

## Laborator 5

### Nivelul rețea

Nivelul rețea are ca sarcină preluarea pachetelor de la sursă și transferul lor către destinație, chiar dacă ajungerea la destinație poate necesita parcurgerea mai multor segmente de rețea. Această funcție contrastează clar cu cea a nivelului legătură de date, care avea un scop mult mai modest și anume de a transfera cadre de la un capăt al unui fir la celălalt.

O adresă Mac ar putea fi comparată cu numele unei persoane, pe când o adresă de rețea cu adresa la care locuiește acea persoană. Dacă acea persoană s-ar muta, numele ar rămâne același, însă adresa automat se va schimba astfel încât aceasta să “pointeze” spre locația acelei persoane. Dispozitivele aflate în rețea, cum ar fi computerele și routerele au atât o adresă MAC cât și o adresă de rețea. La mutarea fizică a unui calculator într-o altă rețea, adresa lui MAC va rămâne aceeași, dar va trebui să îi fie schimbată adresa de rețea (adresa IP este cel mai important exemplu de adresă de rețea și de aici încolo când ne vom referi la adresa de rețea ne vom referi de fapt la adresa IP).

Protocoloalele aflate la nivelul Rețea (Layer 3) folosesc o altă tehnică de identificare față de Nivelul Legătură de Date (Layer 2).

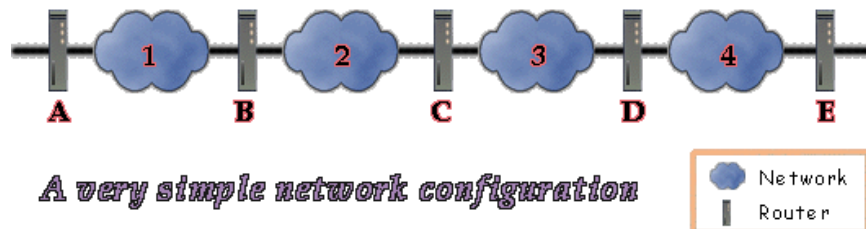
Adresarea folosind adresele MAC (cea de la Layer 2 sau Legătură de Date) este destul de obtuză adesele gazdelor fiind aleatoare, făcând dificilă localizarea dispozitivelor aflate în alte rețele. În schimb adresarea de la nivelul Rețea folosește o schemă de adresare **ierarhică** care permite:

- adresă unică pentru un dispozitiv în cadrul granițelor unei rețele
- o metodă pentru găsirea căii de transmisie a unor date destinate altor rețele

Rețeaua telefonică este un exemplu de rețea în care se folosește o schemă de adresare ierarhică. Sistemul telefonic folosește un cod format din câteva numere pentru o anumită arie geografică (acesta este primul hop). Apoi urmează un număr de trei cifre (de obicei) pentru aria locală (acesta este al doilea hop). În sfârșit ultimele cifre vor reprezenta destinația finală, individuală a apelului telefonic (ultimul hop).

Datele transmise de la o rețea LAN către alta, prin una dintre căile disponibile, sunt rutate. Protocoloalele care acceptă comunicații LAN-LAN pe mai multe căi sunt cunoscute sub numele de protocoale rutabile. Deoarece protocoalele rutabile pot fi folosite pentru a conecta mai multe rețele LAN, creând astfel medii WAN, importanța lor a crescut. Exemplu de protocol rutabil: TCP/IP. Exemplu de protocol nerutabil: NetBeui

În ziua de astăzi routerele au devenit backbone-ul Internet-ului putând conecta diferite tehnologii de rețea. Un router poate fi un computer foarte puternic pe care să ruleze un anumit software pentru rutarea pachetelor, dar de cele mai multe ori este un dispozitiv (o casetă) dedicat exclusiv acestei operații de rutare.

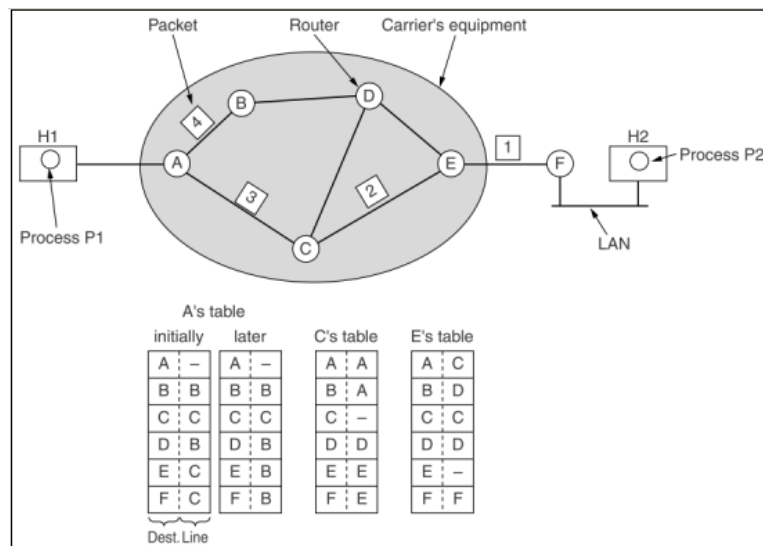


Routerele vor conecta două sau mai multe rețele și vor putea comuta pachete între aceste rețele folosindu-se de adresele logice (IP). Când datele ajung la router de la unul dintre segmente, router-ul va decide conform tabelului său de rutare, către ce segment să le “forward-eze”.

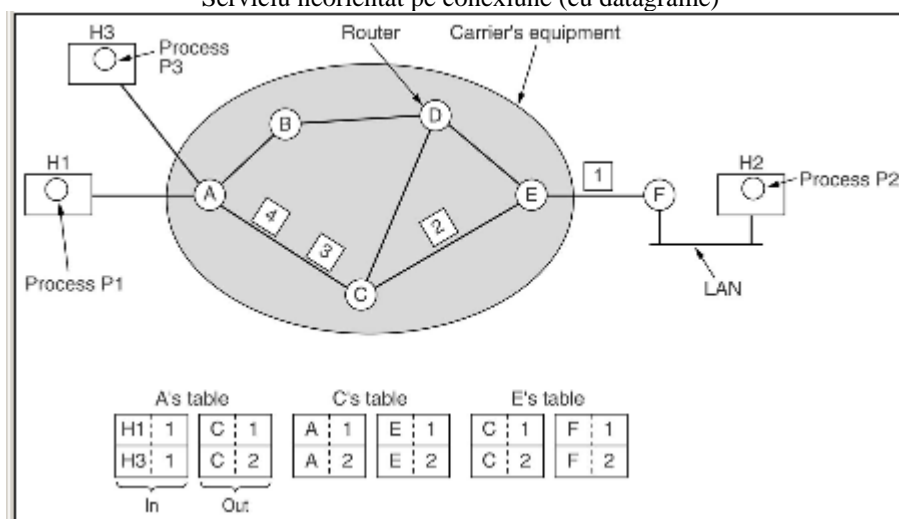
Un router va folosi o tabelă de rutare care este configurată manual sau automat prin schimburi de informații cu alte rutere din rețea.

Ceea ce trebuie precizat este faptul că routerele “cunosc” doar rețele nu și gazde, adică ele vor face comutarea pachetelor pe ideea că o anumită dată trebuie să ajungă la o anumită rețea, dar nu știe la care dintre gazdele acestei rețele va ajunge.

Un router se aseamănă cu un bridge (amândouă au un procesor, memorie RAM și diferite interfețe de I/O, fiecare pentru conectarea unui alt tip de rețea), diferența constă la nivelul OSI la care se lucrează (Bridge = Data Link, Router = Network Link). Această diferență face ca un router să aibă acces la informații pe care un bridge nu le cunoaște.



Serviciu neorientat pe conexiune (cu datagrame)



Serviciu orientat pe conexiune (cu datagrame)

Un ruter poate fi conectat la mai multe rețele prin așazisele interfețe (sau porturi). În rutarea IP, fiecare interfață trebuie să aibă o adresă de rețea (subrețea) diferită, unică.

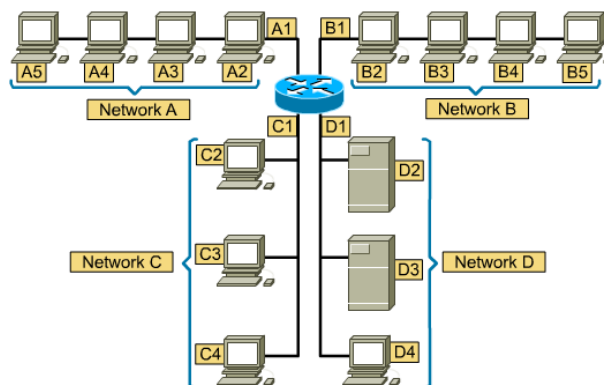
Unitatea de dată folosită în transferul de date este pachetul IP. Un astfel de pachet este format din două componente majore: header-ul (conține adresa sursă și destinație) și partea de date. Alte tipuri de protocoale au formatele lor proprii.

Routerule au două responsabilități majore:

1. Selectarea unei rute optime pentru pachetele ce parcurg rețeaua de la o gazdă la alta. Un router utilizează un algoritm de rutare pentru a determina calea optimă spre destinație. Ex de algoritmi: The Optimality Principle, Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Flooding, Routing for Mobile Hosts, Routing in Ad Hoc Networks. Acești algoritmi întrețin tabele de rutare care conțin informații despre rute, cum ar fi destinația sau următorul hop. Un algoritm de rutare trebuie să satisfacă condițiile de:
  - Optimalitate și simplitate – implică găsirea celei mai bune rute în funcție de anumite metrice specifice fiecărui ruter în parte (număr de hopuri, întârzieri etc.) fără însă să ceară resurse prea mari
  - Robustețe și simplitate – routerule sunt localizate în punctele de joncțiune ale rețelelor (=> de ele depind comunicarea între acele rețele), de aceea trebuie să știe ce măsuri să ia în cazul unei comportări neobișnuite ale anumitor componente hardware
  - Convergență rapidă – routerule trebuie să se pună de acord cu privire la rute optime de-a lungul rețelei pentru a se evita cazul intrării într-o buclă. Când are loc un eveniment în rețea (eroare la un computer sau o rețea, căderea unui anumit segment), automat se vor face schimbări în tabela de rutare, fiecare router trebuind să refacă anumite calcule, bazate pe mesajele primite de la alte rutere din rețea.
  - Flexibilitate – este abilitatea de a se adapta rapid la schimbările sau evenimentele ce au loc în rețea
2. Comutarea (transportarea) pachetelor prin rețele

Când un computer vrea să transmită un pachet într-o altă rețea aflată după un router, va reformata pachetul primit cu adresa fizică a routerului și cu adresa logică (de rețea) al adevăratului destinatar. Când acel cadru ajunge la router acesta îl va despacheta (va da jos încapsularea specifică nivelului Legătură de Date) pentru a avea acces la adresele de rețea (cele specifice nivelului Rețea). Router-ul va căuta în tabela sa de rutare și va lua o decizie în funcție de rezultatul căutării:

- Dacă nu este nici o intrare în tabela de rutare pentru respectiva adresă pachetul va fi distrus
- Dacă găsește o intrare în tabela sa de rutare pentru respectiva adresă va reîncapsula pachetul într-un cadru (frame) specific nivelului Legătură de Date unde va pune la adresa destinație adresa MAC (fizică) a următorului hop. Acest hop nu este neapărat host-ul destinație ci poate fi un alt router care va executa aceeași rutină din nou. Un pachet cu date poate trece pe la mai multe rutere în drumul său, de fiecare dată câmpul destinație al adresei fizice schimbându-se.



Pentru ca două dispozitive aflate într-o rețea să poată comunica, dispozitivul care comunică trebuie să aibă atât adresa IP cât și adresa MAC a dispozitivului destinație, altfel respectivul pachet nu va trece niciodată de Layer 3. Adresa IP este cunoscută, fiind provenită de la aplicațiile care folosesc schimbul de informații în rețea. Problema care apare cel mai frecvent este problema aflării adresei MAC care corespunde unei anumite adrese IP, cunoscută. Această problemă, în cazul unei rețele care folosește TCP/IP se rezolvă folosind protocolul ARP - *Address Resolution Protocol* (integrat în TCP/IP).

Ideea este de a se completa cu intrări pereche de tipul adresa IP - adresă MAC niște tabele numite ARP tables aflate în memoria RAM a unui dispozitiv (calculator) și fiind completată și întreținută de acel dispozitiv. Când un dispozitiv vrea să trimită date într-o rețea pentru o anumită adresă IP el va trebui să-și consulte tabela ARP pentru a putea completa câmpul adresă destinație a cadrelor transmise cu adresa MAC corespunzătoare acelei adrese IP. În cazul în care este găsită o intrare de tip adresă IP- adresă MAC la care adresa IP coincide cu cea căutată problema este rezolvată.

Dacă nu este găsită se va folosi protocolul ARP și se va proceda astfel:

- se va iniția procesul ARP request care începe cu crearea unui pachet ARP request care va fi trimis tuturor dispozitivelor din rețea; acest lucru se face folosind adresa de broadcast FF-FF-FF-FF-FF-FF (hexazecimal) pentru câmpul adresă destinație a pachetului transmis

<b>MAC HEADER</b> Destination FF-FF-FF-FF-FF-FF Source 02-60-8C-01-02-03	<b>IP HEADER</b> Destination 197.15.22.126 Source 197.15.22.33	<b>ARP REQUEST MESSAGE</b> What is your MAC address?
--	--	---

Structura pachetului ARP request

- pentru că pachetul ARP request este transmis în mod broadcast, toate dispozitivele din rețeaua locală vor primi pachetul și-l vor trimite spre analiză nivelurilor OSI superioare. Dacă adresa IP a oricăruia dintre dispozitive este aceeași cu cea aflată în câmpul sursă al pachetului primit, atunci acel dispozitiv îi va trimite adresa lui MAC celui care a inițiat ARP request-ul printr-un pachet numit ARP reply
- această adresă MAC va fi folosită pentru transmiterea datelor care se voiau a fi trimise inițial dar și pentru completarea tabelului ARP a celui care a inițiat ARP request-ul trebuie menționat faptul că și dispozitivul destinație va folosi câmpurile din pachetul ARP request pentru a-și modifica la rândul său tabela ARP

RARP (Reverse ARP) este un protocol folosit de obicei de stațiile fără hard-disk, care-și cunosc adresa MAC, dar nu și cea IP. Este nevoie de un RARP server existent în rețea pentru a-i trimite solicitantului un pachet cu IP-ul cerut.

## BAZE DE NUMERAȚIE

Un număr zecimal obișnuit, constă dintr-un șir de cifre zecimale (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Alegerea lui zece ca bază (radix), a fost făcută pentru că în viața de zi cu zi suntem obișnuiți cu numere zecimale. În lumea calculatoarelor sunt mult mai convenabile alte baze cum ar fi 2, 8 și 16. Aceste baze de numerație sunt denumite binară, octală și respectiv hexazecimală.

Baza 2 => cifrele 0, 1

Baza 8 => cifrele 0, 1, 2, 3, 4, 5, 6, 7

Baza 16 => cifrele 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Numărul 7B9 este evident hexazecimal, deoarece simbolul B poate apare doar în această reprezentare. Oricum numărul 111 poate apare în oricare din sistemele de numerație discutate. Pentru eliminarea ambiguității se utilizează ca indice inferior 2, 8, 10 sau 16 pentru a indica baza de numerație, atunci când ea nu poate fi dedusă din context.

Forma generală a unui număr indiferent de baza n care este scris este:

$$d_n \dots d_2 d_1 d_0 . d_{-1} d_{-2} \dots d_{-k}$$

iar numărul se va putea scrie în funcție de baza considerată după formula:

$$Numar = \sum_{i=-k}^n d_i * r^i \text{ unde } r \text{ este baza (radix)}$$

Scrieți cât reprezintă în format zecimal următoarele numere folosind formula de mai sus. (baza de numerație folosită este precizată în dreapta)

- 1 1 1 1 1 0 1 0 1 0 1 (binar)  
 $1*2^{10} + 1*2^9 + 1*2^8 + 1*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 2005$   
1024 512 256 128 64 0 16 0 4 0 1
- 3 7 2 5 (octal)  
 $3*8^3 + 7*8^2 + 2*8^1 + 1*8^0 = 2005$   
1536 448 16 5
- 7 D 5 (hexazecimal)  
 $7*16^2 + 13*16^1 + 5*16^0 = 2005$   
1792 208 5
- 2005 (zecimal)  
 $2*10^3 + 0*10^2 + 0*10^1 + 5*10^0$

Conversia numerelor zecimale în numere binare se poate face în două moduri diferite.

**Metoda I** metodă derivă direct din definiția numerelor binare. Cel mai mare număr obținut prin ridicarea lui 2 la putere, mai mic decât numărul dat, este scăzut din numărul dat. Procesul se repetă apoi asupra diferenței obținute. Odată obținută decompoziția numărului în puteri ale lui 2, scrierea în baza 2 a numărului poate fi obținută poate fi scrisă prin scrierea lui 1 în pozițiile corespunzătoare puterilor lui 2 și 0 în rest.

Ex:  $2005_{10}$  îl transformăm în binar folosind tabelul de mai jos

$2005 - 1024 = 981 \Rightarrow$  cifra de 1

$981 - 512 = 469 \Rightarrow$  cifra de 1

$469 - 256 = 213 \Rightarrow$  cifra de 1

$213 - 128 = 85 \Rightarrow$  cifra de 1

$85 - 64 = 21 \Rightarrow$  cifra de 1

$21 - 16 = 5 \Rightarrow$  cifra de 0 și 1 (s-a sărit peste 32)

$5 - 4 = 1 \Rightarrow$  cifra de 0, și 1 (s-a sărit peste 8)

$1 - 1 = 0 \Rightarrow$  cifra de 0, și 1 (s-a sărit peste 2)

$$\Rightarrow 1*2^{10} + 1*2^9 + 1*2^8 + 1*2^7 + 1*2^6 + 0*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$$

Deci  $2005_{10} = 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 2$

2 puterea este	0	1	2	3	4	5	6	7	8	9	10	11
	1	2	4	8	16	32	64	128	256	512	1024	2048

**Metoda II** este o care se folosește de împărțirea numărului și caturilor obținute la 2 succesiv și considerarea ultimului cât precum și a resturilor intermediare obținute în sens invers

Ex:  $2005_{10}$  îl transformăm în binar

2005  
1002 rest 1

501	rest	<b>0</b>	
250	rest	<b>1</b>	
125	rest	<b>0</b>	
62	rest	<b>1</b>	
31	rest	<b>0</b>	$\Rightarrow 2005_{10} = 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 2$
15	rest	<b>1</b>	
7	rest	<b>1</b>	
3	rest	<b>1</b>	
<b>1</b>	rest	<b>1</b>	

Dacă este obținută reprezentarea binară a unui număr zecimal atunci reprezentarea octală și hexazecimală a numărului este foarte ușor de obținut. Aceasta se face:

- pt repr. octală: se împarte reprezentarea binară în grupe de 3 începând din stânga care, fiecare în parte reprezintă o cifră în octal
- pt repr. hexazecimală: se împarte reprezentarea binară în grupe de 4 începând din stânga care, fiecare în parte reprezintă o cifră în hexazecimal

Ex:    1 1 1 1 1 0 1 0 1 0 1 2  
       011 111 010 101 2 = 3 7 2 5 5  
       0111 1101 0101 2 = 7 D 5 16

<b>AND</b>	0	1
0	0	0
1	0	1

<b>OR</b>	0	1
0	0	1
1	1	1

<b>XOR</b>	0	1
0	0	1
1	1	0

<b>NOT</b>	0	1
	1	0

**Sarcina de realizat:**

Prezentați anul sau de naștere în format:

1. binar
2. octal
3. hexazecimal
4. zecimal