienți nu scalează bine în medii mari. Din acest motiv, mulți administratori de rețea dintr-o rețea IPv6 vor activa atribuirea dinamică a adreselor IPv6.

Există două moduri în care un dispozitiv poate obține o adresa unică globală IPv6 în mod automat:

- Stateless Address AutoConfiguration (SLAAC).
- DHCPv6.



### 8.15.1 Stateless Address AutoConfiguration (SLAAC)

Stateless Address AutoConfiguration (SLAAC) este o metodă ce permite unui dispozitiv să-și obțină prefixul, lungimea prefixului și informații de adresă de default gateway de la un router IPv6 fără utilizarea unui server DHCP. Folosind SLAAC, dispozitivele se bazează pe mesajele ICMPv6 Router Advertisement (RA) ale routerului local pentru a obține informațiile necesare.

Routerele IPv6 trimit periodic mesaje ICMPv6 Router Advertisement (RA) tuturor dispozitivelor activate cu IPv6 din rețea. Implicit, routerele Cisco trimit mesaje RA la fiecare 200 de secunde tuturor adreselor de grup multicast IPv6. Un dispozitiv IPv6 din rețea nu trebuie să aștepte mesajele periodice RA. Un dispozitiv poate trimite un mesaj Router Solicitation (RS) routerului, folosind adresa de grup multicast all-routers IPv6. Atunci când un router IPv6 primește un mesaj RS, va răspunde imediat cu un RA.

Chiar dacă o interfață de pe un router Cisco poate fi config.ă cu o adresă IPv6, nu îl face un router IPv6. Un router IPv6 este un router ce:

- Trimit pachete IPv6 între rețele.
- Poate fi config.ă cu rute statice IPv6 sau cu un protocol dinamic de rutare IPv6.
- Trimit mesaje ICMPv6 RA.

Rutarea IPv6 nu este activată implicit. Pentru a activa un router ca un router IPv6, comanda de config.ă globală **ipv6 unicast-routing** trebuie să fie folosită.

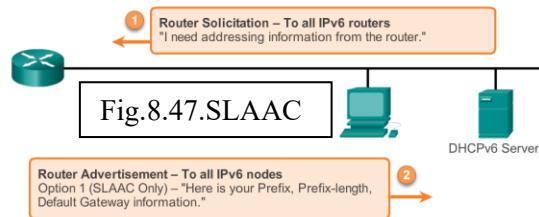
**Notă:** Routerele Cisco sunt activate ca routere IPv4 implicit.

Mesajul ICMPv6 RA conține prefixul, lungimea prefixului și alte informații pentru dispozitivul IPv6. Mesajul ICMPv6 RA informează de asemenea dispozitivul IPv6 de modul cum să-și obțină informațiile de adresare. Mesajul RA poate conține una dintre următoarele opțiuni, astăzi cum se observă și în Fig. :

- **Opțiunea 1 - SLAAC Only** – Dispozitivul ar trebui să folosească prefixul, lungimea prefixului și informațiile de adresa de default gateway conținute în mesajul RA. Nici-o altă informație nu este disponibilă de la serverul DHCPv6.
- **Opțiunea 2 – SLAAC and DHCPv6** – Dispozitivul ar trebui să folosească prefixul, lungimea prefixului și informațiile de adresa de default gateway în mesajul RA. Nici-o altă informație nu este disponibilă de la serverul DHCPv6, cum ar fi adresa de server DNS. Dispozitivul, prin intermediul procesului normal de descoperire și cerere la serverul DHCPv6, obține această informație suplimentară. Acest lucru este cunoscut ca stateless DHCPv6 deoarece serverul DHCPv6 nu trebuie să aloce sau să țină evidența nici-unei atribuiri de adresă IPv6, însă oferă numai informații suplimentare cum ar fi adresa de server DNS.
- **Opțiunea 3 – DHCPv6 only** – Dispozitivul nu trebuie să folosească informațiile din mesajul RA pentru informațiile de adresare. În schimb, dispozitivul va folosi procesul normal de descoperire și cerere la un server DHCPv6 pentru a obține toate informațiile sale de adresare. Acestea includ o adresă unică globală IPv6, lungimea prefixului, o adresa de default gateway și adresele serverelor DNS. În acest caz, serverul DHCPv6 se comportă ca un server stateful DHCP, similar lui DHCP pentru IPv4. Serverul DHCPv6 alocă și ține evidența adreselor IPv6, deci nu atribuie aceeași adresă IPv6 la mai multe dispozitive.

Routerele trimit mesaje ICMPv6 RA folosind adresa de legătură locală ca adresă sursă IPv6. Dispozitivele folosind SLAAC utilizează adresa de legătură locală a routerului ca adresa lor de default gateway.

Router Solicitation and Router Advertisement Messages



### 8.15.2 DHCPv6

Dynamic Host Configuration Protocol pentru IPv6 (DHCPv6) este similar DHCPului pentru IPv4. Un dispozitiv poate primi automat informațiile sale de adresare, inclusiv adresa unică globală IPv6, lungimea prefixului, o adresa de default gateway și adresele serverelor DNS folosind serviciile unui server DHCPv6.

Un dispozitiv poate primi toate sau unele informații de adresare IPv6 de la un server DHCPv6, în funcție de ce opțiune este specificată în mesajul ICMPv6 RA (opțiune SLAAC și DHCPv6 sau opțiunea numai DHCPv6). În plus, OS al hostului ar putea alege să ignore orice se află în mesajul RA de la router și să obțină adresa IPv6 proprie și alte informații direct de la un server DHCPv6.

Înainte de implementarea dispozitivelor IPv6 într-o rețea este o ideea bună de a verifica dacă un host observă opțiunile din mesajul ICMPv6 RA al routerului.

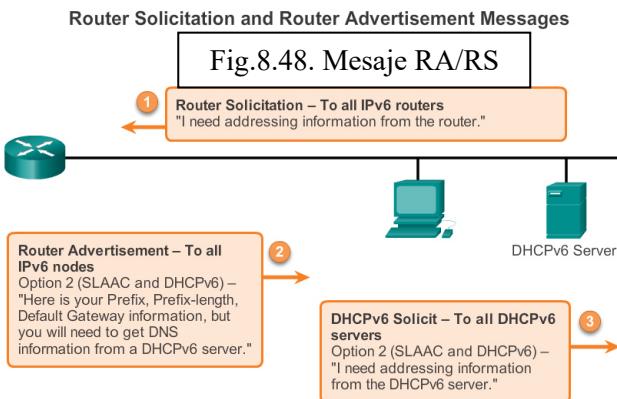
Un dispozitiv își poate obține adresa unicast globală IPv6 în mod dinamic și poate fi configurat cu mai multe adrese statice IPv6 pe aceeași interfață. IPv6 permite ca mai multe adrese IPv6, aparținând aceleiași rețele IPv6, să fie configurate pe aceeași interfață.

Un dispozitiv poate fi de asemenea configurat cu una sau mai multe adrese IPv6 de default gateway. Pentru mai multe informații în ceea ce privește decizia luată în ceea ce privește ce adresă este folosită ca o adresă sursă IPv6 sau care adresă default gateway este utilizată, de documentat RFC 6724, Default Address Selection pentru IPv6.

### 8.15.3 IDul Interfeței

În cazul în care clientul nu folosește informațiile conținute în mesajul RA și se bazează pe DHCPv6, serverul DHCPv6 va oferi întreaga adresă unicast globală IPv6, inclusiv prefixul și IDul interfeței.

Însă, dacă opțiunea 1 sau opțiunea 2 este folosită, clientul nu obține partea reală de ID de interfață a adresei din aceste procese. Dispozitivul client trebuie să determine propriul ID de interfață de 64 de biți, fie prin folosirea procesului EUI-64, fie prin generarea unui număr aleator de 64 de biți.



### 8.15.4 EUI-64

IEEE definește Extended Unique Identifier (EUI) sau procesul modificat EUI-64. Acest proces folosește o adresă Ethernet MAC de 48 de biți a clientului și inserează alti 16 biți în mijlocul adresei MAC de 48 de biți pentru a crea un ID de interfață de 64 de biți.

Adresele Ethernet MAC sunt de obicei reprezentate în hexazecimal și sunt alcătuite din două părți:

- **Organizationally Unique Identifier (OUI)** – OUI este un cod de furnizor de 24 de biți (6 cifre hexazecimale) atribuit de către IEEE.
- **Device Identifier** – Identificatorul dispozitivului este o valoare unică de 24 de biți dintr-un OUI.

Un ID de interfață EUI-64 este reprezentat în binar și este alcătuit din trei părți:

- OUI de 24 de biți de la adresa MAC a clientului, însă al 7-lea bit (the Universally/Locally (U/L) bit) este inversat. Acest lucru înseamnă că dacă al 7-lea bit este un 0 devine 1 și invers.
- Valoarea introdusă FFFE de 16 biți (în hexazecimal).
- Identifierul dispozitivului de 24 de biți de la adresa MAC a clientului.

Procesul EUI-64 este ilustrat în Fig. 1, folosind adresa MAC GigabitEthernet a lui R1: FC99:4775:CEE0.

**Pasul 1.** Împărțim adresa MAC între OUI și identifierul de dispozitiv.

**Pasul 2.** Inserăm valoarea hexazecimală FFFE, reprezentată în binar ca 1111 1111 1111 1110.

**Pasul 3.** Convertim primele două valori hexazecimale ale OUI în binar și inversăm al 7-lea bit. În acest exemplu, 0 de pe bitul 7 este schimbat în 1.

Rezultatul este un ID de interfață generat de EUI-64 : FE99:47FF:FE75:CEE0.

**Notă:** Utilizarea bitului U/L și motivele de inversare a valorii sale sunt discutate în RFC 5342.

Avantajul EUI-64 este că adresa Ethernet MAC poate fi folosită pentru a determina ID-ul interfeței. Permite de asemenea administratorilor de rețea să urmărească ușor o adresă IPv6 și un dispozitiv ce folosește adresa unică MAC. Însă, acest lucru a ridicat probleme de confidențialitate în rândul multor utilizatori. Sunt îngrijorați de faptul că pachetele lor pot fi urmărite la computerul fizic. Având în vedere aceste îngrijorări, poate fi folosit în schimb un ID de interfață generat aleator.

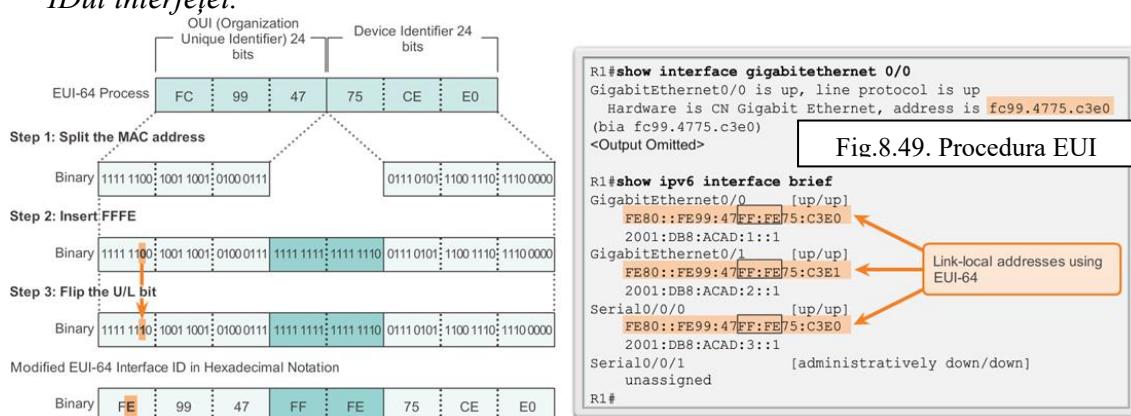
#### 8.15.5 ID de Interfață Generat Aleator

În funcție de sistemul de operare, un dispozitiv poate utiliza un ID de interfață generat aleator în schimbul adresei MAC și a procesului EUI-64. De exemplu, începând cu Windows Vista, Windows folosește un ID de interfață generat aleator în schimbul unuia creat cu EUI-64. Windows XP și sistemele de operare Windows de dinaintea lui, foloseau EUI-64.

Un mod simplu de identificare a faptului că o adresă a fost cel mai probabil creată cu EUI-64, FFFE este localizat în mijlocul ID de interfață, așa cum se observă și în Fig. 2.

După ce ID-ul interfeței este stabilit, fie prin procesul EUI-64, fie prin generarea aleatoare, poate fi combinat cu un prefix IPv6 pentru a crea o adresă unică globală sau o adresă de legătură locală:

- **Adresa unică globală** – Atunci când folosim SLAAC, dispozitivul își primește prefixul de la ICMPv6 RA și îl combină cu ID-ul de interfață.
- **Adresa de legătură locală** – Un prefix de legătură locală începe cu FE80::/10. Un dispozitiv folosește în mod normal FE80::/64 ca prefix/lungime de prefix, urmat de către ID-ul interfeței.



Atunci când folosește SLAAC (numai SLAAC sau SLAAC cu DHCPV6), un dispozitiv își primește prefixul și lungimea prefixului de la ICMPv6 RA. Deoarece prefixul adresei a fost atribuit de către mesajul RA, dispozitivul trebuie să ofere numai partea de ID de interfață a adresei sale. Așa cum spuneam mai devreme, IDul de interfață poate fi generat automat prin procesul EUI-64 sau, în funcție de sistemul de operare, generat aleator. Utilizând informațiile din mesajul RA și IDul de interfață, dispozitivul poate stabili adresa sa unicast globală.

După ce o interfață unicast globală este atribuită unei interfețe, dispozitivul IPv6 va genera automat adresa sa de legătură locală. Dispozitivele activate IPv6 trebuie să aibă, cel puțin, adresa de legătură locală. Reamintim faptul că adresa de legătură locală IPv6 permite unui dispozitiv să comunice cu alte dispozitive activate IPv6 din aceeași subrețea.

Adresele de legătură locală sunt folosite pentru mai multe scopuri, cum ar fi:

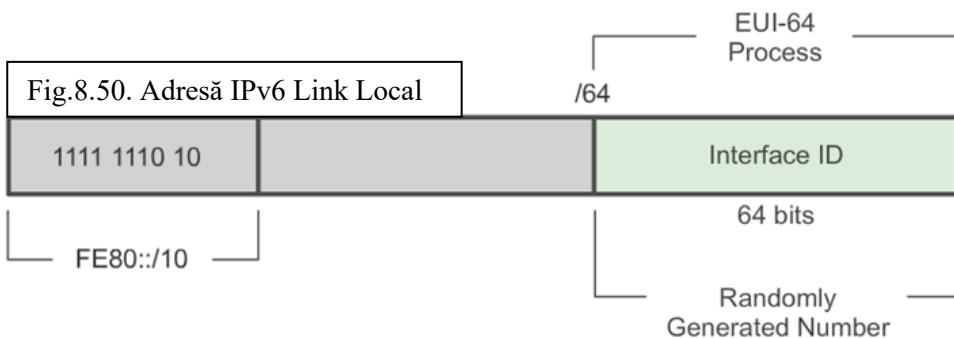
- *Un host folosește adresa de legătură locală a routerului local pentru adresa sa IPv6 de default gateway.*
- *Routerele schimbă mesaje de protocol de rutare dinamic folosind adresele de legătură locală.*
- *Tabelele de rutare ale routerelor folosesc adresa de legătură locală pentru a identifica routerul next-hop atunci când trimit pachete IPv6.*

O adresă de legătură locală poate fi stabilită dinamic sau configată manual ca o adresă statică de legătură locală.

#### 8.15.6 Asignarea Adresei de Link-Local Dinamic

Adresa de legătură locală este creată dinamic folosind prefixul FE80::/10 și IDul de interfață.

Implicit, IOSul routerelor folosesc EUI-64 pentru a genera IDul de interfață pentru toate adresele de legătură locală de pe interfețele IPv6. Pentru interfețele seriale, routerul folosește adresa MAC a unei interfețe Ethernet. Reamintim faptul că o adresă de legătură locală trebuie să fie unică numai pe legătură respectivă sau rețea. Însă, un pas înapoi în utilizarea atribuirii dinamice a adresei de legătură locală este lungimea sa, ce face dificilă identificare și ținerea evidenței adreselor atribuite.



#### 8.15.7 Static Link-Local Address

Configarea adresei de legătură locală manual oferă abilitatea de creare a unei adrese recunoscute și ușor de ținut minte.

Adresele de legătură locală pot fi configurate manual prin folosirea acelorași comenzi de interfață folosită pentru crearea adreselor unicast globale IPv6, dar cu un parametru suplimentar: **Router(config-if)#ipv6 address link-local-address link-local**

Fig.8.50 arată faptul că o adresă de legătură locală are un prefix din spațiul de adrese de la FE80 la FEBF. Atunci când o adresă începe cu acest hextet, parametrul de legătură locală trebuie să urmeze adresa.

Fig.8.51 arată configurația unei adrese de legătură locală prin folosirea comenzi **ipv6 address interface**. Adresa de legătură locală FE80::1 este folosită pentru a face mai ușor de recunoscut faptul că aparține routerului R1. Aceeași adresă IPv6 de legătură locală este configurată pe toate interfețele lui R1. FE80::1 poate fi configurată pe fiecare legătură deoarece este unică pe respectiva legătură.

Similar cu R1, routerul R2 va fi configurat cu FE80::2 ca adresa IPv6 de legătură locală pe toate interfețele sale.

```

R1(config)#interface gigabitethernet 0/0
R1(config-if)#ipv6 address fe80::1 ?
  link-local  Use link-local address

R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#

```

R1#show ipv6 interface brief	
GigabitEthernet0/0	[up/up]
FE80::1	
2001:DB8:ACAD:1::1	
GigabitEthernet0/1	[up/up]
FE80::1	
2001:DB8:ACAD:2::1	
Serial0/0/0	[up/up]
FE80::1	
2001:DB8:ACAD:3::1	
Serial0/0/1	[administratively down/down]
unassigned	

R1#

**Statically configured link-local addresses**

Fig.8.51. Config IPv6 Link Local

Așa cum se observă în Fig.8.51, comanda de verificare a configurației de interfață IPv6 este similară cu comanda folosită în IPv4.

Comanda **show interface** afișează adresa MAC a interfețelor Ethernet. EUI-64 folosește adresa MAC pentru a genera ID-ul de interfață pentru adresele de legătură locală. În plus, comanda **show ipv6 interface brief** afișează detalii pentru fiecare interfață. Afisajul [up/up] din aceeași linie indică starea interfeței de nivel 1/2. Acest lucru este la fel cu Status și Protocol din comanda IPv4.

De remarcat faptul că fiecare interfață are două adrese IPv6. A doua adresă pentru fiecare interfață este adresa unică globală configurată. Prima adresă, cea care începe cu FE80, este adresa unică de legătură locală pentru interfață. Reamintim faptul că adresa unică de legătură locală este adăugată automat interfeței atunci când este atribuită o adresă unică globală.

De asemenea, este de remarcat faptul că adresa de legătură locală de pe Serial 0/0/0 a lui R1 este aceeași cu cea de pe interfață GigabitEthernet 0/0. Interfețele seriale nu au adresa MAC și, prin urmare, IOSul folosește adresa MAC a primei interfețe Ethernet disponibile. Acest lucru este posibil deoarece interfețele de legătură locală trebuie să fie unice doar pe respectiva legătură.

Adresa de legătură locală de pe interfață routerului este de obicei adresa default gateway pentru dispozitivele din respectiva legătură sau rețea.

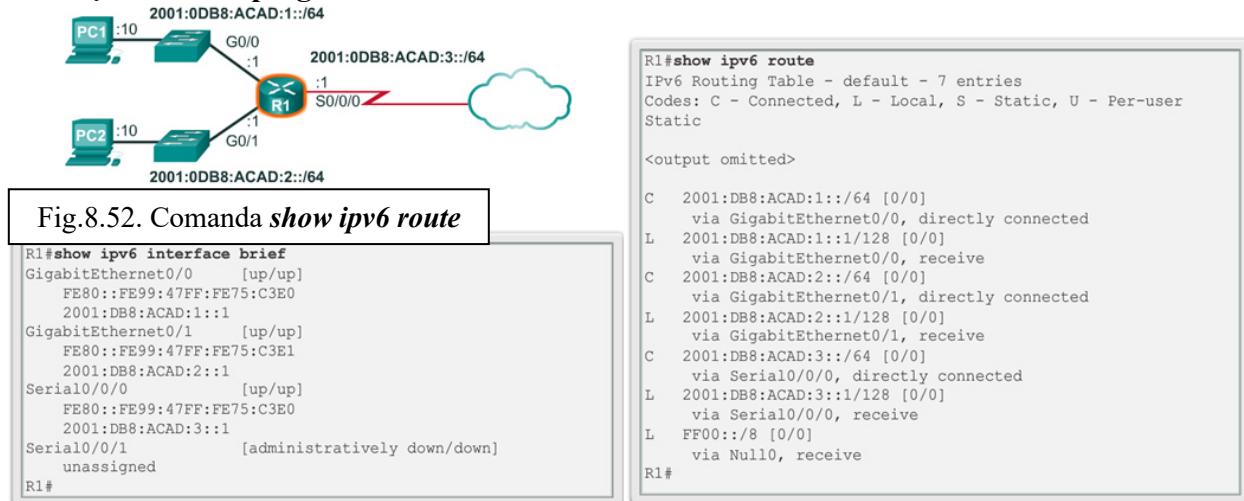
Așa cum se poate observa în Fig.8.52, comanda **show ipv6 route** poate fi folosită pentru a verifica faptul că rețelele IPv6 și adresele de interfețe specifice au fost instalate în tabela de rutare IPv6. Comanda **show ipv6 route** va afișa numai rețelele IPv6, nu și pe cele IPv4.

În tabela de rutare, un C în dreptul unei rute indică faptul că respectiva rută este o rețea direct conectată. Atunci când o interfață de router este configurată cu o adresă unică globală și este în starea "up/up", prefixul și lungimea prefixului IPv6 sunt adăugate în tabela de rutare IPv6 ca rute conectate.

Adresa unică globală IPv6 configurată pe interfață este de asemenea instalată în tabela de rutare ca rută locală. Ruta locală are un prefix /128. Rutele locale sunt folosite de tabela de rutare pentru procesarea eficientă a pachetelor cu o adresă destinație a adresei de interfață a routerului.

Comanda **ping** pentru IPv6 este identică ca cea pentru IPv4, cu excepția faptului că o adresă IPv6 este folosită. Așa cum se observă în Fig.8.53, comanda este folosită pentru verificare conectivității de nivel 3 dintre R1 și PC1. Atunci când de pe un router se dă **ping** către o adresă

de legătură locală, IOSul va cere utilizatorului interfața de ieșire. Deoarece adresa destinație de legătură locală poate fi pe una sau mai multe legături sau rețele, routerul are nevoie să știe pe care interfață să trimită ping.



```
R1#ping 2001:db8:acad:1::10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:1::10, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
R1#
```

**Fig.8.53. Comanda *ping***

### 8.15.8 Adresele IPv6 Multicast

Adresele multicast IPv6 sunt similare cu adresele multicast IPv4. Reamintim faptul că o adresă multicast este folosită pentru a trimite un singur pachet la una sau mai multe destinații (grup multicast). Adresele multicast IPv6 au prefixul FF00::/8.

**Notă:** Adresele multicast pot fi numai adrese destinație și nu adrese sursă.

Există două tipuri de adrese IPv6 multicast:

- *Assigned multicast*.
- *Solicited node multicast*.

#### 8.15.8.1 Assigned Multicast

Adresele multicast alocate sunt adrese de multicast rezervate pentru grupuri predefinite de dispozitive. O adresă multicast alocată este o singură adresă folosită pentru a ajunge la un grup de dispozitive ce rulează un protocol sau serviciu comun. Adresele multicast alocate sunt folosite cu anumite protocole, cum ar fi DHCPv6.

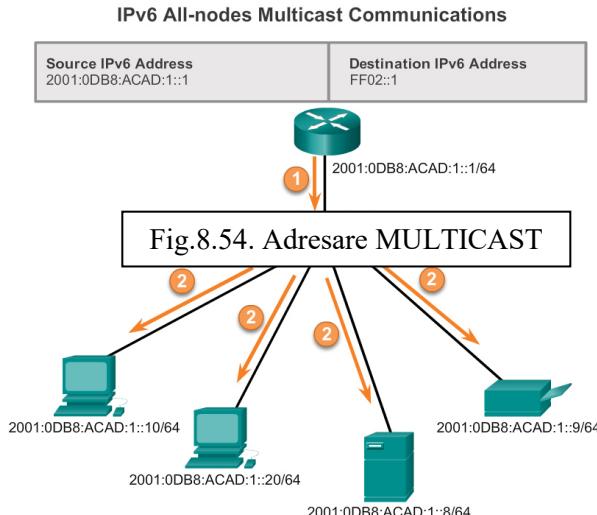
Două grupuri comune multicast alocate IPv6 sunt:

- **FF02::1 All-nodes multicast group** – Acesta este un grup multicast din care fac parte toate dispozitivele activate cu IPv6. Un pachet trimis la acest grup este primit și procesat de către toate interfețele IPv6 din legătură sau rețea. Aceasta are același efect ca o adresă de broadcast din IPv4. Fig. arată un exemplu de comunicare ce folosește all-nodes multicast address. Un router IPv6 trimite mesaje Internet Control Message Protocol version 6 (ICMPv6) RA tuturor nodurilor din

grupul multicast. Mesajul RA informează toate dispozitivele activate IPv6 din rețea despre informațiile de adresare, cum ar fi prefixul, lungimea prefixului și default gateway.

- **FF02::2 All-routers multicast group** – Aceasta este un grup multicast din care fac parte toate routerele IPv6. Un router devine un membru al grupului atunci când este activat ca router IPv6 cu comanda de configurație globală **ipv6 unicast-routing**. Un pachet trimis la acest grup este primit și procesat de către toate routerele IPv6 din legătură sau rețea.

Dispozitivele activate IPv6 trimit mesaje ICMPv6 Router Solicitation (RS) către all-routers multicast address. Mesajul RS cere un mesaj RA de la routerul IPv6 pentru a ajuta dispozitivul în configurația sa de adresă.



Un **"solicited-node"** multicast este similar cu **"all-nodes"** multicast address. Reamintim faptul că all-nodes multicast address este în esență același lucru cu un broadcast IPv4. Toate dispozitivele din rețea trebuie să proceseze traficul trimis la adresă. Pentru a reduce numărul de dispozitive ce trebuie să proceseze traficul, folosim o adresă solicited-node multicast address.

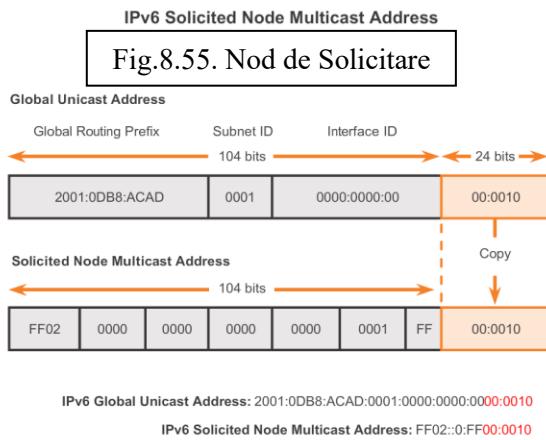
O solicited-node multicast address este o adresă ce are aceeași ultimi 24 de biți cu adresa unică globală IPv6 a dispozitivului. Singurele dispozitive ce trebuie să proceseze pachetele sunt acele dispozitive ce au aceeași ultimi 24 de biți, în extrema dreptă a ID-ului lor de interfață.

O adresă IPv6 solicited-node multicast address este creată automat atunci când este atribuită o adresă unică globală sau una unică de legătură locală. IPv6 solicited-node multicast address este creată prin combinarea prefixului special FF02:0:0:0:0:FF00::/104 cu extrema dreaptă de 24 de biți a adresei unice.

Solicited-node multicast address este alcătuită din două părți:

- **FF02:0:0:0:0:FF00::/104 multicast prefix** – Aceștia sunt primii 104 biți ai all solicited-node multicast address.
- **Least significant 24-bits** – Aceștia sunt ultimii 24 de biți din extrema dreaptă a solicited-node multicast address. Acești biți sunt copiați din cei 24 de biți de extrema dreaptă ai adresei unice globale sau unice de legătură locală a dispozitivului.

Este posibil ca mai multe dispozitive să aibă aceeași solicited-node multicast address. Deși rar, acest lucru se poate întâmpla atunci când dispozitivele au aceeași 24 de biți în extrema dreaptă a ID-urilor de interfață. Acest lucru nu crează nici-o problemă deoarece dispozitivul va procesa în continuare mesajul încapsulat, ce va include adresa completă IPv6 a dispozitivului.



## 8.16 Verificarea Connectivității

### 8.16.1 ICMP

Deși IP nu este un protocol de încredere, suita TCP/IP asigură ca mesajele să fie trimise în cazul în care au loc anumite erori. Aceste mesaje sunt trimise folosind serviciile ICMP. Scopul acestor mesaje este de a oferi feedback despre problemele legate de procesarea pachetelor IP în anumite condiții, nu pentru a face IP de încredere. Mesajele ICMP nu sunt necesare și sunt adesea nepermise în rețea din motive de securitate.

ICMP este disponibil pentru IPv4 și IPv6. ICMPv4 este protocolul de mesagerie pentru IPv4. ICMPv6 oferă aceleași servicii pentru IPv6, însă include funcționalități adiționale. În acest curs, termenul de ICMP va fi folosit pentru ambele, ICMPv4 și ICMPv6.

Tipurile de mesaje ICMP și motivele pentru care sunt transmise sunt multe. Vom discuta unele dintre aceste multe mesaje.

Mesajele ICMP comune ambelor, ICMPv4 și ICMPv6, includ:

- *Confirmarea hostului.*
- *Destinație sau serviciu fără acoperire.*
- *Timp depășit.*
- *Redirecționarea rutei.*

*Confirmarea hostului* – Un ICMP Echo Message poate fi folosit pentru a determina dacă un host este operațional. Hostul local trimite un ICMP Echo Request unui host. Dacă hostul este disponibil, hostul destinație răspunde cu un mesaj Echo Reply. Această utilizarea a mesajelor ICMP Echo este baza utilitarului **ping**.

*Destinație sau serviciu fără acoperire* – Atunci când un host sau gateway primește un pachet ce nu poate fi livrat, poate folosi un mesaj ICMP Destination Unreachable pentru a notifica sursa de faptul că destinația sau serviciul nu este disponibil. Mesajul va include un cod ce indică de ce pachetul nu poate fi livrat.

Unele dintre codurile de destinație inaccesibile pentru ICMPv4 sunt:

- 0 – *net inaccesibil.*
- 1 - *host inaccesibil.*
- 2 - *protocol inaccesibil.*
- 3 - *port inaccesibil.*

**Notă:** ICMPv6 are coduri similare, însă puțin diferite pentru mesajele de destinație inaccesibilă.

*Timp depășit* – Un mesaj ICMPv4 Time Exceeded este folosit de către un router pentru a indica faptul că un pachet nu poate fi expediat mai departe deoarece câmpul TTL al pachetului a fost decrementat până la 0. Dacă un router primește un pachet și decrementează câmpul TTL din pachetul IPv4 la zero, aruncă pachetul și trimite un mesaj ICMPv4 Time Exceeded hostului sursă.

ICMPv6 trimite de asemenea un mesaj Time Exceeded în cazul în care routerul nu poate transmite un pachet IPv6 deoarece pachetul “a expirat”. IPv6 nu are un câmp TTL; folosește câmpul de limită de hop pentru a determina dacă pachetul a expirat.

*Redirecționarea rutelor* – Un router ar putea folosi mesajul ICMP Redirect Message pentru a notifica hosturile din rețea de faptul că o rută mai bună este disponibilă pentru o destinație particulară. Acest mesaj poate fi folosit numai atunci când hostul sursă se află pe aceeași rețea fizică cu ambele gatewayuri.

Ambele, ICMPv4 și ICMPv6, folosesc mesaje de redirecționare de rută.

ICMPv4 Ping to a Remote Host

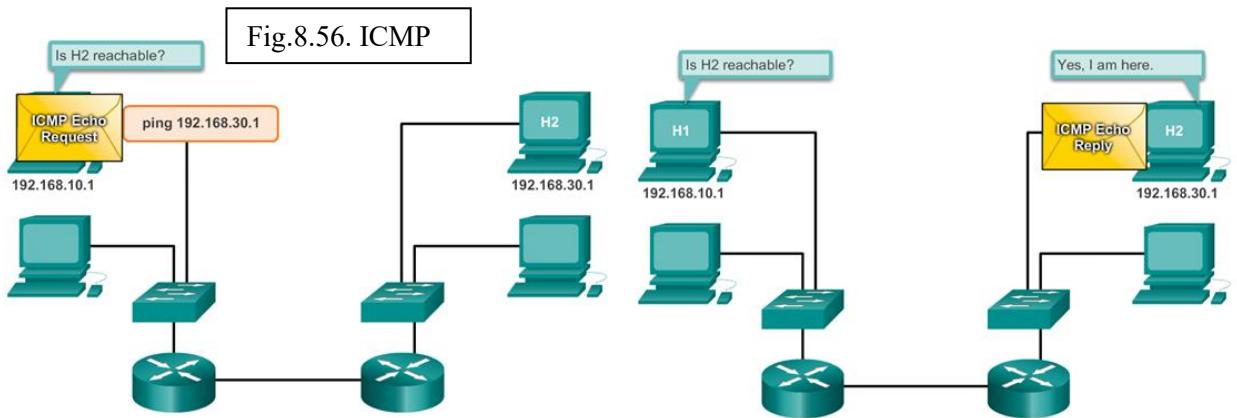


Fig.8.56. ICMP

Mesajele informaționale și de eroare aflate în ICMPv6 sunt foarte similare cu mesajele de control și eroare implementate de către ICMPv4. Însă, ICMPv6 are noi caracteristici și funcționalități îmbunătățite, neîntâlnite în ICMPv4.

ICMPv6 include patru noi protocoale ca parte a Neighbor Discovery Protocol (ND sau NDP):

- *Router Solicitation message.*
- *Router Advertisement message.*
- *Neighbor Solicitation message.*
- *Neighbor Advertisement message.*

#### 8.16.2 Router Solicitation și Router Advertisement Messages

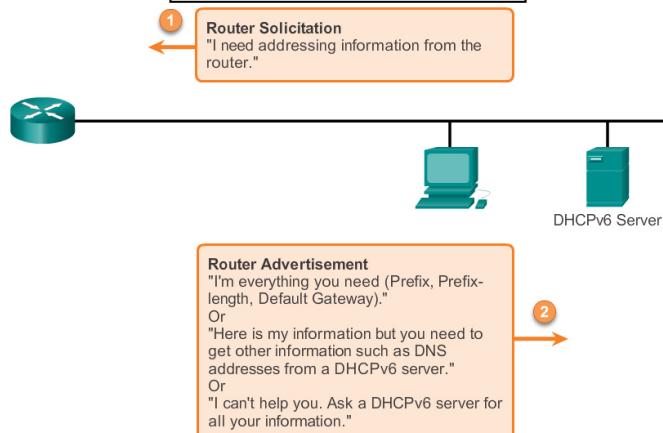
Dispozitivele activate cu IPv6 pot fi împărțite în două categorii, routere și hosturi. Mesajele Router Solicitation și Router Advertisement sunt trimise între hosturi și routere.

- **Mesajul Router Solicitation (RS):** Atunci când un host este configurat pentru a-și obține informațiile de adresare automat cu ajutorul Stateless Address Autoconfiguration (SLAAC), hostul va trimite un mesaj RS routerului. Mesajul RS este trimis ca un mesaj multicast IPv6 all-routers.
- **Mesajul Router Advertisement (RA):** Mesajele RA sunt trimise de către routere pentru a furniza informații de adresă hosturilor folosind SLAAC. Mesajul RA poate conține informații de adresare pentru host, cum ar fi prefixul și lungimea prefixului. Un router va transmite un mesaj RA periodic sau ca răspuns pentru un mesaj RS. Implicit, routerele

Cisco trimite mesaj RA la fiecare 200 de secunde. Mesajele RA sunt trimise la adresa multicast IPv6 all-nodes. Un host ce folosește SLAAC va transmite propriul default gateway la adresa de legătură locală a routerului ce transmite mesajul RA.

Router Solicitation and Router Advertisement Messages

Fig.8.57. Mesaje RS/RA



ICMPv6 Neighbor Discovery Protocol include două tipuri de mesaje suplimentare, mesaje Neighbor Solicitation (NS) și Neighbor Advertisement (NA).

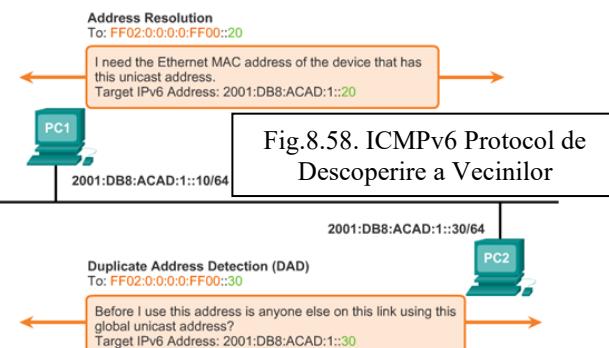
Mesajele Neighbor Solicitation (NS) și Neighbor Advertisement (NA) sunt folosite pentru:

- *Rezoluția adresei.*
- *Detectarea adresei după (DAD).*

*Rezoluția adresei* – Rezoluția de adresă este folosită atunci când un dispozitiv de pe LAN știe adresa unică IPv6 a unei destinații, dar nu știe adresa Ethernet MAC. Pentru a determina adresa MAC a destinației, dispozitivul va transmite un mesaj NS adresei de nod solicitată. Mesajul va include adresa IPv6 cunoscută. Dispozitivul care are adresa IPv6 cunoscută va răspunde cu un mesaj NA ce conține adresa Ethernet MAC.

*Detectarea adresei după* – Atunci când un dispozitiv are atribuită o adresă unică globală sau una unică de legătură locală, este recomandat ca DAD să fie efectuat pe adresa pentru asigurarea de faptul că este unică. Pentru a verifica unicitatea unei adrese, dispozitivul va trimite un mesaj NS cu propria adresă IPv6 ca adresa IPv6 țintă. Dacă alt dispozitiv din rețea are această adresă, va răspunde cu un mesaj NA. Acest mesaj NA va înștiința sursa de faptul că adresa este folosită. Dacă un mesaj NA nu este primit într-o anumită perioadă de timp, adresa unică este unică și utilizabilă.

**Notă:** DAD nu este necesar, însă RFC 4861 recomandă ca DAD să fie efectuat pe adresele unice.



## 8.17 Testare și Verificare

**Ping** este utilitarul de testare ce folosește meseje ICMP echo request și echo reply pentru a testa conectivitatea dintre hosturi. **Ping** funcționază cu ambele tipuri de adrese, IPv4 și IPv6.

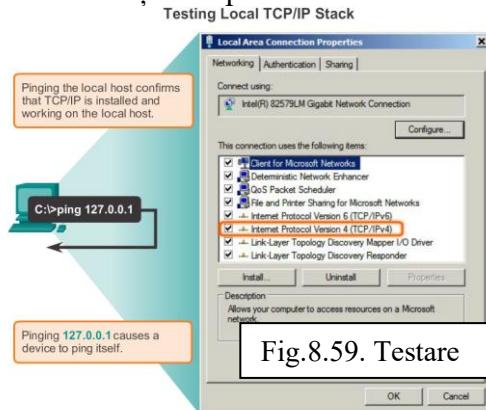
Pentru a testa conectivitatea cu un alt host dintr-o rețea, un echo request este trimis adresei de host prin folosirea comenzi **ping**. Dacă hostul de la adresa specificată primește echo request, răspunde cu un echo reply. Cu fiecare echo reply primit, **ping** oferă feedback în timpul în care cererea a fost trimisă și confirmarea a fost primită. Acest lucru poate fi privit ca o măsură a performanței rețelei.

**Ping** are o valoare de timeout pentru reply. Dacă un răspuns nu a fost primit în acest timp, **ping** oferă un mesaj ce indică faptul că un răspuns nu a fost primit. Acest lucru indică de obicei că există o problemă, dar și faptul că anumite caracteristici de securitate ce blochează mesajele **ping** au fost activate în rețea.

După ce toate cererile au fost trimise, utilitarul **ping** oferă un rezumat ce include rata de succes și timpul mediu de dus-întors de la destinație.

**Pinging în adresa de Local Loopback** – Există unele cazuri speciale de testare și verificare pentru care putem utiliza **ping**. Un caz este testarea configurației interne de IPv4 sau IPv6 pe hostul local. Pentru a efectua acest test, folosim **ping** pe adresa de loopback local 127.0.0.1 pentru IPv4 (:1 pentru IPv6). Testarea IPv4 loopback este evidențiată în Fig. de mai jos.

Un răspuns de la 127.0.0.1 pentru IPv4 (:1 pentru IPv6), indică că un IP este instalat adecvat pe host. Acest răspuns provine de la nivelul rețea. Acest răspuns nu este însă o indicație a faptului că adresele, măștile sau gateways sunt corecte. Nu indică nimic despre statusul nivelului inferior al stivei de rețea. Testează pur și simplu IPul la nivelul de rețea IP. Dacă primim un mesaj de eroare, este o indicație a faptului că TCP/IP nu este operational pe host.

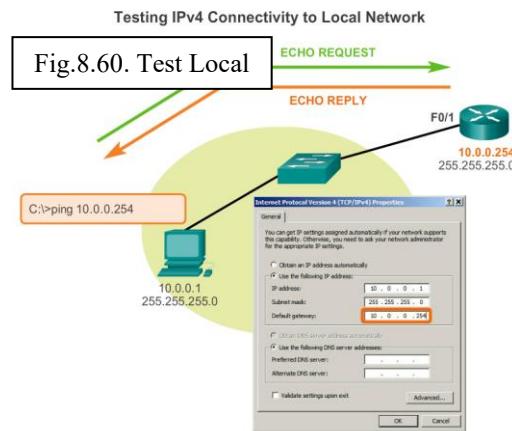


Putem folosi **ping** pentru a testa de asemenea abilitatea unui host de a comunica cu rețeaua locală. Acest lucru se realizează în general prin efectuarea de **ping** spre gateway a hostului. Un **ping** spre gateway indică faptul că un host și interfața routerului sunt operaționale pe rețeaua locală.

Pentru acest test, adresa de gateway este cea mai utilizată deoarece routerul este, în mod normal, operațional. Dacă adresa de gateway nu răspunde, un **ping** poate fi trimis la adresa IP a altui host din rețeaua locală ce este cunoscut ca fiind operațional.

Dacă gateway sau alt host răspunde, hostul local poate comunica cu succes peste rețeaua locală. Dacă gateway nu răspunde, dar un alt host răspunde, se poate indica faptul că este o problema cu interfața routerului (gateway).

O posibilitate este aceea că o adresă greșită de gateway a fost configurată pe host. O altă posibilitate este ca interfața routerului să fie funcțională, însă să aibă securitate aplicată ce împiedică procesarea și răspunsul la cererile de **ping**.

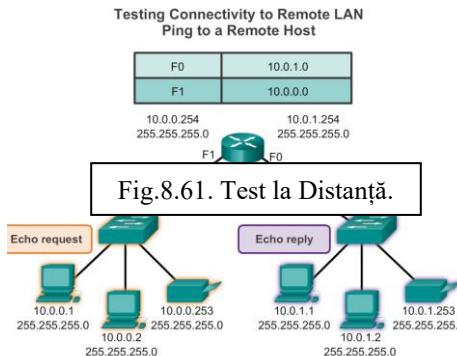


**Ping** poate fi de asemenea folosit pentru a testa abilitatea unui host local de a comunica prin internetwork. Hostul local poate da **ping** către un host IPv4 funcțional de pe o rețea de la distanță, aşa cum se poate vedea și în Fig. .

Dacă acest **ping** este cu succes, funcționarea unei mari bucăți din internetwork poate fi verificată. Un **ping** cu succes în internetwork confirmă comunicarea din rețeaua locală, funcționalitatea routerului (gateway) și funcționalitatea tuturor celorlalte routere ce se află în calea dintre rețeaua locală și rețeaua hostului de la distanță.

În plus, funcționalitatea hostului de la distanță poate fi verificată. Dacă hostul de la distanță nu poate comunica în exteriorul rețelei sale locale, nu va putea răspunde.

**Notă:** Mai mulți administratori de rețea limitează sau interzic intrarea mesajelor ICMP în rețeaua întreprinderii; prin urmare, lipsa unui răspuns **ping** poate fi provocată de restricțiile de securitate.



**Ping** este folosit pentru testarea comunicării dintre două hosturi, însă nu oferă informații despre detalii ale dispozitivelor dintre hosturi. Traceroute (**tracert**) este o utilitarul ce generează o listă de hopuri ce au fost atinse cu succes de-a lungul traseului. Lista poate oferi informații importante de verificare și depanare. Dacă datele ajung la destinație, **tracert** listează interfața fiecărui router din calea celor două hosturi. Dacă datele ajung doar până la un anumit hop din drum, adresa ultimului router ce a răspuns poate oferi o indicație cu privire la locul în care sunt găsite probleme sau restricții de securitate.

**Round Trip Time (RTT)** – Folosirea lui traceroute oferă **Round Trip Time (RTT)** pentru fiecare hop din cale și indică dacă un hop nu răspunde cu succes. **Round trip time** este timpul necesar pentru ca un pachet să ajungă la hostul de la distanță sau pentru ca un răspuns să ajungă la sursă. Simbolul “(\*)” indică un pachet pierdut sau fără răspuns.

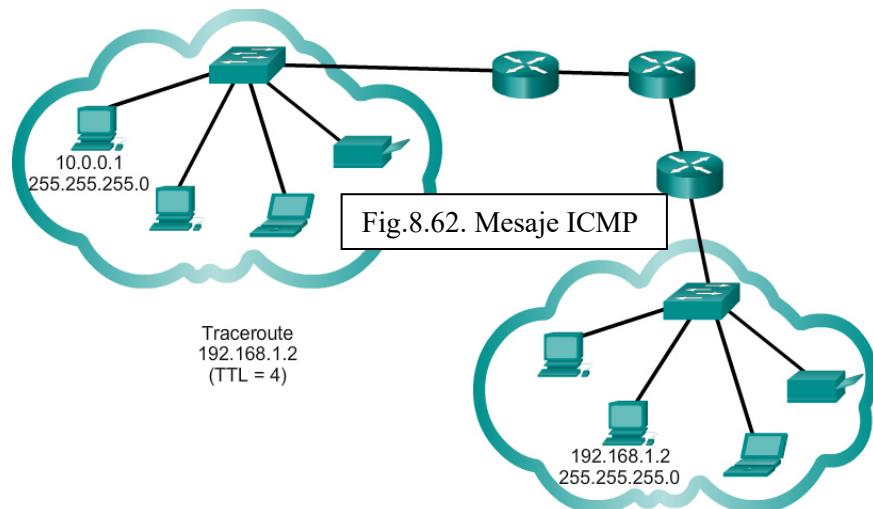
Acstea informații pot fi folosite pentru a localiza un router cu probleme din cale. Dacă afișajul arată tempi de răspuns mari sau pierderi de date de la un anumit hop, este o indicație a faptului că resursele routerului sau conexiunile sale pot fi "stresate".

**IPv4 Time-to-Live (TTL) and IPv6 Hop Limit** – Traceroute folosește o funcție de câmp TTL din IPv4 și Hop limit din IPv6 din headerele de nivel 3, împreună cu mesajul ICMP time exceeded.

Prima secvență de mesaje trimise de la traceroute vor avea un câmp TTL cu valoarea 1. Acest lucru face ca TTL să ajungă la 0 în pachetul IPv4 la primul router. Acest router răspunde apoi cu un mesaj ICMPv4. Traceroute are acum adresa primului hop.

Traceroute crește apoi câmpul TTL progresiv (2, 3, 4...) pentru fiecare secvență de mesaje. Acest lucru oferă **tracert** cu adresa fiecărui hop din cale. Câmpul TTL continuă să crească până când ajunge la destinație sau este incrementat până la un anumit maxim predefinit.

O dată ajuns la destinație, hostul răspunde fie cu un mesaj ICMP port unreachable, fie cu un mesaj ICMP echo reply message, în locul mesajului ICMP time exceeded.



### 8.18 Concluzii Capitolul 8

În acest capitol a fost prezentat modul în care întreprinderile mici și mijlocii sunt conectate la rețele în grupuri. Internet of Everything a fost de asemenea introdus în activitate de modelare de la început.



Adresele IP sunt ierarhice în rețea, subrețea și părțile de host. O adresă IP poate reprezenta o rețea completă, un host specific, sau adresa de broadcast a rețelei.

Înțelegerea notației binare este importantă atunci când se stabilește dacă două hosturi sunt în același rețea. Bițiile din partea de rețea a adresei IP trebuie să fie identici pentru toate dispozitivele din același rețea. Mască de rețea sau prefixul este folosit pentru a determina partea de rețea a unei adrese IP. Adresele IP pot fi atribuite fie static, fie dinamic. DHCP permite atribuirea dinamică a informațiilor de adresare cum ar fi adresa IP, masca de rețea, default gateway sau alte informații de configurație.

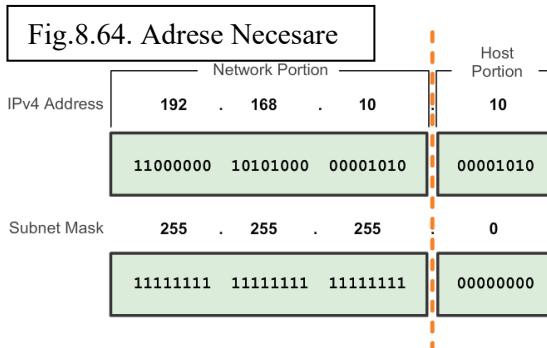
Hosturile IPv4 pot comunica într-unul dintre cele trei moduri diferite: unicast, broadcast și multicast. De asemenea, blocurile de adrese folosite în rețele care necesită acces limitat sau inexistent la Internet sunt numite adrese private. Blocurile de adrese IPv4 private sunt: 10.0.0.0/8, 172.16.0.0/12 și 192.168.0.0/16.

Epuizarea spațiului de adrese IPv4 este factorul motivant pentru trecerea la IPv6. Fiecare adresă IPv6 are 128 de biți, în comparație cu o adresă IPv4 de 32 de biți. IPv6 nu folosește notație de mască de rețea zecimală punctată. Lungimea prefixului este folosită pentru a indica partea de rețea a unei adrese IPv6 folosind următorul format: IPv6 address/prefix length.

Există trei tipuri de adrese IPv6: unicast, multicast și anycast. O adresă IPv6 de legătură locală permite unui dispozitiv să comunique cu alte dispozitive activate IPv6 din același legătură locală sau numai pe respectiva legătură (subrețea). Pachetele cu o adresă sursă sau destinație de legătură locală nu pot fi rutate în afara legăturii de unde este original pachetul. Adresele IPv6 de legătură locală sunt în spațiul FE80::/10.

ICMP este disponibil atât pentru IPv4, cât și pentru IPv6. ICMPv4 este protocolul de mesagerie pentru IPv4. ICMPv6 oferă aceleași servicii pentru IPv6, însă include funcționalități suplimentare.

După ce este implementată, o rețea IP trebuie să fie testată pentru a verifica conectivitatea sa și performanța funcțională a sa.



## CAPITOLUL 9. SUBNETAREA REȚEELOR IP

### Introducere

Proiectarea, implementarea și gestionarea unui plan de adresare IP eficient asigură faptul că rețelele pot funcționa eficient și corect. Acest lucru este valabil mai ales o dată cu creșterea numărului de conexiuni ale hosturilor la o rețea. Înțelegerea structurii ierarhice ale adresării IP și modul în care se poate modifica această ierarhie pentru îndeplinirea eficientă a cerințelor de rutare este o parte importantă a planificării unei scheme de adresare IP.

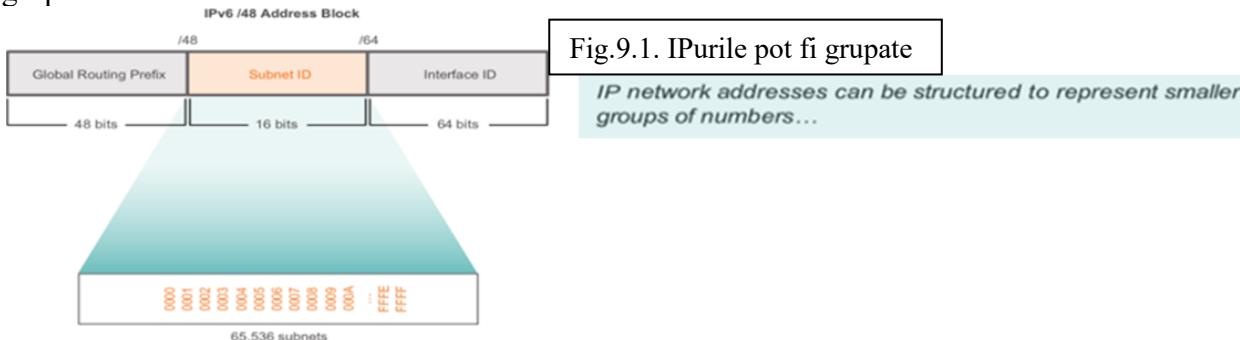
În adresarea originală IPv4, există două nivele de ierarhie: o rețea și un host. Aceste două nivele de adresare permit grupuri de bază de rețea care facilitează routarea de pachete la o rețea destinație. Un router transmite pachete în funcție de partea de rețea a unei adrese IP; o dată ce este localizată rețeaua, partea de host a adresei permite identificarea dispozitivului destinație.

Însă, o dată cu creșterea rețelelor, numeroase organizații adaugă sute, chiar mii de hosturi în rețelele lor, astfel încât cele două nivele de ierarhie devin insuficiente.

Împărțirea unei rețele adaugă un nou nivel la ierarhia rețelei, creând, în esență, trei niveluri: o rețea, o subrețea și un host. Introducerea unui nivel suplimentar în ierarhie creează subgrupuri suplimentare într-o rețea IP ceea ce facilitează livrarea mai rapidă de pachete și adaugă filtrarea prin ajustarea minimizării traficului "local".

Acest capitol examinează, în detaliu, crearea și atribuirea de adrese IP de rețea și subrețea prin utilizarea măștii de rețea.

În acest capitol, se va învăța cum să se grupeze dispozitivele în subrețele, sau grupuri mai mici de rețea, dintr-o rețea mai mare. Adresele IP de rețea pot fi structurate prin reprezentarea în grupuri mici de numere.



### 9.1 Subnetarea Rețeelor IPv4 – Segmentarea Rețelelor

În implementările de rețea la început, era comun organizațiilor să aibă toate computerele și alte dispozitive conectate la o singură rețea IP. Toate dispozitivele din organizație aveau atribuită o adresă de rețea IP cu un ID de rețea corespunzător. Acest tip de configurație este cunoscut ca o proiectare de rețea plată. Într-o rețea mică, cu un număr limitat de dispozitive, o proiectare de rețea plată nu reprezintă o problemă. Însă, odată cu creșterea rețelei, acest tip de configurație poate crea mari probleme.

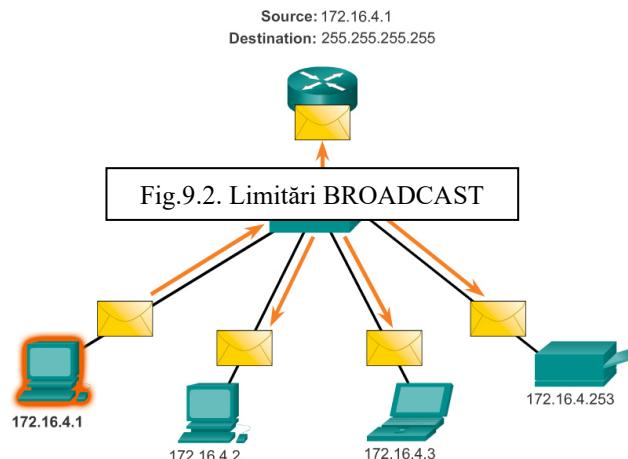
Într-un LAN Ethernet, dispozitivele folosesc adresarea broadcast pentru a localiza serviciile și dispozitivele necesare. Reamintim că solicitările de tip comunicație broadcast sunt trimise tuturor hosturilor dintr-o rețea IP. Dynamic Host Configuration Protocol (DHCP) este un exemplu de serviciu de rețea ce depinde de adresarea broadcast. Dispozitivele trimit comunicării de tip broadcast prin rețea pentru a localiza serverul DHCP. Într-o rețea mare, acest lucru ar putea crea o cantitate mare de trafic ce încetinește operațiile de rețea. În plus, deoarece o comunicație broadcast este adresată tuturor dispozitivelor, toate dispozitivele trebuie să accepte și să

proceseze traficul, având ca rezultat cerințe de procesare crescute. Dacă un dispozitiv trebuie să proceseze o cantitate mare de cerințe broadcast, ar putea încetini chiar și operațiile de dispozitiv. Din aceste motive, rețelele mari trebuie să fie segmentate în subrețele mai mici, localizate în grupuri mai mici de dispozitive și servicii.

Acest proces de segmentare a unei rețele, prin divizarea lor în spații de rețea mai mici, se numește **subnetare**, iar rețelele astfel obținute se numesc **subrețele**. Administratorii de rețea pot grupa dispozitivele și serviciile în subrețele ce sunt determinate de o locație geografică (cum ar fi campusul unei universități), de o unitate organizațională (cum ar fi departamentul IT), după tipuri de dispozitive (imprimante, servere, echipamente WAN) sau orice altă diviziune ce dă un sens rețelei. Subnetarea poate reduce traficul global de rețea și poate îmbunătăți performanța rețelei.

**Notă:** O subrețea este echivalentă cu o rețea și acești termeni pot fi folosiți în mod alternativ. Multe rețele reprezintă o subrețea a unui bloc mai mare de adrese.

Limited Broadcast

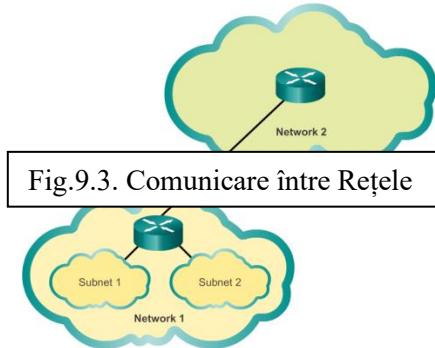


Un router este necesar pentru ca dispozitivele din rețele diferite să poată comunica. Dispozitivele dintr-o rețea folosesc interfață routerului atașată la LANul lor ca default gateway. Traficul destinat unui dispozitiv dintr-o rețea de la distanță va fi procesat de către router și trimis spre destinație. Pentru a determina dacă traficul este local sau la distanță, routerul folosește masca de rețea.

Într-un spațiu de rețea subnetat, acest lucru merge în același mod. Ca și în Fig. , subnetarea creează mai multe rețele logice dintr-un singur bloc de adrese sau adresa de rețea. Fiecare subrețea este tratată ca un spațiu de rețea separat. Dispozitivele din aceeași subrețea trebuie să aibă o adresă, mască de rețea și default gateway ce corespund subrețelei din care fac parte.

Traficul nu poate fi transmis între subrețele fără utilizarea unui router. Fiecare interfață a routerului trebuie să aibă o adresă de host IPv4 ce aparține rețelei sau subrețelei la care este conectată interfața routerului.

Communicating between Networks



### 9.1.1 Subnetarea IP este FUNDAMENTALĂ

Așa cum se observă și în Fig. , planificarea subrețelelor de rețea necesită examinarea nevoilor utilizării rețelei organizației și modul în care subrețelele vor fi structurate. Efectuarea unui studiu de cerință de rețea este punctul de plecare. Acest lucru înseamnă privirea întregii rețele și determinarea secțiunilor principale ale rețelei și modul în care vor fi segmentate. Planul de adresare include deciderea necesităților pentru fiecare subrețea în ceea ce privește dimensiunea, numărul de hosturi, cum adresele de host vor fi atribuite, ce hosturi vor necesita adrese IP statice și ce hosturi pot folosi DHCP pentru obținerea informațiilor de adresare a lor.

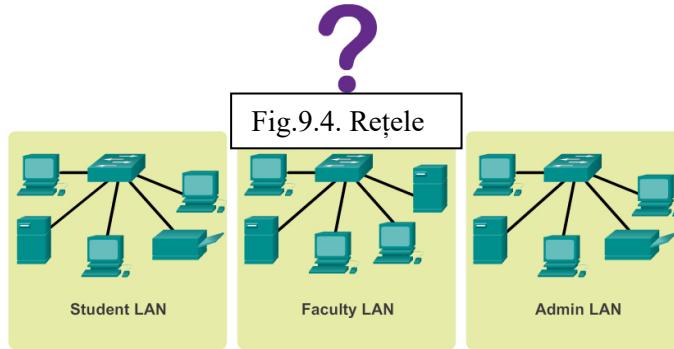
Dimensiunea subrețelei presupune planificarea numărului de hosturi ce necesită adrese IP de host în fiecare subrețea a rețelei private subnetată. De exemplu, într-un design de rețea de campus trebuie să luăm în considerare câte hosturi sunt necesare în LANul administrativ, câte în LAN facultății și câte în LANul de student. Într-o rețea de domiciliu, o considerare ar putea fi realizată prin numărul de hosturi din Main House LAN și numărul de hosturi din Home Office LAN.

Așa cum am discutat mai devreme, spațiul de adrese IP folosit într-un LAN este alegerea administratorului de rețea și necesită considerații atente pentru asigurarea faptului că sunt destule adrese de host disponibile pentru hosturile actuale cunoscute și pentru o extindere viitoare. Amintim faptul că spațiile de adrese private IP sunt:

- *10.0.0.0 cu o masca de rețea 255.0.0.0.*
- *172.16.0.0 cu o masca de rețea 255.240.0.0.*
- *192.168.0.0 cu o masca de rețea 255.255.0.0.*

Cunoascând cerințele de adresare IP vom determina spațiul sau spaile de adrese de host ce le vom implementa. Subnetarea spațiului de adresare privată IP selectat va oferi adresele de host necesare pentru îndeplinirea cerințelor rețelei.

Adresele pulice folosite pentru conectarea la Internet sunt de obicei alocate de la un furnizor de servicii - *ISP*. Deși se aplică aceleași principii pentru subnetare, nu este în general responsabilitatea administratorului de rețea al organizației.



Creem standarde de atribuire a adresei IP din fiecare spațiu de subrețea. De exemplu:

- *Imprimantele și serverele vor avea atribuite adrese IP statice.*
- *Utilizatorul va primi adresele IP de la servere DHCP ce folosesc subrețele /24.*
- *Routerele au atribuite primele adrese de host din range.*

Doi dintre factorii cei mai importanți ce conduc la determinarea blocului de adresare privată necesari sunt numărul de subrețele necesare și maximul numărului de hosturi necesare pe fiecare subrețea. Fiecare dintre aceste blocuri de adresă va permite alocarea de hosturi adecvată în funcție de dimensiunea dată a unei rețele și de hosturile cerute actual și în viitorul apropiat. Cerințele de spațiu de adrese IP vor determina spațiul sau rangeurile hosturilor.

În următoarele exemple vom vedea subnetarea în funcție de blocurile de adrese ce au masca de rețea 255.0.0.0, 255.255.0.0 și 255.255.255.0.



Fig.9.5. Planificare.

### 9.1.2 Subnetarea unei Rețele IPv4

Fiecare adresa de rețea are un spațiu valid de adrese de host. Toate dispozitivele din aceeași rețea vor avea o adresă de host IPv4 din rețea și o mască de rețea comună (sau prefix de rețea).

Prefixul și masca de rețea sunt moduri diferite de reprezentare a același lucru – partea de rețea a unei adrese.

Subrețele IPv4 sunt create prin folosirea unuia sau a mai multor biți ca biți de rețea. Acest lucru se realizează prin extinderea măștii și împrumutarea câtorva biți din partea de host a adresei pentru a crea biți de rețea suplimentari. Cu cât sunt împrumutați mai mulți biți, mai multe subrețele pot fi definite. Pentru fiecare bit împrumutat, numărul de subrețele disponibile se dublează. De exemplu, dacă 1 bit este împrumutat, pot fi create 2 subrețele. Dacă 2 biți sunt împrumutați, se crează 4, dacă 3 biți sunt împrumutați se crează 8 și aşa mai departe. Însă, cu fiecare bit împrumutat, mai puține adrese de host sunt disponibile pe subrețea.

Biții pot fi împrumutați numai din partea de host a adresei. Partea de rețea a adresei este alocată de către furnizorul de servicii și nu poate fi schimbată.

**Notă:** În exemplele din imagini, numai ultimul octet este arătat în binar datorită faptului că numai biții din partea de host sunt împrumutați.

Așa cum se poate observa în Fig.9.6.A, rețeaua 192.168.1.0/24 are 24 de biți în partea de rețea și 8 biți în partea de host, ceea ce indică masca de rețea 255.255.255.0 sau notația /24. Fără subnetare, rețeaua suportă o singură interfață LAN. Dacă este necesar un LAN suplimentar, rețeaua trebuie să fie subnetată.

Așa cum se poate observa în Fig.9.6.B, 1 bit este împrumutat de la cel mai semnificativ bit din partea de host, pentru a extinde partea de host la 25 de biți. Acest lucru crează 2 subrețele identificate prin folosirea unui 0 pe bitul împrumutat pentru prima rețea și un 1 pentru a doua rețea. Masca de rețea pentru ambele rețele folosește un 1 în poziția bitului împrumutat pentru a indica faptul că acest bit este acum parte din zona de rețea.

Așa cum se poate observa în Fig.9.6.C, atunci când convertim octetul binar în zecimal observăm faptul că prima adresă de subrețea este 192.168.1.0 și a doua adresă de subrețea este 192.168.1.128. Deoarece un bit a fost împrumutat, masca de rețea pentru fiecare subrețea este 255.255.255.128 sau /25.

Address	192	168	1	0000	0000
Mask	255	255	255	0000	0000
Fig.9.6.A - Rețea					
Network Portion					Host Portion

Original	192.	168.	1.	0	000	0000	Network: 192.168.1.0/24
Mask	255.	255.	255.	0	000	0000	Mask: 255.255.255.0

Fig.9.6.B- Reprezentare în zecimal

Borrowing 1 bit creates 2 subnets with the same mask.

Net 0	192.	168.	1.	0	000	0000	Network: 192.168.1.0/25
Mask	255.	255.	255.	1	000	0000	Mask: 255.255.255.128

Net 1	192.	168.	1.	1	000	0000	Network: 192.168.1.128/25
Mask	255.	255.	255.	1	000	0000	Mask: 255.255.255.128

Borrow 1 bit from the host portion of the address.

Original      192.      168.      1.      0      000      0000      1 Network  
 Mask          255.      255.      255.      0      000      0000

Fig.9.6.C.Împrumut de Biți

The borrowed bit value is 0 for the Net 0 address.

Net 0	192.	168.	1.	0	000	0000
-------	------	------	----	---	-----	------

The borrowed bit value is 1 for the Net 1 address.      2 Subnets

Net 1	192.	168.	1.	1	000	0000
-------	------	------	----	---	-----	------

The new subnets have the SAME subnet mask.

Mask	255.	255.	255.	1	000	0000
------	------	------	------	---	-----	------

În exemplul anterior, rețeaua 192.168.1.0/24 a fost subnetată pentru a crea două subrețele:

- 192.168.1.0/25.
- 192.168.1.128/25.

În Fig.9.7.A, observăm faptul că routerul R1 are două segmente de LAN atașate la interfețele sale GigabitEthernet. Subrețelele vor fi folosite pentru segmentele atașate la aceste interfețe. Pentru a servi drept gateway pentru dispozitivele din LAN, fiecare interfață a routerului trebuie să aibă atribuită o adresă IP din spațiul de adrese valide pentru subrețea calculată. Este o practică comună utilizarea primei sau a ultimei adrese disponibile dintr-un spațiu de adrese pentru adresa de interfață a routerului.

Prima subrețea, 192.168.1.0/25, este folosită pentru rețeaua atașată la GigabitEthernet 0/0 și a doua subrețea 192.168.1.128/25, este folosită pentru rețeaua atașată la GigabitEthernet 0/1. Pentru a atribui o adresă IP pe fiecare dintre cele două interfețe, este necesară determinarea spațiului de adrese valide pentru fiecare subrețea.

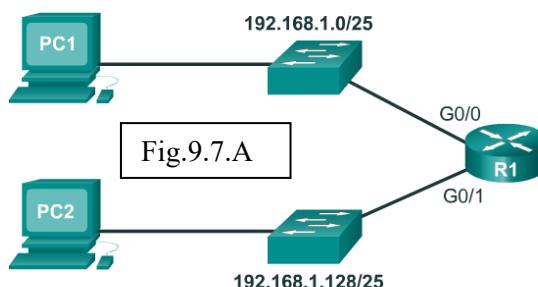
Următoarele sunt liniile directoare pentru fiecare dintre subrețele:

- **Adresa de rețea** – toți biții de 0 în partea de host a adresei.
- **Adresa primului host** – toți biții de 0 plus un bit de 1 cel mai din dreapta în partea de host a adresei.
- **Adresa ultimului host** – toți biții de 1 plus un bit de 1 cel mai din dreapta în partea de host a adresei.
- **Adresa de broadcast** – toți biții de 1 în partea de host a adresei.

Așa cum se poate vedea în Fig.9.7.B, prima adresă de host pentru rețeaua 192.168.1.0/25 este 192.168.1.1, iar ultima adresă de host este 192.168.1.126. Fig.9.7.C arată faptul că prima adresă de host pentru rețeaua 192.168.1.128/25 este 192.168.1.129, iar ultima adresă de host este 192.168.1.254.

Pentru a atribui prima adresă de host din fiecare subrețea interfeței routerului pentru respectiva subrețea, folosim comanda **ip address** în modul de configurație interfață, așa cum este arătat în Fig.9.7.D. De remarcat faptul că fiecare subrețea folosește masca de rețea 255.255.255.128 pentru a indica că partea de rețea a adresei este de 25 de biți.

O configurație de host pentru rețeaua 192.168.1.128/25 este prezentată în Fig.9.7.E. De remarcat faptul că adresa IP de gateway este adresa configurată pe interfața G0/1 a R1, 192.168.1.129, iar masca de rețea este 255.255.255.128.



Address Range for 192.168.1.128/25 Subnet

Network Address	192. 168. 1. 1 . 000 0000	= 192.168.1.128
First Host Address	192. 168. 1. 1 . 000 0001	= 192.168.1.129
Last Host Address	192. 168. 1. 1 . 111 1110	= 192.168.1.254
Broadcast Address	192. 168. 1. 1 . 111 1111	= 192.168.1.255

Fig.9.7.C

Network Address	192. 168. 1. 0 . 000 0000	= 192.168.1.0
First Host Address	192. 168. 1. 0 . 000 0001	= 192.168.1.1
Last Host Address	192. 168. 1. 0 . 111 1110	= 192.168.1.126
Broadcast Address	192. 168. 1. 0 . 111 1111	= 192.168.1.127

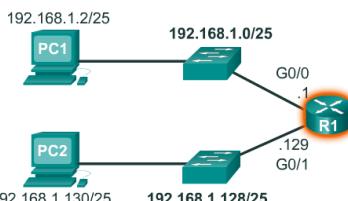
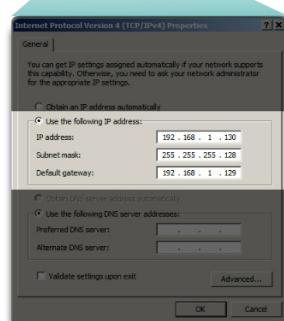
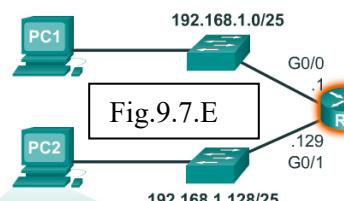


Fig.9.7.D

```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.128
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ip address 192.168.1.129 255.255.255.128
```



**Calcularea Subrețelelor** – Folosim următoarea formulă pentru a calcula numărul de subrețele :  $2^n$  (unde n este numărul de biți împrumutați).

Așa cum se observă și în Fig. 1, pentru exemplul 192.168.1.0/25, calculul arată astfel:  $2^1 = 2$  subrețele

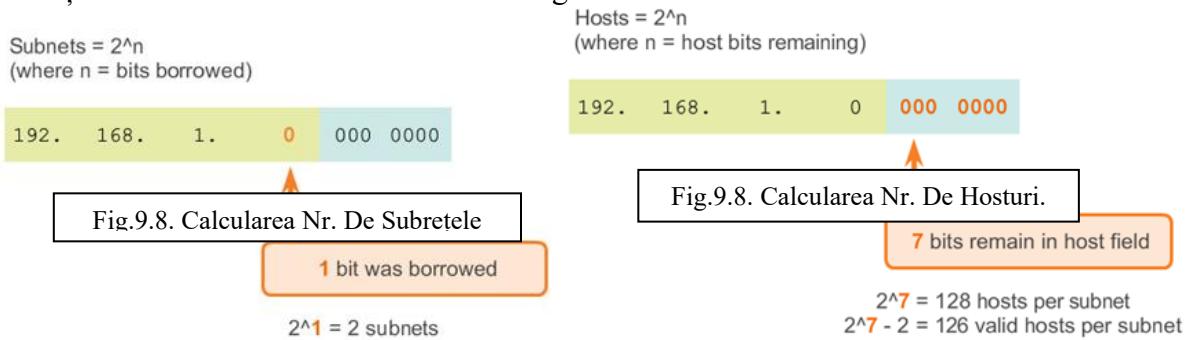
**Calcularea Numărului de Hosturi** – Folosim următoarea formulă pentru a calcula numărul de hosturi pe subrețea :  $2^n$  (unde n este numărul de biți rămași în câmpul de host).

Așa cum se poate observa și în Fig. 2, pentru exemplul 192.168.1.0/25, calculul arată astfel :  $2^7 = 128$  adrese

din care doar 126 pot fi atribuite echipamentelor (*adresa de rețea și broadcast nu pot fi asignate*).

Deoarece hosturile nu pot folosi adresa de rețea sau broadcast dintr-o subrețea, 2 dintre aceste adrese nu sunt asignabile. Acest lucru înseamnă că fiecare subrețea are 126 de adrese de host valide.

Deci în acest exemplu, împrumutând 1 bit de host rezultă crearea a 2 subrețele și fiecare subrețea are un număr de 126 de hosturi asignabile.



Să considerăm o rețea care necesită trei subrețele.

Utilizând același bloc de adrese 192.168.1.0/24, biții de host trebuie să fie împrumutați pentru a crea cel puțin 3 subrețele. Împrumutarea unui singur bit oferă 2 subrețele. Pentru a oferi mai multe rețele, mai mulți biți de host trebuie să fie împrumutați. Calculând numărul de subrețele create dacă împrumutăm 2 biți folosind formula **2<sup>n</sup> numărul de biți împrumutați** rezultă  $2^2 = 4$  subrețele.

Împrumutarea a 2 biți crează 4 subrețele, așa cum se observă și în Fig.9.10.A.

Reamintim faptul că masca de rețea trebuie să fie schimbată pentru a reflecta biții împrumutați. În acest exemplu, când 2 biți sunt împrumutați, masca se extinde cu 2 biți în ultimul octet. În zecimal, masca se reprezintă ca 255.255.255.192, deoarece ultimul octet în binar este 1100 0000.

Pentru a calcula numărul de hosturi, examinăm ultimul octet. După împrumutarea a doi biți pentru subrețea, există 6 biți rămași.

Aplicăm formula de calculare de host , așa cum este prezentată în Fig.9.10.B.

$$2^6 = 64$$

De reținut faptul că dacă toți biții sunt 0 din partea de host a adresei este adresa de rețea, iar toți biții de 1 în partea de host rezultă adresa de broadcast. Prin urmare, există numai 62 de adrese de host care sunt disponibile pentru fiecare subrețea.

Așa cum se poate observa în Fig.9.10.C, prima adresă de host pentru prima subrețea este 192.168.1.1 și ultima adresă de host este 192.168.1.62. Fig.9.10.D prezintă spațiile de adrese pentru subrețelele 0-2. Reamintim faptul că fiecare host trebuie să aibă o adresă IP validă din spațiul definit pentru respectivul segment de rețea. Rețeaua atribuită interfeței routerului va determina cărui segment îi aparține un host.

Fig.910.E prezintă un exemplu de config.re. În această config.re, prima rețea este atribuită interfeței GigabitEthernet 0/0, a doua interfeței GigabitEthernet 0/1, iar a treia interfeței Serial 0/0/0.

Folosind un plan de adresare general, prima adresă de host este atribuită interfeței routerului. Hosturile din fiecare subrețea vor utiliza adresa interfeței routerului ca adresă de default gateway.

- *PC1 (192.168.1.2/26) va utiliza 192.168.1.1 (adresa interfeței G0/0 a R1) ca adresă de default gateway.*
- *PC2 (192.168.1.66/26) va utiliza 192.168.1.65 (adresa interfeței G0/1 a R1) ca adresă de default gateway.*

**Notă:** Toate dispozitivele din aceeași subrețea vor avea o adresă de host IPv4 din spațiul de adrese de host și va folosi aceeași mască de rețea.

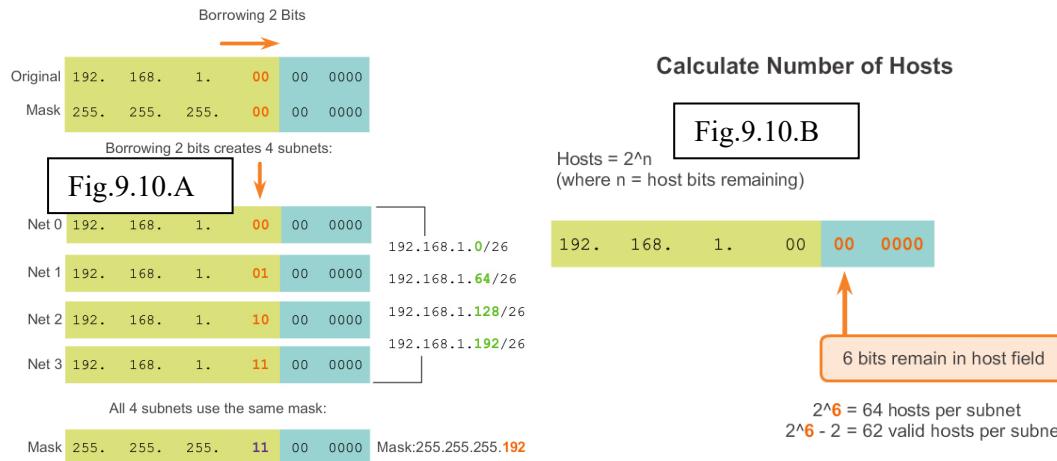
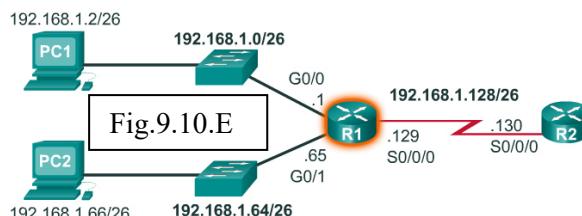


Fig.9.10.D Address Ranges Nets 0 - 2

Address Range for 192.168.1.0/26 Subnet	
Network Address	192.168.1.00000000 = 192.168.1.0
First Host Address	192.168.1.00000001 = 192.168.1.1
Last Host Address	192.168.1.01111110 = 192.168.1.62
Broadcast Address	192.168.1.01111111 = 192.168.1.63

Fig.9.10.C



```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.192
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ip address 192.168.1.65 255.255.255.192
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 192.168.1.129 255.255.255.192
```

Să considerăm o rețea ce necesită cinci subrețele, aşa cum se poate vedea în Fig.9.11.A.

Utilizând același bloc de adrese 192.168.1.0/24, biții de host trebuie să fie împrumutați pentru a crea cel puțin 5 subrețele. Împrumutarea a 3 biți oferă 4 subrețele, așa cum am observat în exemplul anterior. Pentru a oferi mai multe rețele, mai mulți biți trebuie să fie împrumutați. Calculăm numărul de subrețele create dacă 3 biți sunt împrumutați, folosind formala :

$$2^3 = 8 \text{ subrețele.}$$

Așa cum se poate vedea în Fig.9.11.B și Fig.9.11.C, împrumutarea a 3 biți crează 8 subrețele. Atunci când 3 biți sunt împrumutați, masca de rețea este extinsă cu 3 biți în ultimul octet (/27), rezultând masca de rețea 255.255.255.224. Toate dispozitivele din aceste subrețele vor utiliza masca de rețea 255.255.255.224 sau /27.

Pentru a calcula numărul de hosturi, examinăm ultimul octet. După împrumutarea a 3 biți pentru subrețea, rămân 5 biți de host.

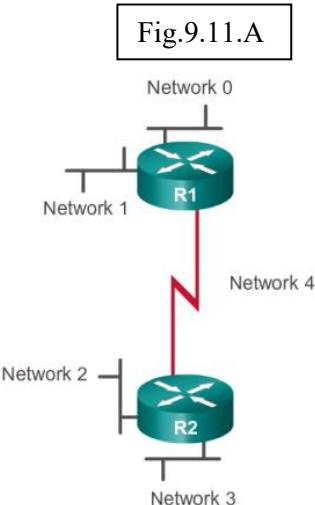
Aplicăm formula de calculare de host  $2^5 = 32$ , însă scădem 2 pentru adresa de rețea (ce are toți de 0 în partea de host) și adresa de broadcast (ce are toți de 1 în partea de host).

Subrețelele sunt atribuite segmentelor de rețea necesare pentru topologie, așa cum se poate observa în Fig.9.11.D.

Folosind un plan de adresare general, prima adresă de host este atribuită interfeței routerului. Hosturile din fiecare subrețea vor utiliza adresa interfeței routerului ca adresă de default gateway.

- PC1 (192.168.1.2/27) va folosi adresa 192.168.1.1 ca adresa să de default gateway.
- PC2 (192.168.1.34/27) va folosi adresa 192.168.1.33 ca adresa să de default gateway.
- PC3 (192.168.1.98/27) va folosi adresa 192.168.1.97 ca adresa să de default gateway.
- PC4 (192.168.1.130/27) va folosi adresa 192.168.1.129 ca adresa să de default gateway.

**5 Subnets Required**

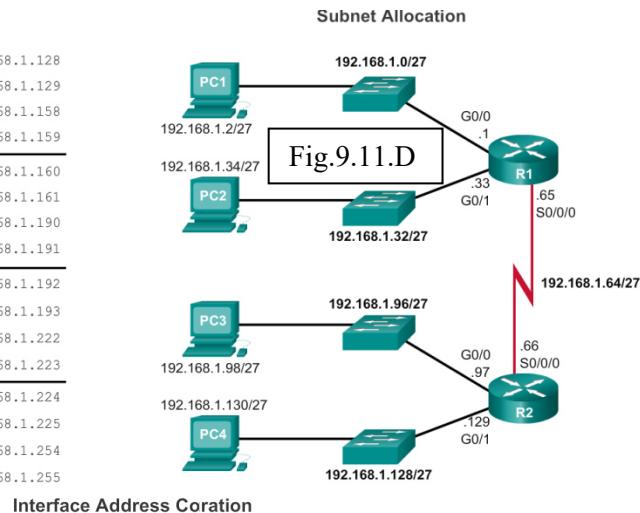


**Fig.9.11.B**

Net 0	Network	192.	168.	1.	000	0	0000	192.168.1.0
	First	192.	168.	1.	000	0	0001	192.168.1.1
	Last	192.	168.	1.	000	1	1110	192.168.1.30
	Broadcast	192.	168.	1.	000	1	1111	192.168.1.31
Net 1	Network	192.	168.	1.	001	0	0000	192.168.1.32
	First	192.	168.	1.	001	0	0001	192.168.1.33
	Last	192.	168.	1.	001	1	1110	192.168.1.62
	Broadcast	192.	168.	1.	001	1	1111	192.168.1.63
Net 2	Network	192.	168.	1.	010	0	0000	192.168.1.64
	First	192.	168.	1.	010	0	0001	192.168.1.65
	Last	192.	168.	1.	010	1	1110	192.168.1.94
	Broadcast	192.	168.	1.	010	1	1111	192.168.1.95
Net 3	Network	192.	168.	1.	011	0	0000	192.168.1.96
	First	192.	168.	1.	011	0	0001	192.168.1.97
	Last	192.	168.	1.	011	1	1110	192.168.1.126
	Broadcast	192.	168.	1.	011	1	1111	192.168.1.127

	Network	192.	168.	1.	100	0	0000	192.168.1.128
Net 4	First	192.	168.	1.	100	0	0001	192.168.1.129
	Last	192.	168.	1.	100	1	1110	192.168.1.158
	Broadcast	192.	168.	1.	100	1	1111	192.168.1.159
	Network	192.	168.	1.	101	0	0000	192.168.1.160
Net 5	First	192.	168.	1.	101	0	0001	192.168.1.161
	Last	192.	168.	1.	101	1	1110	192.168.1.190
	Broadcast	192.	168.	1.	101	1	1111	192.168.1.191
	Network	192.	168.	1.	110	0	0000	192.168.1.192
Net 6	First	192.	168.	1.	110	0	0001	192.168.1.193
	Last	192.	168.	1.	110	1	1110	192.168.1.222
	Broadcast	192.	168.	1.	110	1	1111	192.168.1.223
	Network	192.	168.	1.	111	0	0000	192.168.1.224
Net 7	First	192.	168.	1.	111	0	0001	192.168.1.225
	Last	192.	168.	1.	111	1	1110	192.168.1.254
	Broadcast	192.	168.	1.	111	1	1111	192.168.1.255

Fig.9.11.C



Interface Address Correlation

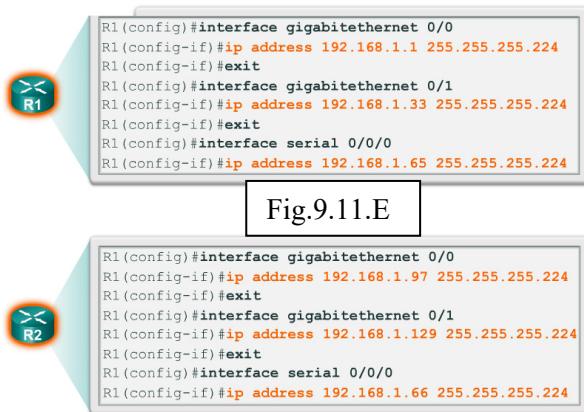


Fig.9.11.E

În exemplele anterioare, am considerat o internetwork ce necesită 3 subrețele și una ce necesită 5 subrețele. Pentru a realiza acest obiectiv de creare a patru subrețele am împrumutat doi biți din 8 biți de host disponibili dintr-o adresă IP ce are masca de rețea 255.255.255.0 sau /24. Masca de rețea rezultată a fost 255.255.255.192, iar un număr total de 4 subrețele posibile au fost create. Aplicând formula de calculare a hosturilor ( $2^6-2$ ) am determinat că fiecare dintre acele 4 subrețele pot avea 62 de adrese de host atribuite nodurilor.

Pentru a obține 5 subrețele, am împrumutat 3 biți din 8 biți de host disponibili dintr-o adresă IP ce are masca de rețea 255.255.255.0 sau /24. Prin împrumutarea celor 3 biți din partea de host a adresei, rămânem cu 5 biți. Masca de rețea rezultată a fost 255.255.255.224 cu un număr total de 8 subrețele posibile și 30 de adrese de host pe subrețea.

Considerăm organizații mari sau campusuri cu o rețea ce necesită 100 de subrețele. La fel ca și în exemplele anterioare, pentru a realiza acest lucru, trebuie să împrumutăm biți din partea de host a adresei IP a internetwork existente. Pentru a calcula numărul de subrețele, trebuie să ne uităm la numărul de biți de host disponibili pentru a-i folosi în formula de calculare  $2^x$  numărul de biți împrumutați - 2. Folosind adresa IP a ultimului exemplu, 192.168.10.0/24, avem 8 biți de host. Pentru a crea 100 de subrețele trebuie să împrumutăm 7 biți.

Calculăm numărul de subrețele dacă 7 biți sunt împrumutați :  $2^7=128$  subrețele.

Însă, împrumutând 7 biți va rămâne un sigur bit de host și dacă vom aplica formula de calcul de host, va rezulta existența niciunui host în respectivele subrețele. Calculăm numărul de hosturi în cazul în care rămâne un singur bit  $2^1=2$ , apoi scădem 2 pentru adresa de rețea și cea de broadcast; rezultă 0 hosturi ( $2^1-2=0$ ).

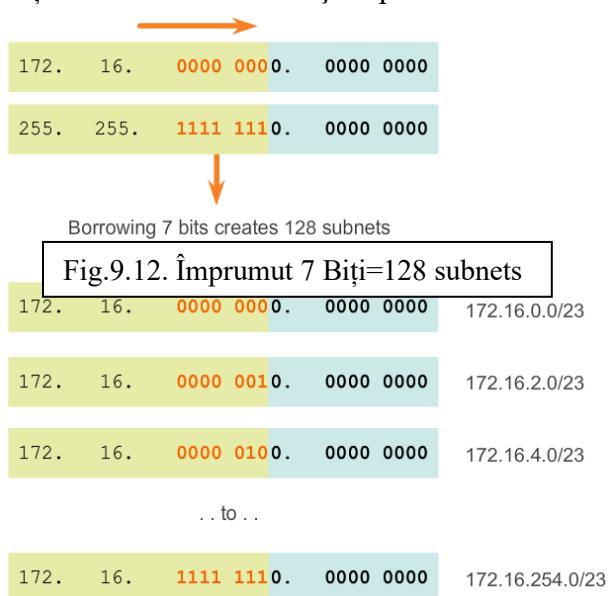
Într-o situație ce necesită un număr mai mare de subrețele, este necesară o rețea ce are mai mulți biți de host de împrumutat, cum ar fi o adresă IP cu o masca de rețea implicită /16 sau 255.255.0.0.

Adresele ce au un spațiu de adrese 128 - 191 în primul octet au o mască implicită 255.255.0.0 sau /16. Adresele din acest spațiu de adrese au 16 biți în partea de rețea și 16 în partea de host. Acești 16 biți sunt biți disponibili pentru împrumutul necesar creării de subrețele.

Utilizând o nouă adresă IP din blocul de adrese 172.16.0.0/16, trebuie să împrumutăm biți de host pentru a crea cel puțin 100 de subrețele. Începând de la stânga la dreapta, cu primul bit de host disponibil, vom împrumuta câte un bit o dată până când ajungem la numărul de biți necesari pentru a crea 100 de subrețele. Împrumutând un bit, vom crea două subrețele, împrumutând 2 biți vom crea 4 subrețele, 3 biți crează 8 subrețele și aşa mai departe. Calculând numărul de subrețele create dacă împrumutăm 7 biți cu ajutorul formulei  $2^7 = 128$  de subrețele.

Împrumutând 7 biți se crează 128 de subrețele, aşa cum este evidențiat în Fig.9.12.

Reamintim faptul că masca de rețea trebuie să fie schimbată pentru a reflecta biții împrumutați. În acest exemplu, când 7 biți sunt împrumutați, masca se extinde cu 7 biți în al treilea octet. În decimal, masca este reprezentată ca fiind 255.255.254.0 sau /23, deoarece al treilea octet este 11111110 în binar și al patrulea octet este 00000000 în binar. Subnetarea se va face în al treilea octet, cu biții de host în al treilea și al patrulea octet.



Pentru a calcula numărul de hosturi, examinăm al treilea și al patrulea octet. După împrumutarea de 7 biți pentru subnet, rămân 3 biți în al treilea octet și 8 biți în al patrulea octet.

Aplicăm formula de calculare de host, aşa cum este efectuat și în Fig.9.13.

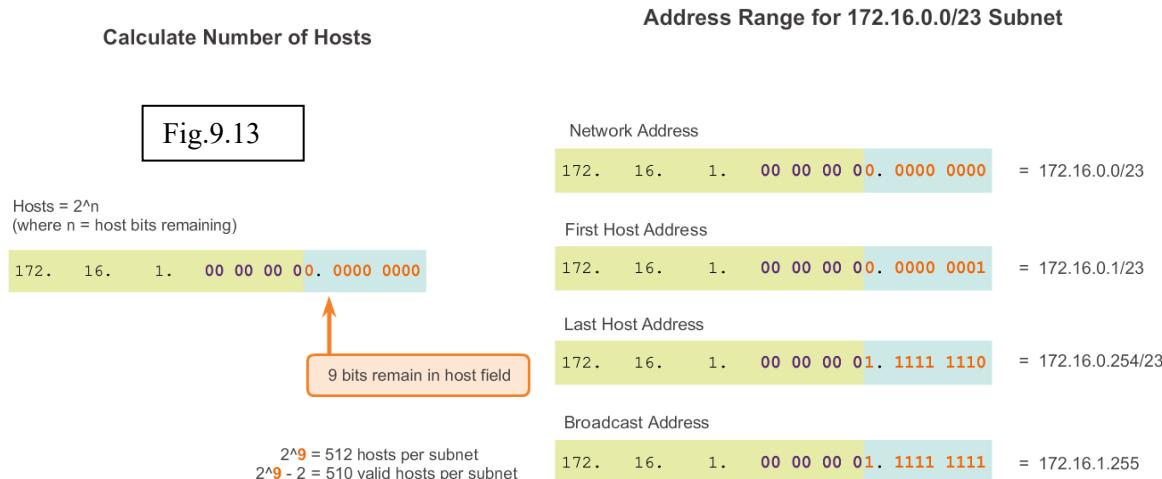
$$2^9 = 512$$

De reținut faptul că atunci când sunt toți biții de 0 în partea de host a adresei este adresa de rețea, iar când sunt toți biții de 1 în partea de host a adresei este adresa de broadcast. Prin urmare, există numai 510 adrese de host disponibile pentru fiecare subrețea.

Așa cum se poate observa și în Fig. 2, prima adresă de host pentru prima subrețea este 172.16.0.1, iar ultima adresă de host este 172.16.1.254. Amintim faptul că fiecare host trebuie să aibă o adresă IP validă din spațiul definit pentru fiecare segment. Subrețea atribuită interfeței routerului va determina cărui segment aparține un host.

**Reminder:**

Biți pot fi împrumutați numai din partea de host a adresei. Partea de rețea a adresei este alocată de către furnizorul de servicii și nu poate fi schimbată. Deci, organizațiile ce necesită un număr mare de subrețele trebuie să comunice nevoia lor la ISP pentru ca ISP-ul să aloce o adresă IP cu o mască implicită cu destui biți astfel încât să se poată crea subrețelele dorite.



Există unele organizații, cum ar fi furnizorii mici de servicii, ce ar putea avea nevoie de mai multe de 100 de subrețele. De exemplu, o organizație ce necesită 1000 de subrețele. Pentru a crea subrețele trebuie să împrumutăm biți din partea de host a adresei IP din internetwork existentă. Ca și înainte, pentru a calcula numărul de subrețele este necesar să ne uităm la numărul de biți de host disponibili. O astfel de situație necesită ca adresa IP atribuită de către ISP să aibă destui biți disponibili pentru calculul a 1000 de subrețele. Adresele IP ce au spațiul între 1-126 în primul octet au masca implicită 255.0.0.0 sau /8. Acest lucru înseamnă că există 8 biți în partea de rețea și 24 de biți în partea de host disponibili pentru împrumut.

Folosind blocul de adrese 10.0.0.0/8, biți de host trebuie să fie împrumutați pentru a crea cel puțin 1000 de subrețele. Începând de la stânga la dreapta, cu primul bit de host disponibil, vom împrumuta un bit o dată până când ajungem la numărul de biți necesari pentru a crea 1000 de subrețele. Prin calcularea numărului de subrețele rezultate în cazul în care 10 biți sunt împrumutați, rezultă  $2^{10} = 1024$  subrețele.

Împrumutul a 10 biți rezultă 1024 subrețele, așa cum se poate vedea în Fig.9.14.

Reamintim faptul că masca de rețea trebuie să se schimbe astfel încât să reflecte biții împrumutați. În acest exemplu, când sunt împrumutați 10 biți, masca de rețea se extinde cu 10 biți în al treilea octet. În decimal, masca este reprezentată ca fiind 255.255.192.0 sau /18, deoarece al treilea octet din masca de rețea este 11000000 în binar și al patrulea octet este 00000000 în binar. Subnetarea va fi efectuată în al treilea octet, însă să nu uităm de biții de host din al treilea și al patrulea octet.

Pentru a calcula numărul de hosturi, examinăm al treilea și al patrulea octet. După împrumutul a 10 biți pentru subnet, rămân 6 biți în al treilea octet și 8 biți în al patrulea octet, rezultând 14 biți rămași.

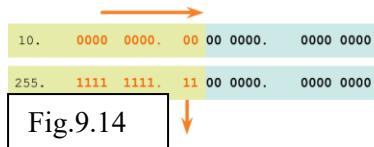
Aplicăm formula de calcul de host, așa cum este efectuată și în Fig.9.15.

$$2^{14} - 2 = 16382$$

Prima adresă de host pentru prima subrețea este 10.0.0.1 și ultima adresă de host este 10.0.63.254. De reținut faptul că fiecare host trebuie să aibă o adresă IP validă din spațiul definit pentru fiecare segment. Subrețea atribuită interfeței routeurului va determina cărui segment aparține un host.

**Notă:** Toate dispozitivele din aceeași subrețea vor avea o adresă de host IPv4 din spațiul de adrese de host și vor utiliza aceeași mască de rețea.

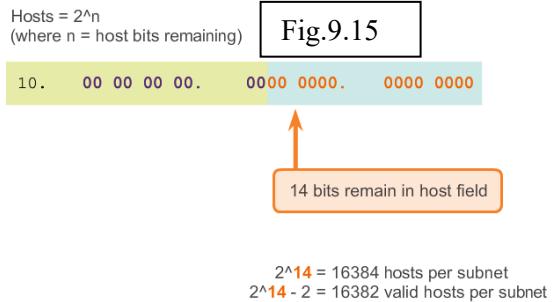
Calculate Number of Hosts



Borrowing 10 bits creates 1024 subnets

10. 0000 0000. 00 00 0000. 0000 0000	10.0.0.0/18
10. 0000 0000. 01 00 0000. 0000 0000	10.0.64.0/18
10. 0000 0000. 11 00 0000. 0000 0000	10.0.192.0/18
10. 0000 0001. 00 00 0000. 0000 0000	10.1.0.0/18
.. to ..	
10. 1111 1111. 10 00 0000. 0000 0000	10.255.128.0/18

Address Range for 10.0.0.0/18 Subnet



Network Address

Fig.9.16

10. 00 00 00 00. 0000 0000. 0000 0000 = 10.0.0.0/18

First Host Address

10. 00 00 00 00. 0000 0000. 0000 0001 = 10.0.0.1/18

Last Host Address

10. 00 00 00 00. 0011 1111. 1111 1110 = 10.0.63.254/18

Broadcast Address

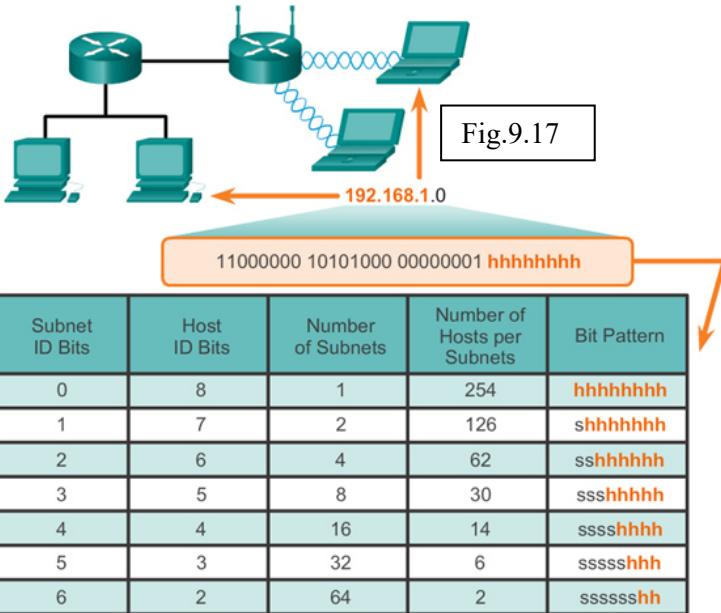
10. 00 00 00 00. 0011 1111. 1111 1111 = 10.0.63.255/18

### 9.1.3 Determinarea Măștii de Rețea

Decizia cu privire la câți biți de host să împrumutăm pentru a crea subrețele este o decizie importantă de planificare. Există două considerații atunci când planificăm subrețelele: numărul de adrese de host necesare pentru fiecare rețea și numărul de subrețele individuale necesare. Fig. arată posibilitatiile de subnetare pentru rețeaua 192.168.1.0. Selectia numărului de biți pentru subnetID afectează atât numărul de posibile subrețele cât și numărul de adrese de host din fiecare subrețea.

De remarcat faptul că este o relație inversă între numărul de subrețele și numărul de hosturi. Cu cât sunt împrumutați mai mulți biți pentru crearea de subrețele, cu atât rămân mai puțini biți de host disponibili; prin urmare, mai puține hosturi pe subrețea. Dacă sunt necesare mai multe adrese, sunt necesari mai mulți biți, având ca rezultat mai puține subrețele.

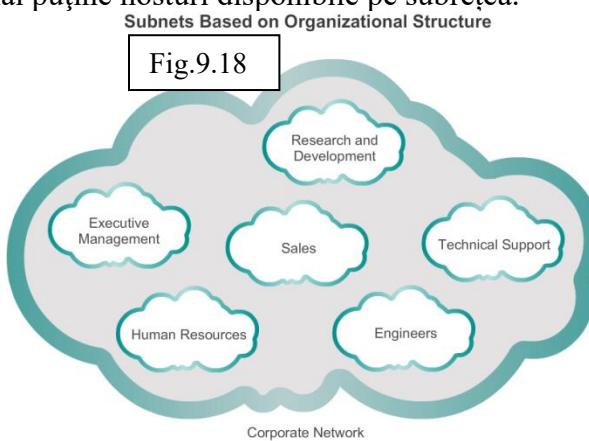
**Numărul de Hosturi** – Atunci când împrumutăm biți pentru a crea mai multe subrețele, pastrăm destui biți de host pentru subrețeaua mai mare. Numărul de adrese de host necesare în subrețeaua mai mare va determina câți biți trebuie să rămână în partea de host. Formula  $2^n$  (unde n este numărul de biți de host rămasi) este folosită pentru a calcula câte adrese vor fi disponibile pe fiecare subrețea. Reamintim faptul că două dintre aceste adrese nu pot fi folosite, deci numărul utilizabil de adrese se calculează ca  $2^{n-2}$ .



Uneori un anumit număr de subrețele este necesar cu accent mai redus pe numărul de adrese de host pe subrețea. Acest lucru ar putea fi cazul în care o organizație alege să separe traficul de rețea în funcție de structura internă sau de departament. De exemplu, o organizație ar putea alege să pună toate dispozitivele de host utilizate de către angajați din departamentul de inginerie într-o rețea și toate dispozitivele de host folosite de management într-o rețea separată. În acest caz, numărul de subrețele este important în determinarea numărului de biți împrumutați.

Reamintim faptul că numărul de subrețele create atunci când biții sunt împrumutați poate fi calculat cu ajutorul formulei  $2^n$  (unde n este numărul de biți împrumutați). Nu există nevoie de scădere a vreunei rețele rezultante deoarece sunt toate utilizabile.

Cheia este echilibrarea numărului de subrețele necesare și numărului de hosturi necesare pentru cea mai mare subrețea. Mai mulți biți împrumutați pentru crearea de subrețele suplimentare înseamnă mai puține hosturi disponibile pe subrețea.

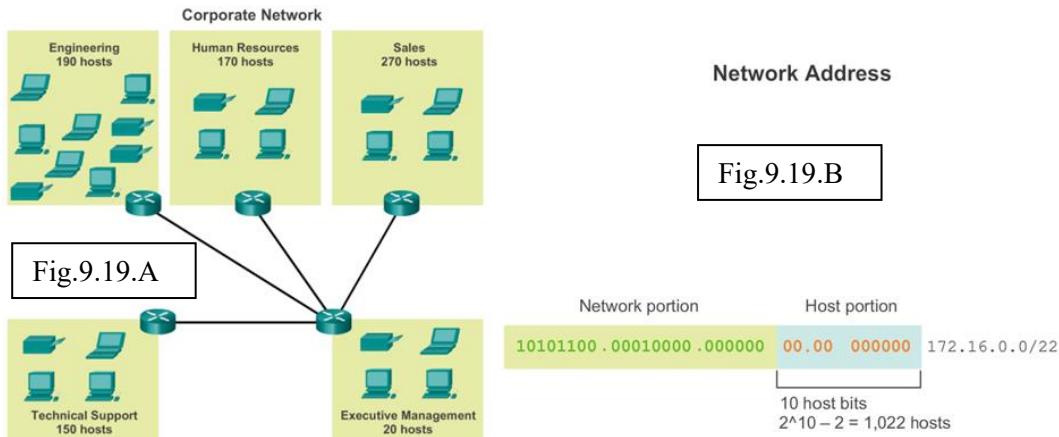


Orice rețea dintr-o organizație este proiectată pentru a suporta un număr finit de hosturi. Subnetarea de bază prevede suficiente subrețele pentru a adapta rețelele și pentru a oferi suficiente adrese de host pe subrețea.

Unele rețele, cum ar fi legăturile point-to-point WAN, necesită numai două hosturi. Alte rețele, cum ar fi un LAN dintr-o clădire mare sau departamentar ar putea necesita să susțină sute de hosturi. Administratorii de rețea trebuie să elaboreze schema de adresare internetwork pentru a fixa numărul de hosturi maxim pentru fiecare rețea.

**Determinarea Numărului Total de Hosturi** – Mai întâi, luăm în considerare numărul total de hosturi necesare în întreaga internetwork corporativă. Un bloc de adrese destul de mare trebuie să fie folosit pentru a cuprinde toate dispozitivele din toate rețelele corporate. Aceste dispozitive includ dispozitivele de utilizator, servere, dispozitive intermediare și interfețele de router.

Considerăm exemplul unei internetwork corporativ ce trebuie să cuprindă un număr total de 800 de hosturi în cinci locații - Fig.9.19.A. În acest exemplu, furnizorul de servicii a alocat următoarea adresă de rețea 172.16.0.0/22 (10 biți de host). Așa cum se poate observa și în Fig.9.19.B, aceasta va oferi 1.022 adrese de host ce cuprind mai mult decât nevoile de adresare pentru această internetwork.



**Determinarea Numărului și Dimensiunii Rețelelor** – Apoi, considerăm numărul de subrețele necesare și numărul de adrese de host necesare pentru fiecare subrețea. Având în vedere topologia de rețea ce cuprinde 5 segmente de LAN și 4 conexiuni internetwork între routere, sunt necesare 9 subrețele. Cea mai mare subrețea necesită 40 de hosturi. Atunci când proiectăm o schemă de adresare, trebuie să anticipăm creșterea în ambele domenii - numărul de subrețele și numărul de hosturi pe subrețea.

Adresa de rețea 172.16.0.0/22 are 10 biți de host. Deoarece cea mai mare subrețea necesită 40 de hosturi, un minim de 6 biți de host trebuie să fie împrumutați. Acest lucru este determinat de formula  $2^6 - 2 = 62$  hosturi. Cei 4 biți rămași pot fi folosiți pentru a aloca subrețele. Folosind formula de determinare a subrețelelor, rezultă 16 subrețele  $2^4 = 16$ . Deoarece internetwork din exemplu necesită 9 subrețele, îndeplinim cerințele și permite o anumită creștere ulterioară.

Atunci când sunt împrumutăți 4 biți nouă mască de rețea este 255.255.255.192 sau /26.

Așa cum se poate observa și în Fig.9.20.A, folosind o lungime de prefix /26, pot fi determinate 16 adrese de subrețea. Numai partea de subnet a adresei este incrementată. Cei 22 de biți originali ai adresei de rețea nu se pot schimba și partea de host va conține numai biți de 0.

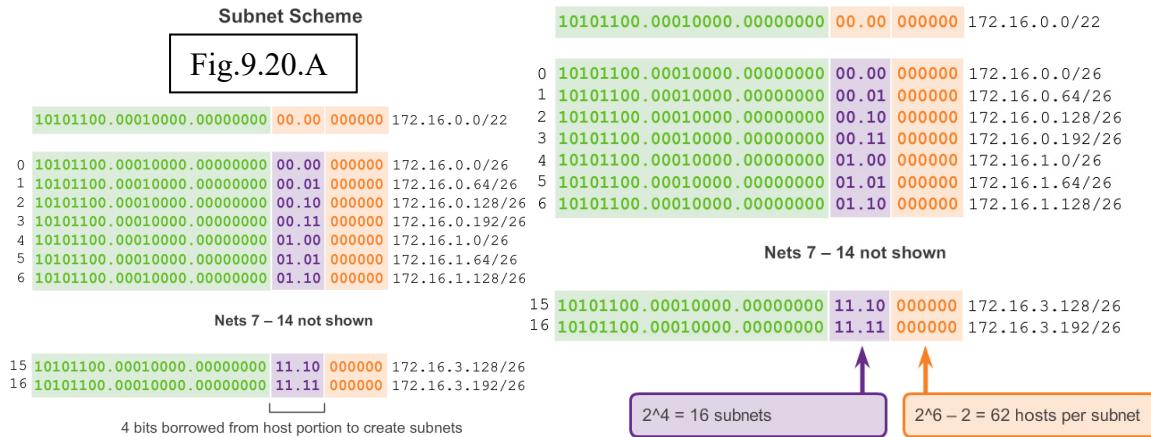
**Notă:** De remarcat faptul că deoarece partea de subrețea se află în octetul trei și patru, una dintre aceste două valori va oscila în adresele de subrețea.

Așa cum se poate observa și în Fig.9.20.B, rețeaua originală 172.16.0.0/22 era o singură rețea cu 10 biți de host ce oferea 1.022 adrese utilizabile pentru atribuirea hosturilor. Prin împrumutarea a 4 biți, 16 subrețele (de la 0000 la 1111) pot fi create. Fiecare subrețea are 6 biți de host sau 64 de adrese de host utilizabile pe subrețea.

Așa cum se poate observa și în Fig.9.20.C, subrețelele pot fi atribuite segmentelor de LAN și conexiunilor router-to-router.

## Subnets and Addresses

Fig.9.20.B



$= 16$  subnets       $2^6 - 2 = 62$  hosts per subnet

$$2^4 = 16 \text{ subnets}$$

$$2^6 - 2 = 62 \text{ hosts per subnet}$$

Fig.9.20.C

Fig.9.20.C

#### **9.1.4 Beneficiile Maștilor cu Lungime Variabilă (Variable Length Subnet Masking-VLSM)**

Folosind subnetarea tradițională, același număr de adrese este alocat pentru fiecare subrețea. Dacă toate subrețele au aceeași cerință cu privire la numărul de hosturi, aceste blocuri de adrese de dimensiune fixă sunt eficiente. Însă, adesea nu se întâmplă aşa.

De exemplu, topologia din Fig.9.21.A necesită șapte subrețele, una pentru fiecare dintre cele patru LANuri și una pentru fiecare dintre cele trei conexiuni WAN dintre routere. Folosind subnetarea tradițională cu adresa dată 192.168.20.0/24, 3 biți pot fi împrumutați din partea de host a ultimului octet pentru a îndeplini cerința de subnetare pentru șapte subrețele. Așa cum se poate observa în Fig.9.21.B, împrumutarea a trei biți crează 8 subrețele cu 5 biți de host și 30 de hosturi utilizabile pe subrețea. Această schemă crează subrețele necesare și îndeplinește cerința de host pentru cel mai mare LAN.

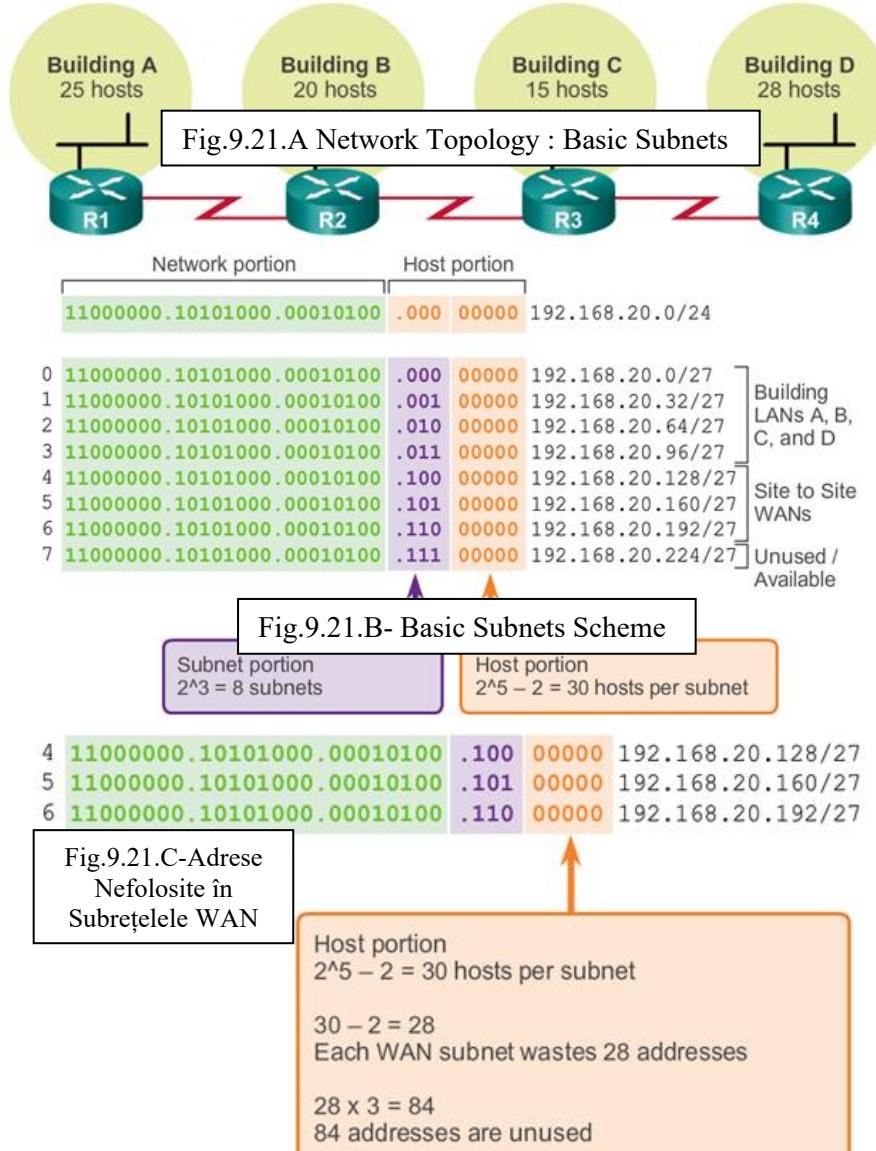
Deși această subnetare tradițională îndeplinește cerințele celor mai mari LAN și împarte spațiul de adrese într-un număr adecvat de subrețele, are ca rezultat o mare pierdere de adrese utilizabile.

De exemplu, numai două adrese sunt necesare pentru fiecare subrețea dintre cele trei legături LAN. Deoarece fiecare subrețea are 30 de adrese utilizabile rămân 28 de adrese în fiecare dintre aceste subrețele. Așa cum se poate observa în Fig.9.21.C, rezultă 84 de adrese nefolosite ( $28 \times 3$ ).

Mai mult, acest lucru limitează creșterea viitoare prin reducerea numărului total de subrețele disponibile. Această utilizare ineficientă a adreselor este caracteristică subnetării tradiționale ale rețelelor de tip classful.

Aplicarea unei scheme de subnetare tradițională la acest scenariu nu este foarte eficientă și este risipitoare. De fapt, acest exemplu este un bun model de exemplificare a modului în care subnetarea poate fi folosită pentru maximizarea utilizării de adrese.

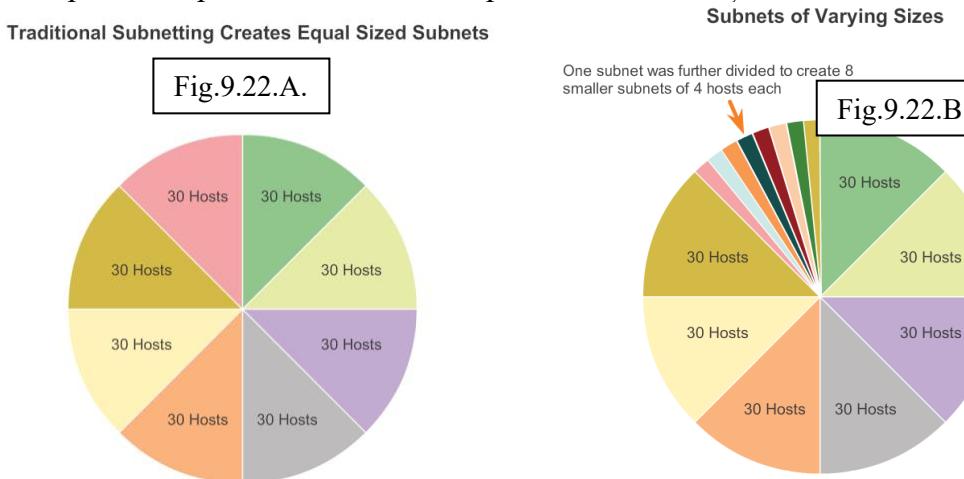
Subnetarea unui rețele, sau folosirea Variable Length Subnet Mask (VLSM), a fost concepută pentru a evita pierderea de adrese.



În toate exemplele anterioare de subnetare, remarcăm faptul că aceeași mască de subrețea a fost aplicată tuturor subrețelelor. Acest lucru înseamnă că fiecare subrețea are același număr de adrese de host disponibile.

Așa cum se ilustrează în Fig.9.22.A, subnetarea tradițională crează subrețele de dimensiuni egale. Fiecare subrețea dintr-o schemă tradițională folosește aceeași mască de rețea. Așa cum este evidențiat în Fig.9.22.B, VLSM permite ca un spațiu de rețea să fie divizat în părți inegale. Cu VLSM, masca de rețea variază în funcție de câți biți trebuie să fie împrumutați pentru o anumită subrețea, astfel rezultând partea "variabilă" a VLSM.

Subnetarea cu VLSM este similară cu cea tradițională în care biții sunt împrumutați pentru a crea subrețele. Formulele de calcul a numărului de hosturi pe subrețea și a numărului de subrețele create se aplică și aici. Diferența este că această subnetare nu este o activitate "single pass". Cu VLSM, rețeaua este subnetată mai întâi și apoi subrețelele sunt subnetate din nou. Acest proces poate fi repetat de mai multe ori pentru a crea subrețele de dimensiuni diferite.



Pentru a înțelege mai bine procesul VLSM, să ne întoarcem la exemplul anterior.

În exemplul anterior, prezentat în Fig.9.23.A, rețeaua 192.168.20.0/24 a fost subnetată în 8 subrețele de dimensiuni egale; șapte din 8 au fost alocate. Patru subrețele au fost folosite pentru LANuri și trei pentru conexiunile WAN dintre routere. Reamintim faptul că risipa spațiului de adrese a fost în subrețelele folosite pentru conexiunile WAN deoarece acele subrețele necesitau numai două adrese utilizabile: una pentru fiecare interfață a routerului. Pentru a evita această risipă, VLSM poate fi folosit pentru a crea subrețele mai mici pentru conexiunile WAN.

Pentru a crea subrețele mai mici pentru conexiunile WAN, una dintre subrețele va fi divizată. În Fig.9.23.B, ultima subrețea 192.168.20.224/27, va fi subnetată mai departe.

Reamintim că atunci când numărul de adrese de host necesare este cunoscut, formula  $2^n - 2$  (unde n este egal cu numărul de biți de host rămas) poate fi folosită. Pentru a oferi două adrese utilizabile, trebuie să rămână 2 biți de host în partea de host.

$$2^2 - 2 = 2$$

Deoarece există 5 biți de host în spațiul de adrese 192.168.20.224/27, 3 biți pot fi împrumutați, rămânând 2 biți în partea de host.

Calculele din acest moment sunt exact la fel cu cele folosite pentru subnetarea tradițională. Biții sunt împrumutați și spațiile de adrese pentru subrețele sunt determinate.

Așa cum se evidențiază în Fig.9.23.B, schema de subnetare VLSM reduce numărul de adrese pe subrețea la o dimensiune adecvată pentru WANuri. Subnetizarea subrețelei 7 pentru WANuri permite ca subrețelele 4, 5, 6 să fie disponibile pentru rețele viitoare și multe alte subrețele disponibile pentru WANuri.

	11000000.10101000.00010100.00000000	192.168.20.0/24	
0	11000000.10101000.00010100.00000000	192.168.20.0/27	
1	11000000.10101000.00010100.00100000	192.168.20.32/27	LANs A, B, C, D
2	11000000.10101000.00010100.01000000	192.168.20.64/27	
3	11000000.10101000.00010100.01100000	192.168.20.96/27	
4	11000000.10101000.00010100.10000000	192.168.20.128/27	Unused / Available
5	11000000.10101000.00010100.10100000	192.168.20.160/27	
6	11000000.10101000.00010100.11000000	192.168.20.192/27	
7	11000000.10101000.00010100.11100000	192.168.20.224/27	

Fig.9.23.A

	11000000.10101000.00010100.00000000	192.168.20.0/24	
0	11000000.10101000.00010100.00000000	192.168.20.0/27	
1	11000000.10101000.00010100.00100000	192.168.20.32/27	LANs A, B, C, D
2	11000000.10101000.00010100.01000000	192.168.20.64/27	
3	11000000.10101000.00010100.01100000	192.168.20.96/27	
4	11000000.10101000.00010100.10000000	192.168.20.128/27	
5	11000000.10101000.00010100.10100000	192.168.20.160/27	Unused / Available
6	11000000.10101000.00010100.11000000	192.168.20.192/27	
7	11000000.10101000.00010100.11100000	192.168.20.224/27	

Fig.9.23.B Schema de Subnetare VLSM

Folosind VLSM, segmentele LAN și WAN pot fi adresate fără nici-o “risipă” inutilă.

Hosturile din fiecare dintre LANuri vor fi atribuite cu o adresă de host validă din spațiul de adresare pentru respectiva subrețea și vor avea masca /27. Fiecare dintre cele patru routere vor avea o interfață LAN cu o subrețea /27 și una sau mai multe interfețe seriale cu o subrețea /30.

Folosind o schema de adresare generală, prima adresă de host IPv4 a fiecărei subrețele este atribuită interfeței LAN a routerului. Interfețele WAN ale routerelor sunt asignate cu adrese IP și masca /30.

Fig.9.24 și respective Fig.9.25 arată configurația de interfață pentru fiecare dintre routere.

Hosturile din fiecare subrețele vor avea o adresă de host IPv4 din spațiul de adrese de host pentru respectiva subrețea și o mască de rețea adecvată. Hosturile vor folosi adresa interfeței routerului LAN atașat ca adresă de default gateway.

- Hosturile din clădirea A(192.168.20.0/27) vor folosi adresa 192.168.20.1 ca adresă de default gateway.
- Hosturile din clădirea B(192.168.20.32/27) vor folosi adresa 192.168.20.33 ca adresă de default gateway.
- Hosturile din clădirea C(192.168.20.64/27) vor folosi adresa 192.168.20.65 ca adresă de default gateway.
- Hosturile din clădirea D(192.168.20.96/27) vor folosi adresa 192.168.20.97 ca adresă de default gateway.

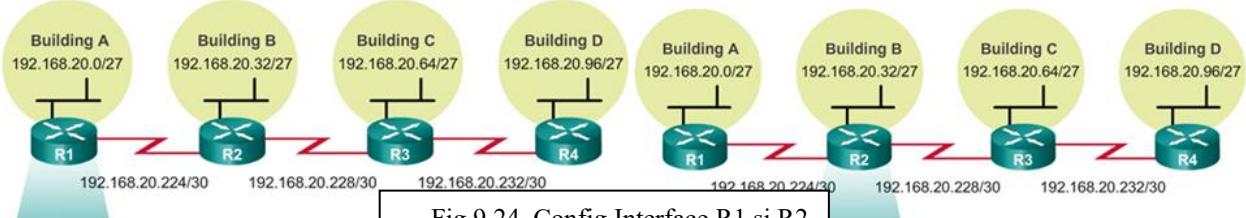


Fig.9.24. Configurație R1 și R2

```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.20.1 255.255.255.224
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 192.168.20.225 255.255.255.252
R1(config-if)#end
R1#
```

```
R2(config)#interface gigabitethernet 0/0
R2(config-if)#ip address 192.168.20.33 255.255.255.224
R2(config-if)#exit
R2(config)#interface serial 0/0/0
R2(config-if)#ip address 192.168.20.226 255.255.255.252
R2(config-if)#exit
R2(config)#interface serial 0/0/1
R2(config-if)#ip address 192.168.20.229 255.255.255.252
R2(config-if)#end
R2#
```

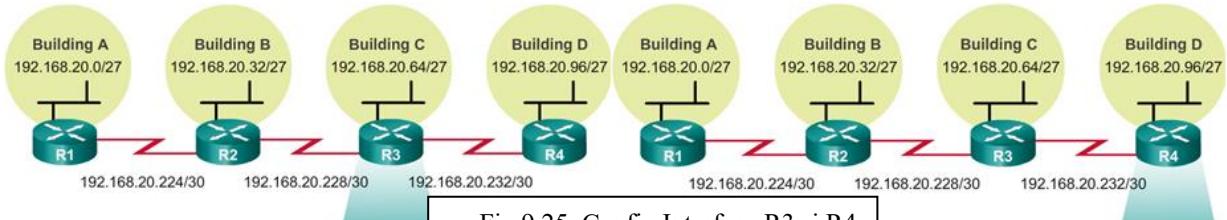


Fig.9.25. Config Interface R3 si R4

```
R3(config)#interface gigabitethernet 0/0
R3(config-if)#ip address 192.168.20.65 255.255.255.224
R3(config-if)#exit
R3(config)#interface serial 0/0/0
R3(config-if)#ip address 192.168.20.230 255.255.255.252
R3(config-if)#exit
R3(config)#interface serial 0/0/1
R3(config-if)#ip address 192.168.20.233 255.255.255.252
R3(config-if)#end
R3#

```

```
R4(config)#interface gigabitethernet 0/0
R4(config-if)#ip address 192.168.20.97 255.255.255.224
R4(config-if)#exit
R4(config)#interface serial 0/0/0
R4(config-if)#ip address 192.168.20.234 255.255.255.252
R4(config-if)#end
R4#

```

Planificarea de adrese poate fi de asemenea îndeplinită cu ajutorul unei varietăți de instrumente. O metodă este folosirea diagramei VLSM pentru identificarea a ce blocuri de adrese sunt disponibile pentru utilizare și care sunt deja atribuite. Acest lucru ajută la prevenirea atribuirii adreselor ce au fost deja alocate. Folosind rețea din exemplul anterior, diagrama VLSM poate fi folosită pentru planificarea atribuirii de adrese.

**Examinarea Subrețelelor cu masca /27** – Așa cum se poate vedea în Fig.9.26, atunci când folosim subnetarea tradițională primele șapte blocuri de adrese au fost alocate pentru LANuri și WANuri. Reamintim faptul că schema prezintă 8 subrețele cu 30 de adrese utilizabile (/27). Deși această schemă funcționa pentru segmentele LAN, există o mare pierdere de adrese în segmentele WAN.

Atunci când proiectăm o schemă de adresare pe o rețea nouă, blocurile de adrese pot fi atribuite într-un mod ce minimizează pierderea și păstrează blocuri neutilizate de adrese continue.

**Asignarea Blocurilor de Adrese cu VLSM** – Așa cum se poate observa în Fig.9.27, pentru folosirea spațiului de adrese cât mai eficient, sunt create subrețele /30 pentru legăturile WAN. Pentru a păstra blocurile de adrese nefolosite împreună, ultima subrețea /27 a fost subnetată la rândul ei în subrețele /30. Primele trei subrețele au fost atribuite legăturilor WAN.

- .224 /30 spațiul de adrese de host de la 225 la 226: legătura WAN dintre R1 și R2.
- .228 /30 spațiul de adrese de host de la 229 la 230: legătura WAN dintre R2 și R3.
- .232 /30 spațiul de adrese de host de la 233 la 234: legătura WAN dintre R3 și R4.
- .236 /30 spațiul de adrese de host de la 237 la 238: disponibile pentru utilizare.
- .240 /30 spațiul de adrese de host de la 241 la 242: disponibile pentru utilizare.
- .244 /30 spațiul de adrese de host de la 245 la 246: disponibile pentru utilizare.
- .248 /30 spațiul de adrese de host de la 249 la 250: disponibile pentru utilizare.
- .252 /30 spațiul de adrese de host de la 253 la 254: disponibile pentru utilizare.

Proiectarea schemei de adresare în acest fel lasă 3 subrețele neutilizate /27 și 5 subrețele neutilizate /30.

Basic Subnetting of 192.168.20.0/24

Fig.9.26

	/27 Network	Hosts
Building A	.0	.1 - .30
Building B	.32	.33 - .62
Building C	.64	.65 - .94
Building D	.96	.97 - .126
WAN R1 – R2	.128	.129 - .158
WAN R2 – R3	.160	.161 - .190
WAN R3 – R4	.192	.193 - .222
Unused	.224	.225 - .254

VLSM Subnetting of 192.168.20.0/24

	/27 Network	Hosts
Bldg A	.0	.1 - .30
Bldg B	.32	.33 - .62
Bldg C	.64	.65 - .94
Bldg D	.96	.97 - .126
Unused	.128	.129 - .158
Unused	.160	.161 - .190
Unused	.192	.193 - .222
	.224	.225 - .254

	/30 Network	Hosts
WAN R1-R2	.224	.225 - .226
WAN R2-R3	.228	.229 - .230
WAN R3-R4	.232	.233 - .234
Unused	.236	.237 - .238
Unused	.240	.241 - .242
Unused	.244	.245 - .246
Unused	.248	.249 - .250
Unused	.252	.253 - .254

## 9.2 Scheme de Adresare – Proiectarea Structurii

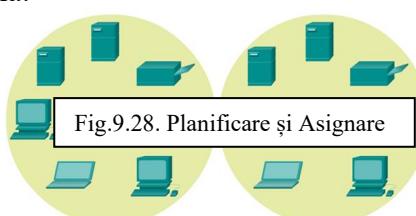
Așa cum se poate observa și în Fig.9.28, alocarea spațiului de adrese de la nivelul rețea dintr-o rețea corporativă trebuie să fie bine concepută. Atribuirea de adrese nu trebuie să fie aleatoare. Există trei principale considerații atunci când vine vorba de planificarea de alocare a adreselor.

- **Prevenirea duplicării de adrese** – *Fiecare host dintr-o internetwork trebuie să aibă o adresă IP unică. Fără o planificare și documentare adecvată, o adresă poate fi atribuită la mai mult de un host, rezultând probleme de acces pentru ambele hosturi.*
- **Asigurarea și controlul accesului** – *Unele hosturi, cum ar fi serverele, oferă resurse hosturilor interne, cât și celor externe. Adresa de nivel 3 atribuită unui server poate fi folosită pentru controlul accesului la serverul respectiv. Însă, dacă adresa este atribuită aleator și nu este bine documentată, controlul accesului este mai dificil.*
- **Monitorizarea securității și performanței** – În mod similar, securitatea și performanța hosturilor de rețea și a rețelei întregi trebuie să fie monitorizate. Ca parte a procesului de monitorizare, traficul de rețea este examinat pentru adresele ce generează sau primesc pachete în mod excesiv. Dacă există o planificare și documentare a adresării de rețea, dispozitivele de rețea problematice pot fi găsite ușor.

**Asignarea Adresării într-o Rețea** – Într-o rețea, există diferite tipuri de dispozitive, cum ar fi:

- Utilizatori finali.
- Servere și periferice.
- Hosturile accesibile din Internet.
- Dispozitive intermediare.
- Gateway.

La elaborarea unei scheme de adresare IP, este recomandată în general să existe un pattern a modului în care adresele sunt alocate pentru fiecare tip de dispozitiv. Acest lucru aduce beneficii administratorilor atunci când adaugă și scot dispozitive, filtrează traficul în funcție de IP, dar și simplifică documentația.



Un plan de adresare de rețea ar putea include diferite spații de adrese din fiecare subrețea, pentru fiecare tip de dispozitiv.

**Adresele pentru Clienți** – Din cauza provocărilor asociate cu managementul adreselor statice, dispozitivele utilizatorilor finali sunt adesea atribuite cu adrese în mod dinamic, folosind Dynamic Host Configuration Protocol (DHCP). DHCP este în general metoda preferată de atribuire a adreselor IP hosturilor din rețelele mari deoarece reduce sarcina personalului ce administrează rețeaua și elimină virtual erorile de introducere.

Un alt beneficiu al DHCP este că o adresă nu este atribuită permanent unui host, ci este numai “închiriată” pentru o perioadă de timp. Dacă trebuie să schimbăm schema de subnetare din rețeaua noastră, nu trebuie să reatribuim static adresele de host individuale. Cu DHCP, trebuie numai să reconfigurăm serverul DHCP cu noile informații. După realizarea acestui lucru, hosturile trebuie doar să își reînnnoiască automat cererile de adresare IP.

**Adresele pentru Servere și Periferice** – Orice resursă de rețea, cum ar fi un server sau o imprimantă, ar trebui să aibă o adresă IP statică, așa cum se vede și în Fig. . Hosturile client accesează aceste resurse folosind adresele IP ale dispozitivelor. Prin urmare, adresele IP previzibile pentru fiecare dintre aceste servere și periferice sunt necesare.

Servelele și perifericele sunt un punct central pentru traficul de rețea. Există mai multe pachete trimise la și de la adresele IPv4 ale acestor dispozitive. La monitorizarea traficului de rețea cu un utilitar precum Wireshark, un administrator de rețea ar trebui să fie capabil să identifice rapid aceste dispozitive. Folosind un sistem de numerotare consistent pentru aceste dispozitive face identificarea mai ușoară.

**Adresele pentru Hosturile care sunt Accessibile din Internet** – În cele mai multe internetworkuri, doar câteva dispozitive sunt accesibile de hosturile din exteriorul întreprinderii. În cea mai mare parte, aceste dispozitive sunt servere de unele tipuri. Pe toate dispozitivele dintr-o rețea care oferă resurse de rețea, adresele IP ar trebui să fie statice.

În cazul serverelor accesibile prin Internet, fiecare dintre acestea trebuie să aibă o adresă din spațiul public asociat. În plus, variații în adresa a unui astfel de dispozitiv îl va face inaccesibil din Internet. În multe cazuri, aceste dispozitive sunt pe o rețea cu adrese private. Acest lucru înseamnă că routerul sau firewallul din perimetru rețelei trebuie să fie configurați să traducă adresa internă a serverului într-o adresă publică. Datorită acestei configurații suplimentare în dispozitivul intermediar, este mai important ca acest dispozitiv să aibă o adresă previzibilă.

**Adresele pentru Echipamentele Intermediare** – Dispozitivele intermediare sunt de asemenea un punct de concentrare pentru traficul de rețea. Aproape tot traficul din sau dintre rețele este pasat prin intermediul dispozitivelor intermediare. Prin urmare, aceste dispozitive de rețea oferă o locație potrivită pentru managementul, monitorizarea și securitatea rețelei.

Dispozitivele intermediare au asignate adrese de nivel 3, fie pentru managementul de dispozitiv, fie pentru funcționarea lor. Dispozitivele, cum ar fi huburi, switchuri și puncte de acces wireless, nu necesită adrese IPv4 pentru a funcționa ca dispozitive intermediare. Însă, dacă trebuie să le accesez pentru configurație, monitorizare sau funcția de depanare, ele trebuie să fie asignate cu adrese.

Deoarece trebuie să știm cum să comunicăm cu dispozitivele intermediare, ele ar trebui să aibă adrese previzibile. Prin urmare, adresele lor sunt în mod normal atribuite manual. În plus, adresele acestor dispozitive ar trebui să fie dintr-un spațiu de adrese diferit din blocul de rețea decât adresele dispozitivelor de utilizator.

**Adresele pentru Gateway (Routere și Firewalluri)** – Spre deosebire de dispozitivele intermediare menționate, routerele și dispozitivele firewall au o adresă IP atribuită pe fiecare interfață. Fiecare interfață se află pe o rețea diferită și servește drept gateway pentru hosturile din respectiva rețea. În mod normal, interfața routerului folosește fie cea mai mare adresă , fie cea

mai mică adresă din rețea. Această atribuire ar trebui să fie uniformă peste toate rețelele din întreprindere astfel încât personalul să știe întotdeauna gatewayul rețelei, indiferent de rețea.

Interfețele routerelor și a firewallurilor sunt un punct important în traficul ce intră și ieșe din rețea. Deoarece hosturile din fiecare rețea folosesc o interfață de router sau dispozitiv firewall drept gateway din rețea, multe pachete trec prin aceste interfețe. Prin urmare, aceste dispozitive joacă un rol important în securitatea rețelei prin filtrarea pachetelor în funcție de adresele IP sursă/destinație. Gruparea diferitelor tipuri de dispozitive în grupuri de adresare logice face ca atribuirea și funcționarea filtrării de pachete să fie mai eficientă.

Network: 192.168.1.0/24		
Use	First	Last
Host Devices	.1	.229
Servers	.230	.239
Printers	.240	.249
Intermediary Devices	.250	.253
Gateway (router LAN interface)	.254	

Fig.9.30 Spațiu de Adresare IP.

### 9.3 Considerații pentru Proiectarea IPv6

#### 9.3.1 Subnetarea unei Rețele IPv6

Subnetarea IPv6 necesită o abordare diferită față de subnetarea IPv4. Principalul motiv este acela că în IPv6 există foarte multe adrese, motiv pentru care subnetarea este complet diferită. Un spațiu de adrese IPv6 nu este subnetat pentru a conserva adrese; mai degrabă, este subnetat pentru suportul designului logic ierarhic al rețelei. Pe când subnetarea IPv4 este pentru gestionarea deficitului de adrese, subnetarea IPv6 este pentru construirea unei ierarhii de adresare în funcție de numărul de routere și de rețele suportate.

Reamintim faptul că un bloc de adrese IPv6 cu un prefix /48 are 16 biți pentru subnet ID, așa cum se poate observa și în Fig.9.31A. Subnetarea utilizând subnet ID de 16 biți conduce la 65.536 /64 posibile subrețele și nu necesită împrumutul niciunui bit din ID-ul de interfață sau din partea de host a adresei. Fiecare subrețea IPv6 /64 conține aproximativ 18 adrese quintillion (o mie la puterea 6), în mod evident, mai mult decât va fi vreodată nevoie într-un singur segment de rețea IP.

Subrețelele create din subnet ID sunt ușor de reprezentat deoarece nu este necesară conversia în binar. Pentru a determina următoarea subrețea disponibilă, doar numărăm în hexazecimal. Așa cum se vede în Fig.9.31.B, acest lucru înseamnă numărarea în hexazecimal în partea de subnet ID.

Prefixul de routare global este același pentru toate subrețele. Numai cvartetul subnet ID este incrementat pentru fiecare subrețea.

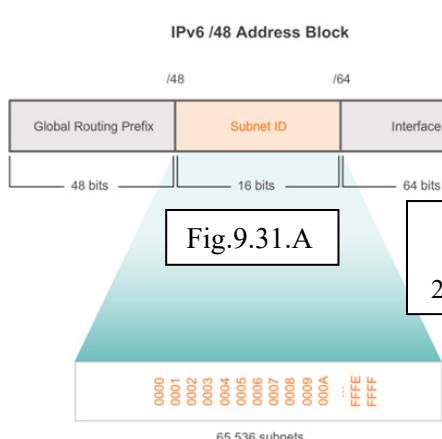
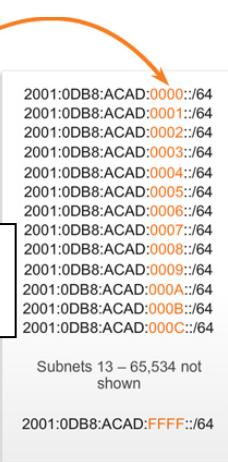


Fig.9.31.B Blocul de Adrese pentru 2001:0BB8:ACAD:/48



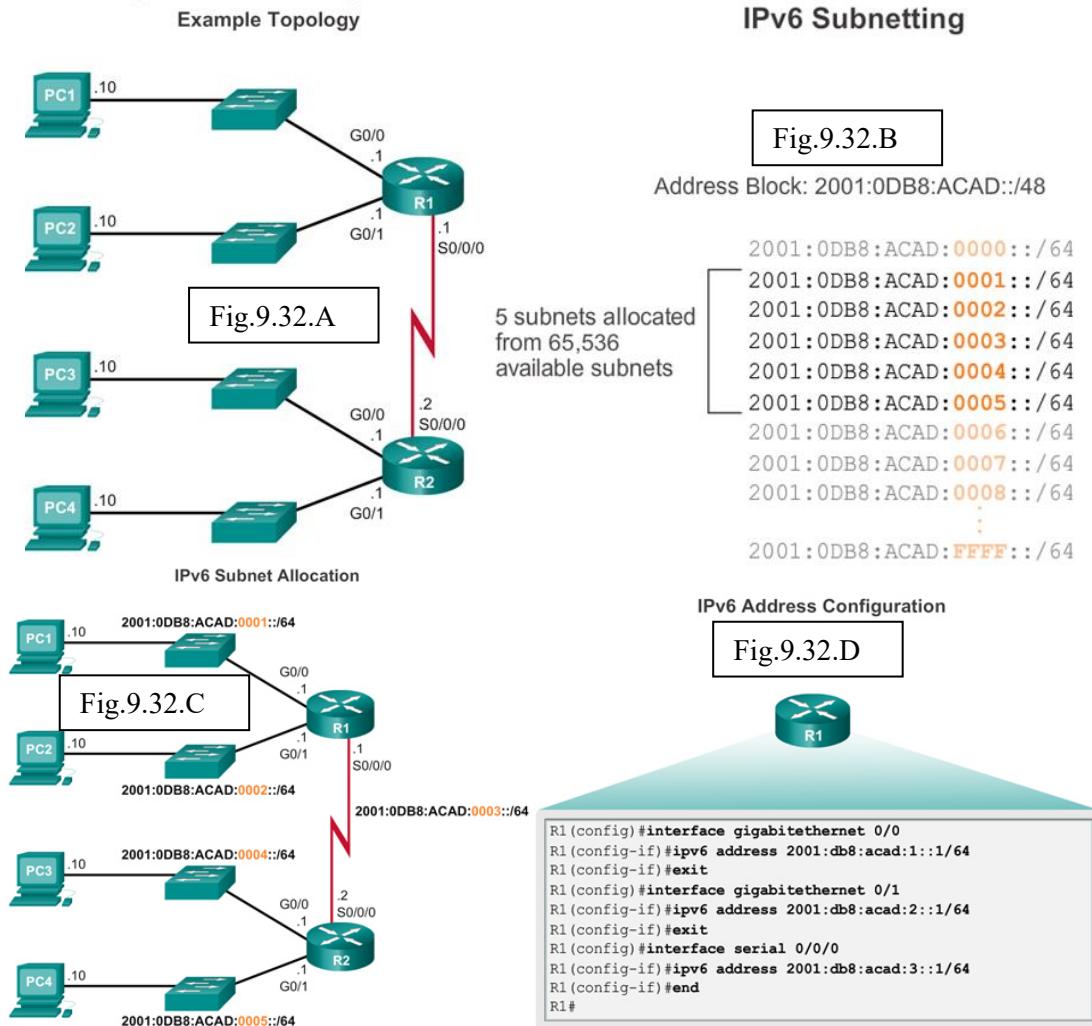
Cu mai mult de 65.000 de subrețele de unde să alegem, sarcina administratorului de rețea devine una de proiectare a unei scheme logice de adresare a rețelei.

Așa cum se poate observa în Fig.9.32.A, topologia exemplului va necesita subrețele pentru fiecare LAN, cât și pentru legătura WAN dintre R1 și R2. Spre deosebire de exemplul pentru IPv4, cu IPv6 subrețea din legătura WAN nu va fi subnetată în continuare. Deși acest lucru ar putea “risipi” adrese, nu reprezintă o preocupare atunci când folosim IPv6.

Așa cum se poate observa în Fig.9.32.B, alocarea de 5 subrețele IPv6, cu subnet ID de la 0001 la 0005 va fi folosită pentru acest exemplu. Fiecare subrețea /64 va oferi mai multe adrese decât vor fi vreodată necesare.

Așa cum se poate observa în Fig.9.32.C, fiecare segment LAN și legătura WAN au atribuite o subrețea /64.

Similar configurării de IPv4, Fig.9.32.D prezintă faptul că fiecare interfață a routerului a fost configurată cu o subrețea diferită IPv6.



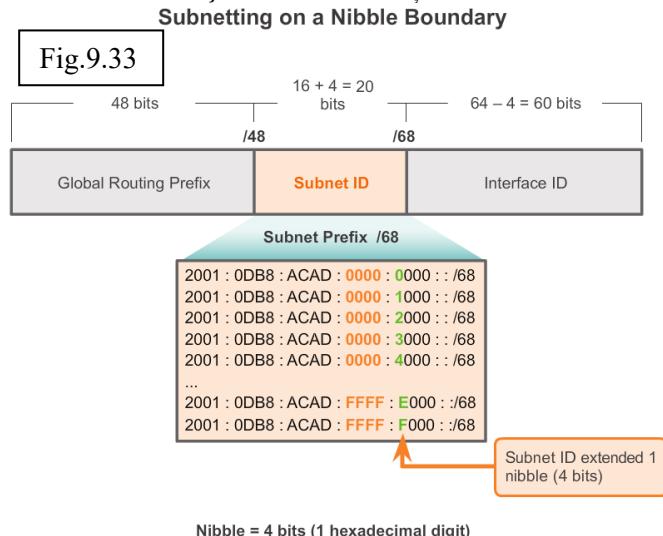
Similar împrumutului de biți din partea de host a unei adrese IPv4, biții IPv6 pot fi împrumutați din ID-ul interfeșei pentru a crea subrețele IPv6 suplimentare. Acest lucru este de obicei efectuat din motive de securitate pentru a crea mai puține hosturi pe subrețea și nu neapărat pentru crearea de subrețele suplimentare.

Atunci când extindem subnet ID prin împrumutarea de biți din interface ID, cea mai bună practică este aceea de a subneta pe “nibble boundary”. Un nibble este o cifră hexazecimală sau 4 biți. Ca și în Fig., prefixul de subnet /64 este extins cu 4 biți sau 1 nibble la /68. Făcând acest lucru, reducem dimensiunea interface ID cu 4 biți, de la 64 la 60 de biți.

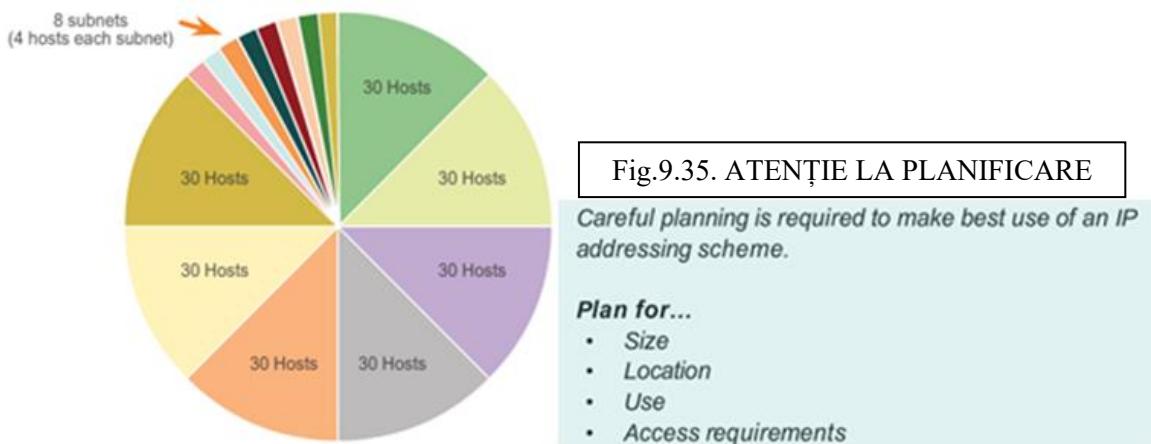
Subnetarea pe “nibble boundaries” înseamnă că folosim numai măști de rețea aliniate nibble. Începând cu /64, măștile de rețea aliniate nibble sunt /68, /72, /76, /80 etc.

Subnetarea pe “nibble boundaries” crează subrețele prin folosirea valorii hexazecimale suplimentare. În exemplu, noul subnet ID constă din 5 valori hexazecimale, începând de la 00000 la FFFF.

Este posibilă subnetarea în interiorul unei “nibble boundary”, într-o cifră hexazecimală, însă nu este recomandat sau necesar. Subnetarea în interiorul nibble nu are avantajul determinării ușoare a prefixului din interface ID. De exemplu, dacă este folosită o lungime de prefix /66, primii doi biți vor fi parte a subnet ID și următorii 2 biți din interface ID.



#### 9.4 Concluzii Capitolul 9



Prin parcurgerea acestui material un tehnician de rețea devine familiar cu implementările de adresare IPv4 și IPv6, gata să preiea o infrastructură de rețea existentă și să aplice cunoștințele și abilitățile sale pentru a finaliza conFig.ția.

Așa cum se poate observa în Fig. , procesul de segmentare a rețelei, prin divizarea sa în mai multe spații de rețea mai mici, se numește subnetare.

Fiecare adresă de rețea are un range valid de adrese de host. Toate dispozitivelor atașate la aceeași rețea vor avea o adresă IPv4 de host pentru respectiva rețea și o mască de rețea comună sau prefix de rețea comun. Traficul nu poate fi livrat între subrețele fără utilizarea unui router. Pentru a determina dacă traficul este local sau la distanță, routerul folosește masca de subrețea.

Prefixul și masca de subrețea sunt moduri diferite de reprezentare a același lucru – partea de rețea a unei adrese.

Subrețele IPv4 sunt create prin folosirea unuia sau a mai multor biți ca biți de rețea. Doi factori foarte importanți ce vor duce la determinarea blocului de adresă IP cu masca de rețea sunt numărul de subrețele necesare și numărul maxim de hosturi necesare pe subrețea. Este o relație inversă între numărul de subrețele și numărul de hosturi. Cu cât sunt împrumutați mai mulți biți pentru a crea subrețele, cu atât rămân mai puțini biți disponibili; prin urmare, mai puține hosturi pe subrețea.

Formula  $2^n$  (unde n este numărul de biți de host rămași) este folosită pentru calcularea numărului de adrese ce vor fi disponibile în fiecare subrețea. Însă, adresa de rețea și adresa de broadcast dintr-un range nu sunt utilizabile; prin urmare, pentru a calcula numărul de adrese utilizabile, formula  $2^{n-2}$  este necesară.

Subnetarea unei subrețele, sau folosirea Variable Length Subnet Mask (VLSM) a fost proiectată pentru evitarea risipei de adrese.

Subnetarea IPv6 necesită o abordare diferită decât subnetarea IPv4. Un spațiu de adrese IPv6 nu este subnetat pentru conservarea adreselor; mai degrabă este subnetat pentru suportul designului logic, ierarhic al rețelei. Deci, subnetarea IPv4 este pentru gestionarea deficitului de adrese, iar subnetarea IPv6 este pentru construirea unei ierarhii de adresare în funcție de numărul de routere și de rețele suportate.

Planificarea cu atenție este necesară pentru cea mai bună utilizarea a spațiului de adrese disponibile. Cerințele de dimensiune, locație, folosire și acces sunt considerații în procesul de planificare de adresare.

După implementare, o rețea IP trebuie să fie testată pentru verificarea conectivității și a performanței funcționale.

Original	192.	168.	1.	0	000	0000	Network: 192.168.1.0/24
Mask	255.	255.	255.	0	000	0000	Mask: 255.255.255.0

Fig.9.36. ATENȚIE LA SUBNETARE

Borrowing 1 bit creates 2 subnets with the same mask.



Net 0	192.	168.	1.	0	000	0000	Network: 192.168.1.0/25
Mask	255.	255.	255.	1	000	0000	Mask: 255.255.255.128
Net 1	192.	168.	1.	1	000	0000	Network: 192.168.1.128/25
Mask	255.	255.	255.	1	000	0000	Mask: 255.255.255.128

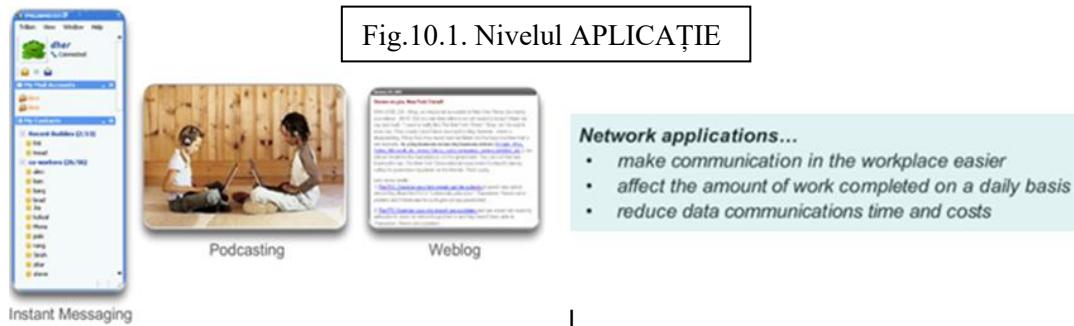
## CAPITOLUL 10. NIVELUL APLICAȚIE

### Introducere

Experimentăm Internetul prin intermediul World Wide Web atunci când jucăm jocuri online, vorbim cu prietenii, comunicăm prin e-mail și facem cumpărături de pe siteuri web. Aplicațiile, cum ar fi cele folosite pentru oferirea serviciilor menționate, oferă interfață umană peste bazele rețelelor. Ele ne permit să trimitem și să primim date cu ușurință. În mod normal le putem accesa și folosi fără a ști cum funcționează. Însă, pentru profesioniștii de rețea, este important să știe modul în care o aplicație este capabilă să formateze, transmită și interpreteze mesajele ce sunt trimise și primite prin rețea.

Vizualizarea mecanismelor care permit comunicarea în întreaga rețea se face mai ușor dacă vom folosi cadrul de lucru stratificat al modelului OSI.

În acest capitol, vom explora rolul nivelului aplicație și modul în care aplicațiile, serviciile și protocoalele din nivelul aplicație fac posibilă comunicarea complexă în rețelele de date.



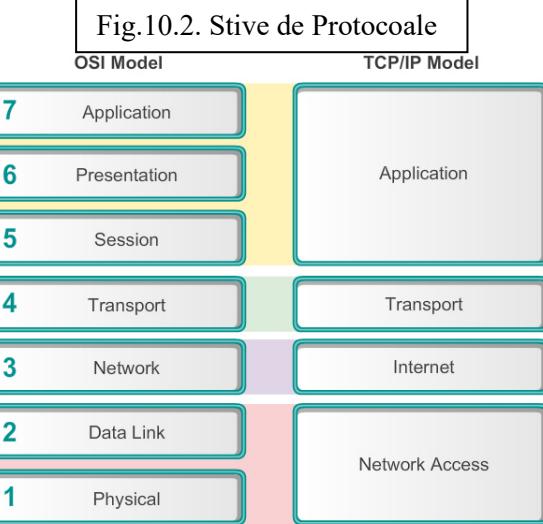
### 10.1 Protocolele de la nivelul aplicație

Așa cum se poate observa și în Fig. , profesioniștii de rețea folosesc modelele OSI și TCP/IP pentru a transmite documente tehnice verbale și scrise. Specialiștii de rețea pot utiliza aceste metode pentru a descrie comportamentul protocoalelor și aplicațiilor.

În modelul OSI, datele sunt pasate de la nivel la nivel, începând cu nivelul aplicație de pe hostul sursă și sunt procesate până la nivelul fizic unde sunt transmise pe canalul de comunicație la hostul destinație unde datele sunt procesate de la nivelul fizic la nivelul aplicație de pe hostul destinație.

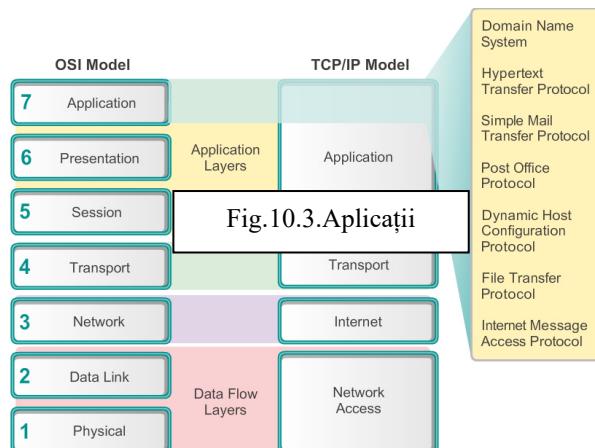
Nivelul aplicație este în vârful ambelor modele, OSI și TCP/IP. Nivelul aplicație TCP/IP include un număr de protocoale ce oferă funcționalitate specifică unei varietăți de aplicații de utilizator. Această funcționalitate a protocoalelor de la nivelul aplicație TPC/IP se potrivește cu cadrul de lucru al celor trei nivele de vârf al modelului OSI: aplicație, prezentare și sesiune. Nivelele modelului OSI 5, 6 și 7 sunt folosite ca referință pentru dezvoltatorii și furnizorii de aplicații software pentru a produce produse, cum ar fi pagini web necesare pentru accesul la rețele.

**Comparing the OSI Model and TCP/IP Models**



## **10.2 Nivelul Aplicație**

Nivelul aplicație este cel mai aproape de utilizatorul final. Așa cum se poate observa în Fig. , acesta este nivelul ce oferă interfață dintre aplicațiile utilizate pentru comunicarea și baza rețelei peste care mesajele noastre sunt transmise. Protocolele de la nivelul aplicație sunt folosite pentru schimbul de date dintre programele ce rulează pe sursă și destinație. Există multe protocoale de nivel aplicație și noi protocoale ce sunt în continuă dezvoltare. Unele dintre cele mai cunoscute protocoale de nivel aplicație sunt HyperText Transfer Protocol (HTTP), File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), Internet Message Access Protocol (IMAP) și Domain Name System (DNS) protocol.



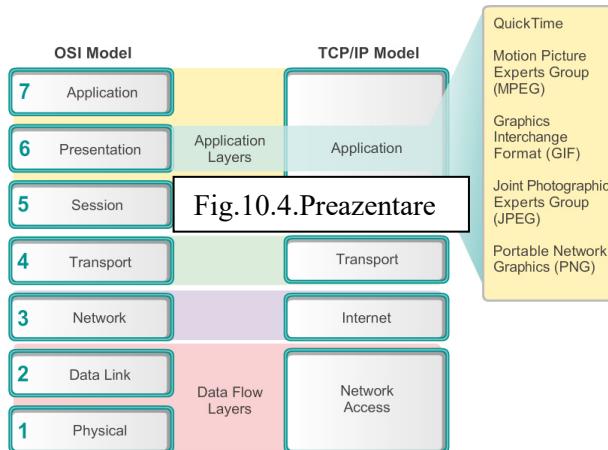
## **10.3 Nivelul Prezentare**

Nivelul prezentare are trei funcții principale:

- Formatează, sau prezintă, datele de la dispozitivul sursă într-o formă compatibilă pentru primirea lor pe dispozitivul sursă.
- Compresează datele într-un mod ce pot fi decomprimate de către dispozitivul destinație.
- Criptează datele pentru transmisie și descripează datele pe dispozitivul destinație.

Aşa cum se poate observa în Fig. , nivelul prezentare formează datele de la nivelul aplicație și setează standarde pentru formatele de fișier. Unele standarde bine-cunoscute pentru video sunt QuickTime și Motion Picture Experts Group (MPEG). QuickTime este o specificație de computer Apple pentru video și audio și MPEG este un standard pentru compresia și codarea video și audio.

Printre cele mai cunoscute formate de imagini grafice care sunt folosite în rețele sunt formatele Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG) și Portable Network Graphics (PNG). GIF și JPEG sunt standarde de compresie și codare pentru imaginile grafice. PNG a fost dezvoltat pentru a adresa anumitor limitări ale formatului GIF și eventual pentru înlocuirea sa.



#### 10.4 Nivelul Sesiune

Aşa cum sugereaza şi numele, funcţiile nivelului sesiune sunt de a crea şi menţine dialogurile dintre aplicaţiile sursă şi destinaţie. Nivelul sesiune gestionează schimbul de informaţii pentru iniţierea dialogurilor, le ține active şi restabileşte sesiunile dacă sunt întrerupte pentru o perioadă mai mare de timp.

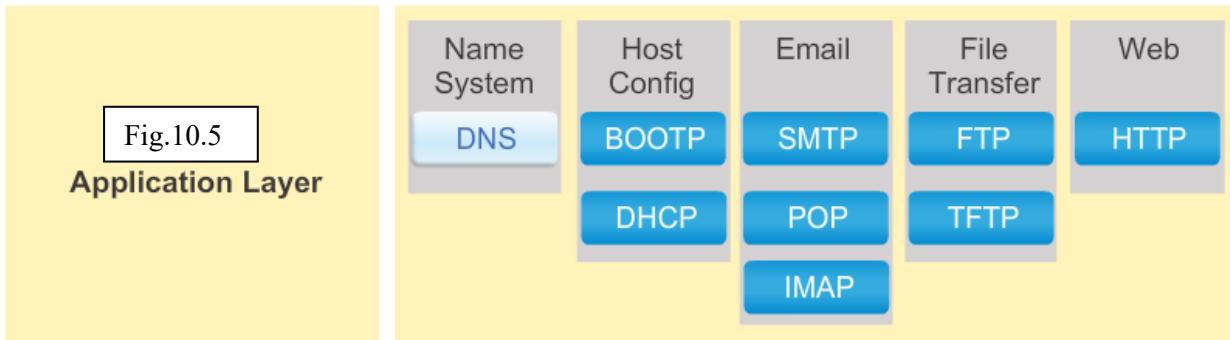
Pe când modelul OSI separă funcţia individuală de aplicație, prezentare și sesiune, aplicațiile TCP/IP cele mai cunoscute și implementate încorporează funcționalitatea acestor trei nivele.

Protocolele aplicație TCP/IP specifică formatul și informațiile de control necesare pentru mai multe funcții de comunicație prin Internet. Printre aceste protocoale TPC/IP sunt:

- **Domain Name System (DNS)** – Acest protocol rezolvă numele Internet în adrese IP.
- **Telnet** – Acesta este folosit pentru a oferi acces de la distanță la servere și dispozitive de rețea, fără securitatea transmisiei.
- **Simple Mail Transfer Protocol (SMTP)** - Acest protocol transferă mesaje mail și anexele acestora.
- **Dynamic Host ConFig.tion Protocol (DHCP)** - Protocol folosit pentru atribuirea unei adrese IP, mască de rețea, default gateway și adrese de server DNS unui host.
- **Hypertext Transfer Protocol (HTTP)** - Acest protocol transferă fișiere ce alcătuiesc paginile web ale World Wide Web.
- **File Transfer Protocol (FTP)** - Un protocol folosit pentru schimbul de fișiere interactiv între sisteme.
- **Trivial File Transfer Protocol (TFTP)** - Acest protocol este folosit pentru transferul de fișiere active neorientat pe conexiune.

- **Bootstrap Protocol (BOOTP)** - *Acest protocol este un precursor al protocolului DHCP. BOOTP este un protocol de rețea folosit pentru a obține informații de adresă IP în timpul bootup.*
- **Post Office Protocol (POP)** - *Un protocol folosit de către clienții de e-mail pentru a preluă e-mail de la un server de la distanță și a-l plasa pe hostul client.*
- **Internet Message Access Protocol (IMAP)** - *Acesta este un alt protocol pentru preluarea de e-mail și distribuirea acestuia pe hostul client.*

Protocolele de la nivelul aplicație sunt utilizate de către dispozitivele sursă și destinație în timpul unei sesiuni de comunicare. Pentru ca aceste comunicații să fie cu succes protocolele de nivel aplicație implementate pe hostul sursă și destinație trebuie să fie compatibile.



## 10.5 Cum Protocolele Aplicație Interacționează cu Aplicațiile Utilizator

La accesarea informațiilor pe un dispozitiv de rețea, fie că este un PC, laptop, tabletă, smartphone sau alt dispozitiv conectat la rețea, datele ar putea să nu fie stocate fizic pe dispozitiv. În acest caz, o cerere de acces a respectivelor informații trebuie să fie efectuată dispozitivului pe care sunt stocate datele. În modelul peer-to-peer (p2p), datele sunt accesate de la un dispozitiv peer fără utilizarea unui server dedicat.

Modelul de rețea P2P include două părți: rețelele P2P și aplicațiile p2p. Ambele părți au caracteristici similare, însă în practică lucrează diferit.

### 10.5.1 Rețelele P2P

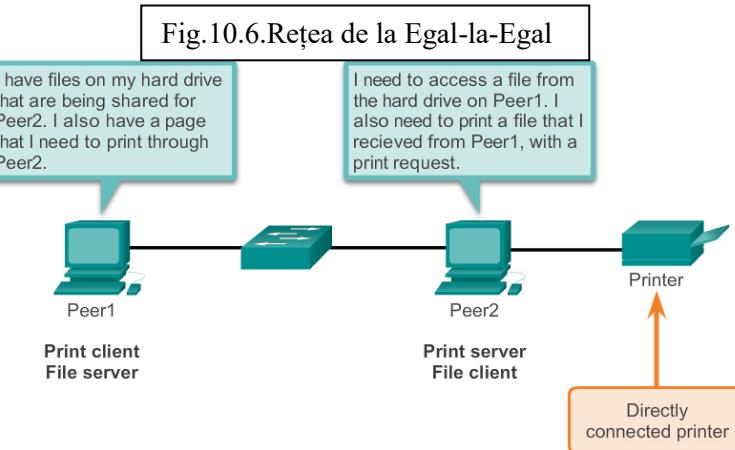
Într-o rețea P2P, două sau mai multe computere sunt conectate printr-o rețea și împart resurse (cum ar fi imprimante și fișiere) fără existența unui server dedicat. Fiecare dispozitiv final conectat (cunoscut ca un peer) poate funcționa atât ca server, cât și ca un client. Un computer își poate asuma rolul de server pentru o tranzacție, în timp ce este un client pentru altă tranzacție. Rolerile clientului și serverului sunt setate pe bază de cerere.

Un exemplu este o rețea de domiciliu simplă cu două computere, aşa cum se arată în Fig. . În acest exemplu, Peer2 are o imprimantă atașată direct prin USB și este setat să împartă imprimanta în rețea astfel încât și Peer1 să poată printa. Peer1 este setat să împartă un drive sau folder în rețea. Acest lucru permite ca Peer2 să acceseze și să salveze fișierele din folderul partajat. În plus față de împărțirea de fișiere, o rețea ca aceasta va permite utilizatorilor să activeze jocuri de rețea sau să împartă o conexiune la Internet.

Rețelele P2P descentralizează resursele dintr-o rețea. În schimbul localizării datelor pe servere dedicate pentru a fi partajate, datele pot fi localizate oriunde și pe orice dispozitiv conectat. Multe sisteme de operare actuale suportă partajarea de fișiere și printarea fără necesitatea unui software de server suplimentar. Însă, rețelele P2P nu folosesc conturi de utilizator centralizate sau servere de acces pentru a gestiona permisiunea. Prin urmare, este dificilă asigurarea securității și politicilor de acces în rețelele ce conțin mai multe computere.

Conturile de utilizator și drepturile de acces trebuie să fie setate individual pe fiecare dispozitiv peer.

### Peer-to-Peer Networking

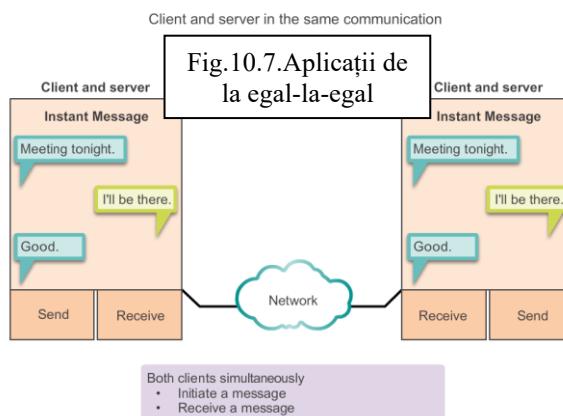


O aplicație **p2p** permite unui dispozitiv să funcționeze și ca server precum ca și client în aceeași comunicație, așa cum se poate observa și în Fig. . În acest model, fiecare client este un server și fiecare server este un client. Ambele pot iniția o comunicație și sunt considerate egale în procesul de comunicare. Însă, aplicațiile **p2p** necesită ca fiecare dispozitiv final să ofere o interfață de utilizator și să ruleze un serviciu de background. Atunci când lansăm o anumită aplicație **p2p**, se încarcă interfața de utilizator necesară și serviciile de background; cu alte cuvinte, dispozitivele pot comunica direct.

Unele aplicații **p2p** folosesc un sistem hybrid în care partajarea de resurse este descentralizată, însă indecșii ce pointează la locațiile resursei sunt stocați într-un director centralizat. Într-un sistem hybrid, fiecare peer accesează un server indexat pentru a afla locația unei resurse stocată pe un alt peer. Serverul indexat de asemenea ajută la conectarea celor două peer, însă după conectare, comunicația are loc între perechi fără comunicație suplimentară cu serverul indexat.

Aplicațiile **p2p** pot fi utilizate în rețelele P2P, rețelele client/server și peste Internet.

### Peer-to-Peer Applications



Cu aplicațiile **p2p**, fiecare computer din rețea ce rulează aplicația funcționează ca un client sau ca un server pentru alte computere din rețea ce rulează aplicația. Aplicații **p2p** sunt:

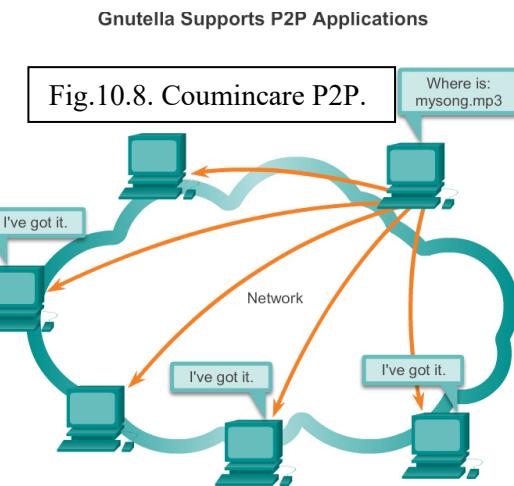
- *eDonkey*.
- *eMule*.
- *Shareaza*.
- *BitTorrent*.

- *Bitcoin*.
- *LionShare*.

Unele aplicații **p2p** sunt bazate pe protocolul Gnutella. Ele permit oamenilor să împartă fișierele de pe hard diskurile lor cu alți utilizatori. Așa cum se vede în Fig. , softwareul de client compatibil Gnutella permite utilizatorilor să se conecteze la servicii Gnutella peste Internet și să localizeze și acceseze resurse partajate de alte perechi Gnutella. Multe aplicații client sunt disponibile pentru accesarea rețelei Gnutella, cum ar fi BearShare, Gnuclues, LimeWire, Morpheus, WinMX și XoloX.

Pe când Gnutella Developer Forum gestionează protocolul de bază, furnizorii de aplicații adesea dezvoltă extensii pentru a face protocolul să funcționeze mai bine pe aplicațiile lor.

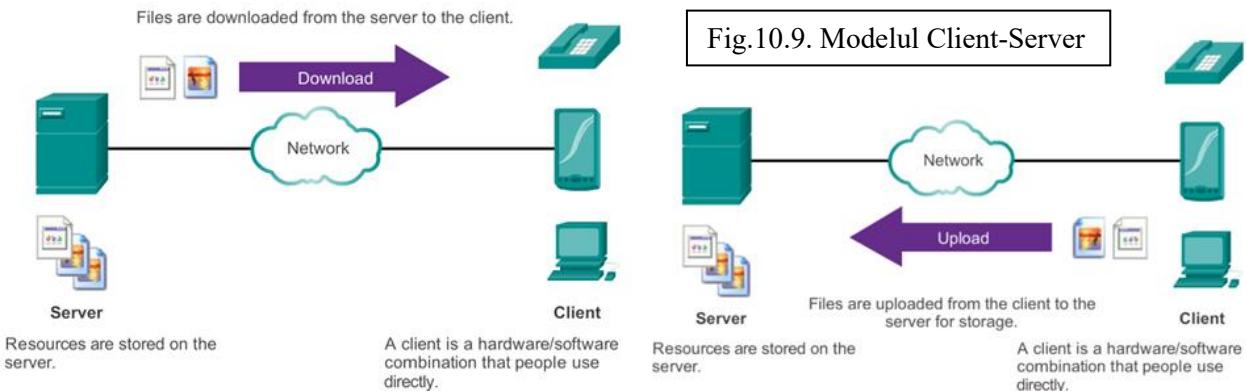
Multe aplicații **p2p** nu folosesc o bază de date centrală pentru a stoca toate fișierele disponibile pe perechi. În schimb, dispozitivele din rețea își “spun” între ele că fișierele sunt disponibile la cerere și folosesc protocolul și serviciile de partajare de fișier pentru a sprijini localizarea resurselor.



În modelul client-server, dispozitivul care cere informațiile este numit client, iar cel ce răspunde la cerere se numește server. Procesele client și server sunt considerate la nivelul aplicație. Clientul începe schimbul prin cererea datelor de la server, ce răspunde prin trimiterea unuia sau a mai multor fluxuri de date la client. Protocolele de la nivelul aplicație descriu formatul cererilor și răspunsurilor dintre clienți și servere. Pentru transferul de date real, acest schimb necesită și autentificarea de utilizator și identificarea fișierului de date ce trebuie transferat.

Un exemplu de rețea client-server este utilizarea unui serviciu de mail al ISP de a trimite, primi și stoca e-mail. Clientul de e-mail de pe un computer de domiciliu inițiază o cerere la serverul de e-mail al ISP pentru orice mail necitit. Serverul răspunde prin trimiterea emailului cerut de către client.

Deși datele sunt în mod normal descrise ca și cum ar “curge” de la server la client, unele datele “curg” întotdeauna de la client la server. Fluxul de date poate fi egal în ambele direcții, sau poate fi chiar mai mare în direcția de la client la server. De exemplu, un client poate să transfere un fișier de la server pentru scopuri de stocare. Așa cum se poate observa în Fig. , transferul de date de la un client la un server se numește **upload**, iar transferul de date de la un server la client **download**.

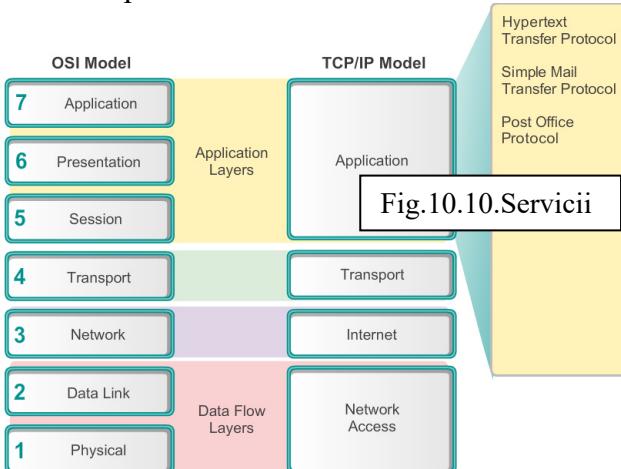


### 10.5.2 Protocole și servicii bine-cunoscute de la nivelul aplicație

Există zeci de protocole de nivel aplicație, însă într-o zi obișnuită se folosesc probabil numai cinci sau șase. Trei protocole de nivel aplicație ce sunt implicate în activitatea de zi cu zi sunt:

- *Hypertext Transfer Protocol (HTTP)*.
- *Simple Mail Transfer Protocol (SMTP)*.
- *Post Office Protocol (POP)*.

ACESTE PROTOCOALE DE NIVEL APlicațIE FAc POsIBILĂ CĂUTAREA ÎN WEB ȘI TRIMITEREA ȘI PRIMIREA DE E-MAIL. HTTP ESTE FOLOSiT PEnTRU A PERMITE UTILIZATORILOR Să SE CONECTEZPe SITEURi WEB DIn INTERNET. SMTP ESTE FOLOSiT PEnTRU A PERMITE UTILIZATORILOR Să TRIMiTă E-MAIL. POP ESTE FOLOSiT PEnTRU PERMITEREA UTILIZATORILOR Să PRIMEASCă E-MAIL.



Atunci când o adresă web sau *Uniform Resource Locator* (URL) este introdusă într-un browser web, browserul web stabilește o conexiune cu serviciul web ce rulează pe serverul ce folosește protocolul HTTP. URLs și *Uniform Resource Identifier* (URIs) sunt numele ce mulți oameni le asociază cu adrese web.

<http://www.fmi.cti-ro/index.html> URL este un exemplu de URL ce este asociat unei anumite resurse; o pagină web numită **index.html** pe un server ce se identifică ca **fmi.cti-ro**.

Browserele web sunt tipurile de aplicații client de pe un computer folosite pentru conectarea la World Wide Web și pentru a accesa resursele stocate pe un server web. Ca multe procese server, serverul web rulează ca serviciu de background și face tipuri diferite de fișiere să fie disponibile.

Pentru a accesa conținutul, clienții web se leagă la server și cer resursele dorite. Serverul răspunde cu resursele și, după primire, browserul interpretează datele și le prezintă utilizatorului.

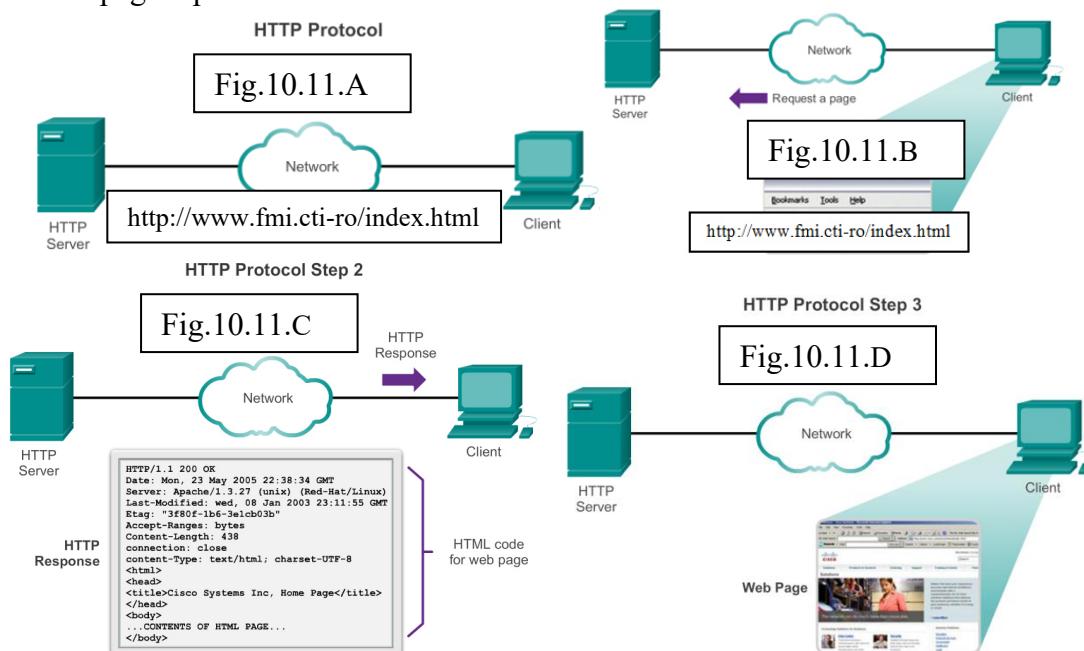
Browserele pot interpreta și prezenta mai multe tipuri de date (cum ar fi text clar sau Hypertext Markup Language, limbajul în care paginile web sunt alcătuite). Alte tipuri de date, însă, ar putea necesita alt serviciu sau program, referit nouă în mod normal ca plug-ins sau add-ons. Pentru a ajuta browserul să determine ce tip de fișier primește, serverul specifică ce tip de date conține fișierul.

Pentru a înțelege mai bine cum interacționează browserul și clientul, putem examina modul în care o pagină web este deschisă într-un browser. Pentru acest exemplu, folosim <http://www.fmi.cti-ro/index.html> URL.

Prima dată, aşa cum se poate observa și în Fig.10.11.A, browserul interpretează trei părți ale URL:

1. **http (protocolul sau schema).**
2. **www. fmi.cti-ro (numele serverului).**
3. **index.html (numele de fișier specific cerut).**

Așa cum se poate vedea în Fig.10.11.B, browserul apoi verifică un server de nume ce convertește **www. fmi.cti-ro** într-o adresă numerică, utilizată pentru conectarea la server. Utilizând cerințele HTTP, browserul trimită o cerere **GET** serverului și cere fișierul **index.html**. Serverul, cum se poate vedea în Fig.10.11.C, trimite un cod HTML pentru această pagină browserului. La final, aşa cum se poate vedea în Fig.10.11.D, browserul deschide codul HTML și formează pagina pentru fereastra de browser.



HTTP este folosit peste World Wide Web pentru transferul de date și este unul dintre cele mai utilizate protocole de aplicație de astăzi. A fost dezvoltat inițial pentru a simplifica publicarea și preluarea paginilor HTML; însă flexibilitatea HTTP l-a facut să fie o aplicație vitală din sistemele distribuite de colaborare pentru informație.

HTTP este un protocol bazat pe cerere/răspuns. Atunci când un client, în mod normal un browser web, trimită o cerere la un server web, HTTP specifică tipul de mesaj pentru comunicație. Trei tipuri comune de mesaj sunt **GET**, **POST** și **PUT** (Fig.).

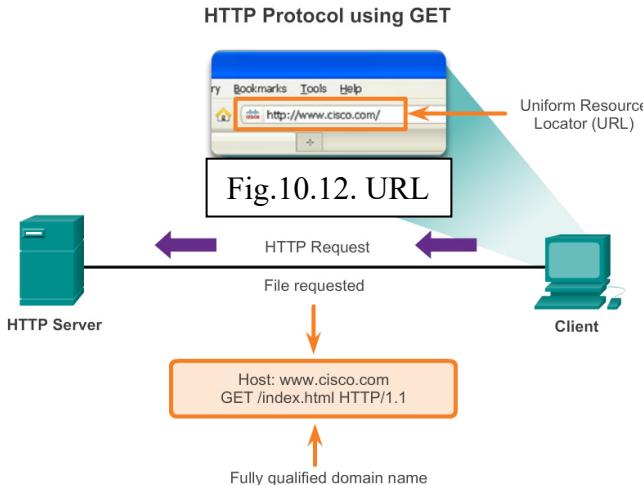
**GET** este o cerere de date a clientului. Un client (browser web) trimită un mesaj **GET** serverului web pentru a cere pagini HTML. Atunci când serverul primește cererea **GET**, răspunde cu o linie de status, cum ar fi **HTTP/1.1 200 OK** și cu un mesaj propriu. Mesajul de la server ar putea include fișierul HTML cerut, dacă este disponibil, sau poate conține o eroare sau un mesaj de informare, cum ar fi "The location of the requested file has changed."

**POST** și **PUT** sunt folosite pentru încărcarea fișierelor de date pe serverul web. De exemplu, atunci când utilizatorul introduce datele într-o formă încorporată unei pagini web (cum ar fi cazul în care se completează o cerere de ordine), mesajul **POST** este trimis serverului web. Datele pe care utilizatorul le-a prezentat în formă sunt incluse în mesajul **POST**.

**PUT** încarcă resursele sau conținutul serverului web. De exemplu, dacă un utilizator încearcă să încarce un fișier sau o imagine pe un website, un mesaj **PUT** este trimis de la client la server cu fișierul atașat sau cu imaginea.

Deși HTTP este remarcabil flexibil, nu este un protocol sigur. Mesajele cerute trimit informații serverului în text clar ce poate fi interceptate și citite. În mod similar, răspunsurile serverului, în mod normal pagini HTML, sunt de asemenea necriptate.

Pentru comunicarea securizată prin Internet, protocolul HTTP Secure (HTTPS) este folosit pentru accesarea și postarea informațiilor de server web. HTTP Secure (HTTPS) poate folosi autentificarea și criptarea pentru securizarea datelor care circulă de la client la server. HTTPS specifică reguli suplimentare pentru transferarea datelor de la nivelul aplicație la nivelul transport. HTTPS folosește același proces cerere client-răspuns server ca HTTP, însă streamul de date este criptat cu Secure Socket Layer (SSL) înainte de a fi transferat peste rețea. HTTPS crează timp de procesare și încarcare suplimentar pe server datorită criptării și decriptării traficului.



Unul dintre principalele servicii oferite de către un ISP este gazdă de e-mail (hosting). Serviciul Email a revoluționat modul în care oamenii comunică prin viteza și simplitatea sa. Pentru a rula pe un computer sau alte dispozitive finale, e-mail necesită multe aplicații și servicii.

Email este o metodă store-and-forward pentru trimiterea, stocarea și preluarea mesajelor electronice dintr-o rețea. Mesajele e-mail sunt stocate în baze de date sau servere de mail. ISPiștii adesea gestionează servere de mail ce suportă mai multe conturi de clienți diferite.

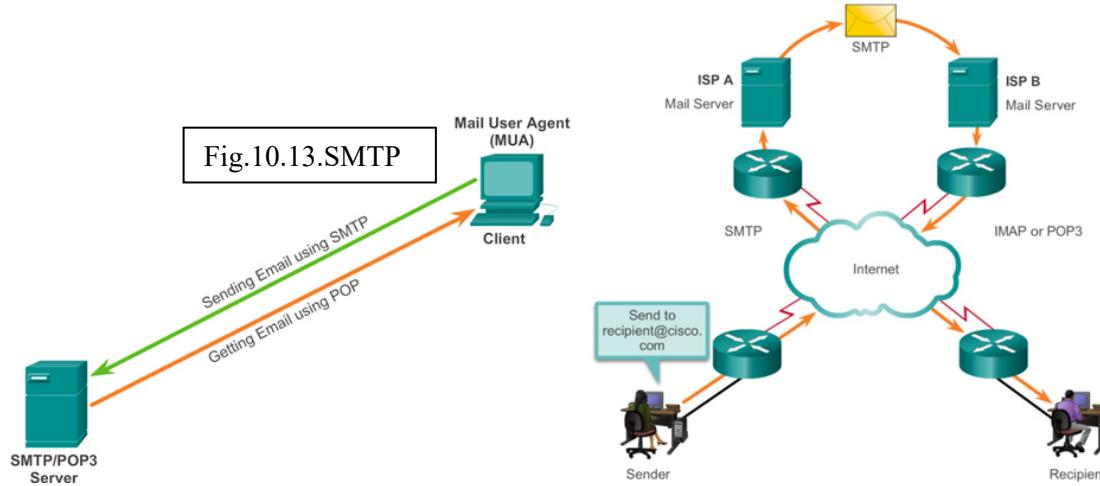
Clienții de e-mail comunică cu serverele de mail pentru a trimite și primi e-mail. Serverele de mail comunică cu alte servere de mail pentru a transporta mesajele de la un domeniu la altul. Un client e-mail nu comunică direct cu alt client e-mail atunci când trimite un e-mail. În schimb, ambii clienți se bazează pe serverul de mail pentru transportarea mesajelor. Acest lucru se întâmplă chiar și atunci când ambii clienți sunt în același domeniu.

Clienții de e-mail trimit mesaje serverului de e-mail config.t în setările de aplicație. Atunci când serverul primește mesajul, verifică dacă domeniul este localizat în baza de date locală. Dacă nu este, trimită o cerere DNS pentru a determina adresa IP a serverului de mail pentru domeniul destinație. E-mailul este apoi transmis mai departe serverului adecvat.

E-mail suportă trei protocoale separate de funcționare: Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP) și Internet Message Access Protocol (IMAP). Procesul de la

nivelul aplicație ce trimite mailul folosește SMTP. Aceasta este cazul de trimitere de la un client la un server, sau de la un server la altul.

Un client preia e-mailul, însă folosind unul dintre cele două protocoale de nivel aplicație: POP sau IMAP.

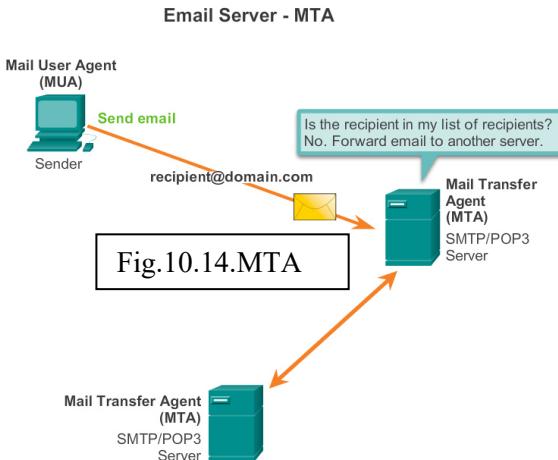


Simple Mail Transfer Protocol (SMTP) transferă mailul în mod eficient și de încredere. Pentru ca aplicațiile SMTP să lucreze eficient, mesajul de mail trebuie să fie transmis adecvat și procesele SMTP trebuie să ruleze pe ambele echipamente, client și server.

Formatele de mesaj SMTP necesită un header de mesaj și un corp al mesajului. Pe când corpul mesajului poate conține orice cantitate de text, headerul mesajului trebuie să aibă o adresă a destinatarului e-mailului corect formatată și o adresă a sursei. Orice alte informații de header sunt opționale.

Atunci când un client trimite e-mail, procesul SMTP client se conectează cu un proces SMTP server pe portul bine cunoscut 25. După realizarea conexiunii, clientul încearcă să trimită e-mailul serverului prin intermediul conexiunii. Atunci când serverul primește mesajul, fie plasează mesajul în contul local, dacă destinația este locală, fie îl transmite mai departe folosind procesul de conexiune SMTP la alt server de mail pentru livrare.

Serverul de e-mail destinație ar putea să nu fie online sau să fie ocupat atunci când mesajele de e-mail sunt transmise. Prin urmare, mesajele SMTP sunt într-o "coadă" pentru a fi transmise mai târziu. Periodic, serverul verifică coada pentru mesaje și încearcă să le trimită din nou. Dacă mesajul nu este livrat după o perioadă predefinită de timp, se întoarce la expeditor ca nelivrabil.

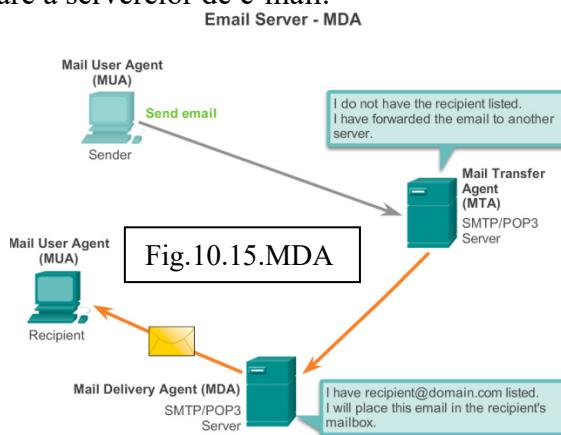


Post Office Protocol (POP) permite unei stații de lucru să preia mailul de pe un server de mail. Cu POP, mailul este descărcat de la server la client și apoi este șters de pe server.

Serverul activează serviciul POP prin “ascultarea pasivă” pe portul 110 TCP pentru cereri de conexiune de la client. Atunci când un client vrea să folosească serviciul, trimite o cerere de stabilire a unei conexiuni TCP la server. Atunci când conexiunea este stabilită, serverul POP transmite “un salut”. Clientul și serverul POP apoi schimbă comenzi și răpsunsuri între ele până când conexiunea este închisă sau abandonată.

Deoarece mesajele de e-mail sunt descărcate pe client și șterse de pe server, nu există o locație centralizată unde mesajele de e-mail sunt păstrate. Deoarece POP nu stocă mesajele, nu este preferat pentru afacerile mici ce necesită o soluție centralizată de backup.

POP3 este preferat pentru un ISP, deoarece ameliorează responsabilitatea de gestionare a unei cantități mari de stocare a serverelor de e-mail.

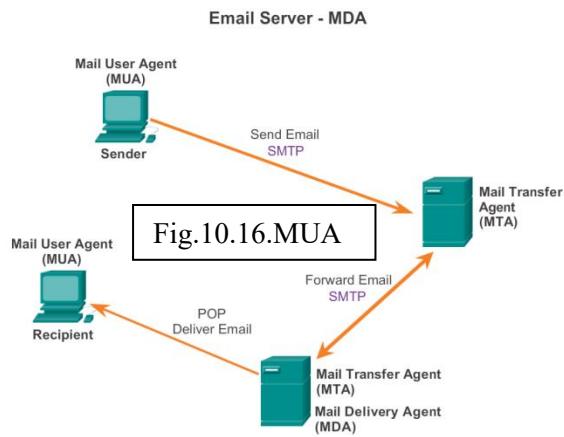


Internet Message Access Protocol (IMAP) este un alt protocol ce descrie o metodă de primire a mesajelor de e-mail. Însă, spre deosebire de POP, atunci când un utilizator se conectează la un server IMAP, copii ale mesajelor sunt descărcate pe aplicația client. Mesajele originale sunt păstrate în server până când sunt șterse manual. Utilizatorii vizualizează copii ale mesajelor pe softwareul lor de client e-mail.

Utilizatorii pot crea o ierarhie de fișiere pe server pentru a organiza și stoca mailul. Acea structură de fișier este duplicată pe clientul e-mail. Atunci când un utilizator decide să șteargă un mesaj, serverul sincronizează acțiunea și șterge mesajul de pe server.

Pentru afacerile mici și mijlocii, există multe avantaje ale utilizării IMAP. IMAP poate oferi stocare pe termen lung a mesajelor de e-mail pe serverele de mail și permite backup centralizat. Permite de asemenea angajaților să acceseze mesajele de e-mail de pe locații multiple, folosind dispozitive diferite sau softwareuri diferite de client. Structura folderului de mailbox pe care un utilizator se așteaptă să o vadă este disponibilă indiferent de modul în care utilizatorul accesează mailbox.

Pentru un ISP, IMAP ar putea să nu fie protocolul preferat. Poate fi scump pentru achiziționare și gestionare a spațiului de disk astfel încât să suporte un număr mare de e-mailuri stocate. În plus, dacă clienții doresc ca mailboxes să fie susținute în mod curent, poate crește costurile ISP.



## 10.6 Furnizarea Serviciilor de Adresare IP

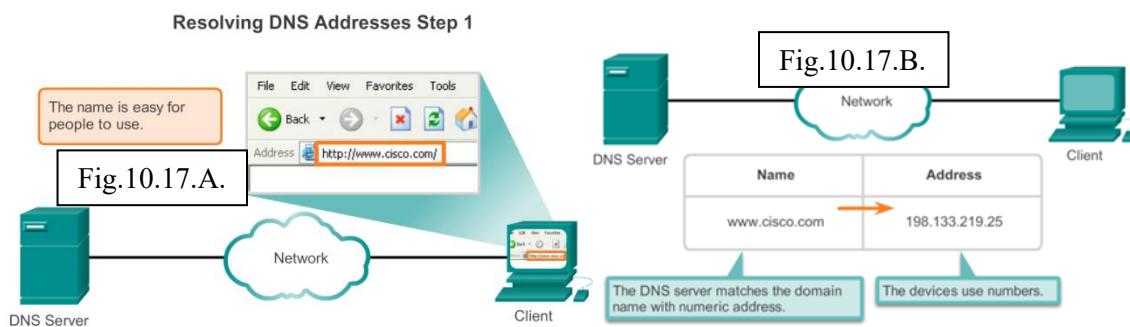
În rețelele de date, dispozitivele sunt etichetate cu adrese IP numerice pentru a transmite și primi date peste rețele. Mulți oameni nu pot ține minte aceste adrese numerice. Numele de domeniu au fost create pentru convertirea adresei numerice într-un nume simplu, recunoscut.

În Internet, aceste nume de domeniu, cum ar fi <http://www.fmi.unibuc.ro>, sunt mult mai ușor de reținut pentru oameni decât **198.133.219.25**, ce este adresa numerică reală pentru acest server. Dacă facultatea decide să schimbe adresa numerică a [www.fmi.unibuc.ro](http://www.fmi.unibuc.ro), este transparentă utilizatorului deoarece numele de domeniu rămâne la fel. Noua adresă este legată simplu la numele de domeniu existent și conectivitatea este menținută. Atunci când rețelele erau mici, era simplă sarcina de gestiune a mapării dintre numele de domeniu și adresele reprezentate. O dată cu creșterea rețelelor și a numărului de dispozitive, sistemul manual a devenit inaplicabil.

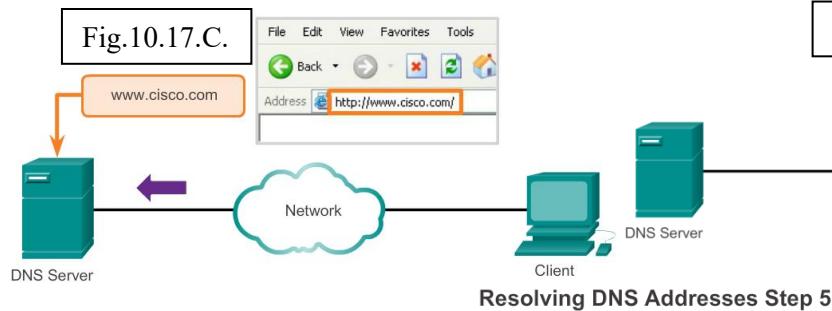
Domain Name System (DNS) a fost creat pentru ca numele de domeniu să fie asociat acestor rețele. DNS folosește un set distribuit de servere pentru a rezolva numele asociate cu adresele numerice.

Protocolul DNS definește un serviciu automat ce asociază numele resurselor cu adresele de rețea numerice necesare. Include formatul pentru cerere, răspuns și date. Comunicațiile protocolului DNS folosesc un singur format numit mesaj. Acest format de mesaj este folosit pentru toate tipurile de cereri de client și răspunsuri de server, mesaje de eroare și transferul de informații ale resurselor dintre servere.

Fig. 10.17.A /Fig.10.17.E prezintă pașii pentru rezolvarea cererilor DNS.

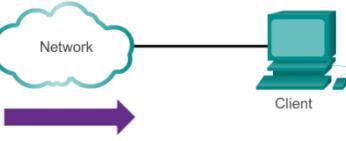


Resolving DNS Addresses Step 3



Resolving DNS Addresses Step 4

Fig.10.17.D.



Resolving DNS Addresses Step 5

Fig.10.17.E.



Un server DNS oferă rezoluție de nume folosind *Berkeley Internet Name Domain* (BIND), sau daemon de nume. BIND a fost dezvoltat inițial de către patru studenți ai University of California Berkley la începutul anilor 1980. Așa cum se poate observa în Fig.10.17C , formatul mesajului DNS folosit de către BIND este cel mai cunoscut format DNS din Internet.

Serverul DNS stochează diferite tipuri de înregistrări de resurse folosite pentru rezolvarea numelor. Aceste înregistrări conțin nume, adresa și tipul de înregistrare.

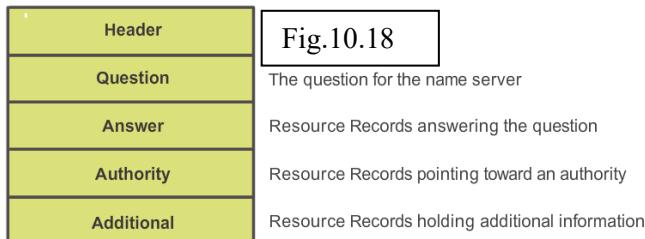
Unele dintre aceste înregistrări sunt:

- **A – O adresă de dispozitiv final.**
- **NS – Un nume de server autoritar.**
- **CNAME – Un nume canonic (sau Fully Qualified Domain Name) pentru un alias; folosit atunci când mai multe servicii au o singură adresă de rețea, însă fiecare serviciu are propria intrare în DNS.**
- **MX - Mail exchange record; mapează un nume de domeniu la o listă de servere de mail exchange pentru respectivul domeniu.**

Atunci când un client face o interogare, procesul BIND al serverului se uită mai întâi în propriile înregistrări pentru a rezolva numele. Dacă nu este capabil să rezolve numele folosind propriile înregistrări stocate, contactează alte servere pentru rezolvarea numelui.

Cererea ar putea să fie pasată mai multor server, ceea ce poate lua mai mult timp și poate consuma lățime de bandă. După ce se găsește o potrivire și este transmisă la serverul original, serverul stochează temporar adresa ce corespunde numelui în memoria cache.

Dacă același nume este cerut din nou, primul server poate returna adresa prin folosirea valorii stocate în name cache. Caching reduce traficul de cerere de date DNS și volumul de lucru al serverelor de pe nivelele de mai sus din ierarhie. Serviciul Client DNS de pe PCurile Windows optimizează performanța rezoluției numelui DNS prin stocarea numelor rezolvate anterior în memorie. Comanda **ipconfig /displaydns** afișează toate înărările DNS stocate pe un sistem PC cu sistem de operare Windows.



Protocolul DNS folosește un sistem ierarhic pentru a crea o bază de date spre a oferi rezoluție de nume. Ierarhia arată ca un copac întors cu rădăcina în vârf și ramurile în jos. DNS folosește numele de domeniu pentru a forma ierarhia.

Structura de nume este împărțită în zone mici, gestionabile. Fiecare server DNS menține un fișier specific de baze de date și este responsabil numai de gestionarea mapărilor de nume în IP pentru respectiva mica parte a întregii structuri DNS. Atunci când un server DNS primește o cerere de traducere de nume ce nu corespunde zonei sale DNS, serverul DNS transferă cererea la alt server DNS din zona adecvată pentru traducere.

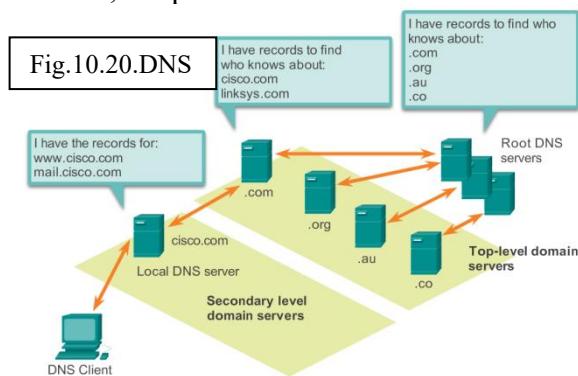
**Notă:** DNS este scalabil deoarece rezoluția de hostname este împărțită mai multor servere.

Domeniile diferite de la nivelul de sus reprezintă fie tipul de organizație, fie țara de origine. Exemple de domenii de la nivelul superior sunt:

- **.au – Australia.**
- **.co – Colombia.**
- **.com – o afacere sau industrie.**
- **.jp – Japan.**
- **.org – o organizație non-profit.**

După domeniile de la nivelul superior sunt numele de domeniu de nivel doi, iar sub ele sunt alte domenii de nivele inferioare. Fiecare nume de domeniu este o cale în jos în copac începând de la rădăcină. De exemplu, aşa cum se poate vedea în Fig. , serverul rădăcină DNS ar putea să nu știe exact unde înregistrarea pentru serverul de e-mail, **mail.fmi.unibuc.ro**, este localizată, însă menține o înregistrare pentru domeniul .ro în domeniul de nivel înalt. Deci, serverele din domeniul .ro ar putea să nu aibă o înregistrare pentru **mail.fmi.unibuc.ro**, însă au o înregistrare pentru domeniu. Serverele din domeniul **fmi.unibuc.ro** au o înregistrare (o înregistrare MX mai precis) pentru **mail.fmi.unibuc.ro**.

DNS se bazează pe ierarhia de servere descentralizate pentru a stoca și menține aceste înregistrări de resurse. Înregistrările de resurse listeză numele de domeniu pe care serverul le poate rezolva și serverele alternative ce pot procesa cererea. Dacă un server dat are înregistrările de resurse ce corespund nivelului sau de domeniului din ierarhie, se numește autoritar pentru respectivele înregistrări. De exemplu, un server de nume din domeniul **cisco.netacad.net** nu va fi autoritar pentru înregistrarea **mail.cisco.com** deoarece respectiva înregistrare este ținută într-un server de nivel de domeniu ridicat; în special serverul de nume din domeniul **cisco.com**.



DNS este un serviciu client/server; însă, diferă de alte servicii client/server. Pe când alte servicii folosesc un client ce este o aplicație (cum ar fi web browser, e-mail client), clientul DNS rulează ca un serviciu de sine stătător. Clientul DNS, uneori numit DNS resolver, suportă rezoluția de nume pentru alte aplicații de rețea și alte servicii ce au nevoie de el.

La configurația unui dispozitiv de rețea, în general oferim una sau mai multe adrese de server DNS pe care clientul DNS le poate folosi pentru rezoluția de nume. În mod normal ISP oferă adresele folosite pentru serverele de DNS. Atunci când o aplicație de utilizator cere să se conecteze la un dispozitiv de la distanță în funcție de nume, clientul DNS cere unuia dintre serverele de nume să rezolve numele într-o adresă numerică.

Sistemele de operare de pe computer au de asemenea un utilitar numit **nslookup** ce permite utilizatorului să ceară manual serverelor de nume să rezolve un hostname dat. Acest utilitar poate fi utilizat de asemenea pentru depanarea problemelor de rezoluție de nume și pentru identificarea stării curente ale serverelor de nume.

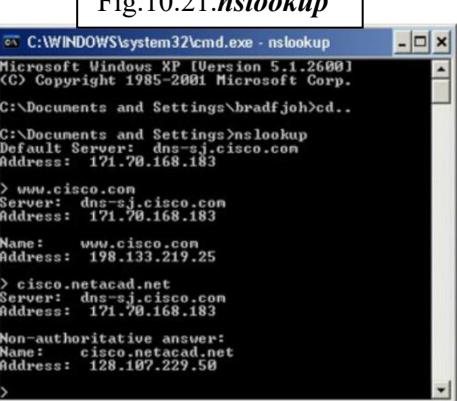
În Fig. , atunci când comanda **nslookup** este efectuată, serverul DNS implicit config.t pentru hostul respectiv este afișat. În acest exemplu, serverul DNS este dns.google.com ce are adresa 8.8.8.8.

Numele unui host sau domeniu poate fi introdus în promptul **nslookup**. În prima interogare din Fig. , o cerere este efectuată pentru **fmi.unibuc.ro**. Serverul de nume oferă adresa 198.133.219.25.

Interrogările arătate în Fig. sunt numai teste simple. Utilitarul **nslookup** are multe opțiuni disponibile pentru testare și verificare extinsă a procesului DNS. La final, introducem exit pentru a ieși din utilitarul **nslookup**.

Using nslookup

Fig.10.21.nslookup



Serviciul Dynamic Host Configuration Protocol (DHCP) permite dispozitivelor dintr-o rețea să obțină adresele IP și alte informații de la un server DHCP. Acest serviciu automatizează atribuirea adreselor IP, măștilor de rețea, de gateway și alți parametrii de rețea. Aceasta se numește adresare dinamică. Alternativa adresării dinamice, este adresarea statică. La utilizarea adresării statice, administratorul de rețea introduce manual informațiile de adresă IP pe hosturile de rețea.

DHCP permite unui host să obțină o adresă IP în mod dinamic atunci când se conectează la rețea. Serverul DHCP este conectat și o adresă este cerută. Serverul DHCP alege o adresă dintr-un range de adrese configurate numit pool și o atribuie ("încărca") hostului pentru o perioadă de timp setată.

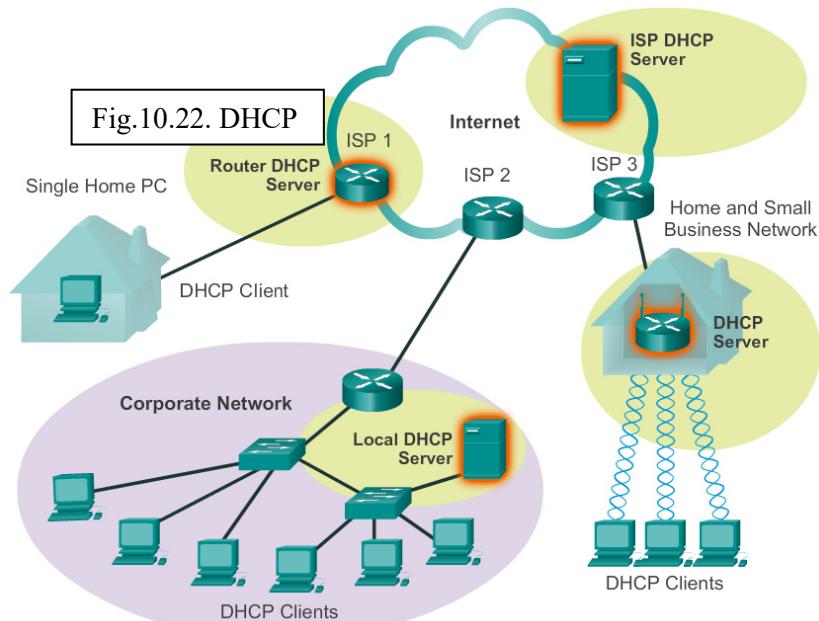
În rețelele locale mari, sau unde populația de utilizatori se schimbă frecvent, DHCP este preferat pentru atribuirea adreselor. Utilizatorii noi pot veni cu laptopuri și obțin o conexiune; alții utilizatori pot avea stații de lucru noi ce trebuie să fie conectate. În loc ca administratorul de rețea să atribuie adrese pentru fiecare stație de lucru, mult mai eficient este ca atribuirea de adrese IP să se facă automat prin DHCP.

Adresele distribuite prin DHCP nu sunt atribuite permanent hosturilor, ci sunt numai închiriate pentru o anumită perioadă de timp. Dacă hostul nu mai este alimentat sau este înălăturat din rețea, adresa se întoarce în pool pentru reutilizare. Acest lucru este util în mod special pentru utilizatorii mobili ce vin și pleacă din rețea. Utilizatorii pot să se mute frecvent dintr-o locație în alta și să restabilească conexiuni la rețea. Hostul poate obține o adresă IP după ce este efectuată conexiunea hardware, printr-un LAN wireless sau cablat.

DHCP face posibilă conectarea la Internet prin wireless în aeroporturi sau cafenele. Atunci când un dispozitiv wireless intră într-un hotspot, dispozitivul client DHCP contactează serverul DHCP local printr-o conexiune wireless și serverul DHCP atribuie o adresă IP dispozitivului.

Așa cum se vede în imagini, mai multe tipuri de dispozitive pot fi servere DHCP atunci când rulează un serviciu software DHCP. Serverul DHCP din cele mai multe rețele mijlocii spre mari este de obicei un server local dedicat bazat pe PC. În rețelele de domiciliu, serverul DHCP este localizat pe routerul local ce conectează rețeaua de domiciliu la ISP. Hosturile locale primesc informații de adresă IP direct de la routerul local. Routerul local primește o adresă IP de la serverul DHCP de la ISP.

DHCP poate presupune un risc de securitate deoarece orice dispozitiv conectat la rețea poate primi o adresă. Riscul face securitatea fizică un factor determinant al alegerii între a utiliza adresarea manuală sau statică. Ambele, adresarea manuală și cea statică, au loc în designul de rețea. Multe rețele folosesc ambele forme de adresare, adresarea manuală și cea statică. DHCP este folosit pentru hosturi de uz general, cum ar fi dispozitivele utilizatorului final; adresarea statică este folosită pentru dispozitive de rețea, cum ar fi gateways, switches, servere și imprimante.



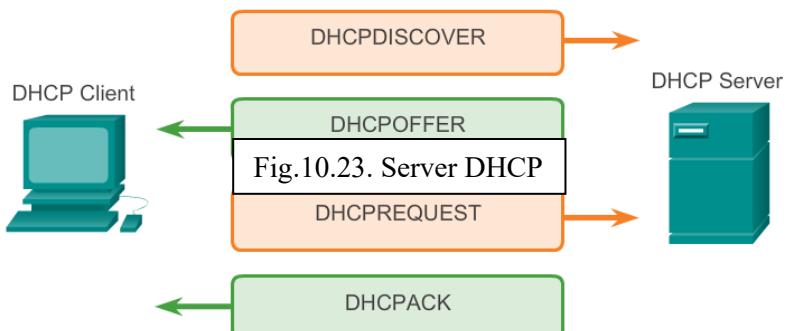
Fără DHCP, utilizatorii trebuie să introducă manual adresa IP, masca de rețea și alte setări de rețea pentru a se conecta la rețea. Serverul DHCP menține un pool de adrese IP și închiriază o adresă oricărui client activat DHCP atunci când clientul este alimentat. Deoarece adresele IP sunt mai degrabă diminice (închiriate) decât statice (permanent atribuite), adresele care nu mai sunt folosite sunt automat returnate în pool pentru realocare. Așa cum se vede în Fig. , atunci când un dispozitiv activat DHCP se conectează la rețea, clientul trimite un mesaj broadcast de descoperire DHCP (DHCPDISCOVER) pentru a identifica orice server DHCP disponibil în rețea. Un server DHCP răspunde cu un mesaj de ofertă DHCP (DHCPOFFER), ce oferă o închiriere clientului.

Mesajul oferit conține adresa IP și masca de rețea ce vor fi atribuite, adresa IP a serverului DNS și adresa IP a default gateway. Oferta de închiriere include de asemenea și durata închirierii.

Clientul ar putea primi mai multe mesaje DHCP OFFER dacă există mai multe servere DHCP în rețeaua locală; prin urmare, trebuie să aleagă între ele și trimite un mesaj de cerere DHCP (DHCP REQUEST) ce identifică serverul explicit și oferta de închiriere pe care clientul a acceptat-o. Un client ar putea alege de asemenea să ceară o adresă pe care a mai avut-o alocată de către server.

Presupunând că adresa IP cerută de către client sau oferită de către server este încă disponibilă, serverul întoarce un mesaj de confirmare DHCP (DHCP ACK) ce confirmă clientului faptul că închirierea este finalizată. Dacă oferta nu mai este disponibilă, poate datorită faptului că alt client a închiriat-o sau a apărut timeout, serverul selectat răspunde cu un mesaj de confirmare negativ DHCP (DHCP NAK). Dacă un mesaj DHCP NAK este întors, procesul de selecție trebuie să înceapă din nou cu un nou mesaj DHCP DISCOVER transmis. După ce clientul a închiriat, trebuie să reînnoiască cererea înainte de expirarea închirierii, printr-un alt mesaj DHCP REQUEST.

Serverul DHCP asigură faptul că toate adresele IP sunt unice (aceeași adresa IP nu poate fi atribuită la două dispozitive diferite din rețea simultan). Folosirea DHCP, permite administratorilor de rețea să reconfigureze ușor adresele IP de client fără a face schimbări manuale asupra clientilor. Multii furnizori de Internet folosesc DHCP pentru alocarea adreselor clientilor lor ce nu necesită o adresă statică.



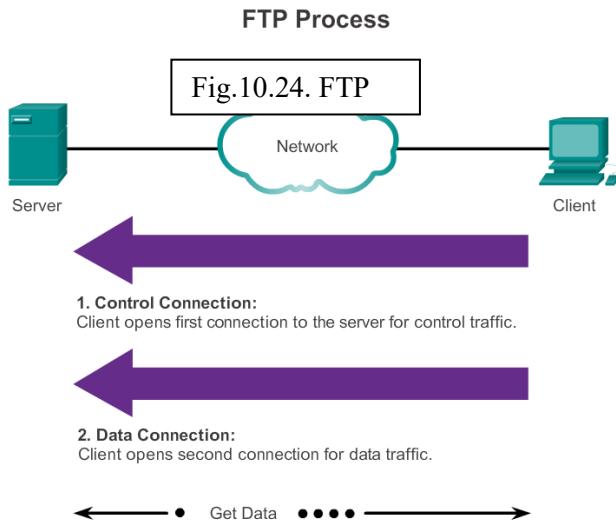
### 10.7 Furnizarea Serviciilor de Partajare de Fișiere

File Transfer Protocol (FTP) este alt protocol comun de la nivelul aplicație. FTP a fost dezvoltat pentru a permite transferul de date dintre un client și un server. Un client FTP este o aplicație ce rulează pe un computer și este utilizată pentru a trimite și a prelua date de la un server ce rulează un FTP daemon (FTPD).

Așa cum se poate observa și în Fig. , pentru un transfer cu succes al datelor, FTP necesită două conexiuni între client și server, una pentru comenzi și răspunsuri, alta pentru transferul real de fișiere:

- Clientul stabilește prima conexiune cu serverul pentru traficul de control, constând din comenzi de client și răspunsuri ale serverului.
- Clientul stabilește a doua conexiune cu serverul pentru transferul real de fișiere. Această conexiune este creată de fiecare dată când există date de transmis.

Transferul datelor poate avea loc în ambele direcții. Clientul poate descărca date de la server sau clientul poate încărca date pe server.



Server Message Block (**SMB**) este un protocol client-server de partajare de fișiere, dezvoltat de IBM la sfârșitul anilor 1980 pentru a descrie structura resurselor de rețea partajate, cum ar fi directoare, fișiere, imprimante și porturi seriale. Este un protocol cerere-răspuns.

Protocolul **SMB** descrie accesul la sistemul de fișier și modul în care clienții pot cere fișiere. Descrie de asemenea comunicarea interproceselor de protocol **SMB**. Toate mesajele **SMB** împart un format comun. Acest format folosește un header de dimensiune fixă, urmat de un parametru de dimensiune variabilă și de componentă de date.

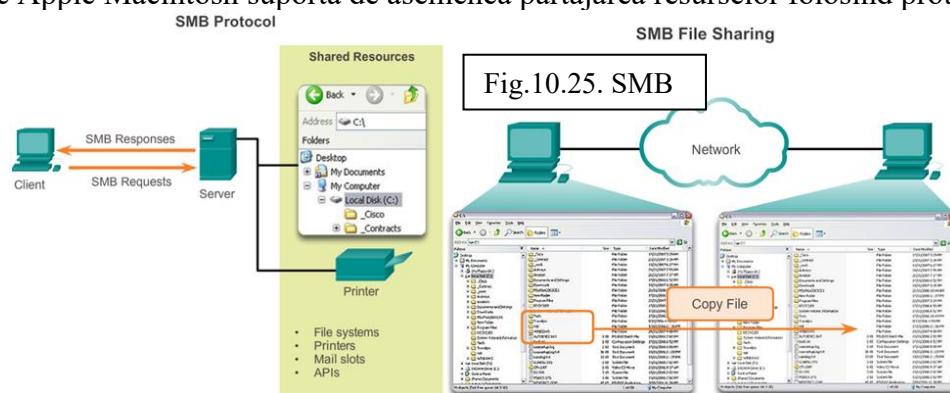
Mesajele **SMB** pot:

- Începe, autentifica și termină sesiuni.
- Controla accesul la fișier și imprimantă.
- Permite unei aplicații să trimită sau să primească mesaje la sau de la un alt dispozitiv.

Serviciile **SMB** de printare și partajare de fișier au devenit sprijinul principal al rețelisticiei Microsoft. O dată cu introducerea seriilor software Microsoft 2000, Microsoft a schimbat structura de bază a utilizării **SMB**. În versiunile anterioare ale produselor Microsoft, serviciile **SMB** foloseau un protocol non-TCP/IP pentru implementarea rezoluției de nume. Începând cu Windows 2000, toate produsele ulterioare folosesc DNS, ce permite ca protocolele TCP/IP să suporte direct partajarea de resursă **SMB**, aşa cum se vede în Fig. 1. Procesul de schimb de fișier **SMB** dintre PC-urile Windows este evidențiat în Fig. 2.

Spre deosebire de partajarea de fișier suportată de File Transfer Protocol (FTP), clienții stabilesc o conexiune pe termen lung cu serverele. După ce este stabilită conexiunea, utilizatorul client poate accesa resursele de pe server ca și cum ar fi local, pe hostul client.

Sistemele de operare LINUX și UNIX oferă de asemenea o metodă de partajare a resurselor cu rețelele Microsoft, folosind o versiune a **SMB** numită SAMBA. Sistemele de operare Apple Macintosh suportă de asemenea partajarea resurselor folosind protocolul **SMB**.



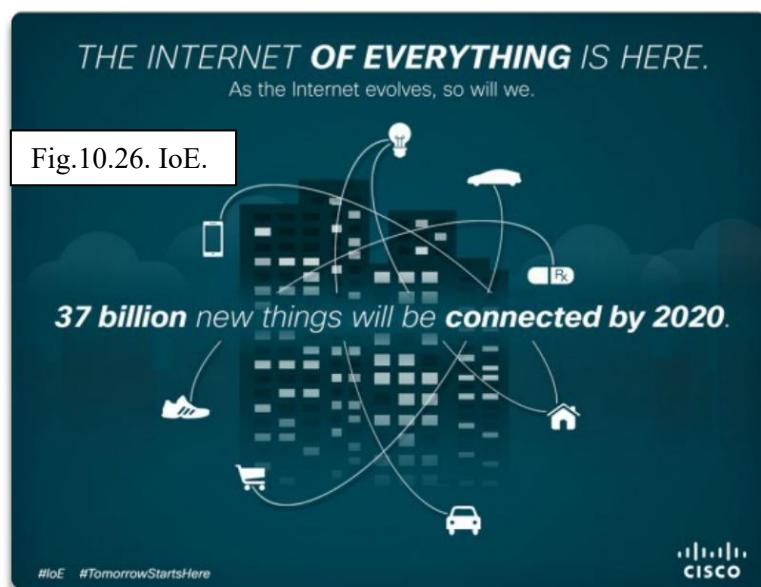
## 10.8 Mesajele pot fi auzite în întreaga lume

Nivelul aplicație este responsabil de accesarea directă a proceselor de bază pe care le gestionează și de livrarea comunicațiilor prin rețea. Acest nivel servește ca sursă și destinație a comunicațiilor peste rețele, indiferent de tipul de rețea de date utilizat. De fapt, progresele cu privire la modului în care ne conectăm la rețea au un efect direct al tipului de aplicații dezvoltate.

Tendințe ca Bring Your Own Device (BYOD), accesul de oriunde, virtualizarea și conexiunile machine-to-machine (m2m) au făcut loc unui noi tip de aplicații. Este estimat că aproximativ 50 miliarde de dispozitive să fie conectate în 2020. Numai în anul 2010, mai mult de 350.000 de aplicații au fost dezvoltate cu mai mult de trei milioane de descărări. Toate acestea conduc la o lume de conexiuni intuitive între oameni, procese, date și lucruri din rețea.

Utilizarea de smart-tagging și conectivitate avansată pentru a digitaliza produsele neinteligente – de la biciclete la sticle, frigidere și mașini – și conectarea lor la Internet, vor permite oamenilor și companiilor să interacționeze în moduri noi și aproape inimaginabile. Obiectele vor fi capabile să primească și să transmită informații utilizatorilor și altor obiecte conectate. Așa cum se poate vedea în Fig. , acest nou val din dezvoltarea Internetului este numit **Internet of Things** !

Peste 100 de milioane de automate, autoturisme, alarme de fum și alte dispozitive deja împart informații automat în zilele noastre, Fig. prezintă analiștii de market de la **Berg Insight** care se asteaptă să crească la 360 de milioane până în 2016. Astăzi, fotocopiatore cu un modul M2M pot cere automat hârtie și toner nou sau pot alerta tehnicienii de un defect – chiar și să le spună ce piese să aducă.



Explozia masivă de aplicații se datorează în mare parte geniului de abordare pe nivele pentru procesarea datelor dintr-o rețea. Mai exact, păstrarea funcionalității nivelului aplicație separat de funcționarea transportului de date, permite ca protocolele de la nivelul aplicație să fie schimbată și noi aplicații să fie dezvoltate, fără ca dezvoltatorul să se îngrijoreze de mecanismele de transport al datelor peste Internet. Aceasta este funcția altor nivele și prin urmare, altor dezvoltatori.

Așa cum se arată în Fig. , atunci când o aplicație trimie o cerere la o aplicație server, mesajul este construit de către nivelul aplicație, însă pasează datele la funcționalitățile altor nivele de pe client pentru livrare. În călătoria lor prin stivă, fiecare nivel încapsulează datele cu un header ce conține protocolele de comunicație pentru respectivul nivel. Aceste protocole, ce

sunt implementate și pe hostul sursă și pe cel destinație, interacționează pentru a oferi livrare end-to-end a aplicațiilor peste rețea.

Protocoloale ca HTTP, de exemplu, suportă livrarea paginilor web la dispozitivele finale. Acum că am învățat toate nivelele și funcțiile lor diferite, putem urmări o cerere de client a unei pagini web de la serverul web pentru a vedea cum aceste funcționalități independente lucrează împreună.

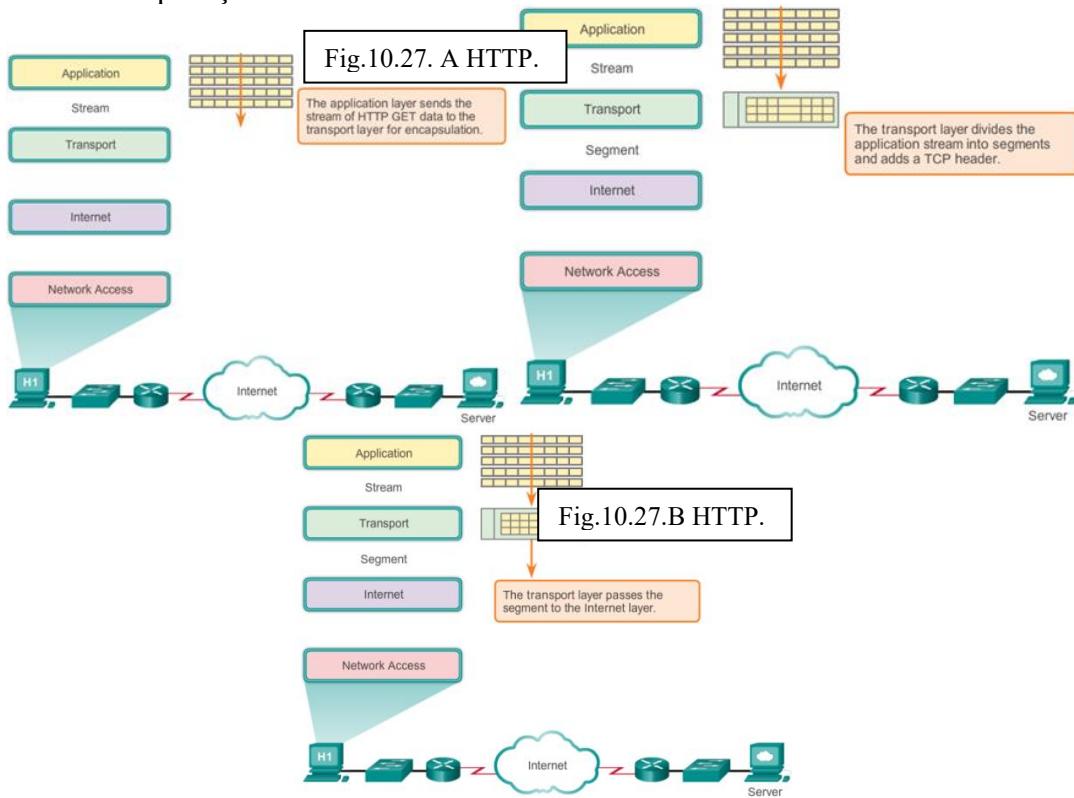
Folosind modelul TCP/IP, un proces complet de comunicare include șase pași:

### **PASUL 1. Crearea datelor**

Primul pas este crearea datelor la nivelul aplicație al dispozitivului sursă. În acest caz, După lansarea cererii clientului web, cunoscută ca HTTP **GET**, datele respective vor fi codate, comprimate și criptate dacă este necesar. Aceasta este sarcina protocolului nivelului aplicație din modelul TCP/IP – însă acesta include funcționalitatea descrisă de către nivelele aplicație, prezentare și sesiune ale modelului OSI. Nivelul aplicație trimite datele ca un stream la nivelul transport.

### **PASUL 2. Segmentarea și încapsularea inițială**

Următorul pas este segmentarea și încapsularea datelor în parcurgerea stivei de protocole. La nivelul transport, mesajul HTTP **GET** va fi împărțit în mai multe piese mai mici și ușor gestionabile și fiecare parte a mesajului va avea un header de nivel transport atașată ei. În interiorul headerului de la nivelul transport sunt indicatori a modului în care se poate reconstrui mesajul. Include de asemenea și un identificator, număr de port 80. Acesta este utilizat pentru a spune serverului destinație că mesajul este destinat pentru aplicația de server web. Un port sursă generat aleator este adăugat pentru a asigura faptul că și clientul poate urmări comunicația și că va fi transmisă la aplicația client corectă.



### PASUL 3. Adresarea

La acest pas, identificatori de adresă sunt adăugați la segmente. Așa cum aici sunt mai multe nivele de protocole ce preparam datele pentru transferul la destinație, există mai multe nivele de adresare pentru asigurarea livrării. Rolul nivelului rețea este adăugarea adresării ce permite transferul de date de la hostul sursă la hostul ce le va utiliza. Nivelul rețea realizează acest lucru prin încapsularea fiecărui segment cu un header de pachet IP. Headerul de pachet IP conține adresele IP ale dispozitivelor sursă și destinație (adresa IP a dispozitivului destinație este de obicei determinată printr-un proces anterior numit *domain name service*). Combinarea de adresă sursă IP și destinație cu numărul de port sursă și destinație se numește socket. Socketul este folosit pentru a identifica serverul și serviciul cerut de către client.

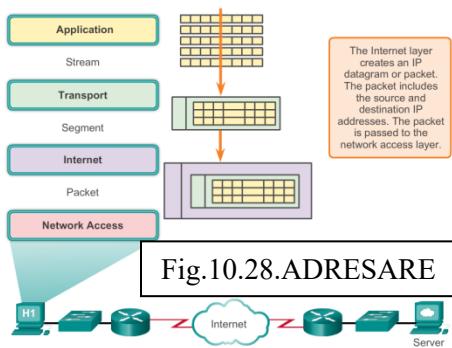


Fig.10.28.ADRESARE

### PASUL 4. Pregătirea pentru transport

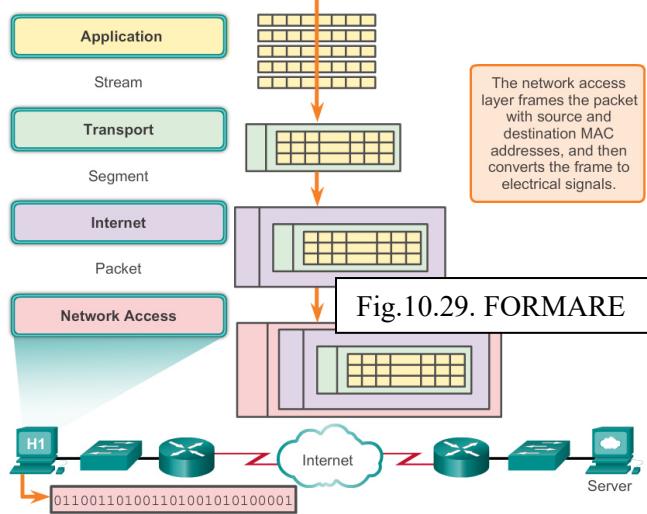
După ce este adăugată adresa IP, pachetul este pasat la nivelul acces la rețea pentru generarea datelor pe mediu, așa cum se poate vedea în Fig. . Pentru a se realiza acest lucru, nivelul acces la rețea trebuie mai întâi să pregătească pachetul pentru transmisie prin plasarea lui într-un frame cu un header și trailer. Acest frame include adresa de host fizică a sursei, adresa fizică a next hop din calea sa spre destinație. Acest lucru este echivalent cu funcționalitatea nivelului 2, sau nivelul legătură de date, a modelului OSI. Nivelul 2 se ocupă de livrarea mesajelor pe o rețea locală. Adresa de nivel 2 este unică pe rețea locală și reprezintă adresa dispozitivului final pe mediul fizic. Într-un LAN ce folosește Ethernet, această adresă se numește adresa Media Access Control (MAC). O dată ce nivelul acces la rețea a pregătit frameul cu adresele sursă și destinație, codifica frameul în biți și apoi în impulsuri electrice sau luminoase ce sunt transmise pe mediul de comunicație.

### PASUL 5. Transportul datelor

Datele sunt transportate prin internetwork, ce constă din mediu și orice dispozitive intermediare. În drumul mesajului încapsulat prin rețea poate parcurge mai multe tipuri de medii de comunicație diferite de rețea. Nivelul de acces la rețea specifică tehnicele de plasare și scoatere a frameului din fiecare mediu, cunoscute ca metode de control al accesului la mediu.

Dacă hostul destinație se află în aceeași rețea cu hostul sursă, pachetul este livrat între cele două hosturi din mediul local fără nevoie unui router. Însă, dacă hostul destinație și cel sursă se află în rețele diferite, pachetul ar putea traversa mai multe rețele, pe mai multe tipuri diferite de medii de comunicație, peste mai multe routere. În călătoria prin rețea, informațiile conținute în frame nu sunt alterate.

La limita fiecărei rețele locale, un dispozitiv de rețea intermediar, de obicei un router, decapsulează frameul pentru a citi adresa de host destinație conținută în headerul pachetului. Routerele folosesc partea de identificator de rețea a adresei pentru a determina ce cale să folosească pentru a ajunge la hostul destinație. O dată ce este determinată calea, routerul încapsulează pachetul într-un nou frame și îl trimită la următorul hop din drumul spre dispozitivul destinație.



#### PASUL 6. Livrarea datelor la aplicația destinație corectă

La final, la dispozitivul destinație, frameul este primit. Decapsularea și reasamblarea datelor are loc, în timpul călătoriei datelor în stiva dispozitivului destinație. Datele sunt pasate continuu la nivelele superioare, de la nivelul de acces la rețea la nivelul rețea, nivelul transport până la nivelul aplicație unde vor fi procesate.

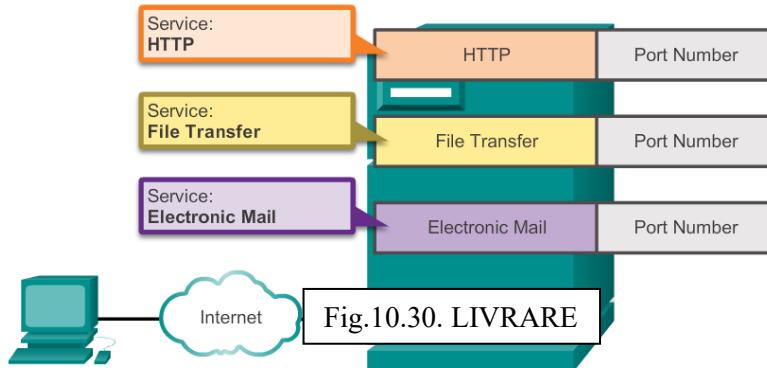
*Dar cum poate fi sigur dispozitivul că este identificat procesul aplicație correct ?*

Așa cum se poate vedea în Fig. , reamintim faptul că la nivelul transport, informațiile conținute în headerul PDU identifică procesul specific sau serviciul ce rulează pe dispozitivul de host destinație ce va procesa datele. Hosturile, fie că sunt clienți, fie că sunt servere din Internet, pot rula mai multe aplicații de rețea simultan. Oamenii ce utilizează PCuri au de obicei un client de e-mail ce rulează în același timp cu un browser web, program de mesagerie instant, streaming media și eventual un joc. Toate aceste programe ce rulează separat sunt exemple de procese individuale.

Vizualizarea unei pagini web implică cel puțin un proces de rețea. Apasarea pe un hyperlink face ca un browser web să comunice cu un server web. În același timp, în background, un client de e-mail poate transmite și primi e-mailuri și un coleg sau prieten poate trimite un mesaj instant.

Să ne imaginăm un computer ce are numai o interfață de rețea conectată la el. Toate fluxurile de date create de către aplicațiile ce rulează pe un PC intră și ies prin respectiva interfață, însă mesajele instant nu apar în mijlocul documentelor de procesor word, iar e-mailurile nu apar într-o interfață a unui joc.

Acest lucru se datorează faptului că procesele individuale ce rulează pe hosturile sursă și destinație comunică între ele. Fiecare aplicație sau serviciu este reprezentat la nivelul 4 printr-un număr de port. Un dialog unic între dispozitive este identificat cu o pereche de numere de port destinație și sursă de nivel 4, reprezentative pentru cele două aplicații ce comunică. Atunci când datele sunt primite pe host, numărul de port este examinat pentru a determina ce aplicație sau proces este optimă pentru date.



O resursă distractiva pentru a te ajuta să vizualizezi conceptele de rețea este filmul de animație "Warriors of the Net" de la TNG Media Lab. Înainte vizualizării video, există câteva lucruri de luat în calcul. Mai întâi, în termenii conceptelor invătate în acest capitol, gandeste-te când în video este într-un LAN sau WAN sau intranet sau Internet; ce sunt dispozitivele finale în comparație cu cele intermediare; cum modelele OSI și TCP/IP se aplică; ce protocoale sunt implicate.

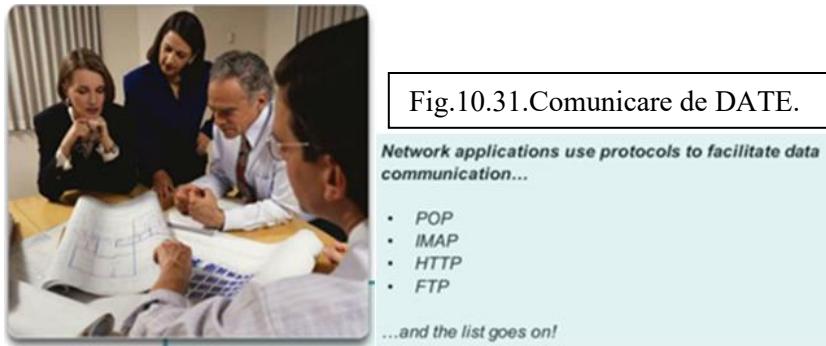
Apoi, pe când numerele de port 21, 23, 25, 53 și 80 ne sunt referite explicit în video, adresele IP sunt referite implicit – poti vedea unde? Unde în video ar putea fi implicate adresele MAC?

La sfârșit, deoarece toate animatiile au de obicei simplificări în ele, există o eroare în video. Pe la minutul 5, afirmația "What happens when Mr. IP doesn't receive an acknowledgement, he simply sends a replacement packet." este făcută. Aceasta nu este o funcție a nivelului 3 IP, ce este un protocol de livrare best effort și unreliable, ci mai degrabă o funcție a protocolului TCP de nivel transport.

Descarcati video de la <http://www.warriorsofthe.net>.



## 10.9 Concluzii Capitolul 10



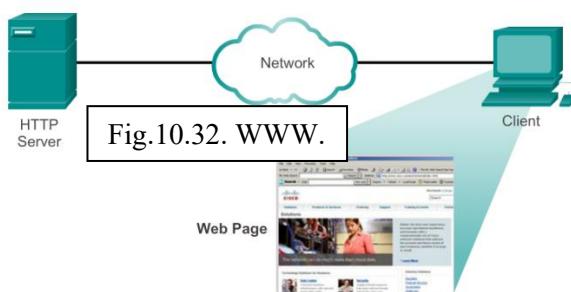
Nivelul aplicație este responsabil de accesarea directă a proceselor de bază pe care le gestionează și livrarea comunicațiilor din rețeaua umană. Acest nivel servește ca sursă și destinație a comunicațiilor din rețelele de date. Aplicațiile, serviciile și protocoalele nivelului aplicație permit utilizatorilor să interacționeze cu rețeaua de date într-un mod ce are înțeles și eficiență.

- *Aplicațiile sunt programe de computer în care utilizatorul interacționează și care inițiază procesul de transfer de date la cererea utilizatorului.*
- *Serviciile sunt programe de background ce oferă conexiune între nivelul aplicație și nivelele inferioare ale modelului de rețea.*
- *Protocoalele oferă o stivă de reguli stabilite și procese ce asigură serviciile ce rulează pe un dispozitiv particular ce poate trimite și primi date de la un rangu de dispozitive de rețea diferite.*

Livrarea de date peste rețea poate fi cerută de la un server de către un client sau între dispozitive ce funcționează într-un aranjament **p2p**, unde relația client/server este stabilită, în funcție de ce dispozitiv este destinație și sursă la un moment dat. Mesajele sunt schimbate între serviciile de nivel aplicație de pe fiecare dispozitiv final în concordanță cu specificațiile de protocol stabilite și de utilizarea acestor relații.

Protocoale cum ar fi HTTP, de exemplu, suportă livrarea paginilor web la dispozitivele finale. SMTP și POP suportă trimiterea și primirea de e-mail. **SMB** și FTP permit utilizatorilor să împartă fișiere. Aplicațiile P2P fac mai ușor pentru consumatori să împartă mediul într-un mod distribuit. DNS rezolvă numele lizibile umane utilizate pentru referirea resurselor de rețea în adrese numerice folosite de către rețea. Norii sunt locații upstream de la distanță ce stochează datele și aplicațiile de host pentru ca utilizatorii să nu necesite aşa multe resurse locale și pentru ca utilizatorii să acceseze conținutul de pe diferite dispozitive, din orice locație.

Toate aceste elemente funcționează împrumăt la nivelul aplicație. Nivelul aplicație permite utilizatorilor să lucreze și să se joace împreună peste Internet.



## CAPITOLUL 11. ESTE O REȚEA

### Introducere

Până în acest moment al cursului, am examinat serviciile ce pot fi oferite, de către o rețea de date, rețelei umane, caracteristicile fiecărui nivel al modelului OSI și funcțiile protoocoalelor TCP/IP și am descris nivelul Ethernet în detaliu, tehnologie LAN universală. Următorul pas este de a învăța asamblarea acestor elemente împreună într-o rețea funcțională și care trebuie menținută activă.

**Notă:** Studenții pot lucra individual, în perechi sau întreaga clasă poate completa această activitate împreună.

### 11.1 Creare și Dezvoltare. Echipamentele în Rețelele Mici

Majoritatea afacerilor sunt afaceri mici. Deci, nu este o surpriză faptul că majoritatea rețelelor sunt rețele mici.

În rețelele mici proiectarea rețelei este de obicei simplă. Numărul și tipul de dispozitive din rețea sunt reduse semnificativ în comparație cu o rețea mai mare. Topologiile de rețea pentru rețelele mici implică de obicei un singur router și unul sau mai multe switchuri. Rețelele mici ar putea avea de asemenea puncte de acces wireless (posibil construite într-un router) și telefoane IP. Ca și conexiune la Internet, în mod normal o rețea mică are o singură conexiune WAN oferită printr-o tehnologie DSL, cablu sau o conexiune Ethernet.

Gestionarea unei rețele mici necesită aceleași aptitudini ca în gestionarea unei rețele mari. Cea mai mare parte a muncii este axată pe gestionarea și depanarea echipamentului existent, cât și pe securitatea dispozitivelor și a informațiilor din rețea. Administrarea unei rețele mici este efectuată fie de un angajat al companiei, fie de o persoană angajată prin contract de către companie, în funcție de dimensiunea afacerii și de tipul de afacere.

O rețea tipică pentru o afacere mică este prezentată în Fig. 11.1.

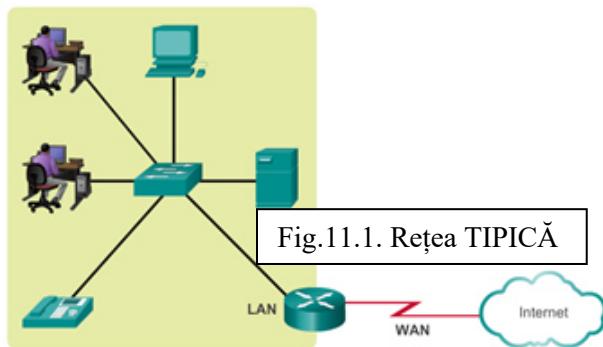


Fig. 11.1. Rețea tipică pentru o afacere mică

Pentru a îndeplini cerințele utilizatorului, chiar și rețelele mici necesită planificare și proiectare. Planificarea asigură faptul că toate cerințele, factorii de cost și opțiunile de dezvoltare sunt luate în considerare.

Una dintre primele considerații de proiectare atunci când implementăm o rețea mică este tipul de dispozitive intermediare folosite pentru suportul rețelei. La alegerea tipului de dispozitive intermediare există un număr de factori ce trebuie să fie luați în considerare, aşa cum se poate observa și în Fig. 11.2.

**Costul** – Costul este în mod normal unul dintre cei mai importanți factori atunci când alegem echipamentul pentru o rețea de dimensiune mică. Costul unui switch sau router este determinat de capacitatea și caracteristicile sale. Capacitatea dispozitivului include numărul și tipuri de porturi disponibile și viteza de backplane. Alți factori ce au impact asupra costului sunt capabilitățile de management ale rețelei, tehnologiile de securitate integrate și tehnologiile optionale de switching avansate. Un alt element cheie ce afectează costul este cătă redundanță să încorporăm în rețea – include dispozitivele, porturile de pe un dispozitiv și cablajul prin cupru sau fibră optică.

**Viteza și tipurile de porturi/interfețe** – Alegerea numărului și tipurilor de porturi de pe un router sau switch este o decizie critică. Întrebările ce trebuie să fie puse includ :

- “Să comandăm porturi destule pentru nevoile de astăzi sau să luăm în considerare cerințele de creștere ?”
- “Avem nevoie de un amestec de viteză UTP ?”
- “Avem nevoie de ambele tipuri de porturi, UTP și de fibră ?”

Computerele mai noi au încorporate Plăci de Rețea (NIC) de 1 Gbps. Porturile de 10 Gbps sunt deja incluse în unele stații de lucru și servere. Deși sunt mai costisitoare, alegerea dispozitivelor de nivel 2 ce se pot adapta vitezelor în creștere permit rețelei să evolueze fără înlocuirea dispozitivelor centrale.

**Extensibilitatea** – Dispozitivele de rețea au configurații fizice atât fixe cât și modulare.

Configurațiile fixe au un număr și tip de porturi sau interfețe specific.

Dispozitivele modulare au sloturi de expansiune ce oferă flexibilitate pentru adăugarea de noi module o dată cu extinderea cerințelor. Multe dispozitive modulare au la început un număr de porturi fixe, cât și de sloturi de expansiune. Switchurile sunt disponibile cu porturi suplimentare speciale pentru legături de mare viteză optionale. De asemenea, deoarece routerele pot fi utilizate pentru conectarea unor numere și tipuri diferite de rețele, trebuie să avem grijă să alegem modulele și interfețele adecvate pentru mediul respectiv.

Întrebări ce trebuie să fie puse sunt:

- “Comandăm dispozitive cu module upgradabile ?”
- “Ce tip de interfețe WAN, dacă există, sunt necesare pe router (routere) ?”



## 11.2 Serviciile și Caracteristicile Sistemului de Operare

În funcție de versiunea sistemului de operare, un dispozitiv de rețea poate suporta anumite caracteristici și servicii, cum ar fi :

- **Securitate – Security.**
- **Calitatea Serviciilor – QoS.**
- **Voce peste Protocolul de Internet – VoIP.**
- **Comutare la nivelul 3 din stiva OSI - Layer 3 switching.**
- **Translatarea Adreselor de Rețea – NAT.**
- **ConFig.rea dinamică a gazdelor – DHCP.**

Routerele pot fi costisitoare din punct de vedere al costurilor, în funcție de interfețele și caracteristicile necesare. Modulele suplimentare, cum ar fi cel de fibră optică, cresc costul dispozitivelor de rețea.

La implementarea unei rețele mici, este necesară planificarea spațiului de adresare IP. Toate hosturile dintr-un internetwork trebuie să aibă o adresă unică. Chiar și într-o rețea mică, atribuirea de adrese din rețea nu trebuie să fie întâmplătoare. Schema de adresare IP trebuie să fie planificată, documentată și menținută în funcție de tipul de dispozitive ce primesc adresa.

Exemple pentru diferite tipuri de dispozitive ce vor face parte din proiectarea alocării IP :

- *Dispozitivele finale pentru utilizatori.*
- *Servere și periferice.*
- *Hosturi accesibile din Internet.*
- *Dispozitivele intermediare.*

Planificarea și documentarea schemei de adresare IP ajută administratorul să urmărească tipurile de dispozitive. De exemplu, dacă toate serverele au atribuite o adresă de host din rangeul de la 50 la 100, este ușoară identificarea traficului de server prin adresa IP. Acest lucru poate fi foarte util la depanarea problemelor traficului de rețea cu ajutorul unui analizor de protocol.

În plus, pentru administratori devine mai ușor să controleze accesul la resursele dintr-o rețea în funcție de adresa IP cu folosirea unei scheme de adresare IP deterministă. Acest lucru poate fi important pentru hosturile ce oferă resurse într-o rețea internă cât și într-o rețea externă. Serverele de web sau e-commerce joacă acest rol. Dacă adresele pentru aceste resurse nu sunt planificate și documentate, securitatea și accesibilitatea la dispozitive nu sunt controlate eficient. Dacă un server are o adresă atribuită aleator, blocarea accesului la această adresă este dificilă, iar clienții ar putea să nu fie capabili să localizeze această resursă.

Fiecare dintre aceste tipuri de dispozitive diferite ar trebui să aibă alocat un bloc logic de adrese din spațiul de adresă al rețelei.



Fig. 11.3. Planificare și asignare adrese IP în funcție de echipament.

O altă parte importantă a designului de rețea este fiabilitatea. Chiar întreprinderile mici se bazează adesea pe rețeaua lor puternică pentru activitatea economică. Un eșec al rețelei poate fi foarte costisitor. Pentru a menține un grad ridicat de fiabilitate, redundanța este necesară în proiectarea rețelei. Redundanța ajută la eliminarea punctelor unice de eșec. Există mai multe moduri de a realiza redundanță într-o rețea. Redundanța poate fi realizată prin instalarea echipamentelor după, dar poate fi de asemenea realizată prin furnizarea de legături de rețea după pentru zonele critice, așa cum se arată în Fig. 11.4.

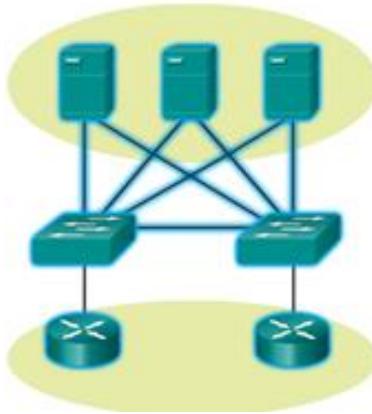


Fig. 11.4 Redundanță într-o ”fermă de servere”

Cu cât este mai mică rețeaua, cu atât este mai mică șansa ca redundanța de echipament să fie accesibilă. Prin urmare, un mod normal de introducere a redundanței este utilizarea de conexiuni redundante switchului între mai multe switchuri de rețea și între switchuri și routere.

De asemenea, serverele adesea au mai multe NICuri ce permit conexiuni redundante la unul sau mai multe switchuri. Într-o rețea mică, serverele sunt dezvoltate adesea ca servere web, servere de fișiere sau servere de e-mail.

Rețelele mici de obicei oferă un punct unic de ieșire spre Internet prin una sau mai multe porți implicite (default gateway). Cu un singur router în topologie, singura redundanță în termenii de căi de nivel 3 este stabilită prin utilizarea a mai multor interfețe Ethernet de pe un router. Însă, dacă routerul ”pică”, întreaga rețea își pierde conectivitatea la Internet. Din acest motiv, ar putea fi un sfat bun pentru o întreprindere mică să plătească pentru un cont de opțiune de cost-scăzut pentru un al doilea furnizor de servicii pentru ”backup”.

Utilizatorii se așteaptă la acces imediat la e-mailuri și la fișierele ce le paratajează sau actualizează. Pentru a ajuta asigurarea acestei disponibilități, dezvoltatorul de rețea ar trebui să parcurgă următorii pași :

**Pasul 1. Securizarea serverelor de fișier și e-mail într-o locație centralizată.**

**Pasul 2. Protejarea locației față de accesul neautorizat prin implementarea măsurilor de securitate fizică și logică.**

**Pasul 3. Crearea de redundanță în ”server farm” pentru a asigura faptul că dacă un dispozitiv ”pică”, fișierele nu sunt pierdute.**

**Pasul 4. ConFig.rea cailor redundante spre servere.**

În plus, rețelele moderne folosesc adesea unele forme de voce și video peste IP pentru comunicarea cu clienții și partenerii de afaceri. Acest tip de rețea convergentă este implementată ca o soluție integrată sau ca o formă suplimentară de date brute suprapuse peste o rețea IP. Administratorul de rețea ar trebui să ia în considerare tipurile variate de trafic și tratarea lor în designul de rețea. Routerul (routerele) și switchul (switchurile) dintr-o rețea mică ar trebui să fie

conFig.te pentru a suporta traficul în timp real, cum ar fi voce sau video, într-o manieră distinctă în comparație cu traficul de date. De fapt, un design de rețea bun va clasifica traficul cu atenție în funcție de prioritate, aşa cum se vede în Fig. 11.5.

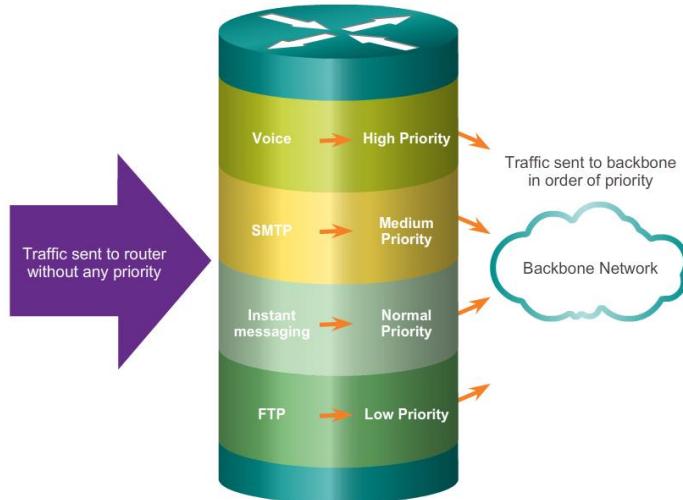


Fig. 11.5 Prioritizarea traficului

Clasele de trafic pot fi :

- *Transferul de fișiere.*
- *E-mail.*
- *Voce.*
- *Video.*
- *Mesagerie.*
- *Tranzacții.*

În final, scopul unui design de rețea bun, chiar și pentru o rețea mică, este de a asigura productivitatea pentru angajați și de a minimiza defecțiunile în timp ale rețelei.

Rețeaua este la fel de utilă ca aplicațiile ce sunt pe ea. Așa cum se vede în Fig. 11.6, la nivelul aplicație există două forme de programe software sau procese ce oferă acces la rețea: aplicațiile de rețea și serviciile de nivel aplicație.

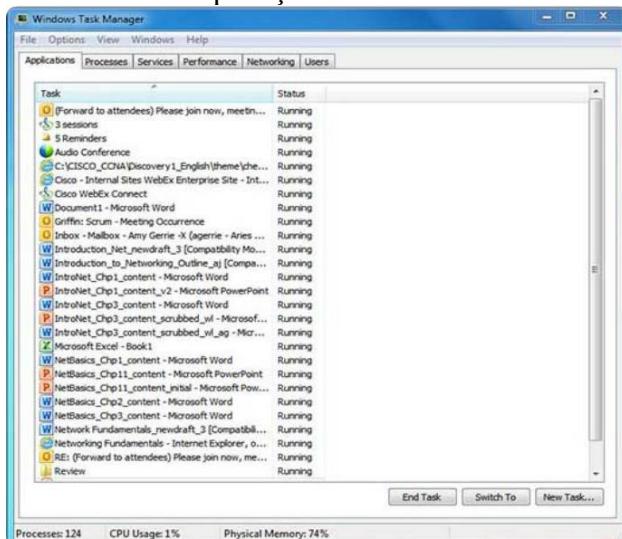


Fig. 11.6 Procese la nivelul aplicație

## 11.2.1 Aplicațiile de rețea

Aplicațiile sunt programe software folosite pentru comunicarea peste rețea. Unele aplicații end-user sunt "network-aware", ceea ce înseamnă că ele implementează protocoale de nivel aplicație și sunt capabile să comunice direct peste nivelele inferioare ale stivei de protocoale. Clientii de e-mail și browserele web sunt exemple de acest tip de aplicație.

### 11.2.1.1 Serviciile de nivel aplicație

Alte programe ar putea avea nevoie de asistență a serviciilor de nivel aplicație pentru a utiliza resursele de rețea, cum ar fi transferul de fișiere sau "spooling" de imprimare prin rețea. Deși transparent pentru un angajat, aceste servicii sunt programe ce interfațează cu rețeaua și pregătesc datele pentru transfer. Tipuri diferite de date, fie text, grafice sau video, necesită servicii de rețea diferite pentru a asigura faptul că sunt pregătite corect pentru procesare de către funcțiile ce au loc la nivelele inferioare ale modelului OSI.

Fiecare aplicație sau serviciu de rețea folosește protocoale ce definesc standarde și formate de date ce vor fi utilizate. Fără protocoale, rețeaua de date nu ar avea un mod comun de formatare și direcționare a datelor. Pentru a înțelege funcționarea serviciilor diferite de rețea, este necesar să devem familiarizați cu protocoalele de bază ce guvernează funcționarea lor.

Cea mai mare parte a muncii unui tehnician, fie într-o rețea mică, fie într-o rețea mare, va fi în modul de implicare cu protocoalele de rețea. Protocoalele de rețea suportă aplicațiile și serviciile folosite de către angajații dintr-o rețea mică. Protocoale comune de rețea sunt:

- *DNS.*
- *Telnet.*
- *IMAP, SMTP, POP (e-mail).*
- *DHCP.*
- *HTTP.*
- *FTP.*

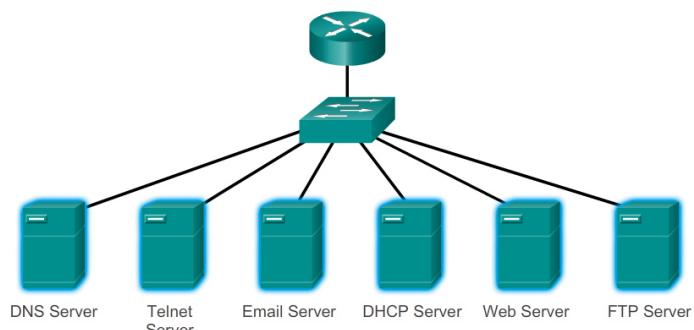


Fig. 11.7 Servere pentru protocoale comune de rețea

Acstea protocoale de rețea cuprind setul de instrumente fundamental pentru un profesionist în rețelistică. Fiecare dintre aceste protocoale definesc :

- *Procesele de la fiecare capăt al unei sesiuni de comunicare.*
- *Tipurile de mesaje.*
- *Sintaxa mesajelor.*
- *Înțelesul câmpurilor informaționale.*
- *Modul în care mesajele sunt trimise și răspunsul așteptat.*
- *Interacțiunea cu nivelul următor inferior.*

Multe întreprinderi au stabilit o politică de folosire a versiunilor securizate ale acestor protocole, acolo unde este posibil, cum ar fi protocolele : HTTPS, SFTP și SSH.

Suplimentar față de protocolele comune de rețea descrise anterior, întreprinderile moderne, chiar și cele mici, folosesc în mod normal aplicații în timp real pentru comunicarea cu clienții și partenerii de business. Deoarece o companie mică nu ar fi capabilă să justifice costul unei soluții de tip "Enterprise Cisco Telepresence", există alte aplicații în timp real, ca cele prezentate în Fig. 11.8, ce sunt convenabile din punct de vedere al prețului și justificabile pentru organizațiile din întreprinderile mici.



Fig. 11.8 Aplicații de comunicare în timp real

Aplicațiile în timp real necesită o planificare mai amănunțită și servicii dedicate (în comparație cu alte tipuri de date) pentru a asigura prioritatea livrării traficului de voce și video. Acest lucru înseamnă că administratorul de rețea trebuie să asigure echipamentul adecvat instalat în rețea și faptul că dispozitivele de rețea sunt configurate pentru a asigura prioritatea livrării. Fig. 11.9 prezintă elementele unei rețele mici ce suportă aplicații în timp real.



Fig. 11.9 Echipamentele de rețea mai puțin văzute de utilizatori

### **11.2.1.2 Infrastructură**

Pentru a suporta aplicațiile în timp real existente și propuse, infrastructura trebuie să cuprindă caracteristicile fiecărui tip de trafic. Dezvoltatorul de rețea trebuie să determine dacă switchurile și cablajul existent pot suporta traficul ce va fi adăugat în rețea. Cablarea ce poate suporta transmisii la viteze gigabit ar trebui să fie capabilă să transporte traficul generat și să nu solicite nici-o schimbare în infrastructură. Alte switchuri ar putea să nu suporte tehnologia "Power over Ethernet" (PoE). Cablarea învechită ar putea să nu suporte cerințele de lățime de bandă. Switchurile și cablarea vor trebui actualizate pentru a suporta aceste aplicații.

### **11.2.1.3 VoIP**

Tehnologia VoIP este implementată într-o organizație ce folosește încă telefoanele tradiționale. VoIP utilizează routere voice-enabled. Aceste routere convertesc vocea analogică din semnalele de telefonie tradițională în pachete IP. După ce semnalele sunt convertite în pachete IP, routerul trimite aceste pachete între locațiile respective. VoIP este mai puțin costisitor decât o soluție integrată de telefonie IP, însă calitatea comunicației lor nu îndeplinește aceleași standarde. Soluțiile de video și VoIP pentru întreprinderile mici pot fi realizate, de exemplu, cu Skype sau versiuni non-enterprise ale Cisco WebEx.

### **11.2.1.4 IP Telephony**

În telefonia IP, telefonul IP efectuează conversia voce-în-IP. Routerele voice-enabled nu sunt necesare într-o rețea cu o soluție integrată de telefonie IP. Telefoanele IP folosesc un server dedicat pentru controlul apelului și signaling. Există acum mulți furnizori cu soluții dedicate de telefonie IP pentru rețelele mici.

### **11.2.1.5 Aplicații în timp real**

Pentru a transporta streaming media eficient, rețeaua trebuie să fie capabilă să suporte aplicații ce necesită livrare sensibilă la întârziere. Real-Time Transport Protocol (RTP) și Real-Time Transport Control Protocol (RTCP) sunt două protocole ce suportă această cerință. RTP și RTCP permit controlul și scalabilitatea resurselor de rețea prin mecanisme de "Quality of Service" (QoS) încorporate. Aceste mecanisme QoS oferă instrumente prețioase pentru minimizarea problemelor de latență pentru aplicațiile de streaming în timp real.

### **11.2.1.6 Creșterea rețelelor mari**

Creșterea este un proces natural pentru multe întreprinderi mici, iar rețelele lor trebuie să crească și ele. Un administrator de rețea pentru o întreprindere mică lucrează fie reactiv, fie proactiv, în funcție de conducătorii companiei, ce includ adesea și administratorul de rețea. Ideal, administratorul de rețea are destul timp de gândire pentru a lua decizii inteligente cu privire la creșterea rețelei o dată cu creșterea companiei.

Pentru scalarea unei rețele, mai multe elemente sunt necesare:

- **Documentarea rețelei** – topologia fizică și logică.
- **Inventarul echipamentelor** – listarea dispozitivelor folosite sau incluse în rețea.
- **Bugetul** – bugetul IT detaliat, inclusiv bugetul de achiziționare de echipamente pe an.
- **Analiza traficului** – protocole, aplicații, servicii și cerințele de trafic respective ar trebui să fie documentate.

Acste elemente sunt utilizate pentru a informa “luarea de decizii” ce însoțesc scalarea unei rețele mici.

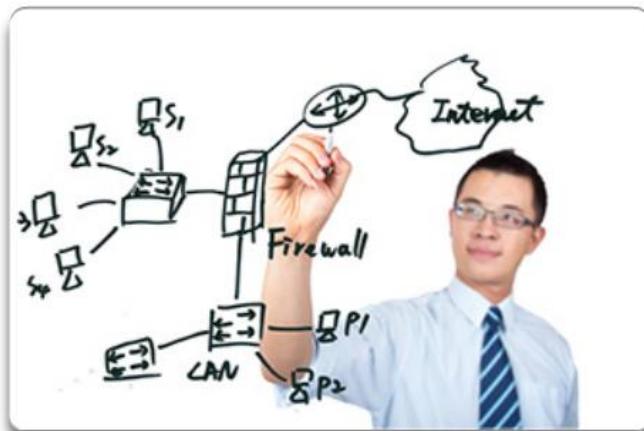


Fig. 11.10 Scalarea rețelelor

Suportul și creșterea unei rețele mici necesită să fim familiari cu protocolele și aplicațiile de rețea ce rulează peste rețea. În timp ce un administrator de rețea are mai mult timp într-un mediu de rețea mică de analizare individuală a utilizării rețelei pentru fiecare dispozitiv de rețea, o abordare mai holistică cu unele tipuri de analizoare de protocole bazate pe hardware sau software este recomandată.

Așa cum se poate observa în Fig.11.11, analizatoarele de protocole permit unui profesionist de rețea să compileze rapid informațiile statistice cu privire la fluxurile de date dintr-o rețea.

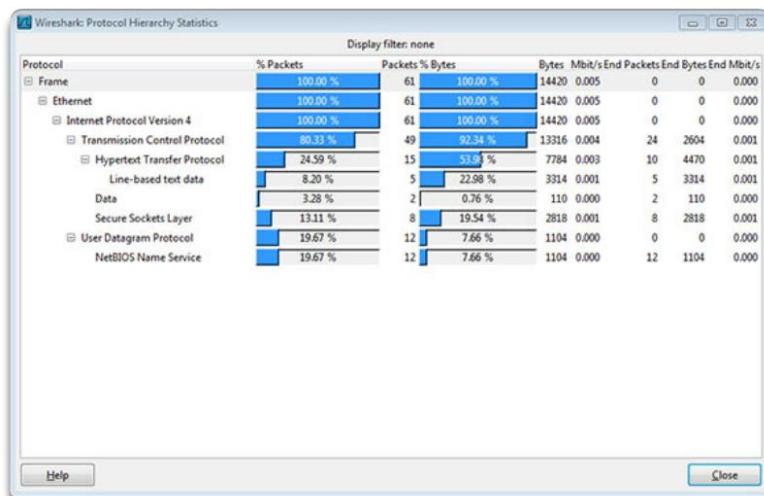


Fig. 11.11 Analizor protocole de rețea

La încercarea determinării modului în care gestionăm traficul, în special o dată cu creșterea rețelei, este important să înțelegem tipul de trafic ce traversează rețeaua, cât și fluxul de trafic curent. Dacă tipurile de trafic sunt necunoscute, analizorul de protocol va ajuta la identificarea traficului și a sursei sale.

Pentru a determina structurile fluxului de trafic, este important să :

- Captăm traficul în perioadele de utilizare de vârf pentru a primi o reprezentare bună a diferitelor tipuri de trafic.
- Efectuăm captura pe diferite segmente de rețea deoarece unele tipuri de trafic vor fi locale, pe un singur segment particular.

Informațiile adunate de către analizorul de protocol sunt analizate în funcție de sursă și destinația traficului, cât și de tipul de trafic transmis. Această analiză poate fi utilizată pentru a lăsa decizii cu privire la gestionarea traficului cât mai eficientă. Acest lucru se poate realiza prin reducerea fluxurilor de trafic inutil sau prin schimbarea patternurilor de flux prin mutarea pe un server, de exemplu.

Uneori, simpla realocare a unui server sau serviciu pe un alt segment de rețea îmbunătățește performanța rețelei și îndeplinește nevoile crescute de trafic. Alte ori, optimizarea performanței rețelei necesită reproiectarea rețelei și intervenția asupra traficului.

Pentru a înțelege tendințele de schimbare a traficului, un administrator de rețea trebuie să fie conștient de cum se schimbă utilizarea rețelei. Așa cum se poate observa în Fig. 11.12, un administrator de rețea dintr-o rețea mică are abilitatea de a obține "capturi de imagine" IT a aplicațiilor utilizate de angajat pe o perioadă de timp. Aceste "snapshots" includ de obicei informații cum ar fi:

- *Versiunea OS+OS.*
- *Aplicațiile non-rețea.*
- *Aplicațiile de rețea.*
- *Utilizarea CPU.*
- *Utilizarea driverului.*
- *Utilizarea RAM.*

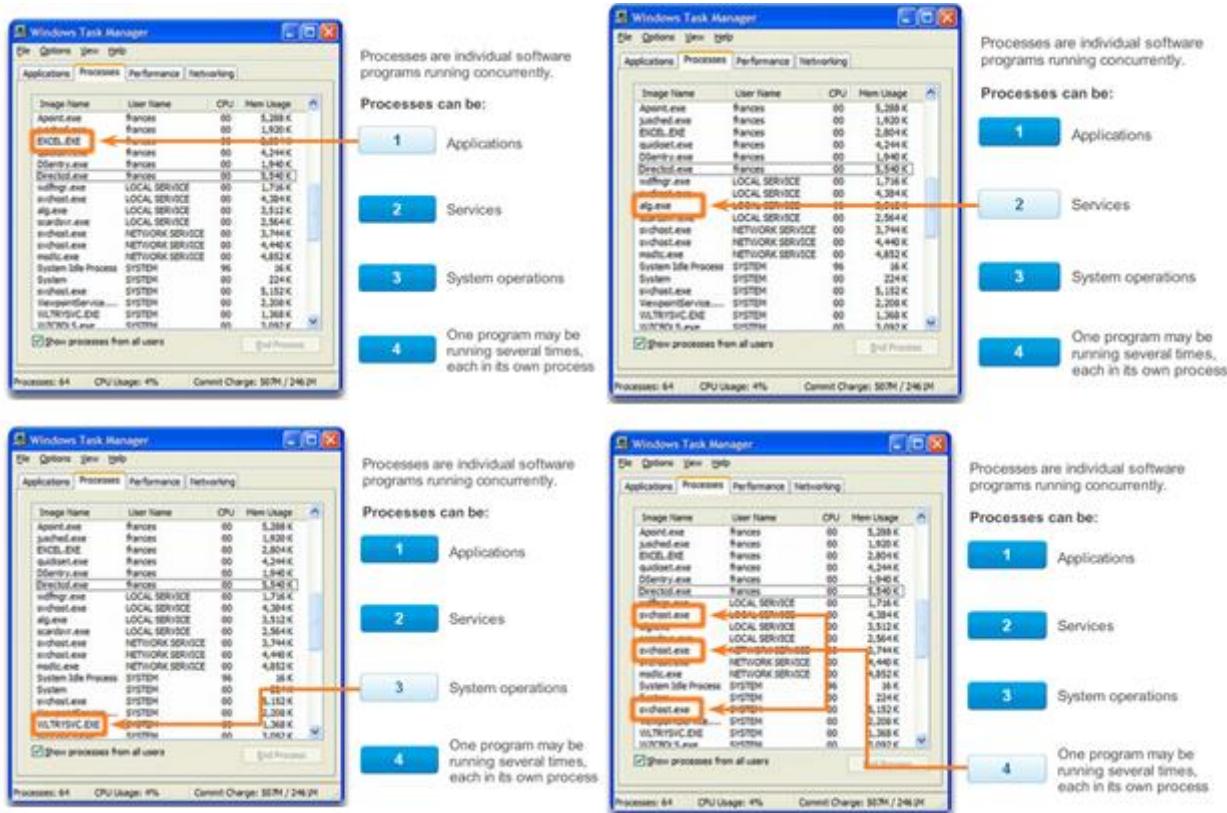


Fig. 11.12 Procese Software

Documentarea prin intermediul snapshots pentru angajații dintr-o rețea mică pe o perioadă de timp va informa administratorul de rețea de creșterea cerințelor de protocole și fluxurile de date asociate. De exemplu, pot exista unii angajați ce folosesc resurse off-site cum ar fi mediul social pentru a poziționa mai bine o companie cu privire la marketing. Atunci când au

început să lucreze pentru companie, acești angajați se axau mai puțin pe reclama pe Internet. Această schimbare în utilizarea resurselor ar putea face ca administratorul de rețea să schimbe alocarea de resurse de rețea în mod corespunzător.

Este responsabilitatea administratorului de rețea să urmărească utilizarea rețelei și cerințele fluxului de trafic și să implementeze modificări de rețea pentru a optimiza productivitatea angajatului o dată cu creșterea rețelei și a întreprinderii.

### 11.3 Păstrarea rețelei în siguranță - Măsuri de securitate a dispozitivelor de rețea

Fie cablate, fie wireless, rețelele de calculatoare sunt esențiale pentru activitățile de zi cu zic. Indivizii și organizațiile depind de computerele și de rețelele lor. Pătrunderea unei persoane neautorizate poate avea ca rezultat intreruperi de rețea costisitoare și pierderi de informații. Atacurile la o rețea pot fi devastatoare și pot rezulta pierderi de timp și bani, datorită deteriorării sau furtului de informații importante sau bunuri.

Intrușii pot câștiga acces la rețea printr-o vulnerabilitate software, atacuri hardware sau prin ghicirea parolei și a numelui de utilizator ale unei persoane. Intrușii care câștigă acces prin modificarea software sau explorarea vulnerabilităților software se numesc hackeri.

După ce un hacker câștigă acces la rețea, pot apărea patru tipuri de amenințări:

- *Furtul de informații.*
- *Furtul de identitate.*
- *Pierdere/manipularea de date.*
- *Întreruperea de servicii.*

Chiar și în rețelele mici, este necesar să luăm în considerare amenințările de securitate și vulnerabilitățile atunci când planificăm implementarea unei rețele.

Când ne gândim la securitatea rețelei, sau securitatea computerului, ne putem imagina atacatorii ce explorează vulnerabilități software. O vulnerabilitate egală în importanță este securitatea fizică a dispozitivelor, aşa cum se poate observa și în Fig. 11.13 Un atacator poate refuza utilizarea resurselor de rețea dacă acele resurse pot fi compromise fizic.



Fig. 11.13 Tipuri de amenințări

Cele patru clase de amenințări fizice sunt:

- **Amenințări hardware** – deteriorarea fizică a serverelor, routerelor, switchurilor, cablării și stațiilor de lucru.
- **Amenințările de mediu** – temperaturile externe (prea frig sau prea cald) sau umiditatea extremă (prea uscat sau prea umed).
- **Amenințările electrice** – vârfurile de tensiune, insuficientă tensiune de alimentare, putere necondiționată (zgomot) și pierderea totală de putere.

- **Amenințări de întreținere** - manipulare necorespunzătoare a componentelor electrice (descărcare electrostatică), pierdere a pieselor de schimb critice, cablare slabă și etichetare slabă.

Unele dintre aceste probleme trebuie să fie abordate într-o politică organizațională. Unele dintre ele reprezintă subiectul unei bune conduceri și un management adecvat într-o organizație.

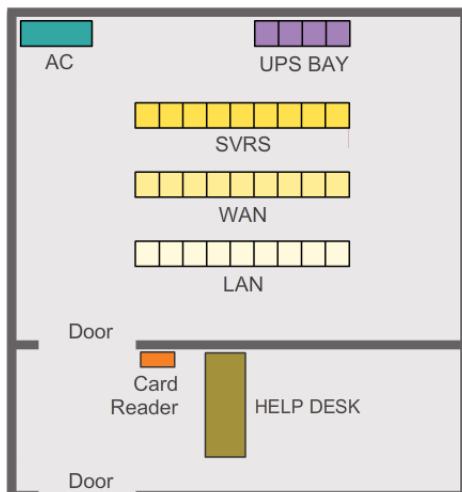


Fig. 11.14 Plan pentru securizarea fizică pentru a limita distrugerea echipamentelor  
Trezi factori de securitate a rețelei sunt :

- **Vulnerabilitatea**-reprezintă gradul de slăbiciune ce este inherent în orice rețea și dispozitiv. Include routerele, switchurile, desktopurile, serverele și chiar dispozitivele de securitate.
- **Amenințările** - includ oameni interesați și calificați în profitarea de fiecare slăbiciune de securitate. Asemenea indivizi continuă să caute noi slăbiciuni și breșe de securitate.
- **Atacurile** - Amenințările sunt realizate printr-o varietate de instrumente, scripturi și programe pentru a lansa atacuri asupra rețelei și dispozitivelor de rețea. În mod normal, dispozitivele de rețea predispușe atacurilor sunt "endpoints", cum ar fi serverele și computerele desktop.

Există trei vulnerabilități principale sau slăbiciuni:

- **Tehnologică**, evidențiată în Fig. 11.15 a

Network security weaknesses:	
<b>TCP/IP protocol weakness</b>	<ul style="list-style-type: none"> <li>• Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Internet Control Message Protocol (ICMP) are inherently insecure.</li> <li>• Simple Network Management Protocol (SNMP) and Simple Mail Transfer Protocol (SMTP) are related to the inherently insecure structure upon which TCP was designed.</li> </ul>
<b>Operating system weakness</b>	<ul style="list-style-type: none"> <li>• Each operating system has security problems that must be addressed.</li> <li>• UNIX, Linux, Mac OS, Mac OS X, Windows Server 2012, Windows 7, Windows 8</li> <li>• They are documented in the Computer Emergency Response Team (CERT) archives at <a href="http://www.cert.org">http://www.cert.org</a></li> </ul>
<b>Network equipment weakness</b>	Various types of network equipment, such as routers, firewalls, and switches have security weaknesses that must be recognized and protected against. Their weaknesses include password protection, lack of authentication, routing protocols, and firewall holes.

Fig. 11.15 a Vulnerabilitate Tehnologică

▪ De conFig.re, evidențiată în Fig. 11.15 b

Configuration Weakness	How the weakness is exploited
Unsecured user accounts	User account information may be transmitted insecurely across the network, exposing usernames and passwords to snoopers.
System accounts with easily guessed passwords	This common problem is the result of poorly selected and easily guessed user passwords.
Misconfigured Internet services	A common problem is to turn on JavaScript in Web browsers, enabling attacks by way of hostile JavaScript when accessing untrusted sites. IIS, FTP, and Terminal Services also pose problems.
Unsecured default settings within products	Many products have default settings that enable security holes.
Misconfigured network equipment	Misconfigurations of the equipment itself can cause significant security problems. For example, misconfigured access lists, routing protocols, or SNMP community strings can open up large security holes.

Fig. 11.15 b Vulnerabilitate de conFig.re

▪ Politica de securitate, evidențiată în Fig. 11.15 c

Policy Weakness	How the weakness is exploited
Lack of written security policy	An unwritten policy cannot be consistently applied or enforced.
Politics	Political battles and turf wars can make it difficult to implement a consistent security policy.
Lack of authentication continuity	Poorly chosen, easily cracked, or default passwords can allow unauthorized access to the network.
Logical access controls not applied	Inadequate monitoring and auditing allow attacks and unauthorized use to continue, wasting company resources. This could result in legal action or termination against IT technicians, IT management, or even company leadership that allows these unsafe conditions to persist.
Software and hardware installation and changes do not follow policy	Unauthorized changes to the network topology or installation of unapproved applications create security holes.
Disaster recovery plan is nonexistent	The lack of a disaster recovery plan allows chaos, panic, and confusion to occur when someone attacks the enterprise.

Fig. 11.15 c Vulnerabilitate prin politici de securitate

Toate cele trei vulnerabilități sau slăbiciuni pot duce la atacuri variate, inclusiv atacuri cu cod rău intenționat sau atacuri de rețea.

### 11.3.1 Vulnerabilități și atacuri de rețea

Atacurile cu cod rău intenționat includ un număr de tipuri de programe de computer ce au fost create cu intenția de a produce pierderi sau deteriorarea de date. Principalele trei tipuri de atacuri cu cod rău intenționat sunt :

- **Virusii** - Un virus este un software rău intenționat ce este atașat altui program pentru a executa o funcție particulară nedorită pe o stație de lucru. Un exemplu este un program ce este atașat la command.com (interpretorul primar pentru sistemele Windows) și șterge anumite fișiere și infectează orice alte versiuni ale command.com pe care le poate găsi.

Virusii necesită în mod normal un mecanism de livrare, un vector, cum ar fi un fișier zip sau unele fișiere executabile atașate unui e-mail, pentru a transporta codul de virus de la un sistem la altul. Elementul cheie ce distinge un vierme de computer de un virus de computer este faptul că interacțiunea umană este necesară pentru a facilita impreăștirea unui virus.

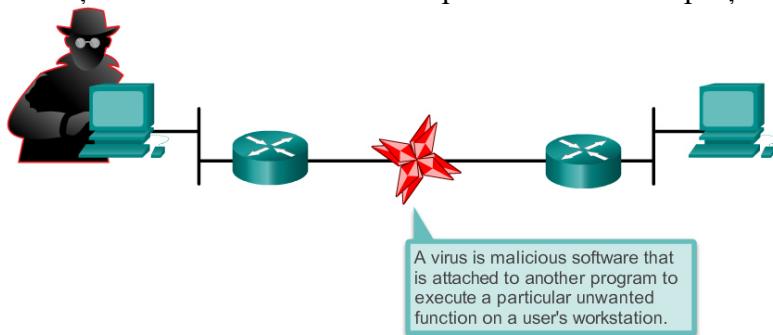


Fig. 11.16 Atac cu viruși

- **Cai Troieni** - Un cal Trojan este diferit numai prin faptul că întreaga aplicație a fost scrisă să arate ca altceva, însă este un instrument de atac. Un exemplu de cal Trojan este o aplicație

software ce rulează un simplu joc pe o stație de lucru. În timp ce utilizatorul este ocupat cu jocul, calul Troian trimite o copie a să fiecărei adrese din rangeul de adrese a utilizatorului. ceilalți utilizatori primesc jocul și îl joacă, astfel împărțind calul Troian tuturor adreselor lor.

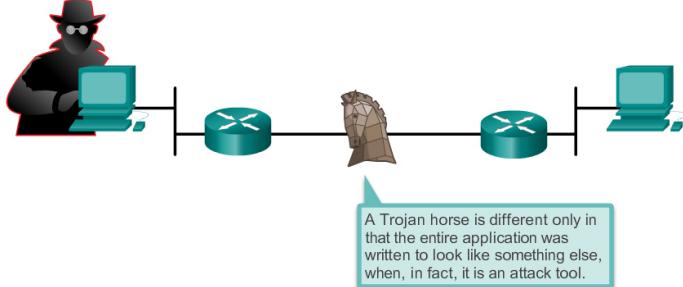


Fig. 11.17 Atac cu cai trojan

- **Viermii** – Viermii reprezintă programe ”self-contained” ce atacă un sistem și încearcă să exploreze o anumită vulnerabilitate de pe țintă. După explorarea cu succes a vulnerabilității, viermele copiază programul său de pe hostul atacat la noul sistem explorat pentru a reîncepe ciclul. Anatomia unui atac cu vierme este:
  - *Vulnerabilitatea permisă* – Un vierme se instalează prin explorarea unei vulnerabilități cunoscute în sisteme, cum ar fi utilizatorii naivi ce deschid atașamente executabile neverificate din emailuri.
  - *Mecanism de propagare* – După câștigarea accesului la un host, un vierme se copiază pe hostul respectiv, apoi caută noi ținte.
  - *Payload* – După ce un host este infectat cu un vierme, atacatorul are acces la host, de obicei ca un utilizator privilegiat. Atacatorii pot folosi o exploatare locală pentru a crește nivelul lor privilegiat la administrator.

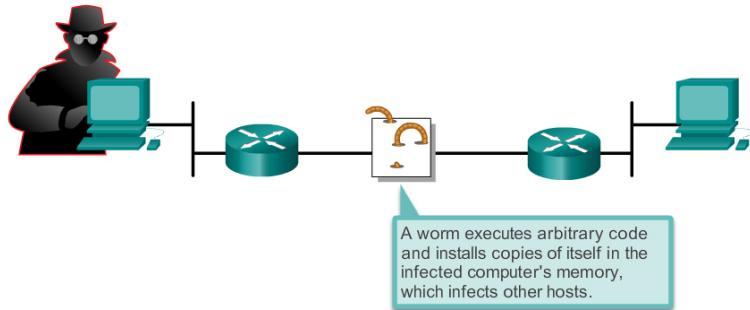


Fig. 11.18 Atac cu viermi

În plus față de atacurile cu cod rău intenționat, este posibil ca rețelele să cadă pradă mai multor atacuri de rețea. Atacurile de rețea pot fi clasificate în trei principale categorii :

1. **Atacuri de recunoaștere** – *descoperirea și mapări neautorizate ale sistemelor, serviciilor sau vulnerabilităților.*
2. **Atacuri de acces** – *manipularea neautorizată a datelor, accesului la sistem sau privilegiilor de utilizator.*
3. **Denial of service** – *dezactivarea sau coruperea rețelei, sistemelor sau serviciilor.*

### 11.3.2 Atacuri de recunoaștere

Atacatorii externi pot folosi instrumente Internet, cum ar fi utilizarea ”nslookup” și ”whois” pentru a determina ușor spațiul de adrese IP atribuit unei întreprinderi sau entități. După ce este determinat spațiul de adrese IP, un atacator poate da **ping** adreselor IP disponibile pentru a identifica adresele active. Pentru automatizarea acestui pas, un atacator ar putea folosi un

instrument de tip ”**ping sweep**”, cum ar fi ”**fping**” sau ”**gping**”, ce dă **ping** sistematic tuturor adreselor de rețea dintr-un range dat sau subrețea. Acest lucru este similar cu urmărirea unei secțiuni din cartea de telefon și apelarea fiecărui număr pentru a vedea cine răspunde.

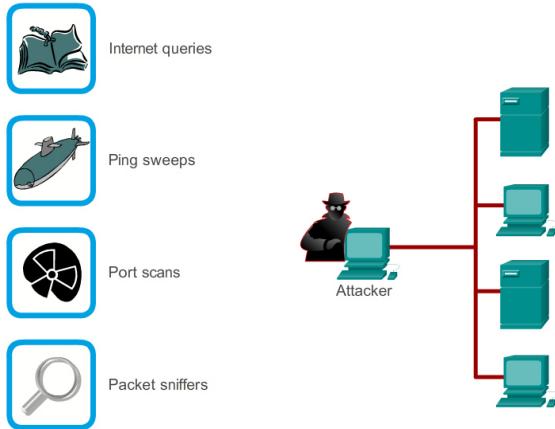


Fig. 11.19 Atacuri de recunoaștere

### 11.3.3 Atacuri de acces

Atacurile de acces explorează vulnerabilități cunoscute în serviciile de autentificare, servicii FTP și servicii web pentru a câștiga acces la conturi web, baze de date confidențiale și alte informații sensibile. Atacurile de acces pot fi clasificate în patru tipuri :

**A.** Unul dintre cele mai cunoscute tipuri de atacuri este atacul de parolă. Atacurile de parolă pot fi implementate folosind un packet sniffer pentru a obține conturile de utilizator și parolele transmise în text clar. Atacurile de parolă pot fi încercări repetitive de logare la o resursă comună, cum ar fi un server sau router, pentru a identifica un cont de utilizator, o parolă sau ambele. Aceste încercări repetitive sunt numite atacuri de tip dicționar sau atacuri brute-force.

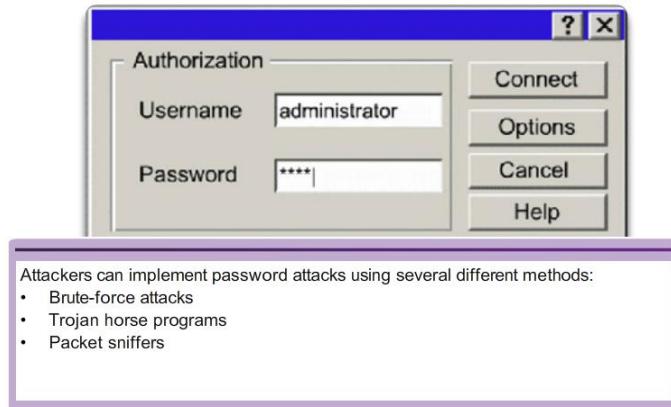
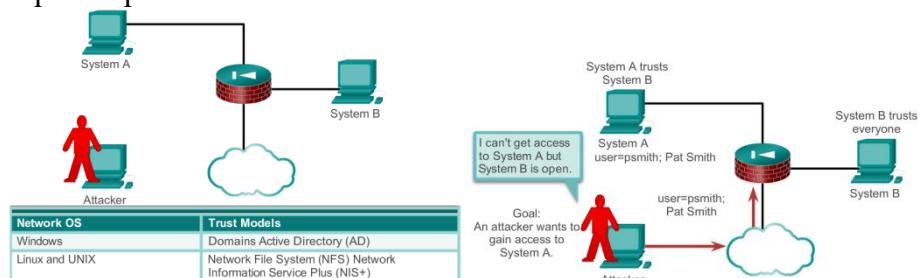


Fig. 11.20 Password attack

### B. Atac prin exploatarea încrederii



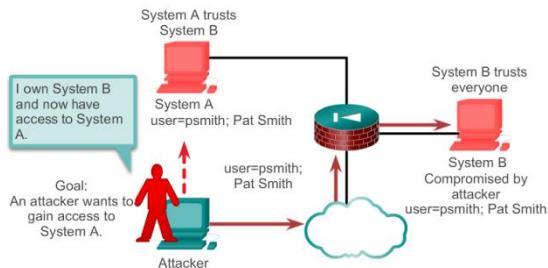


Fig. 11. 21 Trust exploitation

### C. Atac prin redirectarea porturilor

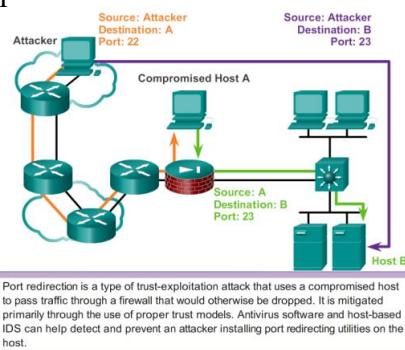


Fig. 11. 22 Port redirection

### D. Atac prin om la mijloc

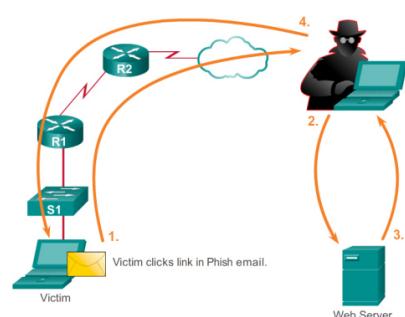
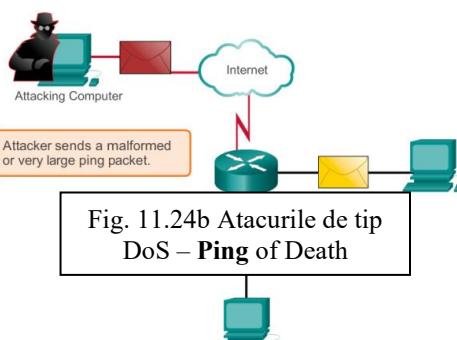
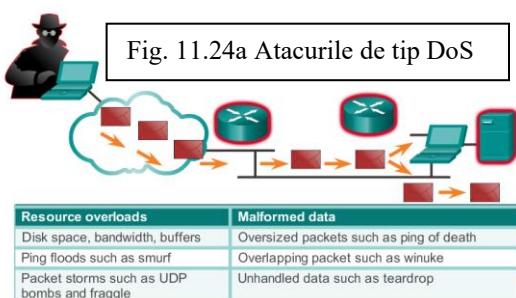


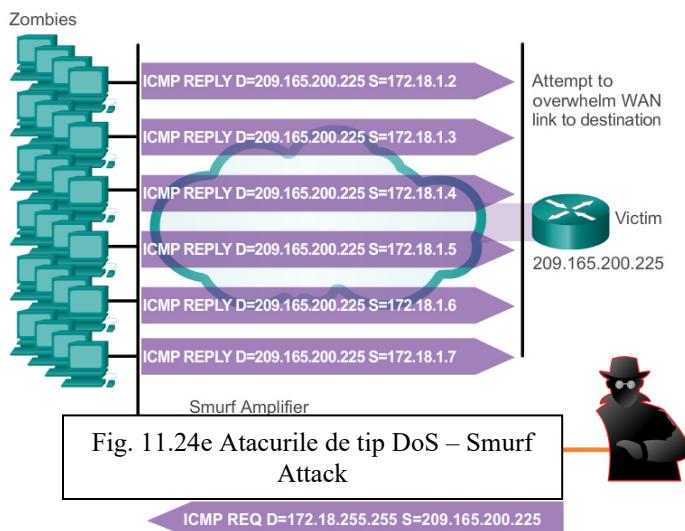
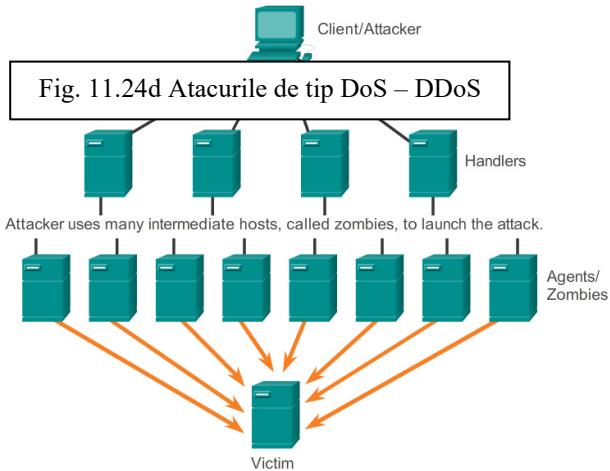
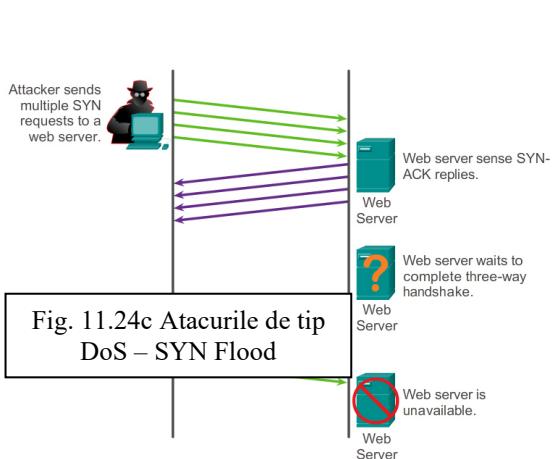
Fig. 11. 23 Man în – the – middle

#### 11.3.4 Negarea serviciilor

Atacurile DoS sunt cele mai mediatizate forme de atac și se află printre cele mai dificile de eliminat. Chiar și într-o comunitate de atacator, atacurile DoS sunt considerate banale și de formă rea deoarece necesită prea puțin efort de executare. Însă datorită ușurinței lor de implementare și a prejudiciului potențial adus, atacurile DoS merită o atenție specială din partea administratorilor de rețea.

Atacurile DoS iau mai multe forme. Ele blochează oamenii autorizați la accesul la un serviciu prin consumarea resurselor sistemului.





### 11.3.5 Combaterea atacurilor de rețea

Software antivirus poate detecta mulți viruși și aplicații de tip cai Troieni și poate preveni împrăștierea acestora în rețea. Softwareul antivirus poate fi instalat la nivel de utilizator și la nivel de rețea.

Menținerea la curent cu cele mai recente dezvoltări în aceste tipuri de atacuri poate de asemenea conduce la o protecție eficientă împotriva acestor atacuri. O dată cu apariția unui nou virus sau aplicații Trojan, întreprinderile trebuie să-și actualizeze ultimile versiuni ale softwareului antivirus.

Combaterea atacului de tip vierme necesită multă silință din partea personalului de administrare a rețelei și sistemului. Următorii pași sunt recomandați pentru combaterea atacului de tip vierme:

- *Izolarea* – Stoparea răspândirii viermelui în rețea. Compartimentăm părțile neinfecțate ale rețelei.
- *Inocularea* – Începem să “peticim” toate sistemele și, dacă este posibil, scanăm sistemele de vulnerabilități.
- *Carantina* – Detectăm fiecare mașină infectată în rețea. Deconectăm, înlăturăm sau blocăm mașinile infectate din rețea.

- *Tratament* – Curățăm și “peticim” fiecare sistem infectat. Unii viermi necesită reinstalare completă de sistem pentru a curăța sistemul.

Cel mai eficient mod de combatere a atacului de tip vierme este prin descărcarea actualizărilor de securitate de la furnizorul de sistem de operare și peticirea tuturor sistemelor vulnerabile. Acest lucru este dificil de realizat cu sisteme necontrolate în rețea locală. Administrarea a numeroase sisteme implică crearea unei imagini software standard (sistemul de operare și aplicații acreditate ce sunt autorizate pentru folosirea pe sistemele client) ce este adăugată pe sisteme noi sau actualizate. Însă, cerințele de securitate se schimbă și sistemele deja adăugate ar putea necesita instalarea ”patches” de securitate actualizate.

O soluție de management a ”patches” de securitate critice este crearea unui server central de ”patch” cu care toate sistemele trebuie să comunice după o anumită perioadă de timp, aşa cum este prezentată în figura 11.25. Orice ”patches” ce nu sunt aplicate pe un host sunt descărcate automat de pe serverul de patch și instalate fără alte intervenții.

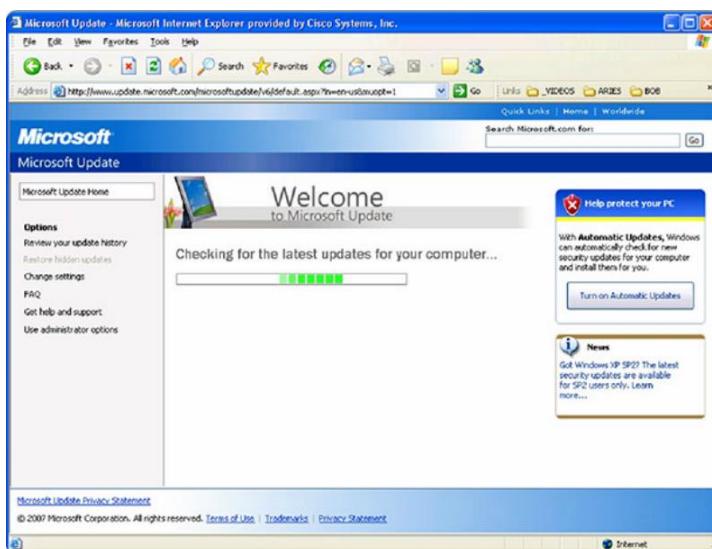


Fig. 11.25 Actualizările SO

Serviciile de securitate de rețea de tip triplu **A - Authentication, Authorization, and Accounting=AAA**) oferă cadrul de muncă inițial pentru setarea controlului accesului la un dispozitiv de rețea. AAA este un mod de control a persoanelor ce au permis accesul la o rețea (autentificare), ce pot face în timp ce se află în rețea (autorizare) și să vadă acțiunile efectuate în acest timp (contabilizare). AAA oferă un grad mai mare de scalabilitate decât comenzi de autentificare de pe consolă, AUX, VTY și din modul privilegiat.

#### *11.3.6 Autentificarea*

Utilizatorii și administratorii trebuie să dovedească că sunt cine spun. Autentificarea poate fi stabilită prin utilizarea unei combinații nume de utilizator-parolă, întrebări de provocare și răspuns, token cards și alte metode. De exemplu: “ Sunt utilizatorul ‘student’. Știu parola pentru a dovedi că sunt utilizatorul ‘student’.”

Într-o rețea mică, autentificarea locală este adesea utilizată. Cu autentificarea locală, fiecare dispozitiv își menține propria bază de date de combinații nume de utilizator/parolă. Însă, când sunt mai multe conturi de utilizator într-o bază de date locală de pe dispozitiv, gestionarea respectivelor conturi de utilizator devine complexă. În plus, o dată cu creșterea rețelei și cu numărul de dispozitive introduse în rețea, autentificarea locală devine dificil de menținut și nu

este scalabilă. De exemplu, dacă există 100 de dispozitive de rețea, toate conturile de utilizator trebuie să fie adăugate pe toate cele 100 de dispozitive.

Pentru rețelele mari, o soluție mai scalabilă este autentificarea externă. Autentificarea externă permite tuturor utilizatorilor să fie autentificați printr-un server extern de rețea. Cele mai populare două opțiuni pentru autentificarea externă a utilizatorilor sunt RADIUS și TACACS+:

- **RADIUS** este un standard deschis cu utilizare scăzută a resurselor CPU și a memoriei. Este folosit de un range de dispozitive de rețea, cum ar fi switchuri, routere și dispozitive wireless.
- **TACACS+** este un mecanism de securitate ce permite autentificarea modulară, autorizarea și contabilitatea serviciilor, care folosește un TACACS+ daemon ce rulează pe un server de securitate.

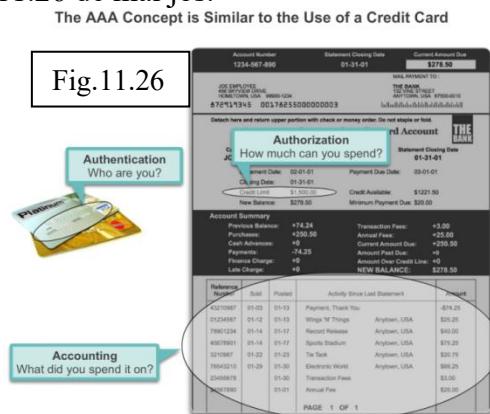
### 11.3.7 Autorizarea

După ce utilizatorul este autentificat, serviciile de autorizare determină ce resurse poate utilizatorul să le acceseze și ce operații îi sunt permise. Un exemplu este: "Utilizatorul 'student' poate accesa serverXYZ folosind numai Telnet."

### 11.3.8 Contorizarea

Contorizarea ține evidența înregistrărilor efectuate de către utilizator, inclusiv ce a accesat, cantitatea de timp în care resursa a fost accesată și orice schimbări efectuate. Contabilitatea ține evidența a modului în care sunt utilizate resursele de rețea. Un exemplu este "Utilizatorul 'student' a accesat serverXYZ folosind Telnet pentru 15 minute."

Conceptul AAA este similar utilizării unui card de credit. Cardul de credit identifică cine îl poate folosi, cât de mult poate cheltui și ține evidența a ce lucruri a comparat utilizatorul, așa cum se poate observa în Fig.11.26 de mai jos.



Pentru a proteja computerele individuale și serverele atașate la rețea, este important să controlăm traficul ce călătorește din și prin rețea.

Un firewall este unul dintre cele mai eficiente instrumente de securitate disponibile pentru protejarea utilizatorilor de rețea interni de amenințările externe. Un firewall este între două sau mai multe rețele și controlează traficul dintre ele și ajută de asemenea la prevenirea accesului neautorizat. Produsele firewall folosesc tehnici variate de determinare a accesului permis și interzis dintr-o rețea.

Acstea tehnici sunt:

- **Filtrarea de pachete** – Previne sau permite accesul în funcție de adresele IP sau MAC.
- **Filtrarea de aplicații** – Previne sau permite accesul unor anumite tipuri de aplicație, în funcție de numerele de port.
- **Filtrarea URL** – Previne sau permite accesul la websiteuri în funcție de anumite URLuri sau cuvinte cheie.
- **Stateful packet inspection (SPI)** – Pachetele ce vin trebuie să fie răspunsuri legitime la cereri de la hosturile interne. Pachetele nesolicită sunt blocate, cu excepția cazului în care sunt permise prin specificare. ISP poate de asemenea include capacitatea de recunoaștere și filtrare a anumitor tipuri de atacuri cum ar fi DoS.

Produsele firewall ar putea suporta una sau mai multe dintre aceste capacitați de filtrare. În plus, firewalurile efectuează adesea Network Address Translation (NAT). NAT traduce o adresă IP internă sau un grup de adrese într-o adresă IP publică, externă ce este trimisă în rețea. Acest lucru permite ca adresele IP interne să fie ascunse de utilizatorii externi.

Produsele firewall vin în diferite forme, aşa cum se poate vedea și în Fig.11.27 :

- **Appliance-based firewalls** – Un firewall bazat pe dispozitiv este un firewall ce este construit într-un dispozitiv dedicat hardware numit și dispozitiv de securitate.
- **Server-based firewalls** – Un firewall bazat pe server constă dintr-o aplicație firewall ce rulează pe network operating system (NOS) cum ar fi UNIX sau Windows.
- **Integrated firewalls** – Un firewall integrat este implementat prin adăugarea funcționalității firewall unui dispozitiv existent, cum ar fi un router.
- **Personal firewalls** - Firewalurile personale se află pe computerele hosturilor și nu sunt dezvoltate pentru implementări LAN. Ele pot fi disponibile implicit de la OS sau pot fi de la un furnizor extern.



Fig.11.27.

O rețea securizată este la fel de puternică precum legătura cea mai slabă a sa. Amenințările de profil înalt discutate cel mai mult în media sunt amenințările externe, cum ar fi viirmii sau atacurile DoS. Însă securizarea rețelei interne este la fel de importantă ca securizarea perimetrului unei rețele. Rețeaua internă este alcătuită din puncte de lucru finale, unele fiind arătate în Fig. . Un endpoint, sau host, este un sistem individual computer sau dispozitiv ce se comportă ca un client de rețea. Endpointurile comune sunt laptopurile, desktopurile, serverele, smart phoneurile și tabletele. Dacă utilizatorii nu își instalează securitatea pe dispozitivele lor, nici-o cantitate de precauții de securitate nu va garanta o rețea sigură.

Securizarea dispozitivelor finale este una dintre cele mai provocatoare sarcini ale unui administrator de rețea deoarece implică natura umană. O companie ar putea avea politici bine documentate puse la punct și angajații trebuie să fie conștienți de acele reguli. Angajații trebuie să fie instruiți de utilizarea adecvată a rețelei. Politicile includ adesea utilizarea de software

antivirus și prevenirea intruziunilor pe host. Mai multe soluții complexe de securizare a endpointului se bazează pe controlul accesului la rețea.

Securitatea endpoint necesită de asemenea securizarea dispozitivelor de nivel 2 din infrastructura de rețea pentru a preveni atacurile de nivel 2 cum ar fi MAC address spoofing, atacurile de overflow a tabelei de adrese MAC și atacurile LAN storm. Acest lucru este cunoscut ca fiind combatere a atacului.



#### 11.4 Securizarea dispozitivelor

O parte a securității de rețea este securizarea dispozitivelor, inclusiv dispozitivele finale și intermediare, cum ar fi dispozitivele de rețea.

Atunci când un nou sistem de operare este instalat pe dispozitiv, setările de securitate sunt la valorile implicate. În multe cazuri, nivelul de securitate este inadecvat. Pentru routerele Cisco, caracteristica Cisco AutoSecure poate fi folosită pentru a ajuta la securizarea sistemului, aşa cum este descrisă în Fig. . Există pași simpli ce ar trebui să fie urmați ce se aplică celor mai multe sisteme de operare:

- *Numele de utilizator și parolele default ar trebui schimbată imediat.*
- *Accesul la resursele sistemului ar trebui să fie restricționat numai la indivizi ce sunt autorizați să utilizeze respectivele resurse.*
- *Orice servicii și aplicații inutile ar trebui închise sau dezinstalate, atunci când este posibil.*

Toate dispozitivele ar trebui să fie actualizate cu patches de securitate imediat ce sunt disponibile. Adesea, dispozitivele de la producător au stat într-un depozit pentru o anumită perioadă de timp și nu au instalate cele mai noi patches. Este important, înainte de punerea în funcțiune, să actualizăm orice software și să instalăm orice patches de securitate.

*Locking Down Your Router*

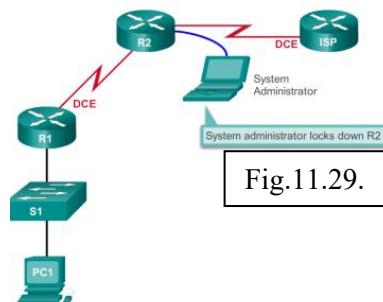


Fig.11.29.

Pentru a proteja dispozitivele de rețea, este important să utilizăm parole puternice. Există mai multe linii de ghidare standard de urmat:

- *Folosim o parolă cu lungime de cel puțin 8 caractere, de preferat 10 sau mai multe caractere. Cu cât este mai lungă parola, cu atât este mai bună.*
- *Facem parolele complexe. Includem o combinație de litere mari cu litere mici, numere, simboluri și spații, dacă este permis.*

- Evită parolele bazate pe repetare, cuvinte comune din dicționar, litere sau numere în segvență, nume de utilizator, nume ale rудelor sau ale animalului, informații biografice, cum ar fi date de naștere, numere ID, numele strămoșilor sau alte piese identificabile ușor.
  - Scriem un cuvânt incorrect în mod deliberat. De exemplu, Smith = Smyth = 5mYth sau Security = SecurIty.
  - Schimbăm parolele des. Dacă o parolă este compromisă în necunoștință, fereastra oportunității pentru atacator de utilizarea a parolei este limitată.
  - Nu notăm parolele și le lăsăm în locuri evidente, cum ar fi pe birou sau monitor.
- Fig. arată exemple de parole puternice și slabe.

Pe routerele Cisco, spațiile de început sunt ignorate pentru parole, însă spațiile după primul caracter nu. Prin urmare, o metodă de creare a unei parole puternice este utilizarea spațiilor în parolă și crearea unei fraze alcătuită din mai multe cuvinte. Aceasta se numește pass phrase. O pass phrase este adesea mai ușor de ținut minte decât o parolă simplă. Este de asemenea mai lungă și mai greu de ghicit.

Administratorii ar trebui să se asigure de faptul că parolele puternice sunt folosite în rețea. Un mod de realizarea al acestui lucru este utilizarea instrumentelor de atac “brute force” folosite de atacatori pentru a verifica puterea parolei.

#### Weak and Strong Passwords

Fig.11.30. PASS

Weak Password	Why it is weak
secret	Simple dictionary password
smith	Mother's maiden name
toyota	Make of a car
bob1967	Name and birthday of a user
Blueleaf23	Simple words and numbers

Strong Password	Why it is strong
b67n42d39c	Combines alphanumeric characters
12^h u4@1p7	Combines alphanumeric characters, symbols, and also includes a space

La implementarea dispozitivelor este important să urmăm toate îndrumările de securitate setate de către organizație. Acestea includ numirea dispozitivelor într-o manieră ce permite documentarea ușoară și urmărirea, dar și care menține o anumită formă de securitate. Nu este înțelept să oferim prea multe informații cu privire la utilizarea dispozitivului în numele de host. Există multe alte măsuri de bază de securitate ce ar trebui să fie urmate.

#### 11.4.1 Additional Password Security

Parolele puternice sunt utile atunci când sunt secrete. Există mai mulți pași ce pot fi urmați pentru a ajuta ca parolele să rămână secrete. Folosirea comenzii din config.rea globală **service password-encryption** previne indivizii neautorizați de a vedea parolele în text clar în fișierul de config.re, așa cum se arată în Fig. . Această comandă are ca efect criptarea tuturor parolelor necriptate.

În plus, pentru a ne asigura că toate parolele au minimul de lungime specificată, folosiți comanda **security passwords min-length** în modul de config.re global.

Un alt mod prin care hackerii pot învăța parolele se face prin simplele atacuri brute-force, încercând mai multe parole până când una se potrivește. Este posibil să prevenim acest tip de atac prin blocarea încercărilor de logare la un dispozitiv dacă un număr setat de eșecuri au loc într-o anumită perioadă de timp.

```
Router(config)# login block-for 120 attempts 3 within 60
```

Această comandă va bloca încercările de logare pentru 120 de secunde, dacă au existat trei încercări cu eşec în 60 de secunde.

**Banners** – Un mesaj banner este similar cu un semn de încălcare a legii. Sunt importante pentru că ar putea avea urmări în justiție, într-o instanță de drept, pe oricine accesează sistemul inadecvat. Ne asigurăm că mesajele de banner respectă politicile de securitate ale organizației.

```
Router(config)# banner motd #Vineri la ora 14:00 serverul va intra în revizie !#
```

**Exec Timeout** – O altă recomandare este setarea de executive timeauturi. Prin setarea exec timeout, spunem dispozitivului să deconecteze automat utilizatorii de pe o linie după ce au fost inactivi pentru o anumită perioadă specificată prin valoarea exec timeout. Exec timeauturile pot fi configurate pe liniile console și vty, precum și porturi auxiliare.

```
Router(config)# line vty 0 15
```

```
Router(config-vty)# exec-timeout 10 55
```

Această comandă va deconecta utilizatorii după 10 minute și 55 de secunde.

```
Router(config) #service password-encryption
Router(config) #security password min-length 8
Router(config) #login block-for 120 attempts 3 within 60
Router(config) #line vty 0 4
Router(config-vty) #exec-timeout 10
Router(config-vty) #end
Router#show running-config
-more-
!
line vty 0 4
password 7 03095A0F034F38435B49150A1819
exec-timeout 10
login
```

## 11. 5 Acces de la distanță prin SSH

Protocolul de gestionare a dispozitivelor de la distanță este Telnet. Telnet nu este securizat. Datele conținute într-un pachet Telnet sunt transmise necriptate. Folosind un instrument ca Wireshark, este posibil ca cineva să “intercepteze” o sesiune Telnet și să obțină o informație de parolă. Din acest motiv, este recomandat să activăm SSH pe dispozitive pentru accesul securizat de la distanță. Este posibilă configurația unui dispozitiv să suporte SSH în patru pași, așa cum se poate vedea și în Fig. .

**Pasul 1.** Ne asigurăm că routerul are un nume de host unic și apoi configurăm numele IP de domeniu al rețelei folosind comanda **ip domain-name Seria25.com** în modul de configurație globală.

**Pasul 2.** Chei secrete one-way trebuie să fie generate de către router pentru a cripta traficul SSH. Cheia este folosită pentru criptarea și decriptarea datelor. Pentru a crea o cheie criptată, folosim comanda **crypto key generate rsa general-keys modulus-modulus-size=1024** în modul de configurație global. Înțelesul specific al diferențelor părți ale acestei comenzi este complex și nu prezintă scopul acestui curs, însă acum doar notăm faptul că modulus determină dimensiunea cheii și poate fi configurația de la 360 biți la 2048 de biți. Cu cât este mai mare modulus, cu atâtă

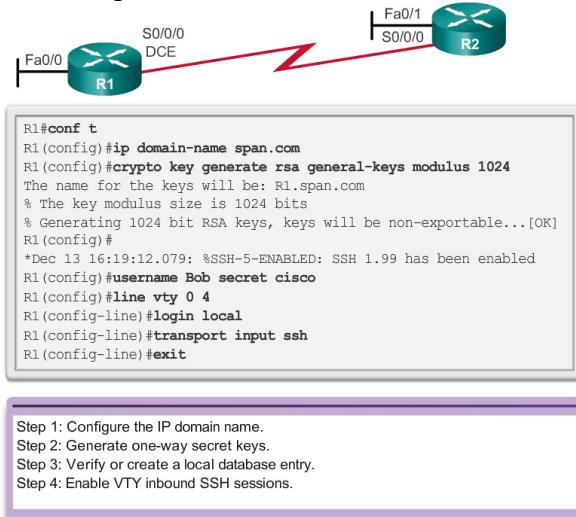
cheia este mai sigură, însă durează mai mult criptarea și decriptarea informațiilor. Lungimea minimă recomandată a modulus este de 1024 de biți.

*Router(config)# crypto key generate rsa (general-keys modulus 1024)*

**Pasul 3.** Creem o bază de date locală cu intrare de nume de utilizator folosind comanda **username name secret secret** din modul de conFig.re global.

**Pasul 4.** Activăm sesiuni SSH pe vty folosind comenziile pe linia vty **login local** și **transport input ssh**.

Serviciul SSH de pe router poate fi acum accesat folosind un software client SSH.



## 11.6 Performanța de bază a rețelei

**Ping** – După ce rețeaua a fost pusă în aplicare, un administrator de rețea trebuie să fie capabil să testeze conectivitatea la rețea pentru a se asigura de faptul că funcționează corect. În plus, este o bună idee ca administratorul de rețea să documenteze rețeaua.

**Comanda ping** – Comanda **ping** este un mod eficient de testare a conectivității. Testarea ne este referită ca testare a stivei de protocoale deoarece comanda **ping** merge de la nivelul 3 al modelului OSI la nivelul 2, apoi la nivelul 1. **Ping** folosește protocolul ICMP pentru a verifica conectivitatea.

Comanda **ping** nu găsește întotdeauna cu precizie natura unei probleme, însă ajută la identificarea sursei problemei, un pas important în depanarea unei probleme de rețea.

Comanda **ping** oferă o metodă de verificare a stivei de protocoale și a conFig.ției adresei IPv4 de pe host, dar și testează conectivitatea cu hosturile locale sau de la distanță, aşa cum se poate observa în Fig. . Există instrumente suplimentare ce pot oferi mai multe informații decât **ping**, cum ar fi Telnet sau Trace route, ce vor fi discutate detaliat mai târziu.

**Indicatorii Ping din IOS** – Un **ping** efectuat de pe IOS va avea unul dintre indicatorii pentru fiecare ICMP echo trimis. Indicatorii cei mai cunoscuți sunt:

- **!** – indică primirea unui mesaj de răspuns ICMP echo.
- **.** – indică faptul că timpul de așteptare a expirat pentru un mesaj de răspuns ICMP echo
- **U** – un mesaj de ICMP unreachable a fost primit.

"!" (semnul de exclamare) indică faptul că **ping** a fost completat cu succes și a verificat conectivitatea de nivel 3.

".." poate indica probleme în comunicare. Ar putea indica faptul că o problemă de conectivitate are loc undeva în cale. Ar putea de asemenea indica faptul că un router din cale nu are o rută spre destinație și nu a trimis un mesaj ICMP de destinație unreachable. De asemenea ar putea indica faptul că **ping** a fost blocat de către securitatea de dispozitiv.

"U" indică faptul că un router în cale nu are o rută spre adresa destinație sau faptul că cererea **ping** a fost blocată și a avut ca răspuns un mesaj ICMP unreachable.

**Testarea loopback** – Comanda **ping** este folosită pentru a verifica configurația IP internă de pe hostul local. Reamintim faptul că acest test este realizat prin utilizarea comenzi **ping** asupra unei adrese rezervate numită loopback (127.0.0.1). Aceasta verifică funcționarea adecvată a stivei de protocol de la nivelul rețea până la nivelul fizic și înapoi, fără punerea unui semnal real pe mediu de comunicare.

Comenzile **ping** sunt introduse pe linia de comandă.

Introducem comanda **ping loopback** cu sintaxa următoare:

C:\>ping 127.0.0.1

Răspunsul pentru o astfel de comandă va prezenta ceva de forma:

*Reply from 127.0.0.1: bytes=32 time<1ms TTL=128*

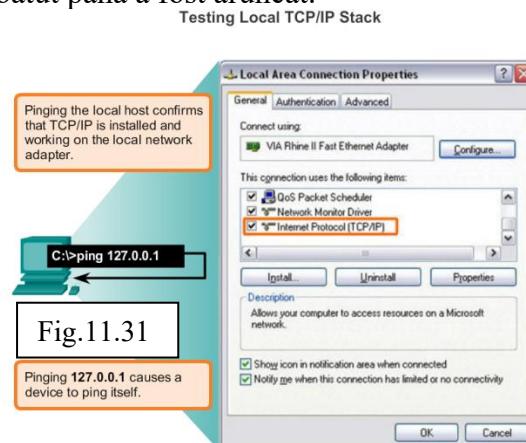
Statisticile pentru **Ping** la adresa 127.0.0.1:

*Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),*

*Approximate round trip times in milli-seconds:*

*Minimum = 0ms, Maximum = 0ms, Average = 0ms*

Rezultatul indică faptul că patru pachete de 32 de byte au fost trimise și returnate de host 127.0.0.1 într-un timp de mai puțin de 1 ms. TTL (Time-to-Live) definește numărul de hopuri pe care pachetul **ping** le-a străbătut până a fost aruncat.



IOS de pe echipamentele intermediare oferă un mod "extins" al comenzi **ping**. Acest mod este realizat prin introducerea comenzi **ping** în modul EXEC privilegiat, fără o adresa IP destinație. O serie de prompturi sunt prezentate, așa cum se poate vedea și în exemplul de mai jos. Prin apăsarea tastei Enter acceptăm valorile indicațiile implicate. Exemplul de mai jos ilustrează modul în care putem forța adresa sură pentru un **ping** să fie la 10.1.1.1 (de văzut R2 din Fig. ); adresa sursă pentru un **ping** standard va fi 209.165.200.226. Prin efectuarea acestui lucru, administratorul de rețea poate verifica de la distanță (de pe R2) faptul că R1 are ruta 10.1.1.0/24 în tabela sa de rutare.

R2# **ping**

Protocol [ip]:

Target IP address: **192.168.10.1**

Repeat count [5]:

Datagram size [100]:

Timeout în seconds [2]:

Extended commands [n]: **y**

Source address or interface: **10.1.1.1**

Type of service [0]:

Set DF bit în IP header? [no]:

Validate reply data? [no]:

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Sweep range of sizes [n]:

Type escape sequence to abort.

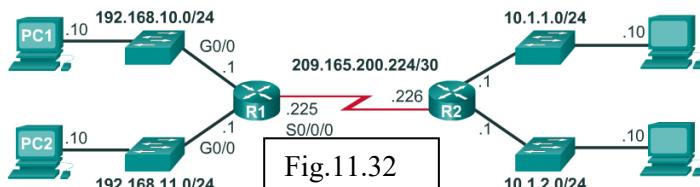
Sending 5, 100-byte ICMP Echos to 192.168.10.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 36/97/132 ms

Introducerea unei perioade mai mari de timeout decât cea implicită permite ca problemele eventuale de latență să fie detectate. Dacă testul **ping** este cu succes pentru o valoare mai mare, o conexiune există între hosturi, însă latența poate fi o problema în rețea.

De reținut faptul că introducerea "y" pe promptul "Extended commands" oferă mai multe opțiuni utile pentru depanare.



Unul dintre cele mai eficiente instrumente de monitorizare și depanare a performanței rețelei este stabilirea unor linii de bază (baseline) de rețea. O "baseline" este un proces de studiere a rețelei la intervale regulate pentru a asigura faptul că rețeaua funcționează corect. O baseline de rețea este mai multe decât un simplu raport ce detaliază "sănătatea" rețelei la un anumit punct. Crearea unei baseline eficiente de performanță a rețelei este realizată în timp. Măsurarea performanței la momente de timp diferite și încărcarea vor ajuta la crearea unei imagini mai bune asupra performanței generale de rețea.

Ieșirea comenzielor de rețea poate contribui cu date pentru baseline de rețea.

O metodă de începere a unei baseline este copierea rezultatelor comenziilor **ping**, **tracert** executate sau a altor comenzi relevante într-un fișier text. Aceste fișiere text pot fi "ștampilate" cu data și salvate într-o arhivă pentru revederea ulterioară a acestora.

O utilizare eficientă a informațiilor stocate este compararea rezultatelor în timp (Fig. 3). Printre elementele de luat în considerare sunt mesajele de eroare și timpii de răspuns de la host la host. Dacă există o creștere considerabilă în timpii de răpsuns, poate exista o problemă de latență la adresa.

Importanța creării documentației nu poate fi accentuată destul. Verificarea conectivității host-la-host, problemele de latență și soluții ale problemelor identificate pot ajuta un administrator de rețea în păstrarea funcționării rețelei cât mai eficient posibilă.

Rețelele corporative ar trebui să aibă baselines extinse; mai extinse decât putem explica în acest curs. Instrumente software de grad profesional sunt disponibile pentru stocarea și menținerea informațiilor baseline. În acest curs, acoperim numai unele tehnici de bază și discutăm scopul acestor baselines.

Capturarea ieșirii comenzi **ping** poate fi realizată din promptul IOS, aşa cum se poate vedea în Fig. 11.33.

**Run the same test**

FEB 8, 2013 08:14:43

```
C:\>ping 10.66.254.159
Pinging 10.66.254.159 with 32 bytes of data:
Reply from 10.66.234.159: bytes=32 time<1ms TTL=128
Ping statistics for 10.66.254.159:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

MAR 17, 2013 14:41:06

```
C:\>ping 10.66.254.159
Pinging 10.66.254.159 with 32 bytes of data:
Reply from 10.66.234.159: bytes=32 time<6ms TTL=128
Ping statistics for 10.66.254.159:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 6ms, Average = 6ms
```

**Compare values**

FEB 8, 2013 08:14:43

```
C:\>ping 10.66.254.159
Pinging 10.66.254.159 with 32 bytes of data:
Reply from 10.66.234.159: bytes=32 time<1ms TTL=128
Ping statistics for 10.66.254.159:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

MAR 17, 2013 14:41:06

```
C:\>ping 10.66.254.159
Pinging 10.66.254.159 with 32 bytes of data:
Reply from 10.66.234.159: bytes=32 time<6ms TTL=128
Ping statistics for 10.66.254.159:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 6ms, Average = 6ms
```

**Router Ping Capture - Saving to a text file**

Fig.11.33

dsh - HyperTerminal

File Edit View Call Transfer Help

Send File... Receive File...

Capture Text... Stop

Send Text File... Pause

Capture to Printer Resume

Interface Serial1 description Serial1 Interface on the RTA router ip address 192.168.4.89.255.255.240

In the terminal session:

1. Start the text capture process.
2. Issue a **ping <ip address>** command.
3. Stop the capture process.
4. Save the text file.

**Tracert** – Trace are ca răspuns o listă de hopuri prin care un pachet este routat în rețea. Forma comenzi depinde de locul unde este efectuată. Atunci când efectuăm comanda trace dintr-un computer Windows, folosind tracert. Atunci când efectuăm comanda trace de pe CLI a unui router, folosim traceroute, aşa cum se poate observa și în Fig. 1.

Ca și comenzi **ping**, comenzi **trace** sunt introduse în linia de comandă și necesită o adresa IP ca argument.

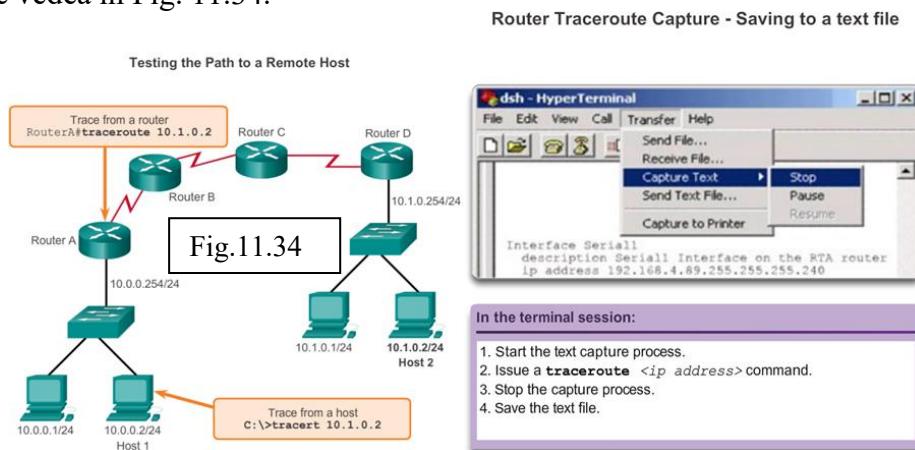
Presupunând că această comandă va fi efectuată de pe un computer Windows, folosim forma **tracert**:

C:\>**tracert 10.1.0.2**

```
Tracing route to 10.1.0.2 over a maximum of 30 hops
1 2 ms 2 ms 2 ms 10.0.0.254
2 * * * Request timed out.
3 * * * Request timed out.
4 ^C
```

Singurul răspuns cu succes a fost de la gateway de pe Router A. Cererile Trace la următorul hop au expirat, însemnând că routerul next hop nu a răspuns. Rezultatele trace indică faptul că eșecul este undeva în internetwork, în exteriorul LANului.

Capturarea ieșirii traceroute poate fi de asemenea efectuată de pe promptul routerului, aşa cum se poate vedea în Fig. 11.34.



## 11.7 Comenzi show

Comenzi IOS CLI **show** afișează informații relevante cu privire la configurația și funcționarea dispozitivului.

Tehnicienii de rețea utilizează comenzi **show** intens pentru vizualizarea fișierelor de config, verificarea statusului interfețelor și proceselor de pe dispozitiv și verificarea statusului operațional al dispozitivului. Comenzi **show** sunt disponibile fie că dispozitivul a fost config.t cu CLI, fie cu Cisco Configuration Professional.

Statusul a aproape fiecărui proces sau funcție a routerului poate fi afișat cu ajutorul comenzi **show**. Unele dintre cele mai populare comenzi **show** sunt:

- **show running-config** (Fig. 11.35).
- **show interfaces** (Fig. 11.36).
- **show arp** (Fig. 11.37).
- **show ip route** (Fig. 11.38).
- **show protocols** (Fig. 11.39).
- **show version** (Fig. 11.40).

Show running-config

```
R1#show running-config
<Output omitted>
Building configuration...
Current configuration : 1063 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname R1
enable secret 5 $1$6w9$vdvpVM6zV10E6tSyLdkR5/
no ip domain lookup
!
interface FastEthernet0/0
description LAN 192.168.1.0 default gateway
ip address 192.168.1.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
description WAN link to R2
ip address 192.168.2.1 255.255.255.0
encapsulation ppp
clock rate 64000
no fair-queue
!
interface Serial0/0/1
no ip address
shutdown
!
interface Vlan1
no ip address
!
router rip
version 2
network 192.168.1.0
network 192.168.2.0
!
banner motd ^CUnauthorized Access Prohibited^C
!
ip http server
!
line con 0
password cisco
```

Fig.11.35

```
!
interface Serial0/0/1
no ip address
shutdown
!
interface Vlan1
no ip address
!
router rip
version 2
network 192.168.1.0
network 192.168.2.0
!
banner motd ^CUnauthorized Access Prohibited^C
!
ip http server
!
line con 0
password cisco
login
line aux 0
line vty 0 4
password cisco
login
```

```
R1#show interfaces
<Output omitted>
FastEthernet0/0 is up, line protocol is up
Hardware is GT96K FE, address is 001b.5325.256e
(hia 001b.5325.256e)
Internet address is 192.168.1.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, 100BaseTX/FX
ARF type: ARPA, ARP Timeout 04:00:00
Last input 00:01:17, output 00:00:01, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes);
Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
196 packets input, 31850 bytes
Received 181 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog
```

Fig.11.36

```
0 input packets with dribble condition detected
392 packets output, 35239 bytes, 0 underruns
0 output errors, 0 collisions, 3 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out

FastEthernet0/1 is administratively down,
line protocol is down

Serial0/0/0 is up, line protocol is up
Hardware is GT96K Serial
Internet address is 192.168.2.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Listen, loopback not set
Keepalive set (10 sec)
Last input 00:00:02, output 00:00:03, output hang never
Last clearing of "show interface" counters 00:51:52
Input queue: 0/75/0/0 (size/max/drops/flushes);
Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
```

```
Hardware is GT96K Serial
Internet address is 192.168.2.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Listen, loopback not set
Keepalive set (10 sec)
Last input 00:00:02, output 00:00:03, output hang never
Last clearing of "show interface" counters 00:51:52
Input queue: 0/75/0/0 (size/max/drops/flushes);
Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
401 packets input, 27437 bytes, 0 no buffer
Received 293 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
389 packets output, 26940 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
6 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

Serial0/0/1 is administratively down, line protocol is down
```

Show arp

Fig.11.37

```
R1#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 172.17.0.1 - 001b.5325.256e ARPA
FastEthernet0/0
Internet 172.17.0.2 12 000b.db04.a5cd ARPA
FastEthernet0/0
```

Show ip route

Fig.11.38

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial0/0/0
R    192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
```

Show protocols	Fig.11.39	Show version	Fig.11.40
<pre>R1#show protocols Global values:   Internet Protocol routing is enabled FastEthernet0/0 is up, line protocol is up   Internet address is 192.168.1.1/24 FastEthernet0/1 is administratively down, line protocol is down FastEthernet0/1/0 is up, line protocol is down FastEthernet0/1/1 is up, line protocol is down FastEthernet0/1/2 is up, line protocol is down FastEthernet0/1/3 is up, line protocol is down Serial0/0/0 is up, line protocol is up   Internet address is 192.168.2.1/24 Serial0/0/1 is administratively down, line protocol is down Vlan1 is up, line protocol is down</pre>		<pre>R1#show version &lt;Output omitted&gt; Cisco IOS Software, 1841 Software (C1841-ADVISERVICESK9-M), Version 12.4(10b), RELEASE SOFTWARE (fc3) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2007 by Cisco Systems, Inc. Compiled Fri 19-Jan-07 15:15 by prod_rel_team  ROM: System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1) R1 uptime is 43 minutes System returned to ROM by reload at 22:05:12 UTC Sat Jan 5 2008 System image file is "flash:c1841-adviserervicesk9-mz.124-10b.bin"  Cisco 1841 (revision 6.0) with 174080K/22528K bytes of memory. Processor board ID FTX1111WQF 6 FastEthernet interfaces 2 Serial(sync/async) interfaces 1 Virtual Private Network (VPN) Module DRAM configuration is 64 bits wide with parity disabled. 191K bytes of NVRAM. 62720K bytes of ATA CompactFlash (Read/Write)  Configuration register is 0x2102</pre>	

După ce fișierul de configurare startup este încărcat și routerul bootează cu succes, comanda **show version** poate fi utilizată pentru verificarea și depanarea unor componente software și hardware de bază folosite în procesul de bootup. Ieșirea comenții **show version** include:

- *Versiunea Cisco IOS utilizată.*
- *Versiunea softwareului bootstrap de sistem, stocat în memoria ROM ce a fost inițial utilizat pentru a boota routerul.*
- *Numele de fișier complet al imaginii Cisco IOS și unde este localizat programul de bootstrap.*
- *Tipul de CPU de pe router și cantitatea de RAM. Ar putea fi necesară actualizarea cantității de RAM atunci când se actualizează softwareul IOS.*
- *Numărul și tipul de interfețe fizice de pe router.*
- *Cantitatea de NVRAM. NVRAM este utilizat pentru a stoca fișierul startup-config.*
- *Cantitatea de memorie flash a routerului. Ar putea fi necesară actualizarea cantității de flash atunci când se actualizează softwareul IOS.*
- *Valoarea curentă config.tă a registrului software de config.ție în hexazecimal.*

Registrul de config.ție spune routerului cum să booteze. De exemplu, setările implicite din fabrică pentru registrul de config.ție este 0x2102. Această valoare indică faptul că routerul încearcă să încarce o imagine IOS din flash și încarcă fișierul startup de config.re din NVRAM. Este posibilă schimbarea registrului de config.re și prin urmare, schimbarea locului în care routerul se uită pentru imaginea IOSului și fișierul startup de config.re în timpul procesului de bootup. Dacă există o a doua valoare în paranteze, precizează valoarea registrului de config.ție din următoarea reîncărcare a routerului.

Fig.11.41

```

Router#show version
Cisco Internetwork Operating System
Software
IOS(tm)2500 Software (C2500-I-L), Version
12.0(17a), RELEASE SOFTWARE (fc1)
Copyright (c)1986-2002 by cisco
Systems, Inc.
Compiled Mon 11-Feb-02 05:55 by kellythw
Image text-base:0x00001000
ROM:system Bootstrap, Version
11.0(10c), SOFTWARE
BOOTFLASH :3000 Bootstrap Software (IGS-
BOOT-R), Version 11.0(10c), RELEASE
SOFTWARE (fc1)
System image file is "flash:c2500-i-
1.120-17a.bin"
Cisco 2500 (68030 processor(revision N)
With 2048K/2048K bytes of memory.
processor bord ID 08860060,with hardware
revision 00000000
Bridging software.
X.25 software,version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile Configuration
memory.
8192K bytes of processor board system
flash (Read ONLY)
Configuration register is 0x2102
Router#

```

Comanda **show version** de pe un switch afișează informații cu privire la versiunea software încărcată curent, împreună cu informații hardware și de dispozitiv. Unele dintre informațiile afișate în această comandă sunt:

- **Software version** - *Versiunea IOS software.*
- **Bootstrap version** - *Versiunea Bootstrap.*
- **System up-time** – *Timpul scurs de la ultimul reboot.*
- **System restart info** – *Metoda de restart (e.g., ciclu energetic, crash).*
- **Software image name** – *Numele fișier IOS.*
- **Switch platform and processor type** – *Numărul modelului și tipul procesorului.*
- **Memory type (shared/main)** – *RAM principal al procesorului și shared packet I/O buffering.*
- **Hardware interfaces** – *Interfețele disponibile pe switch.*
- **Configuration register** – *Setează specificațiile de bootup, viteza de consolă și parametrii aferenți.*

Fig.11.42 arată un exemplu de ieșire tipică a comenzi **show version** afișată de către un switch.

<pre> Switch#show version Cisco IOS Software, C2960 Software (C2960-LANBASE-M), Version 12.2(25)SEE2, RELEASE SOFTWARE (fc1) Copyright (c) 1986-2006 by Cisco Systems, Inc. Compiled Fri 28-Jul-06 04:33 by yenanh Image text-base: 0x00003000, data-base: 0x00AA2F34  ROM: Bootstrap program is C2960 boot loader BOOTLDR: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(25r)SEE1, RELEASE SOFTWARE (fc1)  Switch uptime is 2 minutes System returned to ROM by power-on System image file is "flash:c2960-lanbase-mz.122-25.SEE2/c2960- lanbase-mz.122-25.SEE2.bin"  cisco WS-C2960-24TT-L (PowerPC405) processor (revision B0) with 61440K/4088K bytes of memory. Processor board ID FOC1107Z9ZN Last reset from power-on 1 Virtual Ethernet interface 24 FastEthernet interfaces 2 Gigabit Ethernet interfaces The password-recovery mechanism is enabled. </pre>	<pre> 64K bytes of flash-simulated non-volatile configuration memory. Base ethernet MAC Address : 00:1B:53:03:17:00 Motherboard assembly number : 73-10390-03 Power supply part number : 341-0097-02 Motherboard serial number : FOC11071TTJ Power supply serial number : AZS110605RU Model revision number : B0 Motherboard revision number : C0 Model number : WS-C2960-24TT-L System serial number : FOC1107Z9ZN Top Assembly Part Number : 800-27221-02 Top Assembly Revision Number : C0 Version ID : V02 CLEI Code Number : COM3L00BRA Hardware Board Revision Number : 0x01  Switch Ports Model SW Version SW Image ----- ----- ----- *   1    26   WS-C2960-24TT-L 12.2(25)SEE2 C2960-LANBASE-M  Configuration register is 0xF </pre>
--	---

Fig.11.42

## 11.8 Comenzile pe Host și IOS

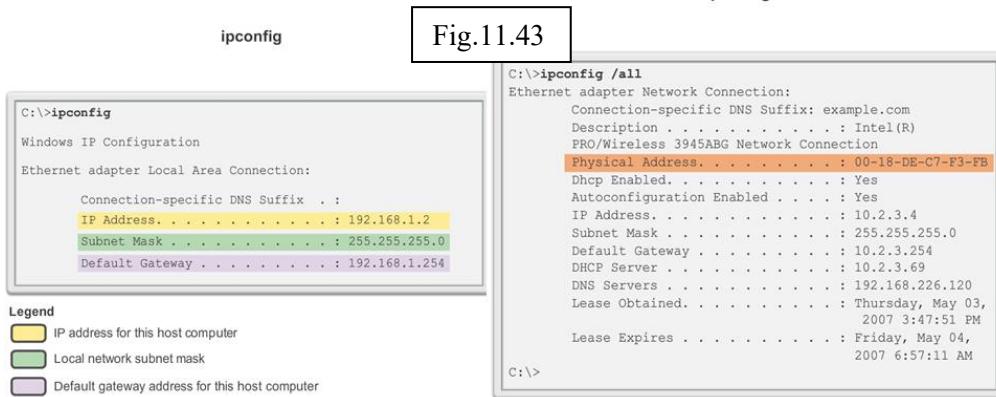
Așa cum se poate vedea în Fig. 1, adresa IP a default gateway de pe hostului poate fi vizualizată prin efectuarea comenzi **ipconfig** pe linia de comandă a unui computer Windows.

Un instrument de examinare a adresei MAC de pe computerul personal este **ipconfig /all**. De remarcat faptul că în Fig. 11.43, adresa MAC a computerului este afișată împreună cu un număr de detalii cu privire la adresarea de nivel 3 a dispozitivului.

În plus, producătorul interfeței de rețea de pe computer poate fi identificat prin partea OUI a adresei MAC. Aceasta poate fi căutat pe Internet.

Serviciul de Client DNS de pe PCurile Windows optimizează performanța rezoluției de nume DNS prin stocarea numelor rezolvate anterior în memorie. Comanda **ipconfig /displaydns** afișează toate intrările DNS stocate pe un sistem de computer Windows.

**ipconfig /all**



```
C:\>ipconfig /all
Windows IP Configuration

Ethernet adapter Network Connection:
  Connection-specific DNS Suffix . :
    IP Address . . . . . : 192.168.1.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.254

C:\>
```

Fig.11.43

Comanda **arp** permite crearea, editarea și afișarea mapărilor adreselor fizice cu adrese IPv4 cunoscute. Comanda **arp** este executată din Windows command prompt.

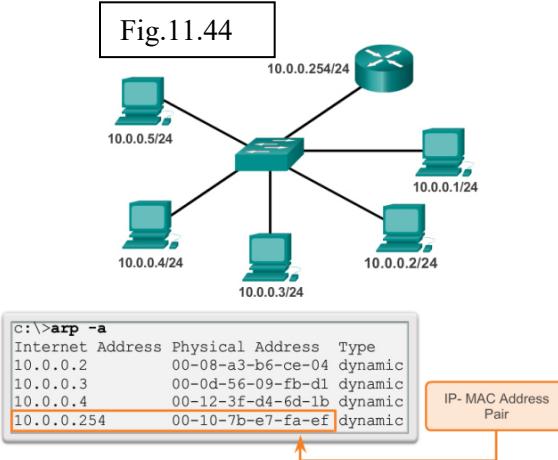
Pentru a executa comanda **arp**, pe command prompt al hostului, introducem C:\host1>**arp -a**

Așa cum se poate observa în Fig. , comanda **arp -a** listează toate dispozitivele aflate curent în ARP cacheul hostului, ce include adresa IPv4, adresa fizică și tipul de adresare (static/dinamic) pentru fiecare dispozitiv.

Cache poate fi “curățat” prin utilizarea comenzi **arp -d** în cazul în care administratorul de rețea vrea să repopuleze cache cu informații actualizate.

**Notă:** ARP cache conține numai informații de la dispozitivele ce au fost accesate recent. Pentru a ne asigura de faptul că ARP cache este populat, **pinguim** un dispozitiv pentru a avea o intrare în ARP table.

Learning About the Nodes on the Network



Examinăm ieșirea comenții **show cdp neighbors** din Fig.11.45, cu topologia din Fig.11.46. Remarcăm faptul că a strâns unele informații detaliate cu privire la R2 și switchul conectat pe interfață Fast Ethernet a R3.

CDP este un protocol proprietar Cisco ce rulează la nivelul legătură de date. Deoarece CDP rulează la nivelul legătură de date, două sau mai multe dispozitive de rețea, cum ar fi routere ce suportă diferite protocoale de nivel rețea, pot învăța unele despre celelalte, chiar dacă nu există o conectivitate de nivel 3.

Atunci când un dispozitiv Cisco bootează, CDP este pornit în mod implicit. CDP descoperă automat dispozitive Cisco vecine ce rulează CDP, indiferent de ce protocol de nivel 3 sau sătă rulează. CDP schimbă informații de dispozitiv hardware și software cu vecinii CDP direct conectați.

CDP oferă următoarele informații cu privire la fiecare dispozitiv vecin CDP:

- **Identifierii de dispozitiv** – De exemplu, *hostname config.t* pe un switch.
- **Lista de adresa** – O adresă de nivel rețea pentru fiecare protocol suportat.
- **Identifier de port** – Numele portului local sau de la distanță în forma unui string de caractere ASCII, cum ar fi *ethernet0*.
- **Lista de capacitați** – De exemplu, dacă dispozitivul respectiv este un router sau switch.
- **Platforma** – Platforma hardware a dispozitivului; de exemplu, un router din seria Cisco 1841.

Comanda **show cdp neighbors detail** afișează adresa IP a unui dispozitiv vecin. CDP va afișa adresa IP a vecinului indiferent dacă putem da **ping** sau nu vecinului. Această comandă este foarte utilă atunci când routerele Cisco nu pot accesa legătura lor comună. Comanda **show cdp neighbors detail** va ajuta la determinarea dacă unul dintre vecini are o eroare de *config.tie IP*.

Pentru situațiile de descoperire de rețea, cunoașterea adresei IP a vecinului CDP este adesea singura informație necesară pentru a accesa de la distanță prin Telnet respectivul dispozitiv.

Din motive evidente, CDP reprezintă un risc de securitate. Deoarece unele versiuni IOS trimit advertisements CDP implicit, este important să simă dezactivăm CDP.

Pentru a dezactiva CDP global, utilizăm comanda **no cdp run** în modul de *config.re* global. Pentru a dezactiva CDP pe o interfață, utilizăm comanda pe interfață **no cdp enable**.

```
R3#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge,
                  B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP,
                  r - Repeater, P - Phone

Device ID Local Intrfce Holdtme Capability Platform Port ID
S3      Fas 0/0        151   S I       WS-C2950  Fas 0/6
R2      Ser 0/0/1      125   R         1841     Ser 0/0/1

R3#show cdp neighbors detail
Device ID: R2
Entry address(es):
  IP address : 192.168.1.2
Platform: Cisco 1841, Capabilities: Router Switch IGMP
Interface: Serial0/0/1, Port ID (outgoing port): Serial0/0/1
Holdtime : 161 sec

Version :
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M),
Version 12.4(10b), RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Fri 19-Jan-07 15:15 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''

-----
Device ID: S3
Entry address(es):
Platform: cisco WS-C2950-24, Capabilities: Switch IGMP
Interface: FastEthernet0/0, Port ID (outgoing port):
FastEthernet0/11
Holdtime : 148 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(9)EA1,
RELEASE SOFTWARE (fc1)
```

Fig.11.45

```

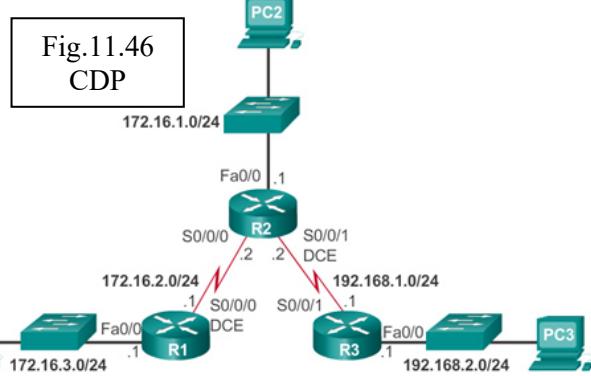
Entry address(es):
Platform: cisco WS-C2950-24, Capabilities: Switch IGMP
Interface: FastEthernet0/0, Port ID (outgoing port):
FastEthernet0/11
Holdtime : 148 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(9)EA1,
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2002 by cisco Systems, Inc.
Compiled Wed 24-Apr-02 06:57 by antonino

advertisement version: 2
Protocol Hello: OUI=0x00000C, Protocol ID=0x0112; payload
len=27, value=0000000FFFFFFF0
10231FF00000000000000AB769F6C0FF0000
VTP Management Domain: 'CCNA3'
Duplex: full

R3#

```



În același mod în care comenzi și utilitare sunt utilizate pentru verificarea unei configurații de host, comenzi pot fi utilizate pentru verificarea interfețelor de pe dispozitivele intermediare. IOSul oferă comenzi de verificare a funcționării interfețelor routerului și switchului.

### 11.9 Verificarea interfețelor de pe router

Una dintre cele mai frecvente comenzi utilizate este comanda **show ip interface brief**. Această comandă oferă o ieșire mai compactă decât comanda **show ip interface**. Oferă un rezumat al informațiilor cheie pentru toate interfețele de rețea de pe router.

În Fig. 1 este arătată topologia utilizată în acest exemplu.

Ieșirea **show ip interface brief** afișează toate interfețele de pe router, adresa IP atribuită pe fiecare interfață, dacă există, și statusul funcțional al interfeței.

Conform ieșirii, interfața FastEthernet 0/0 are o adresa IP 192.168.254.254. Ultimele două coloane din această linie arată statusul de nivel 1 și 2 al acestei interfețe. Cuvântul “**up**” în coloana de Status arată că această interfață este operațională la nivel 1. Cuvântul “**up**” în coloana de protocol indică faptul că protocolul de nivel 2 este funcțional.

De asemenea, remarcăm faptul că interfața Serial 0/0/1 nu a fost activată. Acest lucru este indicat prin **administratively down** din coloana Status.

Ca pe orice dispozitiv final, putem verifica conectivitatea de nivel 3 cu comenzi **ping** și **traceroute**. În acest exemplu ambele comenzi, **ping** și **traceroute**, arată o conectivitate cu succes.

### 11.10 Verificarea interfețelor de pe switch

Comanda **show ip interface brief** poate fi utilizată de asemenea pentru a verifica statusul interfețelor de pe switch. Adresa IP pentru switch este aplicată pe o interfață VLAN. În acest caz, interfața VLAN1 are atribuită adresa IP 192.168.254.250, a fost activată și este operațională.

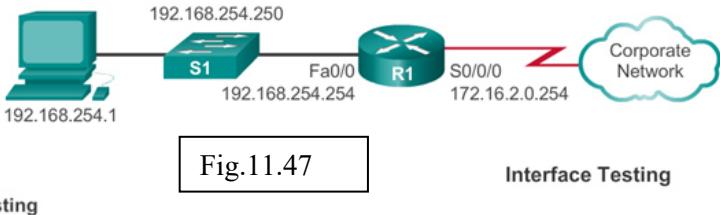
Acest output afișează și faptul că interfața FastEthernet0/1 este down. Acest lucru indică faptul că fie niciun dispozitiv nu este conectat la interfață, fie că dispozitivul care este conectat la această interfață are o interfață de rețea ce nu este funcțională.

Ieșirea arată și faptul că FastEthernet0/2 și FastEthernet0/3 sunt operaționale. Acest lucru este indicat prin faptul că ambele coloane, Status și Protocol, sunt **up**.

Switchul își poate testa de asemenea conectivitatea de nivel 3 cu ajutorul comenziilor **show ip interface brief** și **traceroute**. În acest exemplu, ambele comenzi, **ping** și **traceroute**, arată o conectivitate cu succes.

Este important de reținut faptul că o adresă IP nu este necesară pentru ca un switch să își efectueze sarcina de frame forwarding de nivel 2. O adresă IP este necesară numai atunci când

switchul va fi gestionat prin rețea cu ajutorul Telnet sau SSH. Dacă administratorul de rețea planuiește o conectare de la distanță pe switch dintr-o locație exterioară LANului local, atunci un default gateway trebuie să fie configurat.



```
R1# show ip interface brief
Interface      IP-Address      OK? Method Status      Protocol
FastEthernet0/0 192.168.254.254  YES NVRAM up        up
FastEthernet0/1 unassigned       YES unset  down      down
Serial0/0/0     172.16.0.254   YES NVRAM up        up
Serial0/0/1     unassigned       YES unset  administratively down
down

R1# ping 192.168.254.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.254.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R1# traceroute 192.168.0.1
Type escape sequence to abort.
Tracing the route to 192.168.0.1
 1 172.16.0.253 8 msec 4 msec 8 msec
 2 10.0.0.254 16 msec 16 msec 8 msec
 3 192.168.0.1 16 msec * 20 msec

S1# show ip interface brief
Interface      IP-Address      OK? Method Status      Protocol
Vlan1          192.168.254.250  YES manual up        up
FastEthernet0/1 unassigned       YES unset  down      up
FastEthernet0/2 unassigned       YES unset  up        up
FastEthernet0/3 unassigned       YES unset  up        up
<output omitted>

S1# ping 192.168.254.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.254.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

S1# traceroute 192.168.0.1
Type escape sequence to abort.
Tracing the route to 192.168.0.1
 1 192.168.254.254 4 msec 2 msec 3 msec
 2 172.16.0.253 8 msec 4 msec 8 msec
 3 10.0.0.254 16 msec 16 msec 8 msec
 4 192.168.0.1 16 msec * 20 msec
```

## 11.11 Gestionarea fișierelor de conFig.re IOS

Pentru a implementa și securiza o rețea mică, în sarcina administratorului de rețea intră și gestionarea fișierelor de conFig.re. Gestionarea fișierelor de conFig.re este importantă pentru scopurile de recuperare și backup în cazul producerii unui eveniment de defecțiune a unui dispozitiv.

Cisco IOS File System (IFS) oferă o singura interfață tuturor sistemelor de fișier utilizate de un router, inclusiv:

- *Sistemele de fișier de memorie flash.*
- *Sistemele de fișier de rețea (TFTP și FTP).*
- *Orice alte endpoint de citire sau scriere a datelor, cum ar fi NVRAM, conFig.ția running, ROM și altele.*

Cu Cisco IFS, toate fișierele pot fi vizualizate și clasificate (fișier imagine, text și așa mai departe), inclusiv fișierele de pe serverele de la distanță. De exemplu, este posibilă vizualizarea unui fișier de conFig.re de pe un server de la distanță pentru a verifica faptul că este fișierul corect de conFig.re, înainte de încărcarea acestuia pe router.

Cisco IFS permite administratorului să se deplaceze în directoare diferite și să listeze fișierele într-un director și să creeze subdirectoare în memoria flash sau pe un disk. Directoarele disponibile depind de dispozitiv.

Fig.11.48 afișează ieșirea comenzi **show file systems**, ce listă toate sistemele de fișiere disponibile pe un router Cisco 1941, în acest exemplu. Această comandă oferă informații utile, cum ar fi cantitatea de memorie liberă și disponibilă, tipul de sistem de fișier și permisiunile sale. Permisiunile includ read only (ro), write only (wo) și read and write (rw), arătate în coloana Flags din ieșirea comenzi.

Deși există multe sisteme de fișiere listate, cele ce prezintă interes pentru noi sunt sistemele de fișier tftp, flash și nvram.

De remarcat faptul că sistemul de fișier flash are un asterisk ce îl precede. Acest lucru indică faptul că flash este sistemul de fișier implicit actual. Bootable IOS este localizat în flash; prin urmare simbolul # este anexat la flash indicând faptul că acesta este un disk bootabil.

**Fișierul Sistemului Flash** – Fig.11.49 listează conținutul sistemului de fișier default actual, care în acest caz este flash, indicat prin asterisks precedente listate în Fig. anterioară. Există mai multe fișiere localizate în flash, însă ne interesează în mod special ultima listare. Aceasta este numele imaginii de fișier curent Cisco IOS ce rulează în RAM.

**Fișierul Sistemului NVRAM** – Pentru a vedea conținutul NVRAM, trebuie să schimbăm sistemul de fișier default curent folosind comanda **cd** (change directory), aşa cum se poate vedea în Fig.11.50. Comanda **pwd** (present working directory) verifică faptul că vedem directorul NVRAM. Comanda **dir** (directory) listează conținutul NVRAM. Deși există mai multe fișiere de config.re listate, un interes specific este asupra fișierului startup-conFig.ation.

File Systems					Flash
<b>Fig.11.48</b>					<b>Fig.11.49</b>
<pre>Router#show file systems File Systems:</pre>					<pre>Router#dir Directory of flash0:/</pre>
<pre>Size(b)      Free(b)     Type   Flags  Prefixes -           -          opaque  rw    archive: -           -          opaque  rw    system: -           -          opaque  rw    tmpsys: -           -          opaque  rw    null: -           -          network rw    tftp: * 256487424  183234560  disk    rw    flash0: flash:#</pre>					<pre>1 -rw-    2903 Sep 7 2012 06:58:26 +00:00 cpconfig-          19xx.cfg 2 -rw-    3000320 Sep 7 2012 06:58:40 +00:00 cpexpress.tar 3 -rw-    1038 Sep 7 2012 06:58:52 +00:00 home.shtml 4 -rw-    122880 Sep 7 2012 06:59:02 +00:00 home.tar 5 -rw-    1697952 Sep 7 2012 06:59:20 +00:00 securedesktop-          ios-3.1.1.45-k9.pkg 6 -rw-    415956 Sep 7 2012 06:59:34 +00:00 ssclient-win-          1.1.4.176.pkg 7 -rw-    67998028 Sep 26 2012 17:32:14 +00:00 c1900-          universalk9-          mz.SPA.152-4.M1.bin</pre>
					256487424 bytes total (183234560 bytes free)
NVRAM					
<b>Fig.11.50</b>					
<pre>Router#cd nvram: Router#pwd nvram:/ Router#dir Directory of nvram:/</pre>					
<pre>253 -rw-      1156    &lt;no date&gt;  startup-config 254 ----      5       &lt;no date&gt;  private-config 255 -rw-      1156    &lt;no date&gt;  underlying-config 1  -rw-      2945    &lt;no date&gt;  cwpw_inventory 4  ----      58      &lt;no date&gt;  persistent-data 5  -rw-      17      &lt;no date&gt;  ecfm_ieee_mib 6  -rw-      559     &lt;no date&gt;  IOS-Self-Sig#1.cer</pre>					
262136 bytes total (254779 bytes free)					

Cu sistemul de fișiere flash de pe switch Cisco 2960, putem copia fișierele de config.re și înregistra (încărca și descărca) imagini software.

Comanda de vizualizarea a sistemelor de fișiere de pe un switch Catalyst este aceeași ca cea de pe un router Cisco: **show file systems**, aşa cum este arătat și în Fig.11.51.

Mai multe comenzi UNIX de bază sunt suportate pe switchurile și routerele Cisco: **cd** pentru schimbarea unui sistem de fișiere sau directore, **dir** pentru a afișa directoarele de pe un sistem de fișier și **pwd** pentru a afișa directorul curent.

Switch#show file systems					
File Systems:					
*	32514048	20887552	flash	rw	flash:
-	-	opaque	rw	vb:	
-	-	opaque	ro	bs:	
-	-	opaque	rw	system:	
-	-	opaque	rw	tmpsys:	
65536	48897	nvram	rw	nvram:	
-	-	opaque	ro	xmodem:	
-	-	opaque	ro	ymodem:	
-	-	opaque	rw	null:	
-	-	opaque	ro	tar:	
-	-	network	rw	tftp:	
-	-	network	rw	rcp:	
-	-	network	rw	http:	
-	-	network	rw	ftp:	
-	-	network	rw	scp:	
-	-	network	rw	https:	
-	-	opaque	ro	cns:	

Fig.11.51

### 11.11.1 Back up și restaurarea fișierelor de config.re

Fișierele de config.re pot fi salvate/arhivate într-un fișier text utilizând Tera Term.

Cum se poate vedea și în Fig. , pașii sunt:

- **Pasul 1.** Din meniul File, apăsăm **Log**.
- **Pasul 2.** Alegem locația de salvare a fișierului. Tera Term va începe să capteze text.
- **Pasul 3.** După ce captura a început, executăm comanda **show running-config** sau **show startup-config** sau în promptul EXEC privilegiat. Textul afișat în fereastra de terminal va fi direcționat în fișierul ales.
- **Pasul 4.** După ce captura este completă, alegem **Close** din Tera Term: Log window.
- **Pasul 5.** Vizualizăm fișierul pentru a verifica faptul că nu a fost corupt.

### 11.11.2 Restaurarea config.ților text

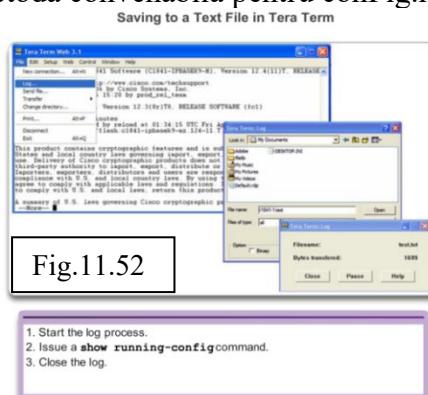
O config.ție poate fi copiată de pe un fișier pe dispozitiv. La copierea de pe un fișier text pe o fereastră terminal, IOS execută fiecare linie a config.ției text ca o comandă. Acest lucru înseamnă ca fișierul va necesita editare pentru a ne asigura de faptul că parolele criptate sunt în text clar și de faptul că textul non-comandă precum "--More--" și mesajele IOS sunt înlăturate. Acest proces va fi prezentat în laborator.

Pe CLI, dispozitivul trebuie să fie setat în modul de config.re globală pentru a primi comenzi din fișierul text ce sunt copiate în fereastra terminal.

Folosind Tera Term, pașii sunt:

- **Pasul 1.** Din meniul file, click **Send file**.
- **Pasul 2.** Localizăm fișierul ce va fi copiat în dispozitiv și apăsăm **Open**.
- **Pasul 3.** Tera Term va copia fișierul în dispozitiv.

Textul din fișier va fi aplicat ca și comandă în CLI și va deveni running-config.ion de pe dispozitiv. Aceasta este o metodă convenabilă pentru config.rea manuală a unui router.



### *11.11.3 Backupul ConFig.ților cu TFTP*

Copii ale fișierelor de conFig.re ar trebui să fie stocate ca fișiere de backup în cazul apariției unei probleme. Fișierele de conFig.re pot fi stocate pe un server Trivial File Transfer Protocol (TFTP) sau pe un drive USB. Un fișier de conFig.re ar trebui să fie de asemenea inclus în documentația de rețea.

Pentru a salva conFig.ția running sau conFig.ția startup pe un server TFTP folosim fie comanda **copy running-config tftp**, fie comanda **copy startup-config tftp**, arătate în Fig. . Urmăm pași de mai jos pentru a copia conFig.ția running pe un server TFTP:

- **Pasul 1.** Introducem comanda **copy running-config tftp**.
- **Pasul 2.** Introducem adresa IP a hostului pe care fișierul de conFig.re va fi stocat.
- **Pasul 3.** Introducem numele ce va fi atribuit fișierului de conFig.re.
- **Pasul 4.** Apăsăm tasta Enter pentru a confirma fiecare alegere.

### *11.11.4 Restaurarea conFig.ților cu TFTP*

Pentru a restaura conFig.ția running sau conFig.ția startup de pe un server TFTP folosim fie comanda **copy tftp running-config**, fie comanda **copy tftp startup-config**. Urmăm acești pași pentru a restaura conFig.ția running de pe un server TFTP:

- **Pasul 1.** Introducem comanda **copy tftp running-config**.
- **Pasul 2.** Introducem adresa IP a hostului pe care fișierul de conFig.re va fi stocat.
- **Pasul 3.** Introducem numele ce va fi atribuit fișierului de conFig.re.
- **Pasul 4.** Apăsăm tasta Enter pentru a confirma fiecare alegere.

```
Router#copy running-config tftp
Remote host []? 131.108.2.155
Name of configuration file to write[tokyo-config]?tokyo.2
Write file tokyo.2 to 131.108.2.155? [confirm]
Writing tokyo.2 !!!!!!! [OK]
```

Caracteristica de stocare Universal Serial Bus (USB) permite anumitor modele de routere Cisco să suporte drivere flash USB. Caracteristica USB flash oferă o capacitate de stocare secundară opțională și un dispozitiv suplimentar de boot. Imaginele, conFig.țile și alte fișiere pot fi copiate pe sau de pe un Cisco USB de memorie flash cu aceeași încredere prin care se stochează și restaurează fișierele cu ajutorul cardului Compact Flash. În plus, routerele cu servicii integrate modulare pot boota orice imagine IOS Software salvată pe memoria USB flash.

Modulele flash Cisco USB sunt disponibile în versiuni de 64MB, 128 MB și 256MB.

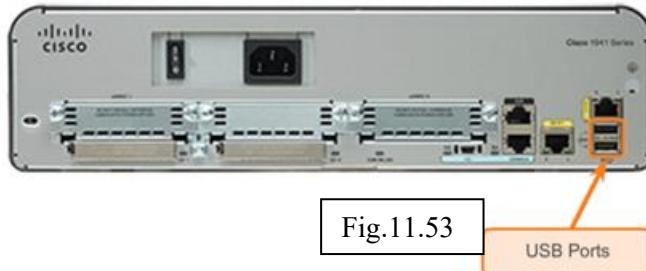
Pentru a fi compatibil cu un router Cisco, un drive USB flash trebuie să fie formatat într-un format FAT16. Dacă nu, comanda show file systems va afișa o eroare ce indică un sistem de fișier incompatibil.

Următorul exemplu este pentru utilizarea comenzii dir pe un sistem de fișier USB:

```
Router# dir usbflash0:
Directory of usbflash0:/
1 -rw- 30125020 Dec 22 2032 05:31:32 +00:00 c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
```

Ideal, USB flash poate păstra mai multe copii ale IOSului și mai multe conFig.ții de router. USB flash permite unui administrator să mute mai ușor și să copieze respectivele fișiere IOS și conFig.țile de la router la router, de mai multe ori, iar procesul de copiere poate avea loc de mai multe ori mai rapid decât peste un LAN sau WAN. Reținem faptul că IOS ar putea să nu

recunoască dimensiunea adecvată a USB flash, însă acest lucru nu înseamnă neapărat faptul că flash este neacceptat. În plus, porturile USB de pe un router sunt de obicei USB 2.0, cum se vede și în Fig.11.53.



### 11.11.5 Configurații de backup cu un drive USB flash

La copierea pe un port USB, este o bună idee de efectuare a comenzi **show file systems** pentru verificarea faptului că USB drive este acolo și confirmă numele, aşa cum se poate vedea în Fig. 11.54.

Apoi, folosim comanda **copy run usb flash0:/** pentru a copia fișierul de config.re pe driveul flash USB. Ne asigurăm că utilizăm numele lui flash drive, aşa cum este indicat în sistemul de fișier. Slash este opțional, însă indică directorul rădăcină al driveului flash USB.

IOS va solicita numele de fișier. Dacă fișierul există deja pe drive flash USB, routerul va solicita suprascriere, aşa cum se poate vedea în Fig. 11.55

Folosim comanda **dir** pentru a vedea fișierul pe driveul USB și folosim comanda **more** pentru a vedea conținutul, aşa cum se poate vedea în Fig.11.56.

### 11.11.6 Restaurarea config.ților cu un drive flash USB

Pentru a copia înapoi fișierul, va fi necesară editarea fișierului USB R1-Config cu un editor de text pentru a-l face un fișier config valid; altfel, există o multime de intrări ce reprezintă comenzi invalide și nici-o interfață nu va fi ridicată.

**R1#copy usbflash0:/R1-Config running-config**

*Destination filename [running-config]?*

```
R1#show file systems
File Systems:
  Size(b)  Free(b)   Type   Flags  Prefixes
  -        -         opaque  rw      archive:
  -        -         opaque  rw      system:
  -        -         opaque  rw      tmpsys:
  -        -         opaque  rw      null:
  -        -         network rw      tftp:
  -        -         disk    rw      flash0: flash:#*
  -        -         disk    rw      flash1:
  262136   249270   nvram   rw      nvram:
  -        -         opaque  wo      syslog:
  -        -         opaque  rw      xmodem:
  -        -         opaque  rw      ymodem:
  -        -         network rw      rcp:
  -        -         network rw      http:
  -        -         network rw      ftp:
  -        -         network rw      scp:
  -        -         opaque  ro      tar:
  -        -         network rw      https:
  -        -         opaque  ro      cns:
  4050042880 3774152704 usbflash  rw      usbflash0:
```

An orange arrow points from a callout box labeled "Shows the USB port and name: \"usbflash0\"." to the "usbflash" entry in the table.

Fig.11.54

```
R1#copy running-config usbflash0:
Destination filename [running-config]? R1-Config
5024 bytes copied in 0.736 secs (6826 bytes/sec)
```

Copying to USB flash drive, and no file pre-exists.

Fig.11.55

```
R1#copy running-config usbflash0:
Destination filename [running-config]? R1-Config
%Warning: There is a file already existing with this name
Do you want to over write? [confirm]
5024 bytes copied in 1.796 secs (2797 bytes/sec)
```

Copying to USB flash drive, and the same configuration file already exists on the drive.

```

R1#dir usbflash0:/  

Directory of usbflash0:/  

  1  drw-    0 Oct 15 2010 16:28:30 +00:00 Cisco  

  16 -rw-  5024 Jan  7 2013 20:26:50 +00:00 R1-Config  

4050042880 bytes total (3774144512 bytes free)  

R1#more usbflash0:/R1-Config  

!  

! Last configuration change at 20:19:54 UTC Mon Jan  7 2013 by  

admin version 15.2  

service timestamps debug datetime msec  

service timestamps log datetime msec  

no service password-encryption  

!  

hostname R1  

!  

boot-start-marker  

boot-end-marker  

!  

logging buffered 51200 warnings  

!  

no aaa new-model  

!
no ipv6 cef

```

Fig.11.56

### 11.11.7 Servicii de rutare integrate

Utilizarea rețelei nu se limitează la întreprinderi mici și organizații mari.

Un alt mediu ce profită din ce în ce mai mult de tehnologia de rețea este locuința. Rețelele de domiciliu sunt utilizate pentru oferirea conectivității și împărțirea Internetului printre mai multe sisteme de computer personale și laptopuri din locuințe. De asemenea permit indivizilor să profite de multiple servicii, cum ar fi partajarea imprimării la o imprimantă de rețea, stocarea centralizată a fotografiilor, muzicii și filmelor pe un dispozitiv network attached storage (NAS); de asemenea permite altor dispozitive de utilizator, cum ar fi tablete, telefoane mobile și chiar electrocasnicelor, precum un televizor, să aibă acces la serviciile de Internet.

O rețea de domiciliu este foarte asemănătoare cu o rețea de întreprindere mică. Însă, unele rețele de domiciliu și mai multe rețele de întreprindere mică nu necesită dispozitive de volum mare, cum ar fi un router dedicat și switchuri. Dispozitivele de dimensiune mai mică, atât timp cât oferă aceeași funcționalitate de routare și switching, reprezintă singura necesitate. Din acest motiv, multe rețele de domiciliu și întreprindere mici utilizează serviciul unui dispozitiv multi-function.

Pentru scopul acestui curs, dispozitivele multi-function vor fi referite ca routere integrate.

Un router integrat este ca și cum am avea mai multe dispozitive diferite conectate împreună. De exemplu, conexiunea dintre un switch și router are loc, însă are loc intern. Atunci când un pachet este trimis de la un dispozitiv la altul, în aceeași rețea, switchul integrat va trimite automat pachetul la dispozitivul destinație. Dacă un pachet este trimis la un dispozitiv de pe o rețea de la distanță, switchul integrat va transmite pachetul conexiunii routerului intern. Routerul intern va determina apoi cea mai bună cale și va transmite pachetul în mod corespunzător căii determinate.

Multe routere integrate oferă capacitați atât de conexiune wireless, cât și switching cablat, și servesc ca access point (AP) în rețeaua wireless, aşa cum se poate vedea și în Fig.11.57. Conectivitatea wireless este o modalitate polulară, flexibilă și eficientă din punct de vedere a costului pentru locuințe și întreprinderi pentru oferirea serviciilor de rețea dispozitelor finale.

Imaginiile listează unele avantaje și considerații comune pentru utilizarea wireless.

Pentru a suporta rutarea, switching și conectivitatea wireless, multe caracteristici suplimentare pot fi disponibile pe un router integrat, cum ar fi : serviciu DHCP, un firewall și chiar network attached storage services.

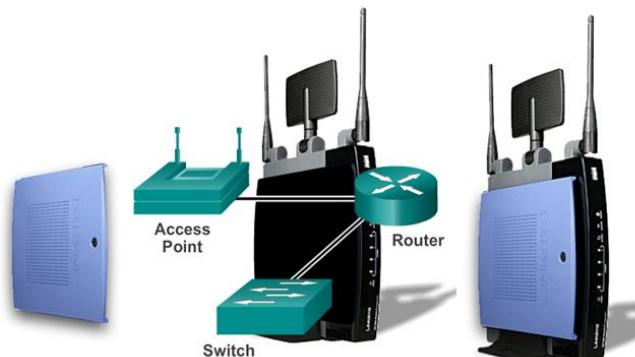
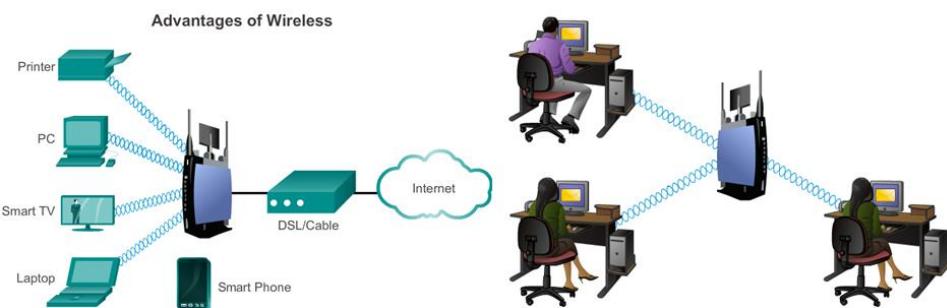


Fig.11.57

Limitations of Wireless



Routerele integrate pot varia de la dispozitive mici destinate pentru locuințe și aplicații de întreprindere mică la dispozitive mai puternice ce suportă filiale de întreprindere.

Un exemplu de acest tip de router integrat este un Linksys wireless router, arătat în Fig.. Acest tip de router integrat este simplu în design și nu are în mod normal componente separate. Acest lucru reduce costul dispozitivului. Însă, în cazul producerii unui eșec, nu este posibilă înlocuirea unei singure componente stricte. Prin urmare, se crează un singur punct de eșec și nu sunt optimizate pentru oricare funcție.

Un alt exemplu de router integrat este Cisco integrated services router sau ISR. Familia de produse Cisco ISR oferă un rangu mare de produse, inclusiv cele destinate pentru medii de small office și home office, cât și cele destinate pentru rețele mari. Multe dintre ISR oferă modularitate și au componente separate pentru fiecare funcție, cum ar fi o componentă de switch și o componentă router. Acest lucru permite componentelor individuale să fie adăugate, înlocuite și actualizate dacă este necesar.

Toate routerele integrate permit setări de bază de configurație cum ar fi parole, adrese IP și setări DHCP, ce sunt aceleași fie că dispozitivul este utilizat pentru a conecta hosturi prin cablu, fie prin wireless. Însă, dacă utilizăm funcționalitatea wireless, parametrii de configurație suplimentari sunt necesari, cum ar fi setări pentru modul wireless, SSID și canalul wireless.

Linksys: Model WRT300N2

Linksys: Model WRT300N2



Fig.11.58

**Front View**

The Linksys is a simplified, low-cost device that carries out the functionality of multiple network devices (switch, router, wireless access point).

Light emitting diodes (LEDs) indicate the connection status of each port.

Click the LEDs for a description.

**Rear View**

When connecting a local network using a multifunction device it is important that all local devices are connected to the switch ports.

Click the ports for a description.

## 11.12 Wireless

Modul wireless se referă la setările wireless IEEE 802.11 standard pe care rețeaua le va vedea. Există patru modificări ale standardului IEEE 802.11 ce descriu caracteristici diferite pentru comunicațiile wireless; ele sunt IEEE 802.11a, IEEE 802.11b, IEEE 802.11g și IEEE 802.11n. Fig. 1 listează mai multe informații cu privire la fiecare standard.

Multe routere wireless integrate suportă 802.11b, 802.11g și 802.11n. Cele trei tehnologii sunt compatibile, însă toate dispozitivele din rețea trebuie să funcționeze cu același standard comun tuturor dispozitivelor. De exemplu, dacă un router 802.11n este conectat la un laptop cu 802.11n, rețeaua va funcționa ca un standard 802.11n. Însă, adăugată o imprimantă wireless 802.11b la rețea, ambele, routerul și laptopul, se vor întoarce să folosească standardul mai înalt 802.11b pentru a comunica toate echipamentele. Prin urmare, păstrarea dispozitivelor wireless mai vechi în rețea va face ca întreaga rețea să funcționeze mai lent. Este important să reținem acest lucru pentru atunci când ne decidem dacă pastrăm sau nu dispozitivele wireless mai vechi.

### 11.12.1 Service Set Identifier (SSID)

Pot exista mai multe alte rețele wireless în aria de acoperire a unei rețele. Este important ca dispozitivele wireless să se conecteze la WLAN corect. Acest lucru se realizează prin utilizarea unui Service Set Identifier (SSID).

SSID este cu nume case-sensitive, alpha-numeric pentru rețeaua wireless din locuință. Numele poate avea până la 32 de caractere în lungime. SSID este utilizat pentru a spune dispozitelor wireless căruia WLAN aparțin și cu ce alte dispozitive pot comunica. Indiferent de tipul de instalare WLAN, toate dispozitivele wireless dintr-un WLAN trebuie să fie configurate cu același SSID pentru a comunica.

### 11.12.2 Canalul wireless

Canalele sunt create prin divizarea spectrumului RF disponibil. Fiecare canal este capabil să transporte o conversație diferită. Aceast lucru este similar cu modul în care canalele de televiziune sunt transmise pe un singur mediu. Mai multe APs pot funcționa în apropierea unuia față de altul, atât timp ce utilizează canale diferite de comunicare.



Măsuri de securitate ar trebui să fie planificate și configurate înaintea conectării AP la rețea sau ISP.

Așa cum se poate observa în Fig. 1, unele dintre măsurile de securitate de bază sunt:

- Schimbarea valorilor default pentru SSID, numele de utilizator și parole.
- Dezactivarea broadcast SSID.
- Configurarea criptării folosind WEP sau WPA.

Criptarea este procesul de transformare a datelor astfel încât și dacă sunt interceptate, sunt inutile.

#### 11.12.3 Wired Equivalency Protocol (WEP)

WEP este o caracteristică avansată de securitate ce criptează traficul de rețea care circulă prin intermediul aerului. WEP folosește chei preconFig.te pentru a cripta și decripta datele, aşa cum se poate vedea în Fig. 2.

O cheie WEP este introdusă ca un string de numere și litere și este în general de 64 de biți sau 128 de biți. În unele cazuri, WEP suportă și chei de 256 de biți. Pentru a simplifica crearea și introducerea acestor chei, mai multe dispozitive includ o opțiune de Passphrase. Passphrase este un mod ușor de reținere a cuvântului sau frazei utilizată pentru a genera automat o cheie.

Pentru ca WEP să funcționeze, AP, cât și toate dispozitivele wireless ce au permisiune de acces la rețea trebuie să aibă aceeași cheie WEP introdusă. Fără această cheie, dispozitivele nu vor fi capabile să înțeleagă transmisia wireless.

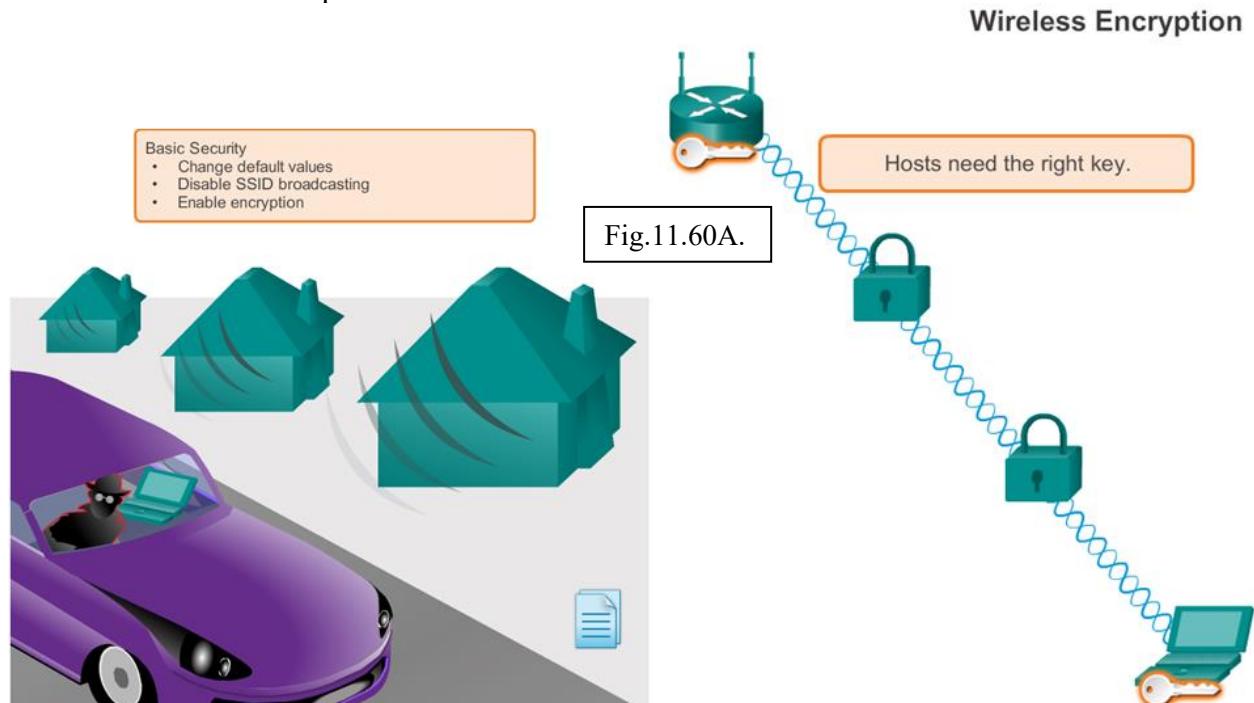
Există unele slăbiciuni cu WEP, inclusiv utilizarea unei chei statice pe toate dispozitivele ce au activat WEP. Există aplicații disponibile atacatorilor ce pot fi folosite pentru descoperirea cheii WEP. Aceste aplicații sunt disponibile în Internet. O dată ce atacatorul a aflat cheia, are acces complet la toate informațiile transmise.

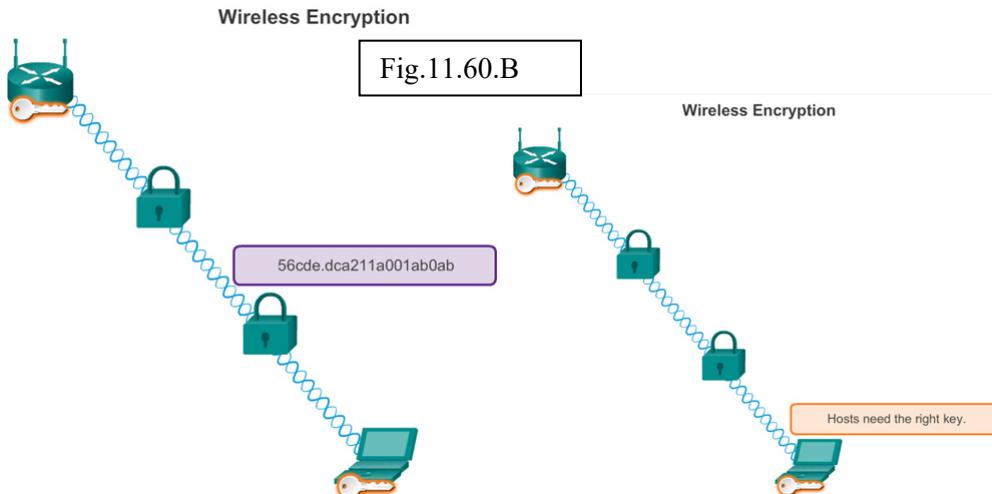
O modalitate de a combate această vulnerabilitate este schimbarea frecventă a cheii. Un alt mod este utilizarea unei forme mai avansate și securizate de criptare numită Wi-Fi Protected Access (WPA).

#### 11.12.4 Wi-Fi Protected Access (WPA)

WPA utilizează de asemenea chei de criptare de 64 de biți până la 256 de biți. Însă, WPA, spre deosebire de WEP, generează chei noi dinamice de fiecare dată când un client stabilește o conexiune cu AP. Din acest motiv, WPA este considerat mai sigur decât WEP deoarece este mult mai dificil de "spart".

Există mai multe implementări de securitate ce pot fi configurate pe un AP wireless, inclusiv filtrarea de adresă MAC, autentificarea și filtrarea traficului. Însă, aceste implementări de securitate nu intră în scopul acestui curs.





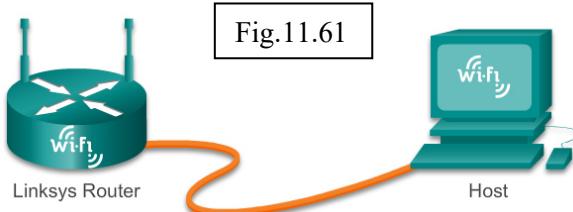
### 11.13 ConFig.rea routerului integrat

Un router Linksys wireless este un dispozitiv comun utilizat în rețele de locuință și de întreprindere mici și va fi utilizat în acest curs pentru demonstrarea conFig.ărilor de bază pe un router integrat. Un dispozitiv Linksys tipic oferă de la cinci la opt porturi Ethernet pentru conectivitate cablată, pentru a acționa ca un wireless access point. Dispozitivul Linksys acționează de asemenea ca un server DHCP și ca un mini-webserver ce suportă web bazându-se pe graphical user interface (GUI).

#### 11.13.1 Accesarea și conFig.rea unui Linksys Router

Inițial, accesăm routerul prin cablarea unui computer la unul dintre porturile routerului LAN Ethernet, aşa cum se poate vedea în Fig. . O dată cablat, dispozitivul conectat automat va obține informații de adresare IP, inclusiv o adresă de default gateway, de la routerul integrat. Adresa de default gateway este adresa IP a dispozitivului Linksys. Verificăm setările de rețea ale computerului folosind comanda **ipconfig /all** pentru a obține această adresă. Acum putem introduce adresa IP într-un browser web de pe computer pentru a accesa web-based conFig.ation GUI.

Dispozitivul Linksys are un conFig.ăre implicită ce permite switching și servicii de rutare de bază. Este de asemenea conFig.ăt implicit ca un server DHCP. Sarcini de conFig.ăre de bază, cum ar fi schimbarea numelui de utilizator și a parolei隐密的, schimbarea adresei default IP a Linksys și chiar rangeurile de adresă IP DHCP implicită, ar trebui să fie efectuate înaintea ca AP să fie conectat la o rețea live.



Pentru a activa conectivitate wireless, modul wireless, SSID, canalul RF și orice mecanism de Securitate droit, trebuie să fie conFig.ăte.

Mai întâi, selectăm modul de wireless corect, aşa cum se poate vedea în Fig.11.62. La selectarea modului sau standardului wireless, fiecare mod include o anumită cantitate de

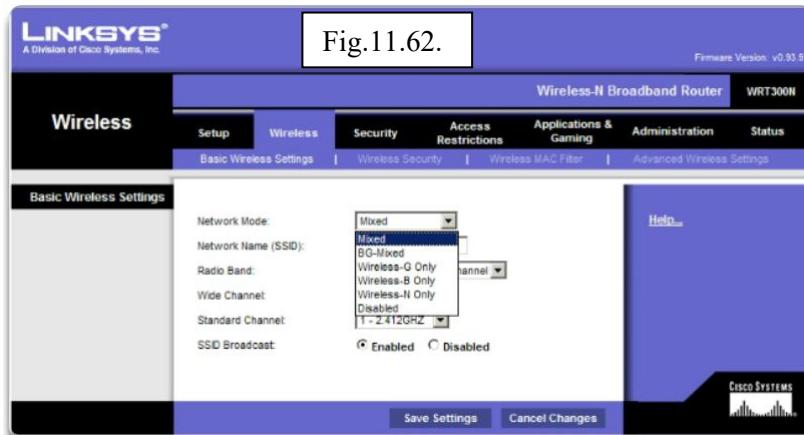
overhead. Dacă toate dispozitivele din rețea folosesc același standard, selectarea modului asociat cu respectivul standard limitează cantitatea de overhead suportată. De asemenea, crește securitatea prin interzicerea dispozitivelor cu standarde diferite să se conecteze. Însă, dacă dispozitivele ce utilizează standarde diferite trebuie să acceseze rețea, poate fi selectat modul mixed. Performanța rețelei va scădea având în vedere overhead suplimentar al tuturor modurilor suportate.

Apoi, setăm SSID. Toate dispozitivele ce doresc să participe la WLAN trebuie să utilizeze același SSID. Din motive de securitate, SSID default ar trebui să fie schimbat. Pentru a permite detectia ușoară a WLAN de către clienți, SSID este broadcast implicit. Este posibilă dezactivarea caracteristicii de broadcast a SSID. Dacă SSID nu este broadcast, clienții wireless vor trebui să aibă această valoare configată manual.

Alegerea canalului RF utilizat pe routerul integrat trebuie să se facă relativ cu alte rețele wireless din împrejurime.

Rețelele wireless adiacente trebuie să utilizeze canale ce nu se suprapun pentru a optimiza throughputul. Multe puncte de acces oferă acum o alegere de a permite routerului să localizeze automat canalul cel mai puțin congestionat.

La final, alegem mecanismul de criptare preferat și introducem o cheie sau passphrase.



### 11.13.2 Configurarea unui client wireless

Un host wireless, sau un client, este definit ca orice dispozitiv ce conține wireless NIC și software de client wireless. Acest software de client permite ca hardware să participe în WLAN. Dispozitivele includ: unele telefoane mobile, laptopuri, desktop PCuri, imprimante, televizoare, tablete etc.

Pentru ca un client wireless să se conecteze la WLAN, setarea de configurație de client trebuie să corespundă cu cea de pe routerul wireless. Acest lucru include SSID, setările de securitate și informațiile de canal (dacă respectivul canal a fost setat automat). Aceste setări sunt specificate în softwareul de client.

Softwareul de client wireless utilizat poate fi software integrat în sistemul de operare al dispozitivului sau poate fi un utilitar software wireless, de sine stătător, descarcabil, destinat pentru interacțiunea cu wireless NIC.

O dată ce softwareul de client este configurație, verificăm legătura dintre client și AP.

Deschidem ecranul Link Information wireless pentru a afișa informații cum ar fi: connection data rate, statusul conexiunii și canalul wireless utilizat, aşa cum se poate vedea în Fig.. Caracteristica Link Information, dacă este disponibilă, afișează puterea semnalului curent și calitatea semnalului wireless.

În plus, pentru a verifica statusul conexiunii, verificăm faptul că datele pot fi într-adevăr transmise. Unul dintre cele mai comune teste de verificare a transmisiei de date cu succes este testul **ping**. Dacă **ping** e cu succes, transmisia datelor este posibilă.



### 11.14 Concluzii Capitolul 11

Pentru a îndeplini cerințele de utilizator, chiar și rețelele mici necesită planificare și design, aşa cum se poate vedea în Fig.11.64. Planificarea asigură faptul că toate cerințele, factorii de cost și opțiunile de implementare sunt luate în considerare. O parte importantă a designului de rețea este încrederea, scalabilitatea și disponibilitatea.



#### When planning any network consider...

- Cost
- Ports
- Speed
- Expandability
- Manageability

Fig.11.64

Suportarea și creșterea unei rețele mici necesită ca noi să fim familiari cu protocolele și aplicațiile de rețea ce rulează în rețea. Analizoarele de protocol permit unui profesionist de rețea să compileze rapid informații statistice cu privire la fluxurile de trafic dintr-o rețea. Informațiile adunate de către analizorul de protocol sunt analizate în funcție de sursa și destinația traficului, cât și în funcție de tipul de trafic transmis. Această analiză poate fi folosită de către un tehnician de rețea pentru a lua decizii despre modul în care să gestioneze traficul mai eficient. Protocolele de rețea comune includ: DNS, Telnet, SMTP, POP, DHCP, HTTP și FTP.

Este important să luăm în considerare amenințările de securitate și vulnerabilitățile atunci când planificăm o implementare de rețea. Toate dispozitivele de rețea trebuie să fie securizate.

Acestea includ routerele, switchurile, dispozitivele de utilizator final și chiar dispozitivele de securitate. Rețelele trebuie să fie protejate împotriva softwareului rău intenționat, cum ar fi virușii, caii Troieni și vermii. Softwareul antivirus poate detecta mulți viruși și aplicații cai Troian și poate preveni împrăștierea lor în rețea. Cel mai eficient mod de combatere a atacului vierme este descărcarea actualizărilor de securitate de la un furnizor de sistem de operare și “peticirea” tuturor sistemelor vulnerabile.

Rețelele trebuie să fie de asemenea protejate de atacurile de rețea. Atacurile de rețea pot fi clasificate în trei mari categorii: de recunoaștere, atacuri de acces și denial of service. Există multe moduri de protecție a unei rețele de atacurile de rețea.

- Serviciile de securitate de rețea Authentication, Authorization, și Accounting (AAA, sau “triple A”) oferă cadrul primar al setării controlului accesului pe un dispozitiv de rețea. AAA este un mod de control asupra celor ce au permis accesul la rețea (autentificare), ce pot face în acest timp (autorizare) și vizualizarea acțiunilor efectuate în timpul accesării rețelei (contabilitate).
- Un firewall este unul dintre cele mai eficiente instrumente de securitate disponibile pentru protejarea utilizatorilor interni din rețea de amenințările externe. Un firewall se află între două sau mai multe rețele și controlează traficul dintre ele și ajută la prevenirea accesului neautorizat.
- Pentru a proteja dispozitivele de rețea, este important să utilizăm parole puternice. De asemenea, atunci când accesăm dispozitivele de rețea de la distanță, este recomandat să activăm SSH, în schimb Telnet nesecurizat.

După ce rețeaua a fost implementată, un administrator de rețea trebuie să fie capabil să monitorizeze și să mențină conectivitatea rețelei. Există mai multe comenzi disponibile pentru acest lucru. Pentru testarea conectivității rețelei cu destinații locale și de la distanță, sunt utilizate comenzi precum **ping**, **telnet** și **traceroute**.

Pe dispozitivele cu IOS, comanda **show version** poate fi utilizată pentru a verifica și depana unele componente hardware și software de bază folosite în timpul procesului de bootup. Pentru a vizualiza informații pentru toate interfețele de pe un router, comanda **show ip interface** este utilizată. Comanda **show ip interface** poate fi de asemenea utilizată pentru vizualizarea unui output mai compact decât al comenzi **show ip interface**. **Cisco Discovery Protocol** (CDP) este un protocol proprietar Cisco ce rulează la nivelul legătură de date. Deoarece CDP operează la nivelul legătură de date, două sau mai multe dispozitive de rețea Cisco, cum ar fi routerele ce suportă protocoale diferite de nivel rețea, pot învăța unele despre celelalte chiar dacă nu există conectivitate de nivel 3.

Fișierele de configurație IOS, cum ar fi startup-config sau running-config, ar trebui să fie arhivate. Aceste fișiere pot fi salvate într-un fișier text sau stocate pe un server TFTP. Unele modele de routere au de asemenea un port USB și un fișier ce poate fi încărcat pe un drive USB. Dacă este necesar, aceste fișiere pot fi copiate pe router și/sau switch de pe serverul TFTP sau driverul USB.

Utilizarea rețelei nu este limitată la întreprinderi mici și organizații mari. Un alt mediu ce profită din ce în ce mai mult de tehnologia de rețea este locuința. O rețea de domiciliu este similară cu o rețea de întreprindere mică. Însă, multe rețele de domiciliu (și multe rețele de întreprindere mică) nu necesită dispozitive de volum ridicat, cum ar fi routere dedicate sau switchurile. În schimb, multe rețele de domiciliu utilizează un singur dispozitiv multi-function. Pentru scopul acestui curs, dispozitivele multi-function vor fi referite ca routere integrate. Multe routere integrate oferă capabilități atât de conexiune wireless, cât și de switching cablat și servesc ca access point (AP) în rețeaua wireless. Pentru a permite conectivitate wireless, modul wireless, SSID, canalul RF și orice mecanism de securitate dorit trebuie să fie implementate.

# BIBLIOGRAFIE

- [1] A. Tanenbaum, *Rețele de calculatoare (ediția a patra)*, Byblos, Tg.Mureș, 2003
- [2] R. Stevens, B. Fenner, A. Rudoff, *UNIX Network Programming* Volume 1, Third Edition: The Sockets Networking API, Addison Wesley, 2003
- [3] M. Popa, *Bazele modelării Rețelelor de Calculatoare*, Ed. Universității din București,, București, 2004.
- [4] T. Thomas, *Primi pași în Securitatea Rețelelor*, ciscopress.com, 2005.
- [5] S. Buraga, G. Ciobanu, *Atelier de programare în rețele de calculatoare*, Polirom, Iași, 2001: <http://www.infoiasi.ro/~lrc/>
- [6] D. Comer, *Internetworking With TCP/IP*, vol.I: Principles, Protocols, and Architecture (2nd edition), Prentice Hall, New Jersey, 1991.
- [7] D. Comer, D. Stevens, Internetworking with TCP/IP: vol.III: Client-Server Programming and Applications, Prentice Hall, New Jersey, 1993.
- [8] S. Androusellis-Theotokis, D. Spinellis, A Survey of Peer-to-Peer Content Distribution Technologies, ACM Computing Surveys, 36(4):335-371, December 2004
- [9] M. Mallick, Mobile and Wireless Design Essentials, John Wiley & Sons, 2003
- [10] S. Raab et al., Mobile IP Technology and Applications, Cisco Press, 2005
- [11] J. Doyle, CCIE Professional Development: Routing TCP/IP, Volume I, Macmillan Technical Publishing, 1998
- [12] K. Robbins, S. Robbins, UNIX Systems Programming: Communication, Concurrency, and Threads, Prentice Hall PTR, 2003
- [13] R. Stevens, TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley Longman, 1994 – sursele pot fi gasite la adresa <http://www.kohala.com/start/tcpipiv1.tar.Z>
- [14] R. Stevens, TCP/IP Illustrated, Volume 2: The Implementation, Addison-Wesley Longman, 1995
- [15] R. Stevens, TCP/IP Illustrated, Volume 3: TCP for Transaction, HTTP, NNTP, and the UNIX Domain Protocols, Addison-Wesley Longman, 1996
- [16] D. Acostăchioie, Programare C și C++ pentru Linux, Polirom, Iași, 2002: <http://www.biosfarm.ro/~dragos/prg>
- [17] D. Acostăchioie, Securitatea sistemelor Linux, Polirom, Iași, 2003: <http://www.biosfarm.ro/~dragos/sec>
- [18] J. Martin, J. Leben, TCP/IP Networking, Prentice Hall, New Jersey, 1994
- [19] A. Abbas, Grid Computing: A Practical Guide to Technology and Applications, Charles River Media, 2004
- [20] D. Acostăchioie, S. Buraga, Utilizare Linux, Polirom, Iași, 2004: <http://www.infoiasi.ro/~linux/>
- [21] D. Acostăchioie, Administrarea și conFig.rea sistemelor Linux (ediția a treia), Polirom, Iași, 2006: <http://www.biosfarm.ro/~dragos/admin>
- [22] M. Ben-Ari, Principles of Concurrent Programming, Prentice Hall International, 1982
- [23] S. Dixit, R. Prasad (eds.), Wireless IP and Building the Mobile Internet, Artech House, 2003
- [24] A. Grama et al., Introduction to Parallel Computing (2nd edition), Addison Wesley, 2003
- [25] E. Hall, Internet Core Protocols: The Definitive Guide, O'Reilly, 2000
- [26] G. Held, Ethernet Networks (4th edition), John Wiley & Sons, 2003
- [27] A. Jones, J. Ohlund, Network Programming for Microsoft Windows, Microsoft Press, 1999
- [28] A. Kshemkalyani, M. Singhal, Distributed Computing. Principles, Algorithms, and Systems, Cambridge University Press, 2008
- [29] C. McNab, Network Security Assessment, O'Reilly, 2004
- [30] D. Naik, Internet. Standards and Protocols, Microsoft Press, 1998
- [31] K. Robbins, S. Robbins, Unix Systems Programming: Communication, Concurrency, and Threads, Prentice Hall PTR, New Jersey, 2003
- [32] M. Rockkind, Advanced UNIX Programming, Prentice Hall, New Jersey, 1985
- [33] N. Shi (ed.), Mobile Commerce Applications, Idea Group Publishing, 2004

- [34] R. Stevens, Advanced UNIX Programming in the UNIX Environments, Addison-Wesley, Reading MA, 1992
- [35] A. Tanenbaum, Modern Operating Systems, Addison-Wesley, Reading MA, 2001
- [36] A. Wells, Grid Application Systems Design, CRC Press, 2008
- [37] \* \* \*, Peer-to-Peer Application Development, Hungry Minds, 2002
- [38] Utilitare pentru shell (traducere de Adrian Haisan)
- [39] Utilitare pentru rețea (tutorial tradus de Diana Popovici)
- [40] TCP/IP și rețelele (tutorial tradus de Ionuț Lucaș)
- [41] Principii UNIX și Internet (HOWTO) (traducere de Florin Bandaș)
- [42] ConFig.rea TCP/IP (traducere de Luminița Moruz și Gabriel Manolache)
- [43] Specificația protocolului IP (traducere de Raluca Gordân)
- [44] Împărțirea în subrețele (traducere de Corina Rotaru)
- [45] Protocole de rutare (o prezentare de Ecaterina Valică și Raluca Moroșan)
- [46] Specificația protocolului ICMP (traducere de Valentin Cilibiu)
- [47] Specificația protocolului RIP (traducere de Leontina Munteanu și Andreea Tutoveanu)
- [48] Specificații formale pentru Exterior Gateway Protocol (traducere de Daniel Onacă)
- [49] Specificația protocolului TCP (traducere de Cătălin Bulancea)
- [50] Specificația protocolului TELNET (traducere de Raluca Motrescu)
- [51] Specificația protocolului FTP (traducere efectuată de Adrian Haisan și Cătălin Mihai Apostu)
- [52] Specificația protocolului TFTP (traducere de Raluca Lăcătușu)
- [53] Specificația protocolului POP3 (traducere de Adina Slușer)
- [54] Fire de execuție în Linux (tutorial tradus de Ana-Maria Cucu)
- [55] Introducere în firele de execuție POSIX (traducere de Bogdan Manolache)
- [56] Programarea rețea în Windows – Winsock FAQ (traducere de Cristian Baciu și Sabin Nemțisor)
- [57] Introducere în MySQL (traducere de Carmen Leonte)
- [58] NFS și NIS (traducere de Ana-Roxana Tubultoc)
- [59] Principles of system administration
- [60] Booting, Init, and Shutdown
- [61] Users and Logins
- [62] System Names and Access Permissions
- [63] Filesystems and Disks
- [64] Managing Processes
- [65] Managing Resources
- [66] TCP/IP and Networks
- [67] Configuring TCP/IP
- [68] TCP/IP Utilities