

Rețele de Calculatoare

Introducere în rețele de calculatoare

Sumar al primului Laborator

1

Elemente
organizatorice

Ce și cum facem în cadrul
laboratorului

2

Statistici

Cum arată Internet-ul în
2020 și respectiv în 2021

3

Evoluția Internetului

Elemente de comunicare

4

Subiectele lucrărilor de
laborator

Enumerare a ceea ce vom
studia pe parcurs

5

Instrumente folosite

Ce instrumente vom folosi
pe parcursul acestui
laborator



Elemente Organizatorice

1. Notare:

Rămâne ca partea de Notare să fie discutată și detaliată complet în săptămâna 4.

2. Sesiunile de laborator vor fi înregistrare

1. Înregistrările vor fi puse la dispoziția studentilor prin CV

3. Prezența va fi făcută – lucrările de laborator trebuie efectuate

1. Prezența va fi facută prin intermediul Campusului Virtual

2. În ultima săptămâna deși nu se efectuează o lucrare distinctă de laborator, prezența este obligatorie, pentru a putea discuta laboratorul.

2020 This Is What Happens In An Internet Minute



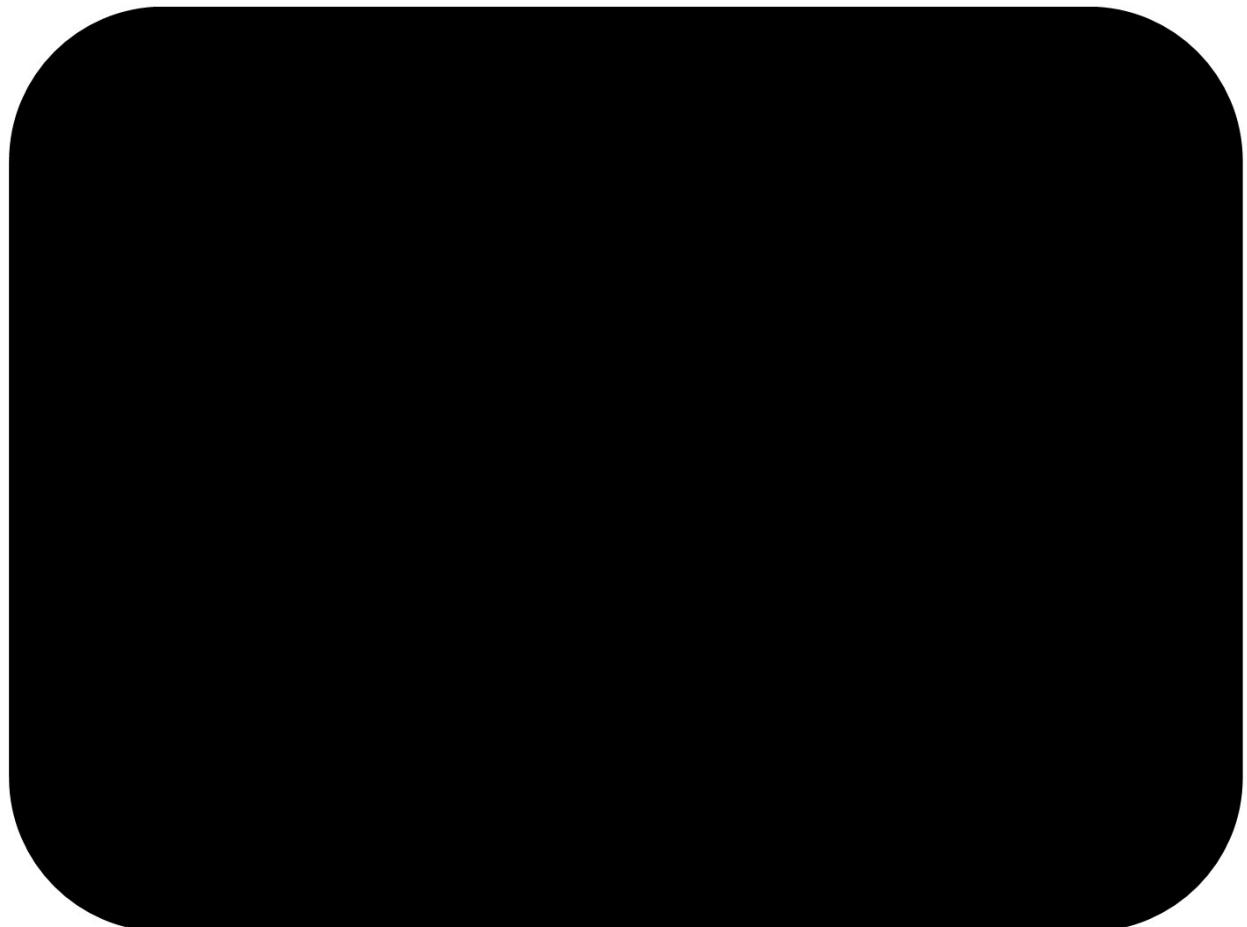
2021 This Is What Happens In An Internet Minute





De ce Rețele de Calculatoare?

Elementele comunicării

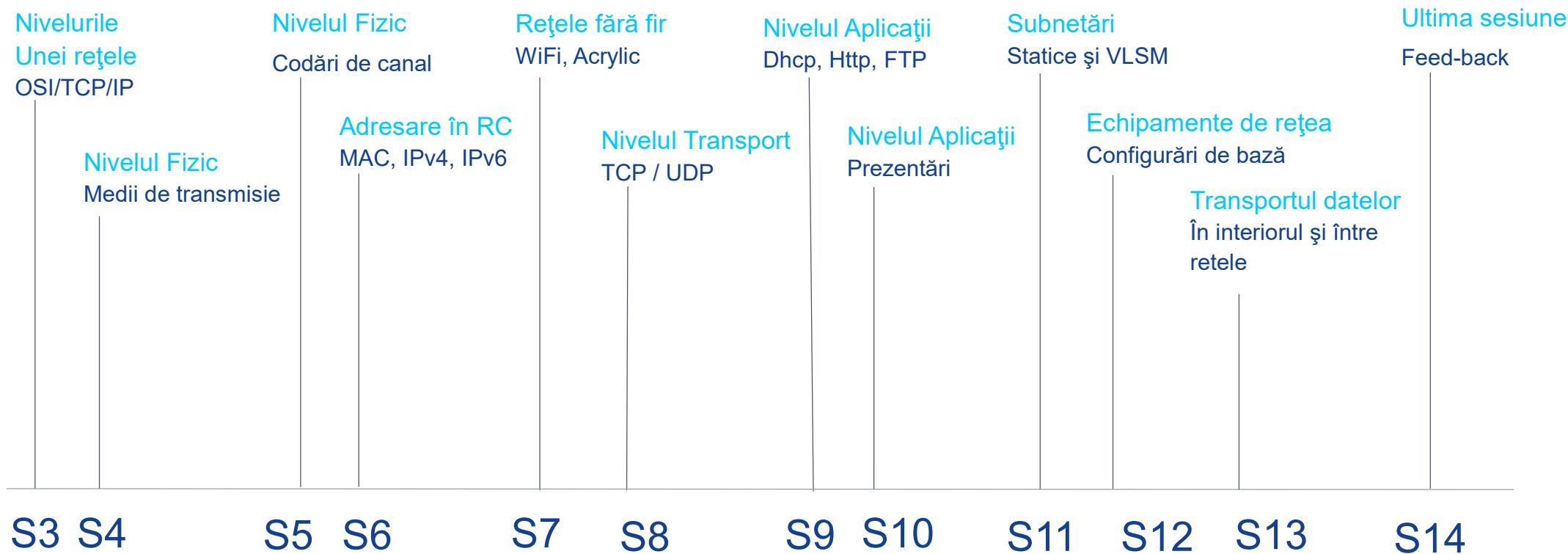


Cum funcționează Internetul



Subiectele discutate în cadrul laboratorului

Timeline



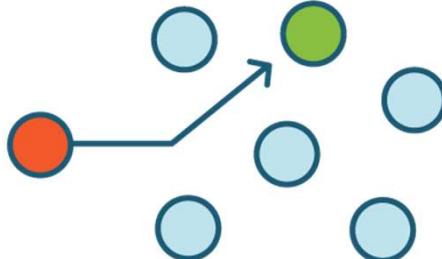
Moduri de Comunicare

Referitor la numărul de receptori

1

Unicast

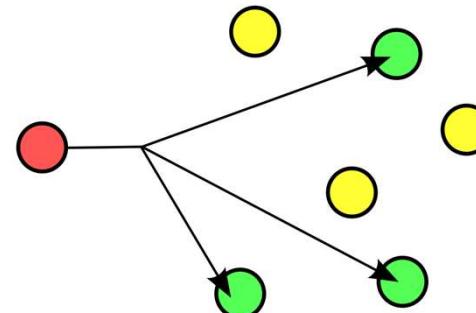
Un emițător la un singur receptor



2

Multicast

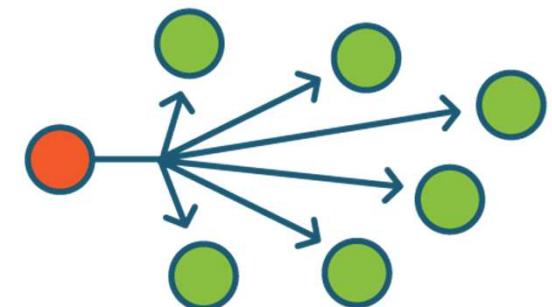
Un emițător la mai mulți, dar nu toți receptorii



3

Broadcast

Un emițător la toți receptorii



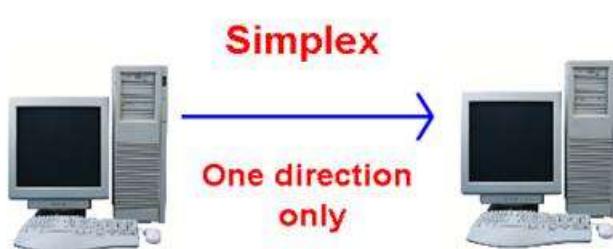
Moduri de Comunicare

Referitor la sensul comunicării

1

Simplex

O entitate emite, cealaltă
recepționează



2

Half-duplex

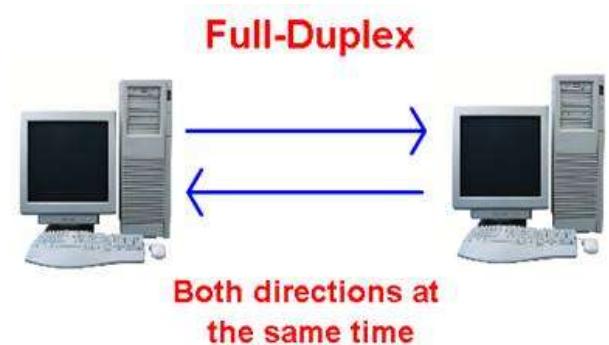
O entitate emite, cealaltă
recepționează, apoi cele 2 își
schimbă rolurile



3

Full-duplex

Ambele entități pot transmite
și recepționa în același timp



Topologii de comunicare

Point to point

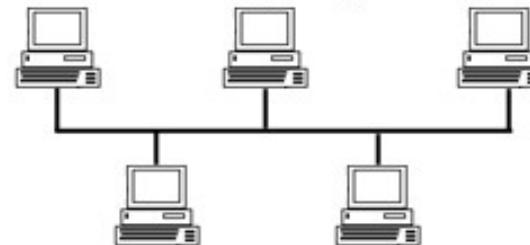


Point to Point



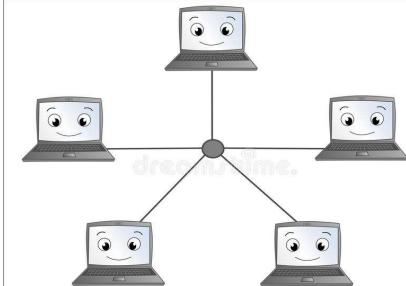
Bus

Bus Topology

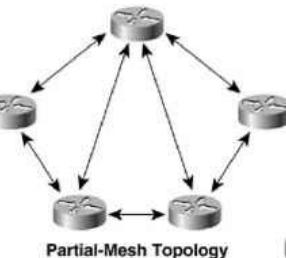


<http://www.computerhope.com>

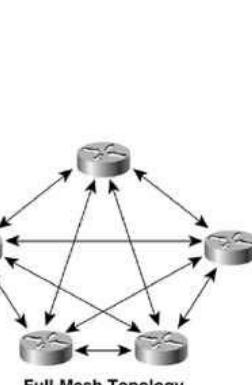
Star



Star



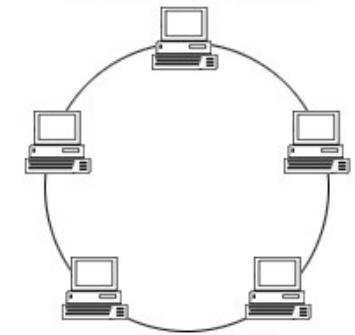
Partial-Mesh Topology



Full-Mesh Topology

Mesh

Ring Topology

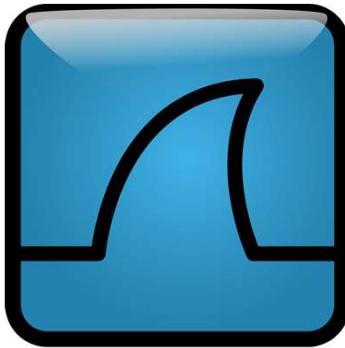


ComputerHope.com

Ring

Tool-uri folosite pe parcursul laboratorului

Wireshark



<https://www.wireshark.org/>

The screenshot shows the Wireshark interface with a list of captured network packets. The columns include No., Time, Deka, Source, Destination, Protocol, and Info. The Info column displays detailed protocol analysis for each packet. Below the list, the "Selected" pane shows the selected packet's raw bytes and ASCII dump. The bottom of the window has a "Filter:" dropdown set to "tcp" and several status indicators.

No.	Time	Deka	Source	Destination	Protocol	Info
13	14.817570	14.817570	192.168.0.10	192.168.0.2	TCP	1242 > 80 [SYN] Seq=1404510823 Ack=0 win=65
14	14.817689	0.000119	192.168.0.2	192.168.0.10	TCP	80 > 1242 [SYN, ACK] Seq=3661615104 Ack=1404
15	14.818178	0.000489	192.168.0.10	192.168.0.2	TCP	1242 > 80 [ACK] Seq=1404510824 Ack=3661615105
16	14.819035	0.000817	192.168.0.10	192.168.0.2	HTTP	GET / HTTP/1.1
17	14.975815	0.156780	192.168.0.2	192.168.0.10	TCP	80 > 1242 [ACK] Seq=3661615105 Ack=1404511233
23	19.382555	4.406740	192.168.0.10	192.168.0.2	TCP	1242 > 80 [FIN, ACK] Seq=1404511234 Ack=3661
24	19.382634	0.000079	192.168.0.2	192.168.0.10	TCP	80 > 1242 [ACK] Seq=3661615105 Ack=1404511233
52	54.234482	34.851848	192.168.0.2	192.168.0.10	HTTP	HTTP/1.1 403 Forbidden (text/html)
53	54.235272	0.000790	192.168.0.10	192.168.0.2	TCP	1242 > 80 [RST] Seq=1404511235 Ack=366044707
54	58.137063	3.901791	192.168.0.10	192.168.0.2	TCP	1244 > 135 [SYN] Seq=1414452237 Ack=0 win=65
55	58.137176	0.000113	192.168.0.2	192.168.0.10	TCP	135 > 1244 [SYN, ACK] Seq=3672465192 Ack=141
56	58.137527	0.000351	192.168.0.10	192.168.0.2	TCP	1244 > 135 [ACK] Seq=1414452238 Ack=36724651
57	58.137992	0.000465	192.168.0.10	192.168.0.2	DCERPC	Bind: call_id: 57 UUID: IOXIDResolver
58	58.188933	0.050941	192.168.0.2	192.168.0.10	DCERPC	Bind_ack: call_id: 57 accept_max_xmit: 5840
59	58.189601	0.000668	192.168.0.10	192.168.0.2	IOXIDR	ComplexPing request AddToSet=0 DelFromSet=1
60	58.202631	0.013030	192.168.0.2	192.168.0.10	IOXIDR	ComplexPing response -> Unknown (0x00000778)
61	58.203457	0.000826	192.168.0.10	192.168.0.2	IOXIDR	ComplexPing request AddToSet=0 DelFromSet=1

Frame 16 (464 bytes on wire, 464 bytes captured)
 Ethernet II, Src: 00:04:61:4a:1e:95, Dst: 00:0b:5d:20:cd:02
 Internet Protocol, Src Addr: 192.168.0.10 (192.168.0.10), Dst Addr: 192.168.0.2 (192.168.0.2)
 Transmission control Protocol, Src Port: 1242 (1242), Dst Port: 80 (80), seq: 1404510824, Ack: 3661615105, Len: 410
 Hypertext Transfer Protocol
 GET / HTTP/1.1
 Host: 192.168.0.2
 User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.5) Gecko/20031007
 Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,image/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.7,*/*
 Accept-Language: en-us,en;q=0.5
 Accept-Encoding: gzip,deflate
 Accept-Charset: iso-8859-1,utf-8;q=0.7,*;q=0.7
 Keep-Alive: 300
 Connection: keep-alive

0000 00 00 5d 20 cd 02 00 04 61 4a 1e 95 08 00 45 00 .J....a3....E.
 0001 01 c2 d1 6d 40 00 80 06 a6 6b c0 a8 00 0a c0 a8 ...m@... ,k?....
 0020 00 02 04 da 00 50 53 b7 22 68 da 3f d0 01 50 18PS. "h?..P.
 0030 ff ff 46 26 00 04 47 45 54 20 2f 20 48 54 54 50 ..F&..GE T / HTTP
 0040 2f 31 28 31 0d 0a 48 6f 73 74 3a 20 31 39 32 26 /1.1.HD ST: 192.
 hexdump 71 74 78 76 70 73 77 0d 0a 7e 77 2c 77 72 41 47 169.0.7 user: 40

Tutoriale:

<https://www.youtube.com/watch?v=TkCSr30UojM>

<https://www.concise-courses.com/security/wireshark-basics/>

Tool-uri folosite pe parcursul laboratorului

Cisco Packet Tracer



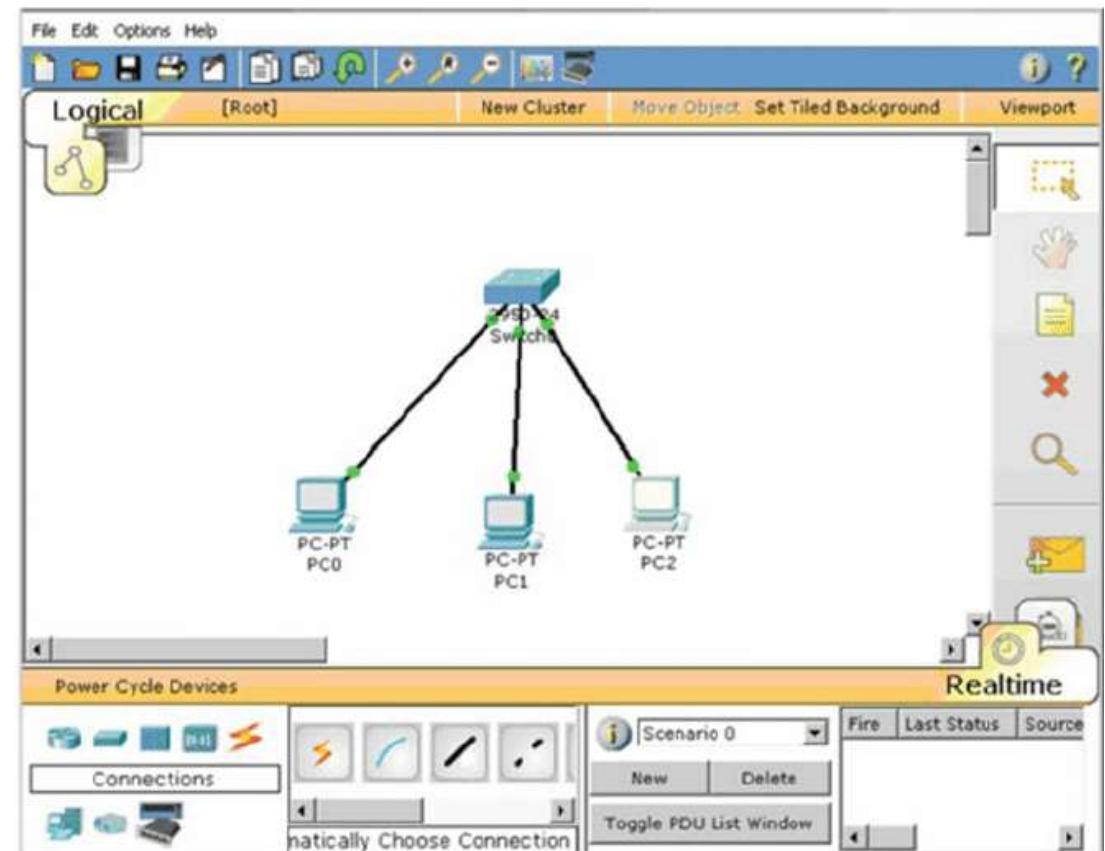
<https://learningnetwork.cisco.com/docs/DOC-29644>

<https://www.itechtics.com/download-cisco-packet-tracer-7-1-free-direct-download-links/>

Tutoriale:

<https://www.youtube.com/watch?v= grVvYk-NG4>

<https://www.youtube.com/watch?v=JtVD0ZXiuhE>



Tool-uri folosite pe parcursul laboratorului

Acrylic WiFi Analyzer

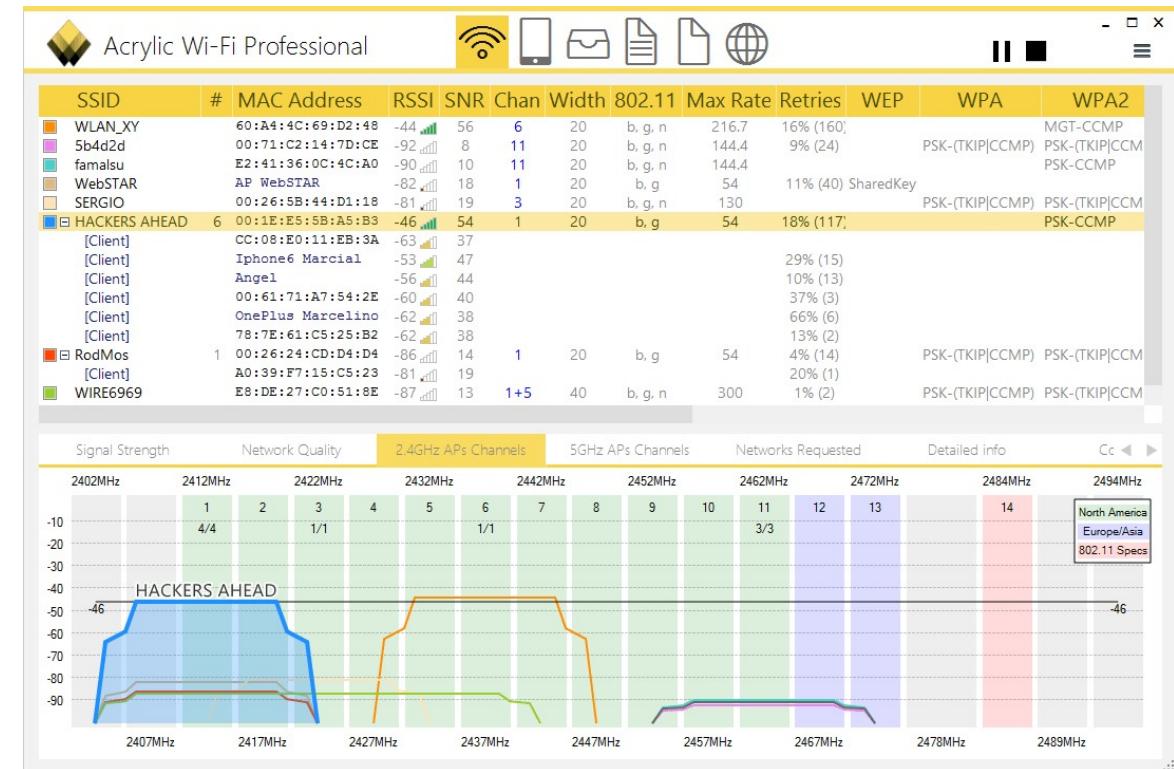


<https://www.acrylicwifi.com/en/>

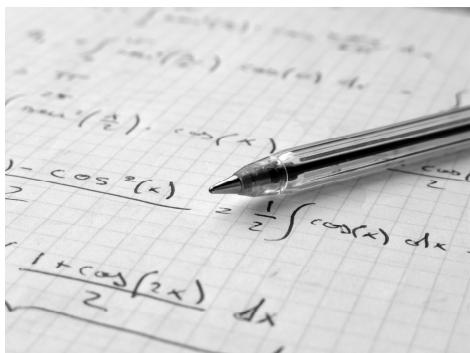
Tutoriale:

<https://www.acrylicwifi.com/en/support-webinars-wifi-wireless-network-software-tools/video-tutorials/>

<https://www.acrylicwifi.com/en/wlan-wifi-wireless-network-software-tools/wifi-analyzer-acrylic-professional/documentation-and-tutorials/>



Tool-uri folosite pe parcursul laboratorului Hârtie, creion și minte



Good old fashion: Pen and paper

Pe parcursul laboratoarelor vom mai utiliza câteva tool-uri și simulatoare, fiecare având link-uri la topic-ul aferent în CV, unde găsiți și o pagină dedicată tool-urilor.

IPv4 SUBNETTING

Subnets				Decimal to Binary							
CIDR	Subnet Mask	Addresses	Wildcard	Subnet Mask	Wildcard						
/32	255.255.255.255	1	0.0.0.0	255 1111 1111	0 0000 0000						
/31	255.255.255.254	2	0.0.0.1	254 1111 1110	1 0000 0001						
/30	255.255.255.252	4	0.0.0.3	252 1111 1100	3 0000 0011						
/29	255.255.255.248	8	0.0.0.7	248 1111 1000	7 0000 0111						
/28	255.255.255.240	16	0.0.0.15	240 1111 0000	15 0000 1111						
/27	255.255.255.224	32	0.0.0.31	224 1110 0000	31 0001 1111						
/26	255.255.255.192	64	0.0.0.63	192 1100 0000	63 0011 1111						
/25	255.255.255.128	128	0.0.0.127	128 1000 0000	127 0111 1111						
/24	255.255.255.0	256	0.0.0.255	0 0000 0000	255 1111 1111						
/23	255.255.254.0	512	0.0.1.255	Subnet Proportion							
/22	255.255.252.0	1,024	0.0.3.255								
/21	255.255.248.0	2,048	0.0.7.255	<table><tr><td>/26</td><td>/27</td><td>/28</td><td>/29</td><td>/30</td><td>/31</td></tr></table>		/26	/27	/28	/29	/30	/31
/26	/27	/28	/29	/30	/31						
/20	255.255.240.0	4,096	0.0.15.255								
/19	255.255.224.0	8,192	0.0.31.255								
/18	255.255.192.0	16,384	0.0.63.255								
/17	255.255.128.0	32,768	0.0.127.255								
/16	255.255.0.0	65,536	0.0.255.255								
/15	255.254.0.0	131,072	0.1.255.255								
/14	255.252.0.0	262,144	0.3.255.255								
/13	255.248.0.0	524,288	0.7.255.255								
/12	255.240.0.0	1,048,576	0.15.255.255								
/11	255.224.0.0	2,097,152	0.31.255.255								
/10	255.192.0.0	4,194,304	0.63.255.255	Classful Ranges							
/9	255.128.0.0	8,388,608	0.127.255.255	A	0.0.0.0 - 127.255.255.255						
/8	255.0.0.0	16,777,216	0.255.255.255	B	128.0.0.0 - 191.255.255.255						
/7	254.0.0.0	33,554,432	1.255.255.255	C	192.0.0.0 - 223.255.255.255						
/6	252.0.0.0	67,108,864	3.255.255.255	D	224.0.0.0 - 239.255.255.255						
/5	248.0.0.0	134,217,728	7.255.255.255	E	240.0.0.0 - 255.255.255.255						
/4	240.0.0.0	268,435,456	15.255.255.255	Reserved Ranges							
/3	224.0.0.0	536,870,912	31.255.255.255	RFC 1918	10.0.0.0 - 10.255.255.255						
/2	192.0.0.0	1,073,741,824	63.255.255.255	localhost	127.0.0.0 - 127.255.255.255						
/1	128.0.0.0	2,147,483,648	127.255.255.255	RFC 1918	172.16.0.0 - 172.31.255.255						
/0	0.0.0.0	4,294,967,296	255.255.255.255	RFC 1918	192.168.0.0 - 192.168.255.255						

Terminology

CIDR

Classless interdomain routing was developed to provide more granularity than legacy classful addressing; CIDR notation is expressed as /XX

VLSM

Variable-length subnet masks are an arbitrary length between 0 and 32 bits; CIDR relies on VLSMs to define routes

by Jeremy Stretch

v2.0



That's all for today, see you next time!

Rețele de Calculatoare

Nivelurile unei rețele

Sumar al laboratorului

1

Tipuri de rețele

- În funcție de locația datelor
- În funcție de raza de acoperire

2

Modelul OSI

Structura modelului

3

Modelul TCP/IP

Structura modelului

4

Maparea modelelor

Maparea celor 2 modele,
introducere în PDU

5

Wireshark

Cum identificăm straturile
cu ajutorul wireshark



Tipuri de rețele

În funcție de locația de acces a datelor

Studiu de caz

Emag.ro

Intranet

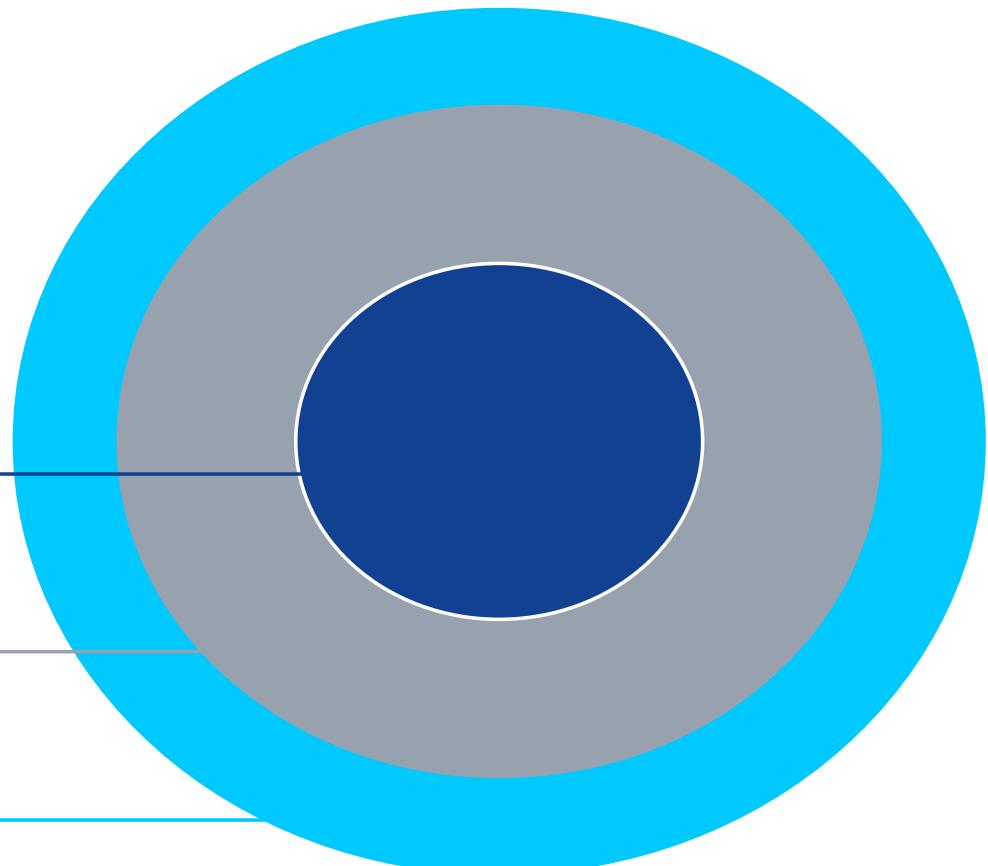
Destinat utilizării strict interne: servere mail, depozitare date, etc

Extranet

Destinat utilizării din exterior pentru resursele interne

Internet

Destinat utilizării de către utilizatorii externi, fără acces la structurile interne



Tipuri de rețele

În funcție de aria de acoperire

WLAN

Wireless Local Area Network

WLAN

LAN

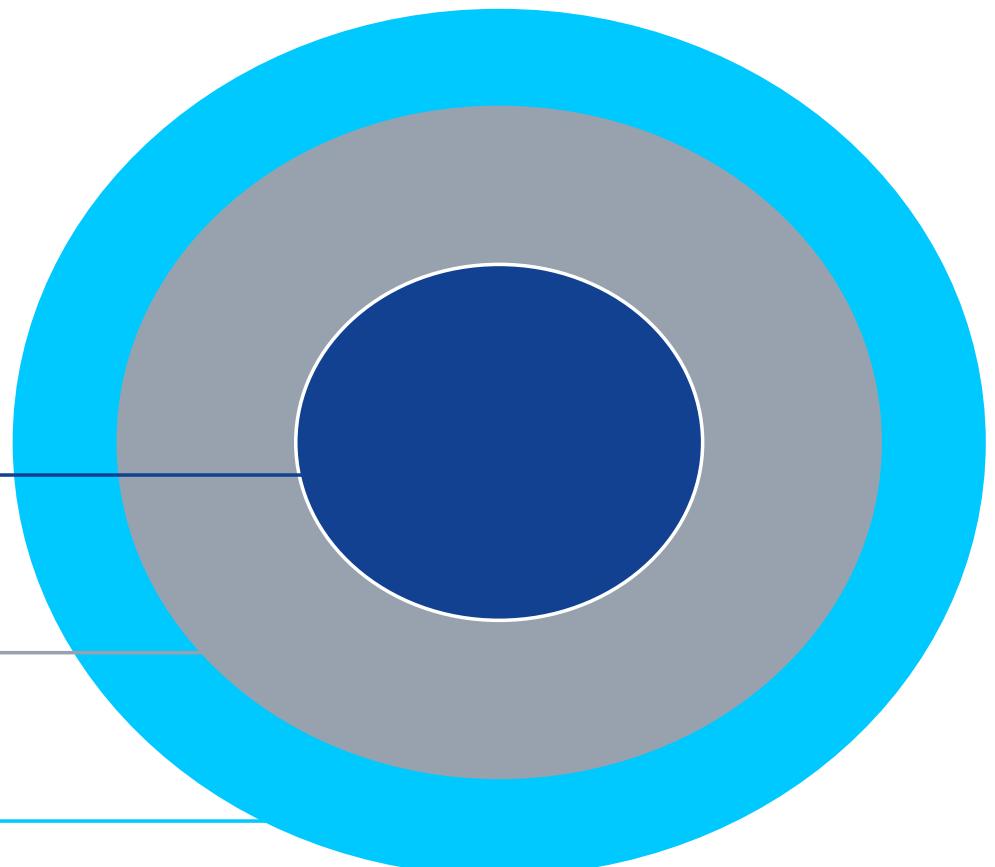
Local Area Network

MAN

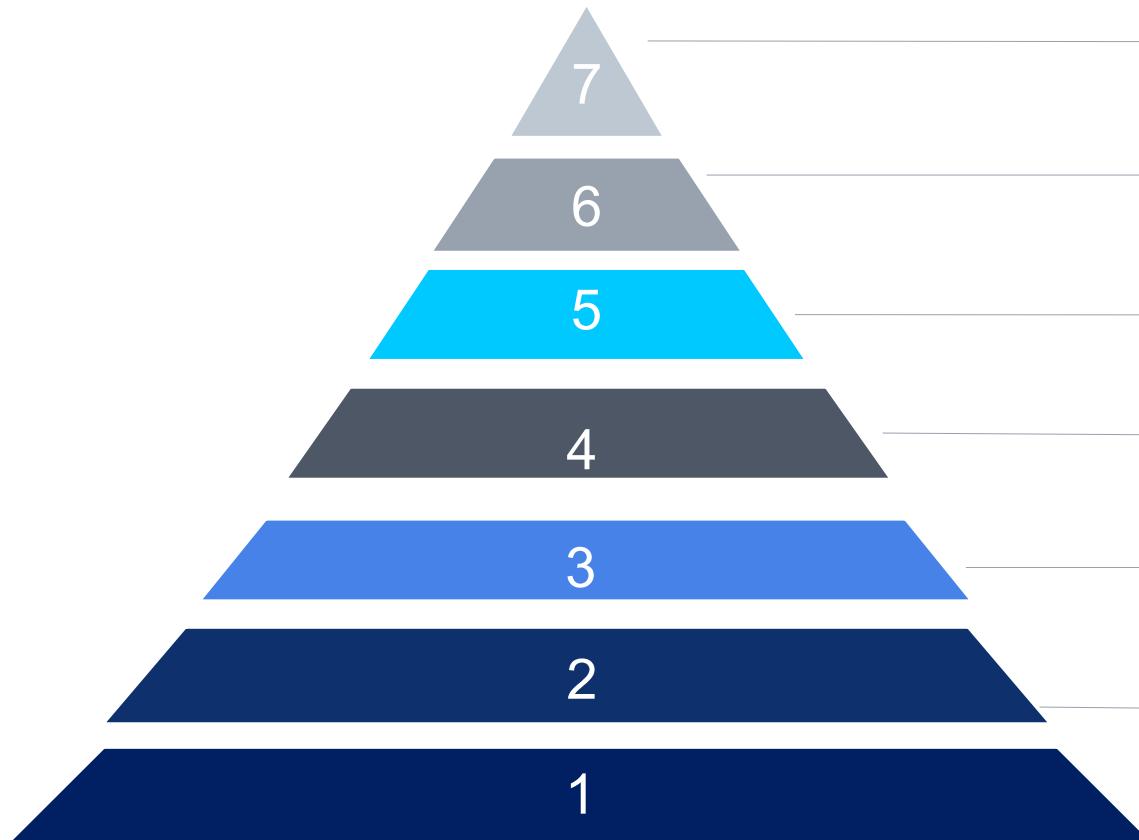
Metropolitan Area Network

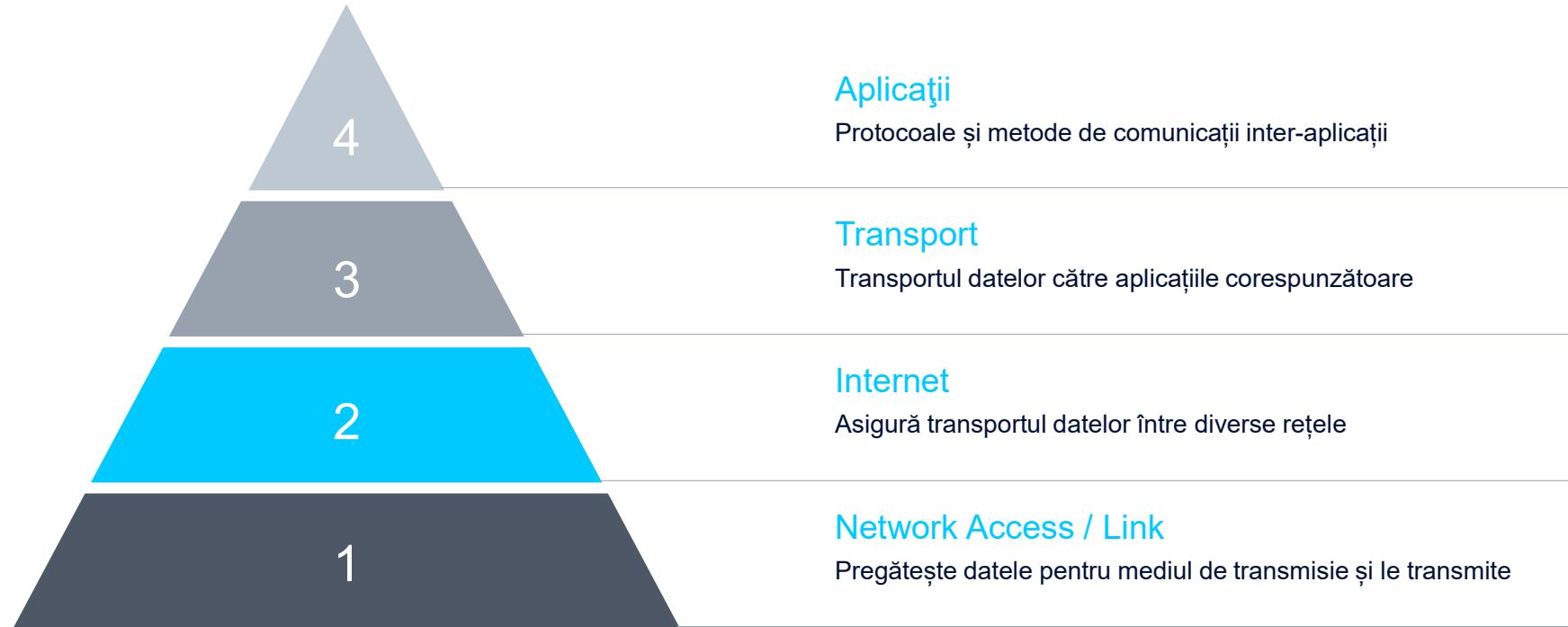
WAN

Wide Area Network



Studiu de caz:
Netcity București
<https://net-city.ro/en/>





Maparea între cele 2 modele

ATENȚIE

- La mapările dintre cele 2 modele
- Vor apărea întrebări din ele

Nr nivel	OSI Model	TCP/IP Model	Nr nivel
7	Application	Application	4
6	Presentation		
5	Session		
4	Transport (First to offer end-to-end connection)	Transport	3
3	Network (Ip and path determination)	Internet	2
2	Data Link (Physical Adressing)	Network Access / Link	1
1	Physical (communication media)		

Internet

Aplicații

Pachete

Date

Transport

Network Access

Segmente

Cadru/frame

PDU

- Protocol Data Unit.
- Unitatea de măsură a datelor aferentă fiecarui strat.

Decapsularea Datelor

Pornim de la nivelul inferior

1

Network
Access



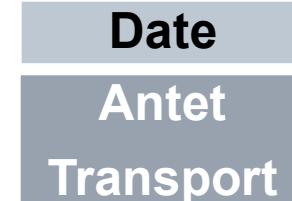
2

Internet



3

Transport



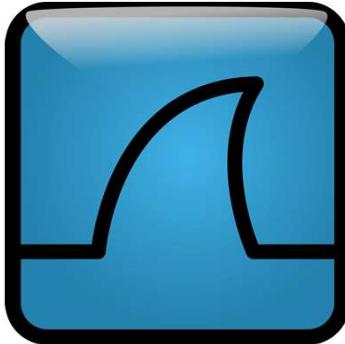
4

Aplicații

Date

Antet
Network Access

Tool folosit pe parcursul laboratorului Wireshark



<https://www.wireshark.org/>

Tutoriale:

<https://www.youtube.com/watch?v=TkCSr30UojM>

<https://www.concise-courses.com/security/wireshark-basics/>

The screenshot shows the Wireshark interface with a list of captured network packets. The columns include No., Time, Deka, Source, Destination, Protocol, and Info. The Info column displays detailed protocol analysis for each packet. Below the list, the "Selected" pane shows the raw bytes of selected packet 16, which is an HTTP GET request for "/". The "Details" pane shows the breakdown of the selected packet, and the "Hex" pane shows the raw hex dump of the bytes.

No.	Time	Deka	Source	Destination	Protocol	Info
13	14.817570	14.817570	192.168.0.10	192.168.0.2	TCP	1242 > 80 [SYN] Seq=1404510823 Ack=0 win=65
14	14.817689	0.000119	192.168.0.2	192.168.0.10	TCP	80 > 1242 [SYN, ACK] Seq=3661615104 Ack=1404510824
15	14.818178	0.000489	192.168.0.10	192.168.0.2	TCP	1242 > 80 [ACK] Seq=1404510824 Ack=3661615105
16	14.819035	0.000817	192.168.0.10	192.168.0.2	HTTP	GET / HTTP/1.1
17	14.975815	0.156780	192.168.0.2	192.168.0.10	TCP	80 > 1242 [ACK] Seq=3661615105 Ack=1404511235
23	19.382555	4.406740	192.168.0.10	192.168.0.2	TCP	1242 > 80 [FIN, ACK] Seq=1404511234 Ack=3661615105
24	19.382634	0.000079	192.168.0.2	192.168.0.10	TCP	80 > 1242 [ACK] Seq=3661615105 Ack=1404511235
52	54.234482	34.851848	192.168.0.2	192.168.0.10	HTTP	HTTP/1.1 403 Forbidden (text/html)
53	54.235272	0.000790	192.168.0.10	192.168.0.2	TCP	1242 > 80 [RST] Seq=1404511235 Ack=366044707
54	58.137063	3.901791	192.168.0.10	192.168.0.2	TCP	1244 > 135 [SYN] Seq=1414452237 Ack=0 win=65
55	58.137176	0.000113	192.168.0.2	192.168.0.10	TCP	135 > 1244 [SYN, ACK] Seq=3672465192 Ack=141
56	58.137527	0.000351	192.168.0.10	192.168.0.2	TCP	1244 > 135 [ACK] Seq=1414452238 Ack=36724651
57	58.137992	0.000465	192.168.0.10	192.168.0.2	DCERPC	Bind: call_id: 57 UUID: IOXIDResolver
58	58.188933	0.050941	192.168.0.2	192.168.0.10	DCERPC	Bind_ack: call_id: 57 accept_max_xmit: 5840
59	58.189601	0.000668	192.168.0.10	192.168.0.2	IOXIDR	ComplexPing request AddToSet=0 DelFromSet=1
60	58.202631	0.013030	192.168.0.2	192.168.0.10	IOXIDR	ComplexPing response -> Unknown (0x00000778)
61	58.203457	0.000826	192.168.0.10	192.168.0.2	IOXIDR	ComplexPing request AddToSet=0 DelFromSet=1

Frame 16 (464 bytes on wire, 464 bytes captured)
 Ethernet II, Src: 00:04:61:4a:1e:95, Dst: 00:0b:5d:20:cd:02
 Internet Protocol, Src Addr: 192.168.0.10 (192.168.0.10), Dst Addr: 192.168.0.2 (192.168.0.2)
 Transmission control Protocol, Src Port: 1242 (1242), Dst Port: 80 (80), seq: 1404510824, Ack: 3661615105, Len: 410
 Hypertext Transfer Protocol
 GET / HTTP/1.1
 Host: 192.168.0.2
 User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.5) Gecko/20031007
 Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,image/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.7,*/*
 Accept-Language: en-us,en;q=0.5
 Accept-Encoding: gzip,deflate
 Accept-Charset: iso-8859-1,utf-8;q=0.7,*;q=0.7
 Keep-Alive: 300
 Connection: keep-alive

0000 00 00 5d 20 cd 02 00 04 61 4a 1e 95 08 00 45 00 .J....a3....E.
 0001 01 c2 d1 6d 40 00 80 06 a6 6b c0 a8 00 0a c0 a8 ...m@... ,k?....
 0002 00 02 04 da 00 50 53 b7 22 68 da 3f d0 01 50 18PS. "h?..P.
 0003 ff ff 46 26 00 00 47 45 54 20 2f 20 48 54 54 50 ..F&..GE T / HTTP
 0004 2f 31 28 31 0d 0a 48 6f 73 74 3a 20 31 39 32 26 /1.1.HD ST: 192.
 0005 71 24 28 26 20 20 22 0d 0e ee 72 2t 72 41 47 169.0.7.....

Partea practică

Wireshark

Welcome to Wireshark

Open

Posibilitate de a deschide trase (fișiere de captură) mai vechi

C:\Users\cmisici\OneDrive - Nokia\Cursuri\Cursuri\UVT\Retele calculatoare_UVT\Trace\trace-dhcp.pcap (2208 Bytes)
C:\Users\cmisici\OneDrive - Nokia\Cursuri\Cursuri\UVT\Retele calculatoare_UVT\Trace\test2.pcapng (716 KB)
C:\Users\cmisici\Downloads\trace-dhcp (1).pcap (not found)
C:\Users\cmisici\Downloads\trace-dhcp.pcap (not found)
C:\Users\cmisici\OneDrive - Nokia\Cursuri\Cursuri\UVT\Retele calculatoare_UVT\Trace\trace-tcp.pcap (1126 KB)
C:\Users\cmisici\OneDrive - Nokia\Cursuri\Cursuri\UPT\Retele UPT\L2_Introducere in nivelurile unei retele\trace-protocol-layers.pcap (16 KB)

Capture

Adăugarea unui filtru de captură
De exemplu a unui protocol căutat

...using this filter: Enter a capture filter ...

All interfaces shown ▾

Local Area Connection* 2
Bluetooth Network Connection
Npcap Loopback Adapter
Ethernet
Local Area Connection* 7
Local Area Connection* 10

Interfața de rețea pe care dorim să o
“spionăm”
PS: cu cât este mai variată, cu atât avem
mai multe date ce circulă pe acolo



No.	Time	Source	Destination	Protocol	Length	Info
1225	23.111654	172.217.22.206	192.168.0.87	UDP	67	443 → 52678 Len=25
1226	23.114358	192.168.0.87	78.96.7.88	DNS	86	Standard query 0x9959 PTR 17.104.114.52.in-addr.arpa
1227	23.145886	192.168.0.87	95.77.94.88	DNS	86	Standard query 0x9959 PTR 17.104.114.52.in-addr.arpa
1228	23.151699	172.217.22.206	192.168.0.87	UDP	123	443 → 52678 Len=81
1229	23.151793	172.217.22.206	192.168.0.87	UDP	451	443 → 52678 Len=409
1230	23.152077	192.168.0.87	172.217.22.206	UDP	75	52678 → 443 Len=33
1231	23.152780	172.217.22.206	192.168.0.87	UDP	74	443 → 52678 Len=32
1232	23.152780	172.217.22.206	192.168.0.87	UDP	248	443 → 52678 Len=206
1233	23.153488	192.168.0.87	172.217.22.206	UDP	75	52678 → 443 Len=33
1234	23.264726	78.96.7.88	192.168.0.87	DNS	165	Standard query response 0x9959 No such name PTR 17.104.114.52.in-addr.arpa
1235	23.266581	192.168.0.87	52.114.104.17	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00>
1236	23.288255	95.77.94.88	192.168.0.87	DNS	165	Standard query response 0x9959 No such name PTR 17.104.114.52.in-addr.arpa
1237	23.450155	192.168.0.227	255.255.255.255	UDP	217	49154 → 6666 Len=175
1238	23.526796	192.168.0.87	172.217.23.36	UDP	461	60109 → 443 Len=419
1239	23.572436	172.217.23.36	192.168.0.87	UDP	68	443 → 60109 Len=26

Posibilitate de a adăuga un filtru de vizualizare
(display filter) suplimentar celui de captură

Lista pachetelor
inspectate atât inbound
cât și outbound

```
> Frame 1226: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
> Ethernet II, Src: LcfChEfe_4f:54:ae (98:fa:9b:4f:54:ae), Dst: CompalBr_c2:85:7f (ac:22:05:c2:85:7f)
> Internet Protocol Version 4, Src: 192.168.0.87, Dst: 78.96.7.88
> User Datagram Protocol, Src Port: 65421, Dst Port: 53
> Domain Name System (query)
```

0000 ac 22 05 c2 85 7f 98 fa 9b 4f 54 ae 08 00 45 00 OT .. E.
0010 00 48 84 8b 00 00 80 11 00 00 c0 a8 00 57 4e 60	. H WN`
0020 07 58 ff 8d 00 35 00 34 16 fd 99 59 01 00 00 01	. X...5.4 ... Y....
0030 00 00 00 00 00 02 31 37 03 31 30 34 03 31 31 1 7 104 11
0040 34 02 35 32 07 69 6e 2d 61 64 64 72 04 61 72 70	4 52 in- addr.arp
0050 61 00 00 0c 00 01	a.....

Detalierea pachetului
pe straturi TCP/IP

Viziunea binară (cod
hexazecimal) a datelor

Partea practică

Cum mapăm straturile TCP/IP peste wireshark

- > Frame 1: 471 bytes on wire (3768 bits), 471 bytes captured (3768 bits) on interface 0
- > Ethernet II, Src: CompaqBr_c2:85:7f (ac:22:05:c2:85:7f), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
- > Internet Protocol Version 4, Src: 192.168.0.1, Dst: 239.255.255.250
- > User Datagram Protocol, Src Port: 42109, Dst Port: 1900
- > Simple Service Discovery Protocol

Sumar al
cadrului

Network Access

Internet

Transport

Aplicații

Partea practică

Name Resolution: Identificarea numelor

The screenshot shows the Wireshark preferences window. The left sidebar has a tree structure with 'Name Resolution' selected. The main area is titled 'Name Resolution' and contains the following options:

- Resolve MAC addresses
- Resolve transport names
- Resolve network (IP) addresses
- Use captured DNS packet data for address resolution
- Use an external network name resolver

Below these options is a 'Maximum concurrent requests' field set to 500.

A blue callout box points from the 'Resolve MAC addresses' checkbox to the text: "Adresele MAC se găsesc la campul Ethernet II".

At the bottom, there is a list of network frames:

- > Frame 1: 471 bytes on wire (3768 bits), 471 bytes captured (3768 bits) on interface 0
- > Ethernet II, Src: CompalBr_c2:85:7f (ac:22:05:c2:85:7f), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)



That's all for today, see you next time!

Rețele de Calculatoare

Nivelul fizic – Medii de transmisie

Sumar al laboratorului

1

Încapsularea datelor

Reminder de data trecută

2

Medii de transmisie

Ethernet

Fibra optică

Wireless (transmisii fără fir)

3

Ethernet

Mufarea cablurilor

Media Convertoare

4

Fibra Optică

Exemplu de cablu de 32 perechi

Probleme posibile la F.O.



Procesul de încapsulare

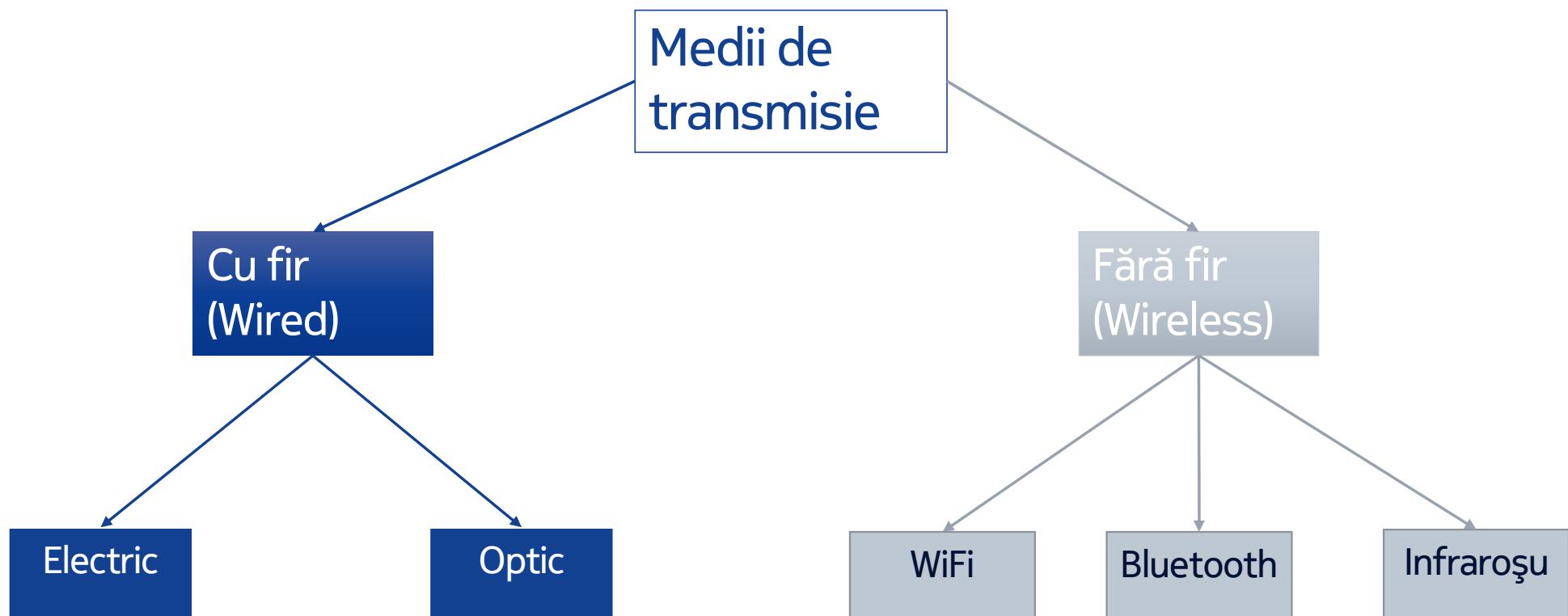
Recapitulare

Aplicații	4
Transport	3
Internet	2
Network Access	1



Medii de transmisie

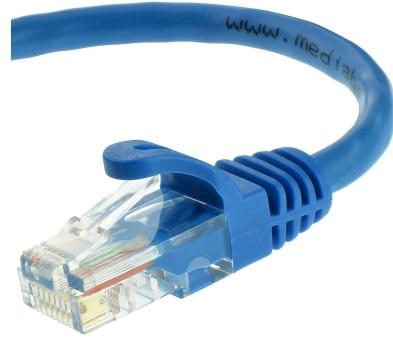
Clasificare



Medii de transmisie

Sumar

Mediul electric –
ex: cablul Ethernet



Cat5e



Cat6



Cat6a



Cat7

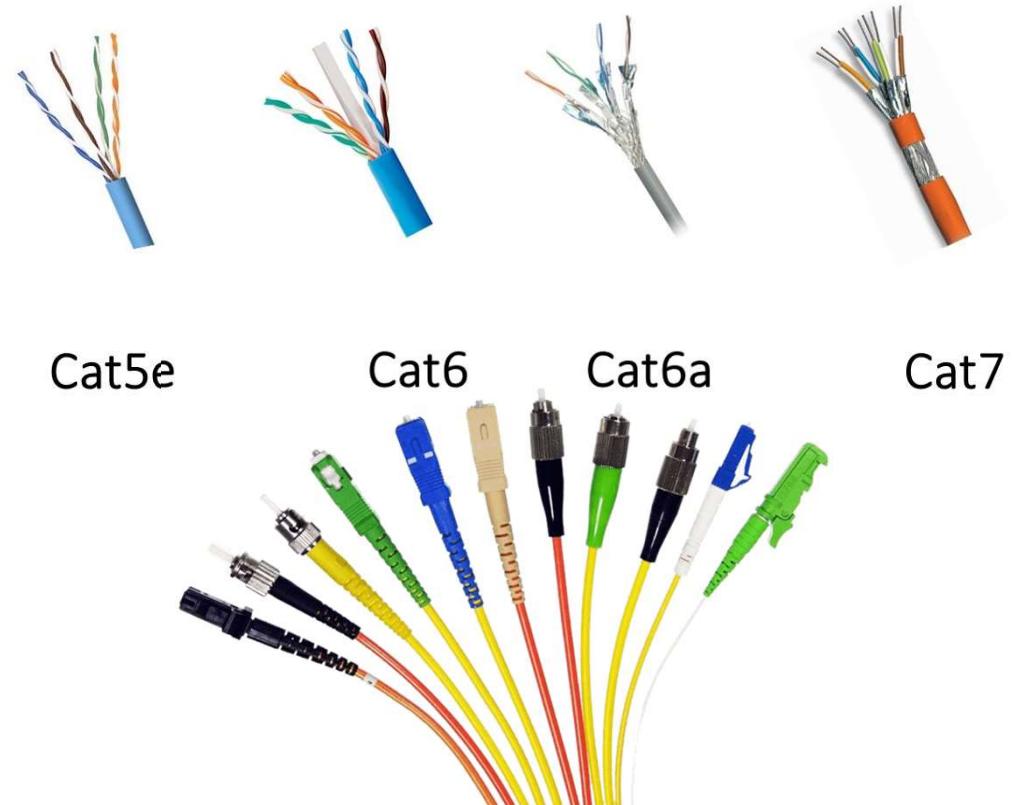
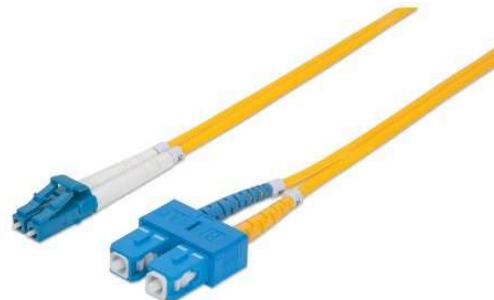
Medii de transmisie

Sumar

Mediu electric –
ex: cablul Ethernet



Mediu optic -
Fibră Optică



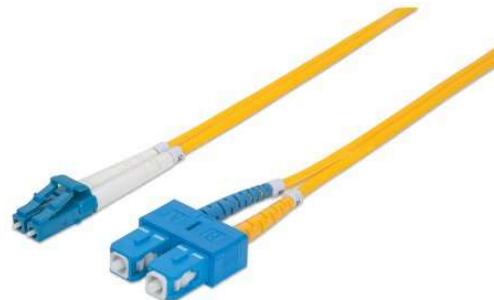
Medii de transmisie

Sumar

Mediu electric –
ex: cablul Ethernet



Mediu optic -
Fibră Optică



Legături fără fir (wireless)
Discutăm într-un lab separat



Medii de transmisie

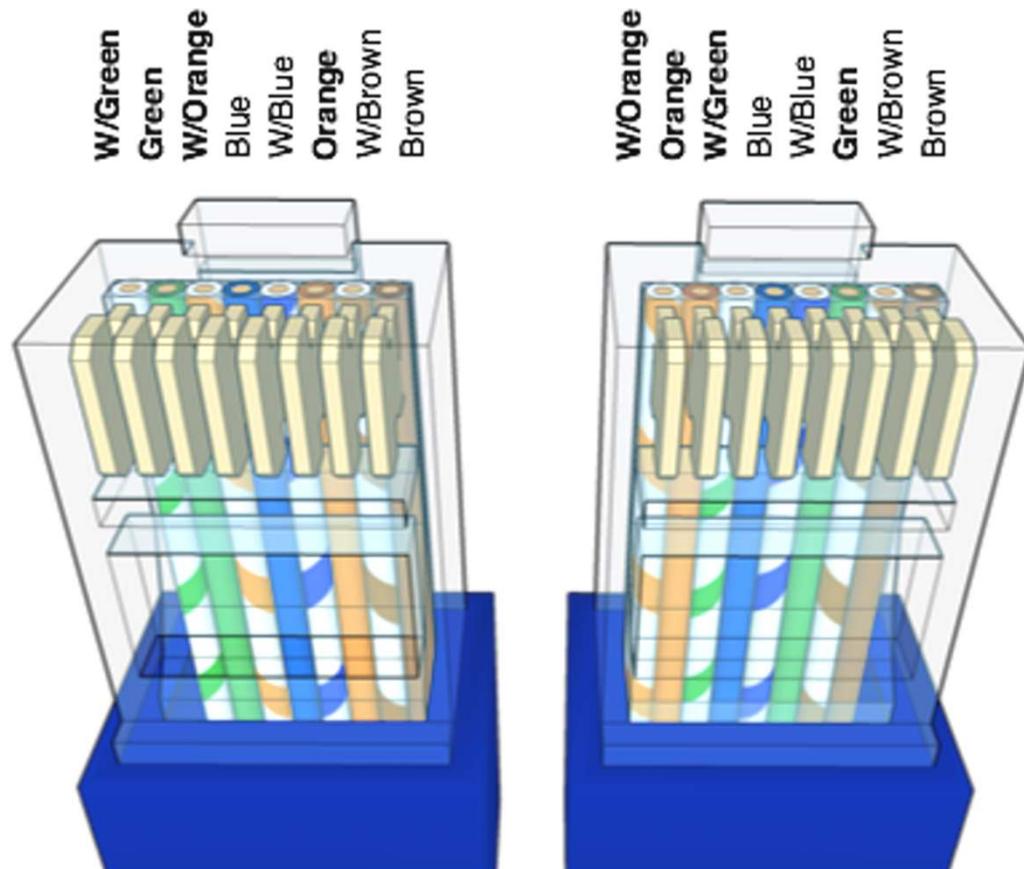
Cabluri Ethernet - 1

Cat.	Viteza de transmisie	Câte DVD-uri descărca într-o oră
Cat 3	10Mbps	1
Cat 5	10/100Mbps	1-10
Cat 5e	1Gbps	100
Cat 6	1Gbps	100
Cat 6a	10Gbps	1000
Cat 7	10 Gbps	1000
Cat 7a	10Gbps	1000
Cat 8	25Gbps	X= calculati

Sursa:

<https://network-telecom.com/ethernet-cable-wiring-ontario/>

8 © 2020 Nokia

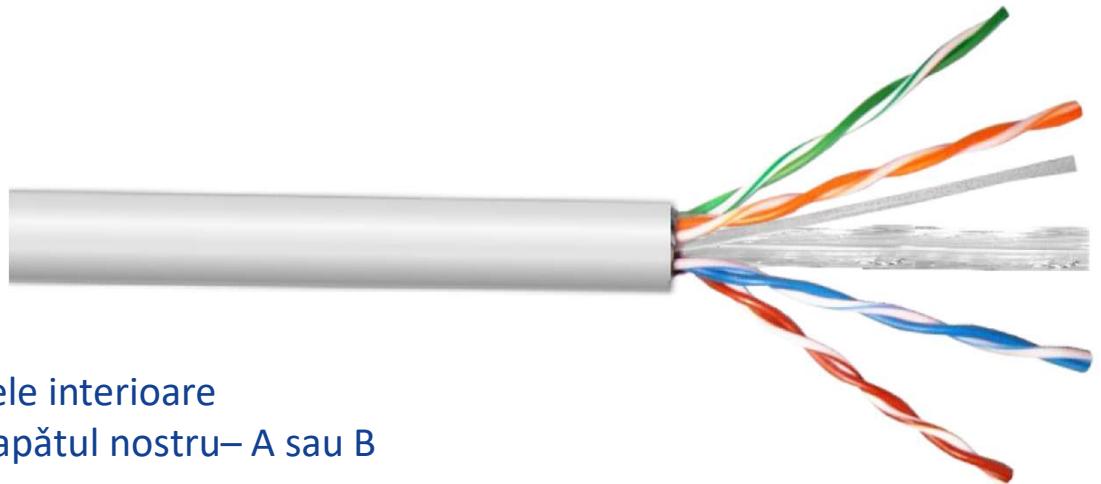


Extra reading: T568-A (Preferred)
<http://www.fiberopticshare.com/guide-choosing-suitable-etherent-cables.html>

NOKIA

Medii de transmisie

Cablul Ethernet - 2



Etapele sertizării corecte a unui cablu Ethernet:

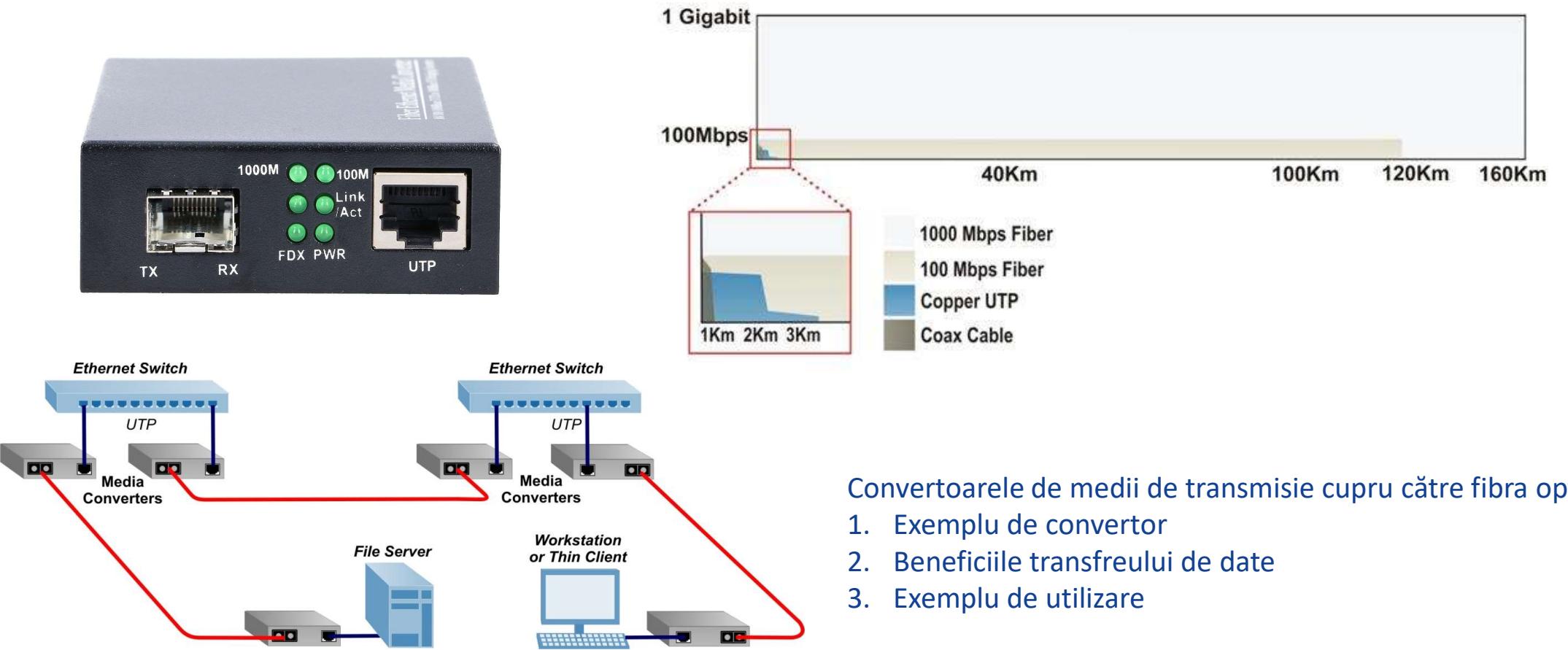
1. Tăierea mufelor defecte - perpendicular pe cablu
2. Desfacerea învelișului de plastic ce delimitază firele interioare
3. Identificarea tipului de cablu ce trebuie făcut pe capătul nostru – A sau B
4. Aranjarea firelor interioare conform modelului necesar
5. Sertizarea mufei RJ-45 pe pozitie
6. Verificarea cablului cu testorul de cablu

Exemplu video:

<https://www.youtube.com/watch?v=WvP0D0jiyLg>

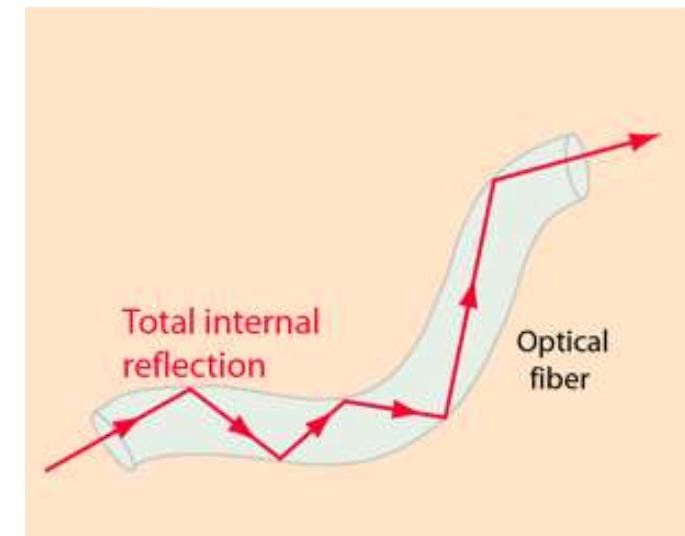
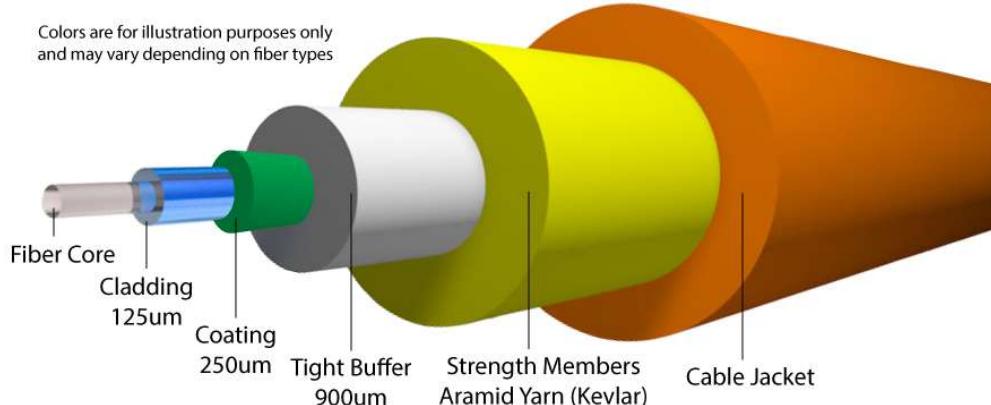
Media Converteare

Ethernet către fibra optică (copper-to-fiber)



Medii de transmisie

Fibra Optică - funcționare



Funcționarea Fibrei Optice se bazează pe legile Reflexiei și Refracției

Pe Campusul Virtual există o pagină dedicată fibrei optice.

Resursă externă:

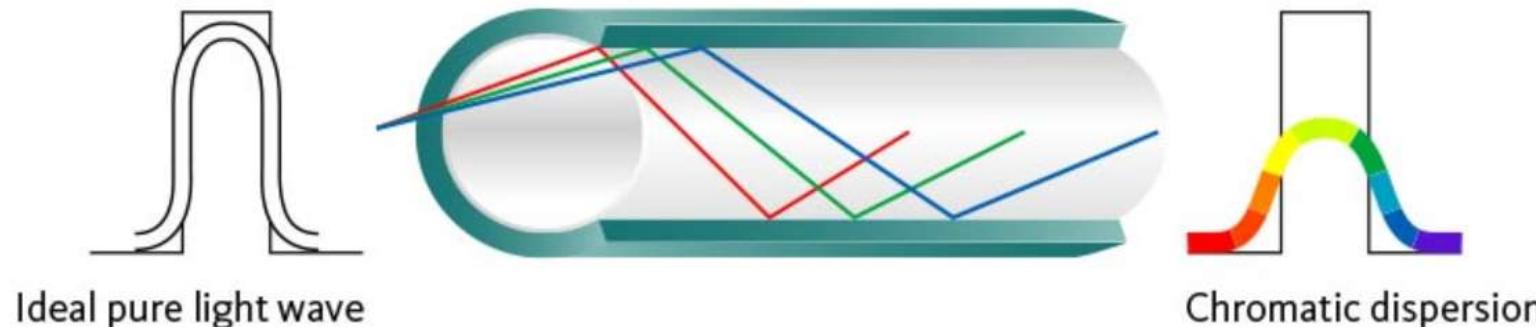
<http://hyperphysics.phy-astr.gsu.edu/hbase/optmod/fibopt.html>

Medii de transmisie

Fibra Optică – probleme apărute

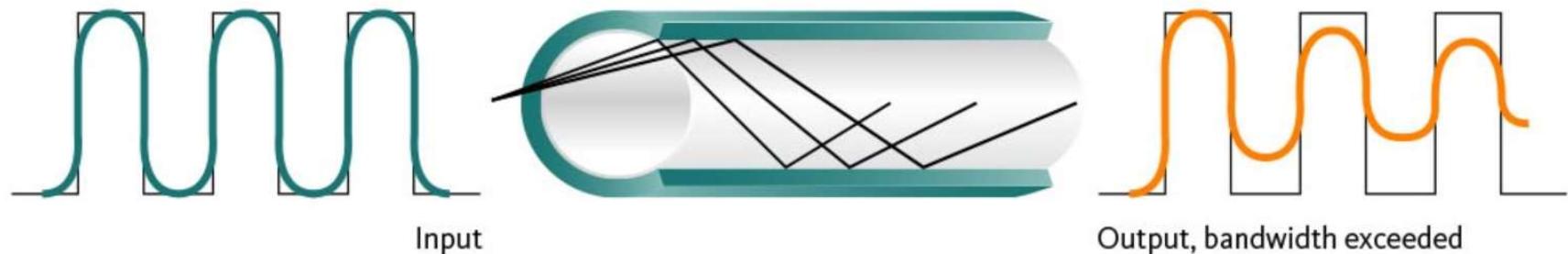
1

Dispersie
Cromatică



2

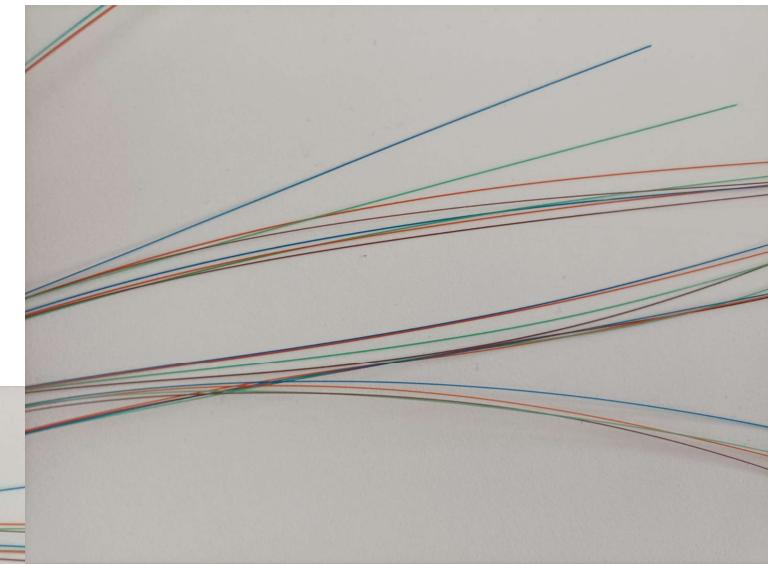
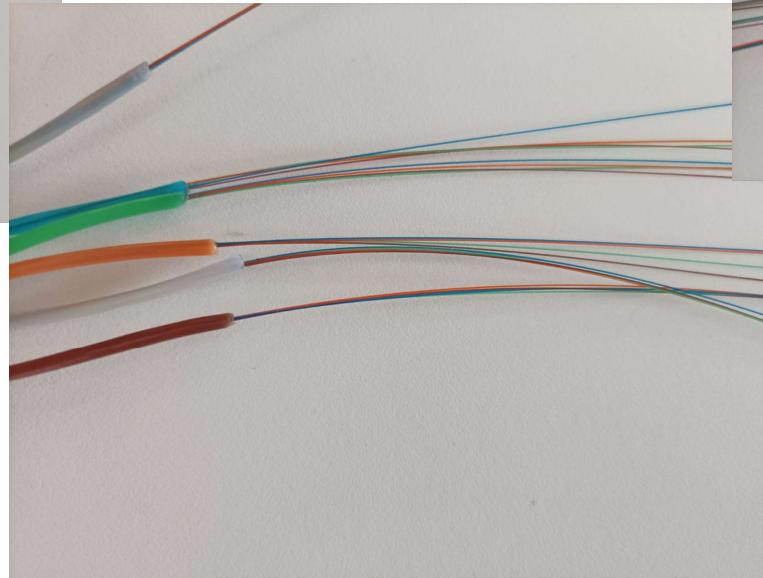
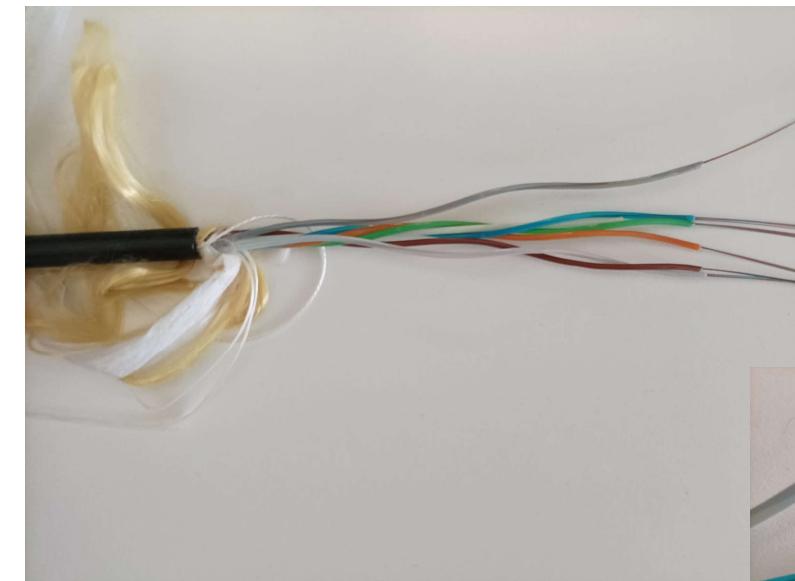
Dispersie
Modală



ATENTIE la Campusul Virtual:
Exemple legate de functionarea si sudura fibrelor optice

Medii de transmisie

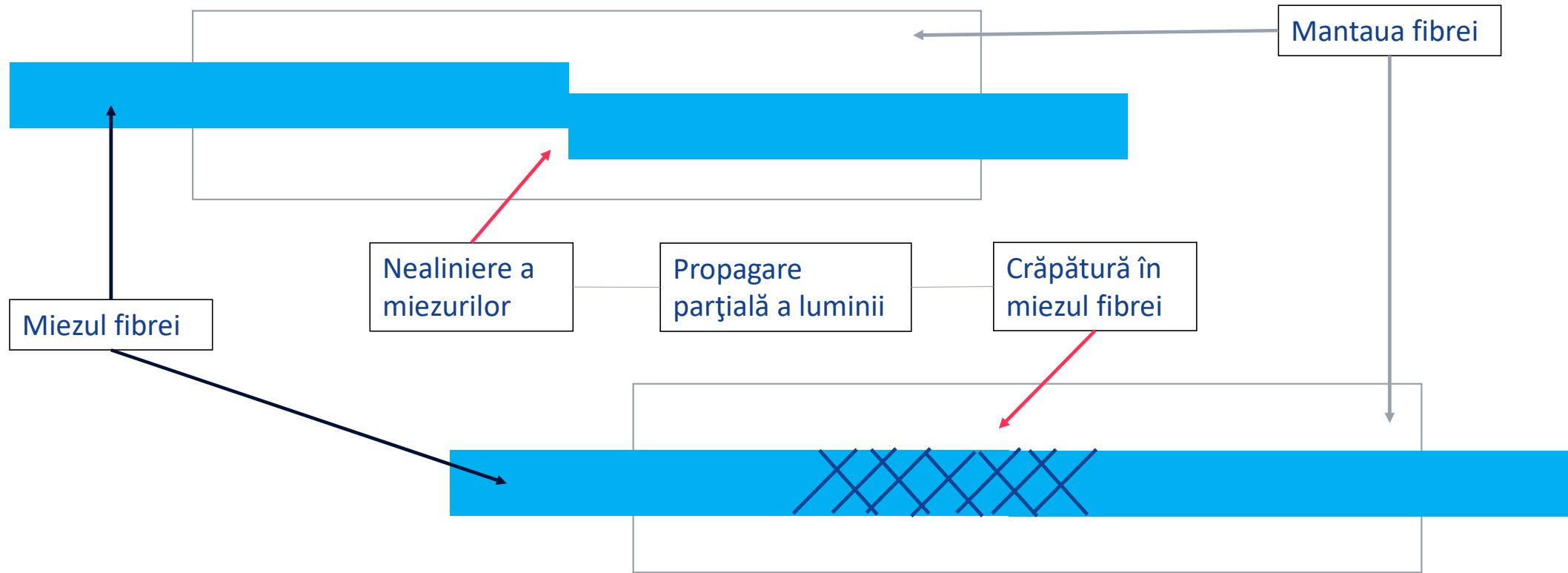
Fibra Optică – Exemplu de cablu de 32 perechi



Atenție la culori – prin ele se face diferența atunci când trebuie să lipesc / să date 2 capetei ai unei F.O.

Medii de transmisie

Fibra Optică – Probleme rezultare din sudură



Fibre Optice

Dăunătorii fibrelor optice



Fibra Optică

Echipamente utilizate pentru lucru cu FO



1

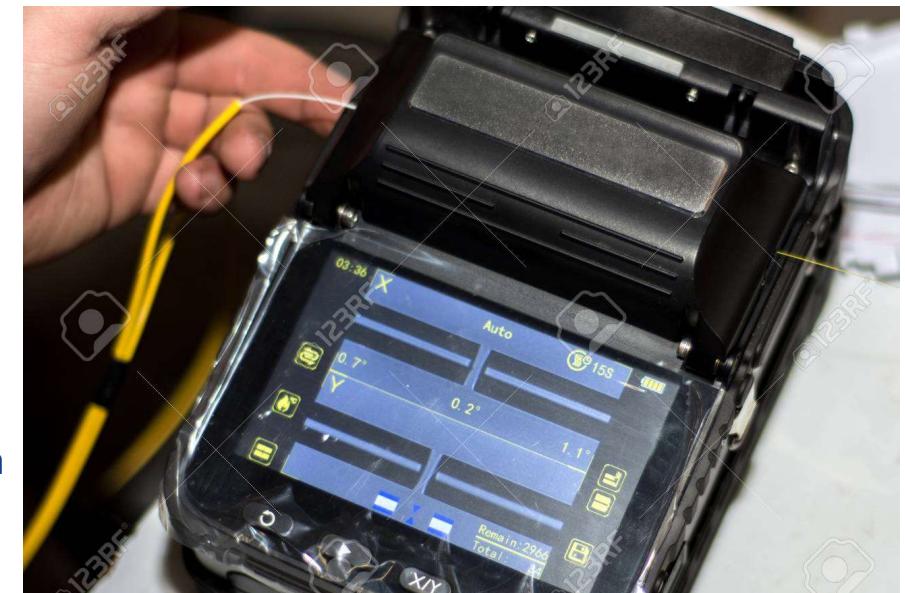


2



3

1. Detector luminos de defecte
2. Instrument de tăiat și îndepărtaț mantaua
3. Aparat detecție distanță a defectelor
4. Aparat de sudură F.O.





That's all for today, see you next time!

Rețele de Calculatoare

Codarea informației

Sumar al laboratorului

1

Tehnici de codare

Unde facem codarea

2

Tipuri de coduri

Codarea sursei

Coduri de canal

Criptografie

Coduri de linie

Tehnici de Modulație

3

AM

FM

PM

QAM

OFDM

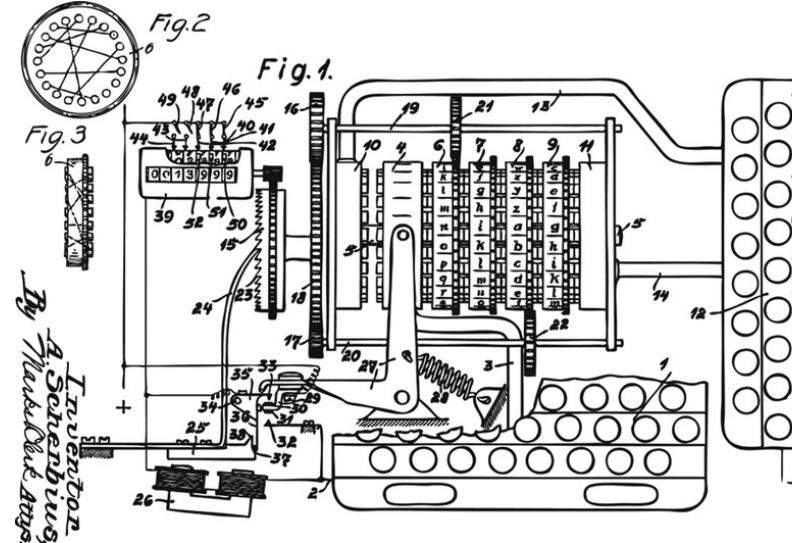


Teoria codării

Teoria codării – studiul despre proprietățile codurilor și aplicarea lor în aplicații specifice

Utilizări ale codării datelor:

- Compresia datelor
- Controlul erorilor
- Criptografie
- Codarea de linie



Mașinărie de codare Enigma – WW2

Sursa:

<https://web.stanford.edu/class/cs106j/handouts/36-TheEnigmaMachine.pdf>



Transmiterea datelor

Unde găsim coduri

Emitător → Canal de transmisie → Receptor



Compresia datelor
+
Criptografie



Coduri de linie



Controlul erorilor
+
Criptografie

Codarea la emisie

Compresia datelor

Scopul codării la sursă este de a restructura datele în aşa fel încât să le reducem dimensiunile.



Size: 17.4 MB (18,283,218 bytes)

Imagine originală NEF

Size: 13.8 MB (14,574,738 bytes)

Imagine originală JPG

Size: 545 KB (558,514 bytes)

Imagine downloadată de pe Fb

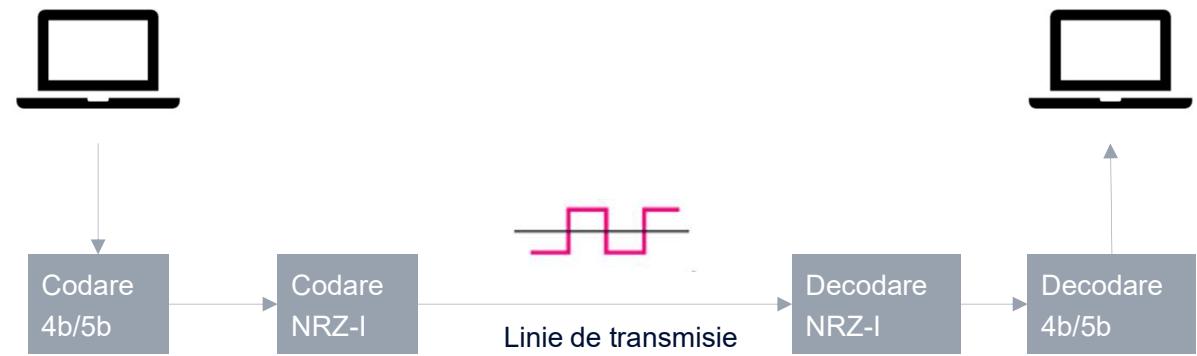
Codarea de canal

Controlul erorilor

Scopul codurilor de canal este de a transmite cât mai repede și cu minim de erori informația

Astfel codurile de canal sunt diferite pentru diversele medii de transmisie

- Coduri pentru Ethernet:
 - 4b/5b pt 100Mbps
 - 8b/10b pt 1Gbps
- Coduri pentru FO
- Coduri pentru WiFi



Coduri de canal

4b/5b

Exemplu:

Pornind de la numele propriu determinați secvența 4b/5b corespunzătoare

Soluție:

1. Folosim un convertor online ascii to hex/binary
 1. <https://www.rapidtables.com/convert/number/ascii-to-binary.html>
2. Folosim tabelul alăturat pentru a găsi secvența 4b/5b

Datele		Codul 4b/5b
Hexazecimal	Binar	
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101



Criptografie

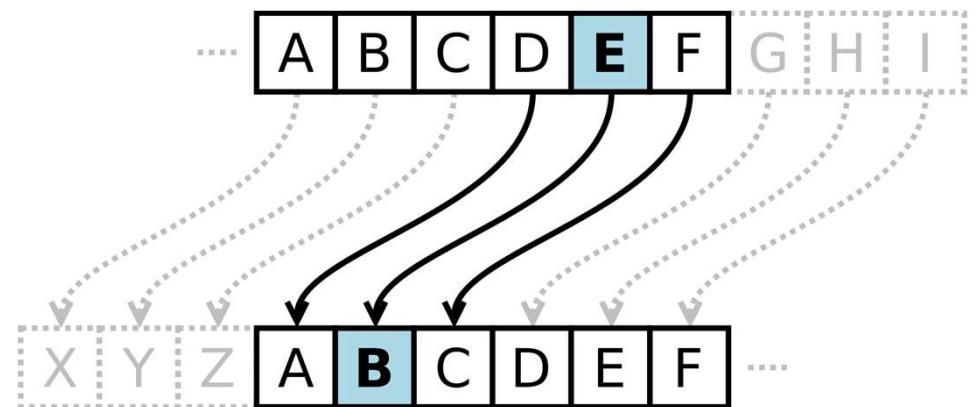
Exemple



Carte de decifrare din secolul 16, Franta

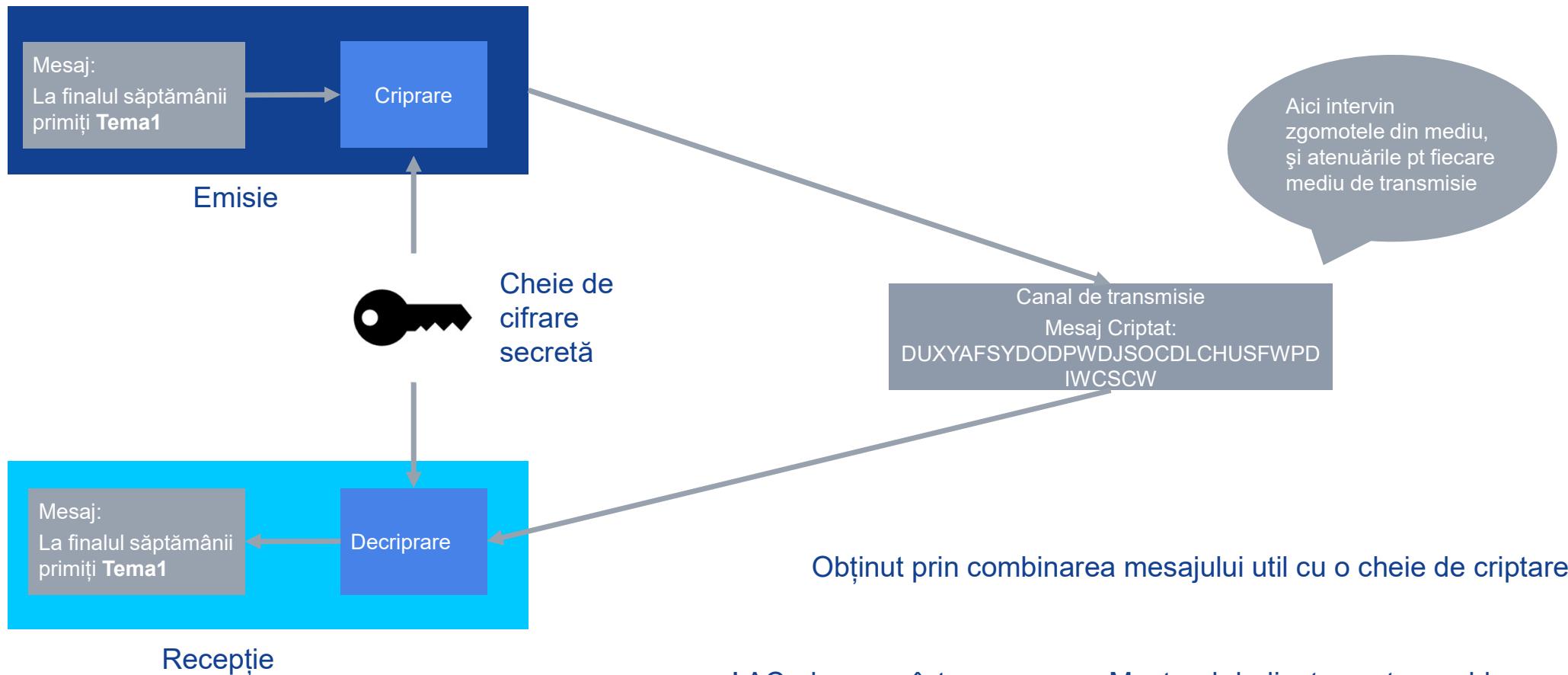
Scopul codării criptografice este de a ascunde mesajul

Cifrul lui Cezar – shift-are a simbolurilor la stânga



Criptografie

Mecanismul de bază



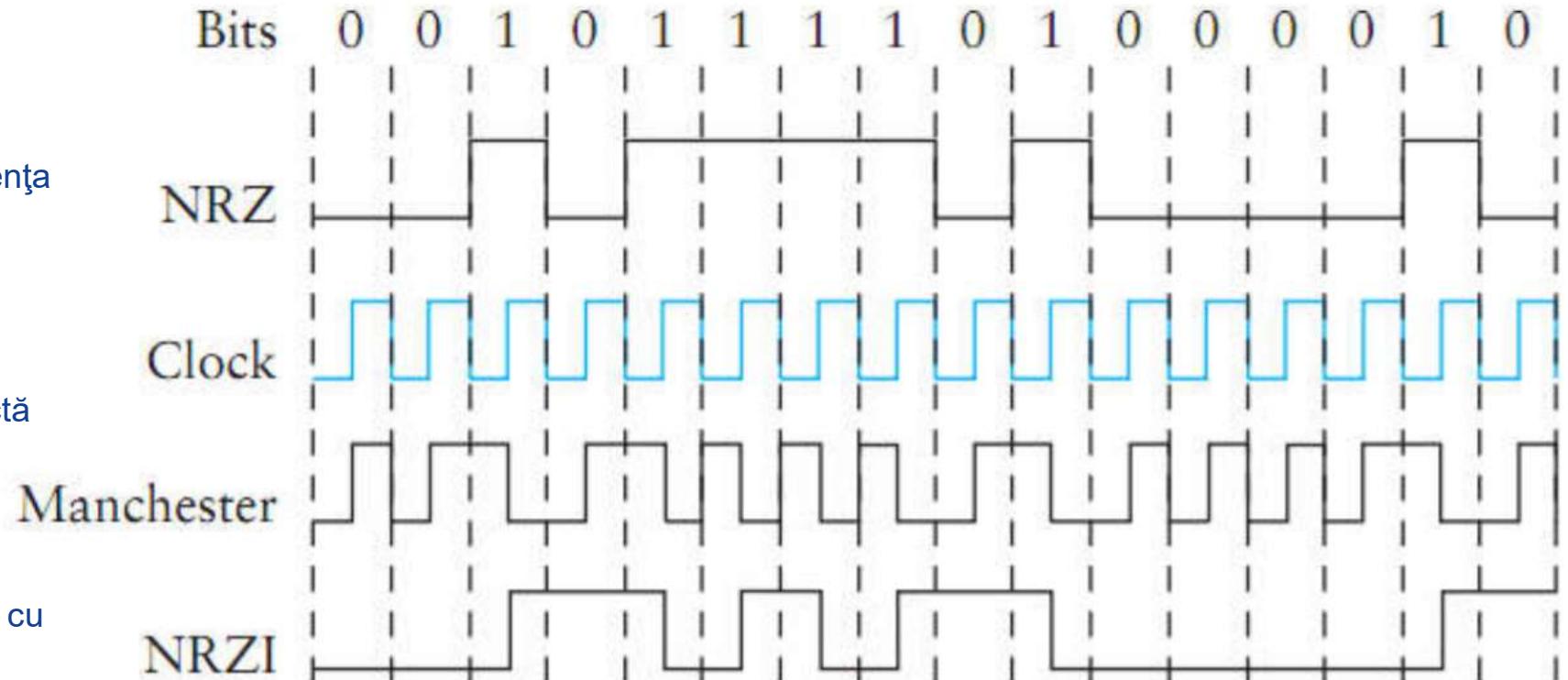
Coduri de linie

Atenție la ceea ce transmitem

Odată cu datele transmise se transmite și frecvența de tact - clock

Semnal lung pe 1 poate duce la nerefacerea corectă a tactului

Semnal lung pe 0 poate fi confundat cu lipsa semnalului



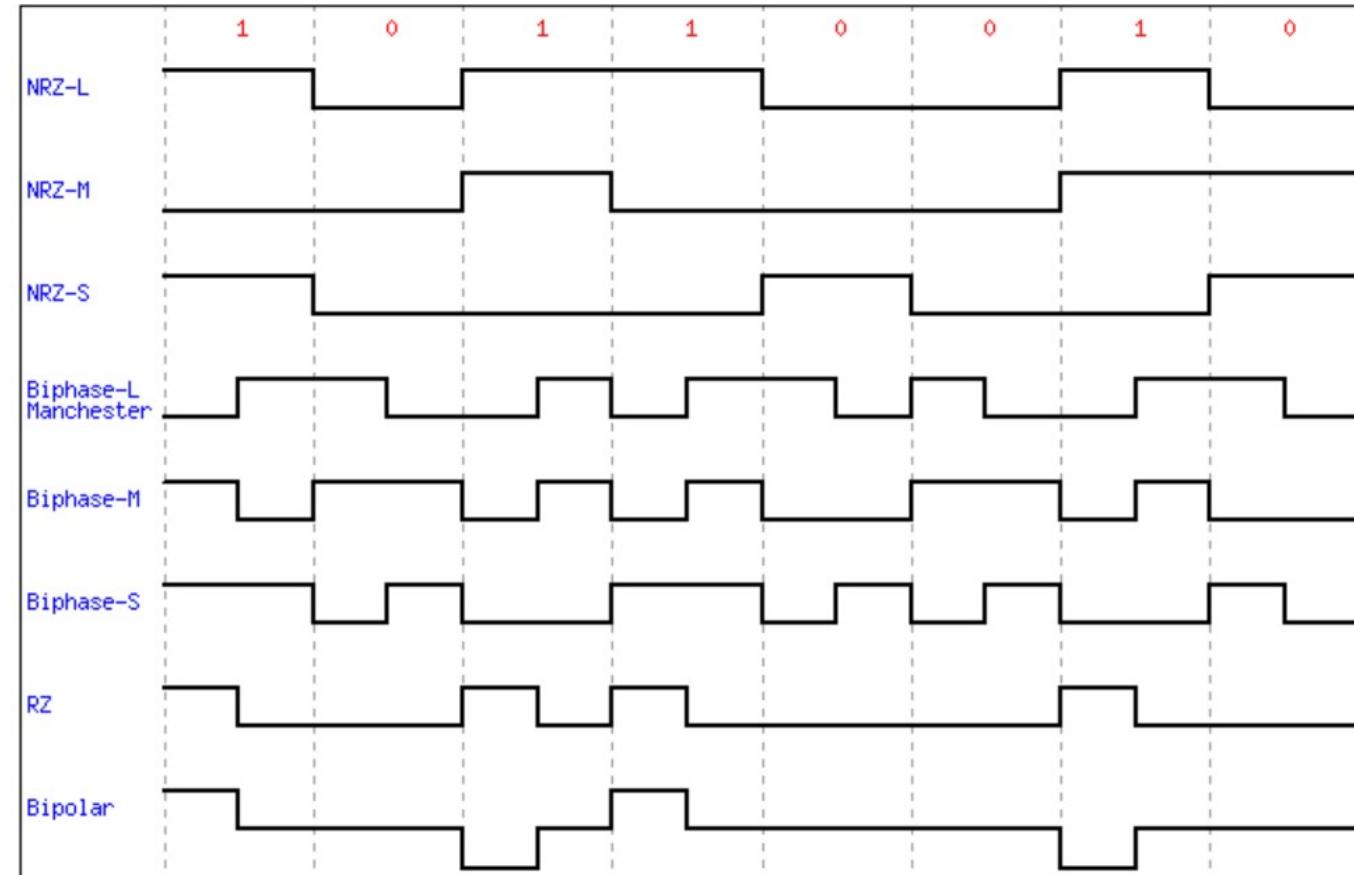
Coduri de linie

Exemple

ATENȚIE la Campusul Virtual:
Pagina de Binary encoding tool vă
dă posibilitatea de a experimenta

Secvența binară codată:
10110010

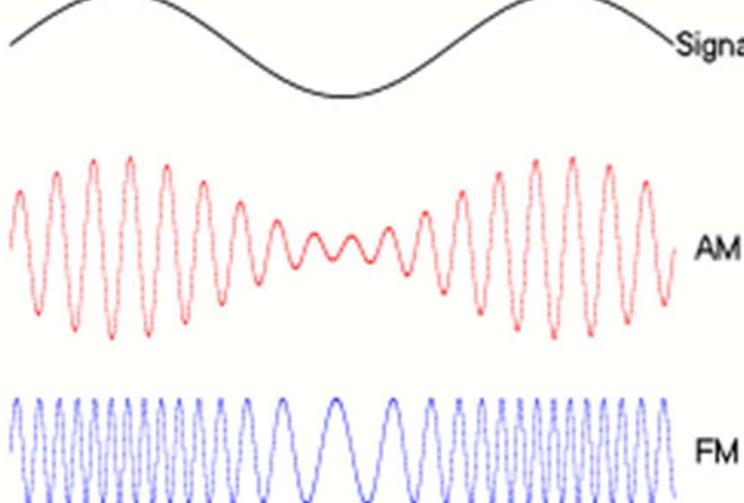
În cadrul laboratorului ne vom
concentra pe:
RZ, NRZ-I, Manchester, 4b/5b



Tipuri de modulații

Modulație de amplitudine(AM)

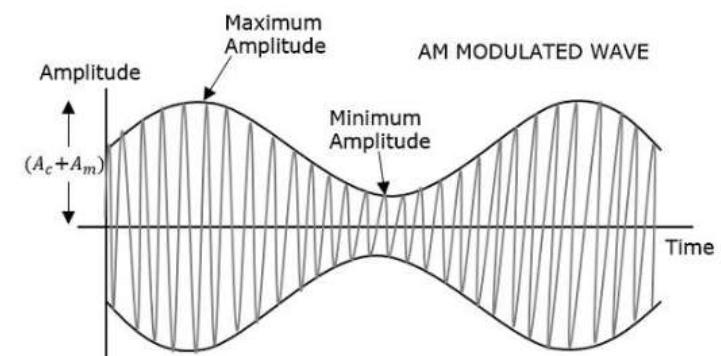
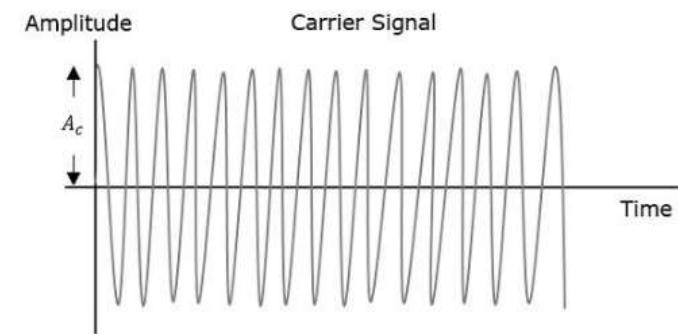
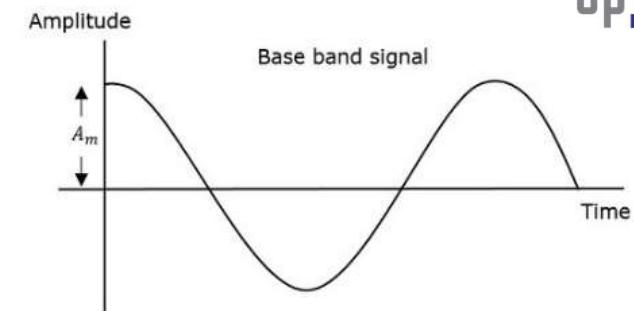
$$s(t)=A \cdot \sin(\omega t + \varphi)$$



A - Amplitudine

ω - Frecvență

φ - Faza semnalului



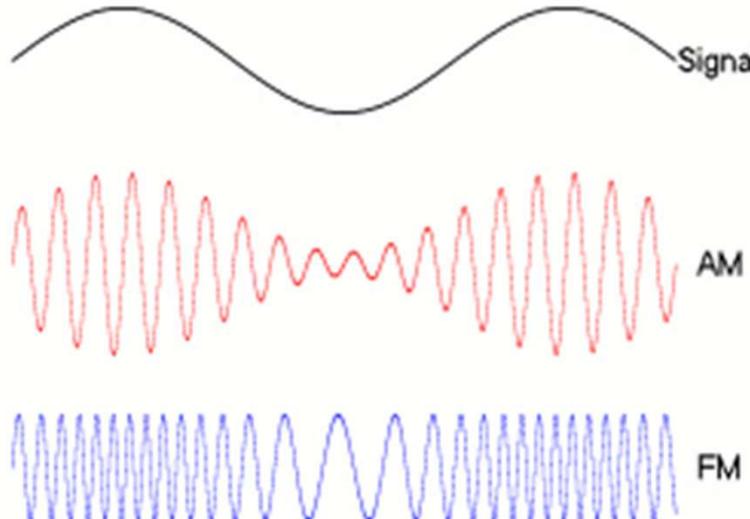
Sursa:

https://www.tutorialspoint.com/analog_communication/analog_communication_amplitude_modulation.htm#

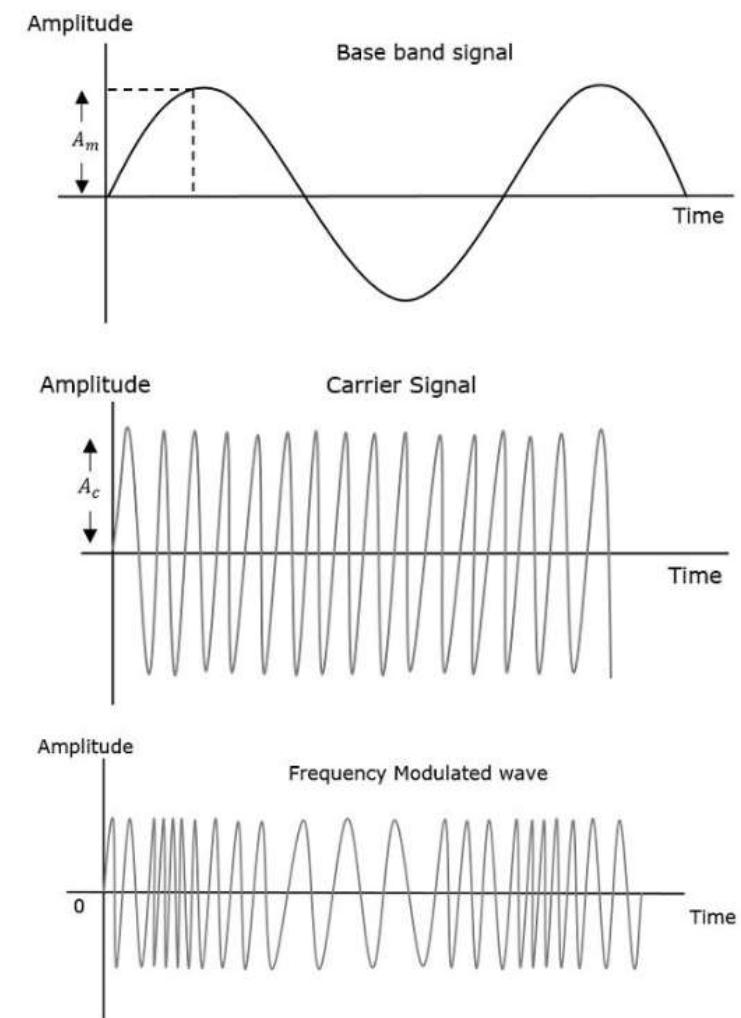
Tipuri de modulații

Modulație de Frecvență (FM)

$$s(t) = A * \sin(\omega t + \varphi)$$



A - Amplitudine
ω - Frecvență
φ - Faza semnalului



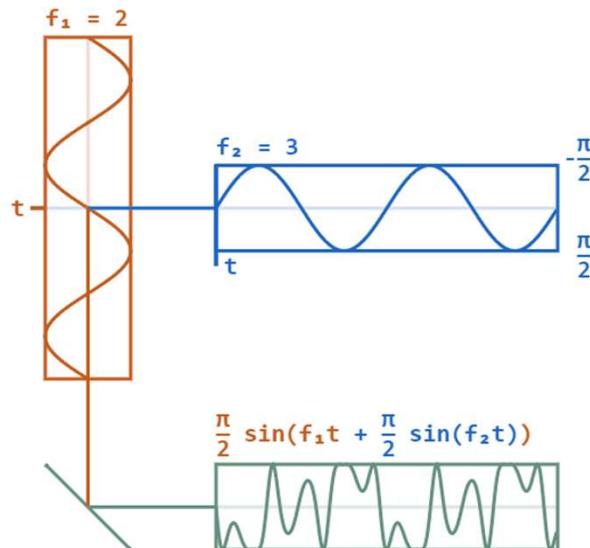
Sursa:

https://www.tutorialspoint.com/analog_communication/analog_communication_amplitude_modulation.htm#

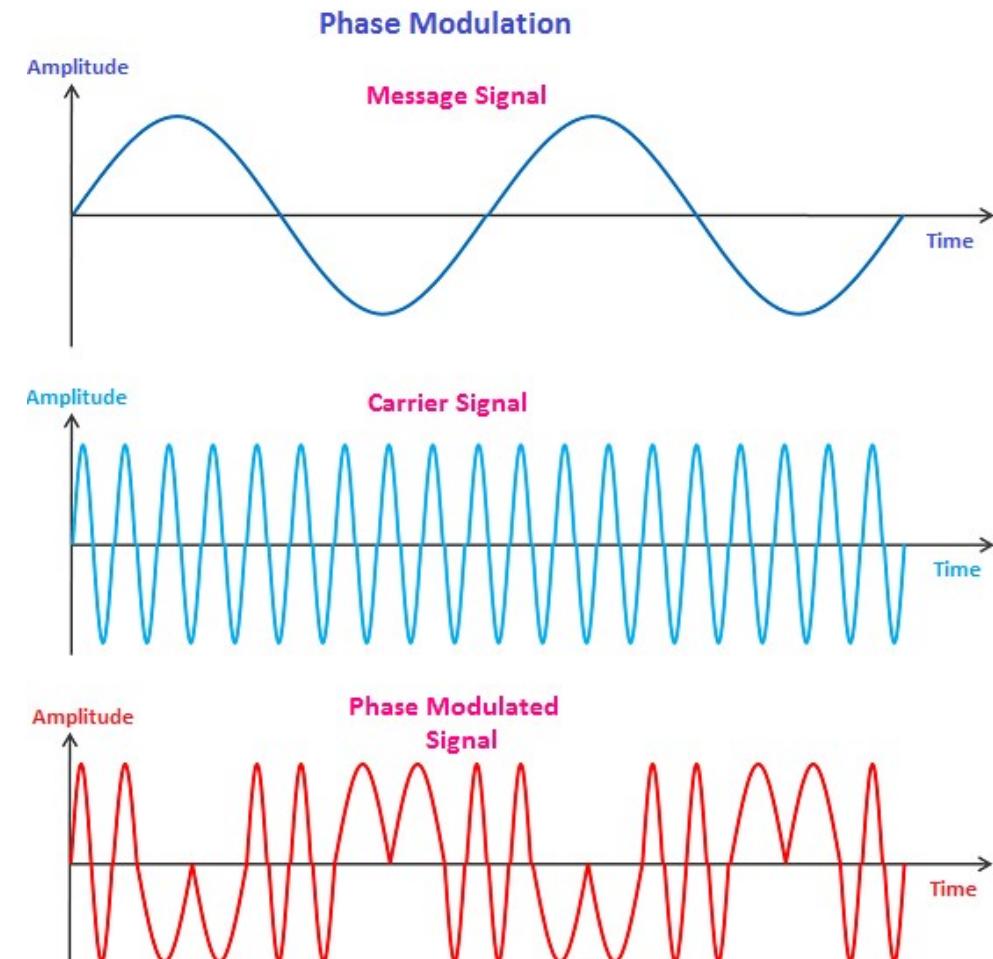
Tipuri de modulații

Modulație de Fază (PM)

$$s(t) = A * \sin(\omega t + \varphi)$$



A - Amplitudine
 ω - Frecvență
φ - Faza semnalului

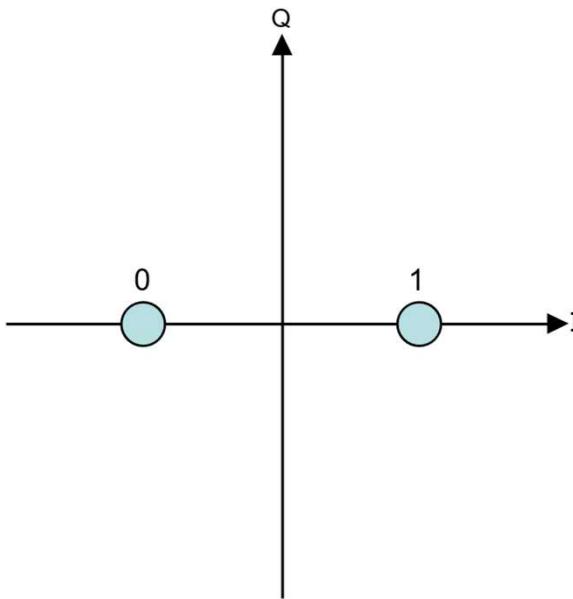


Sursa:

<https://www.physics-and-radio-electronics.com/blog/phase-modulation/>

Tipuri de modulații

Modulații complexe

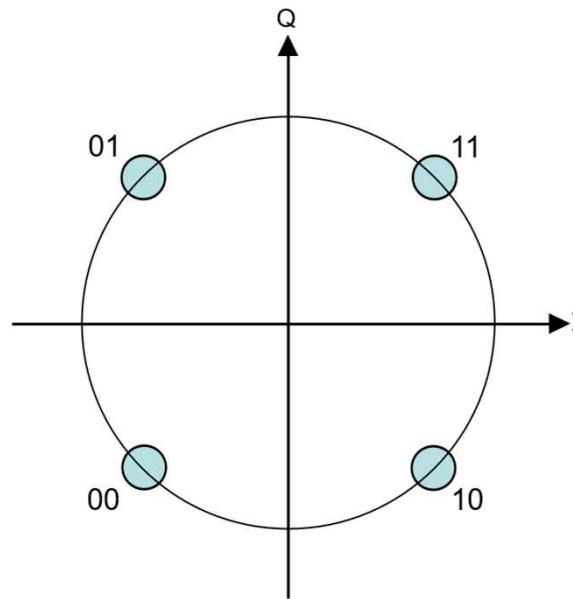


Binary phase-shift keying
(BPSK)

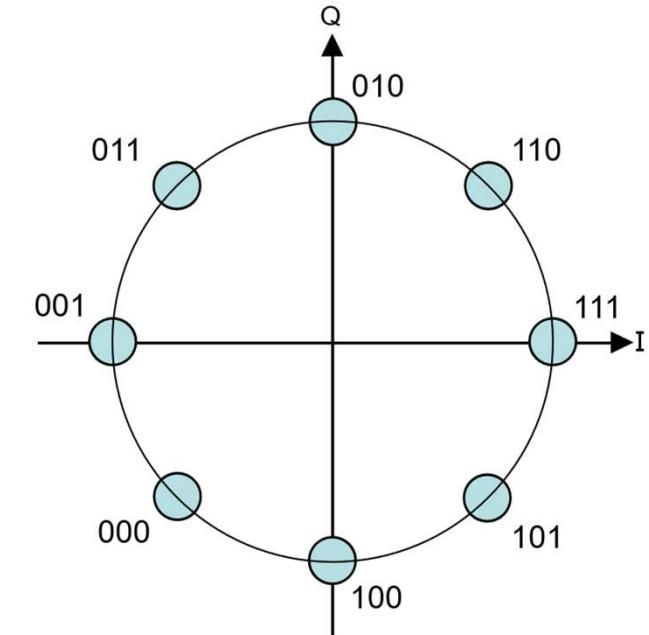
Utilizat în diverse situații pe WiFi

1Mbps – differential BPSK

2Mbps – differential QPSK



Quadrature phase-shift keying
(QPSK)



8 – PSK with Grey encoding
(DPQPSK)

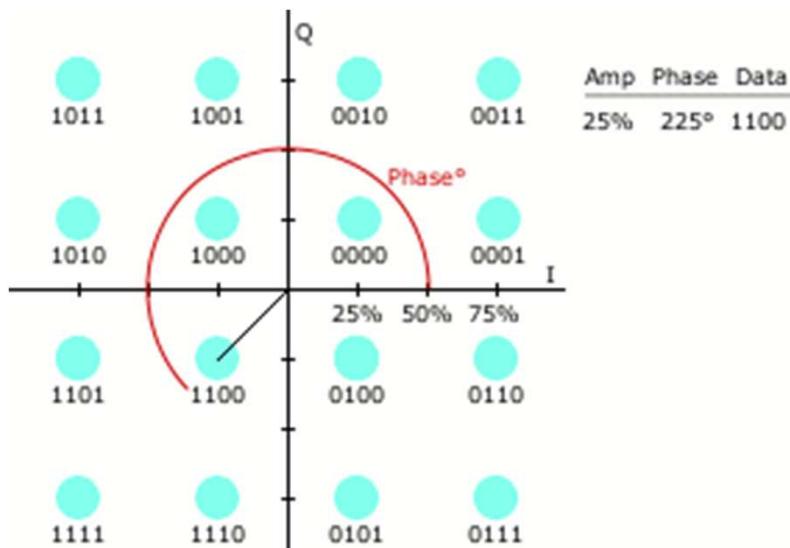
Utilizat în diverse situații pe Bluetooth

4 DQPSK – 2Mbps

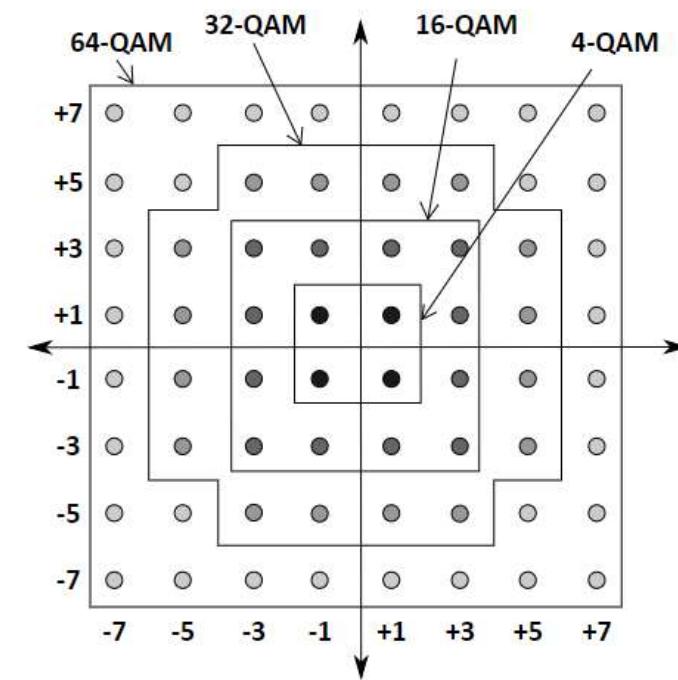
8-DPSK – 3Mbps

Tipuri de modulații

Modulații de quadratură - QAM



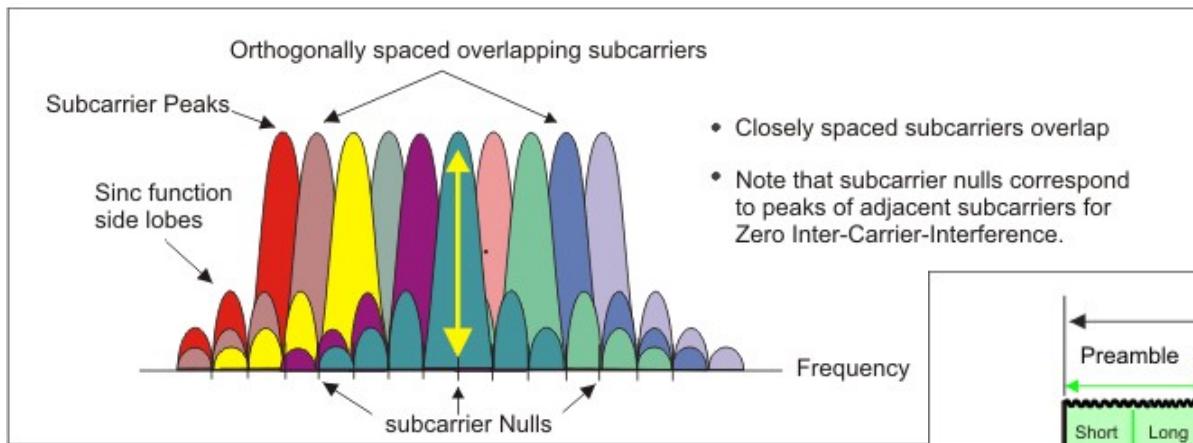
Decodarea modulației QAM



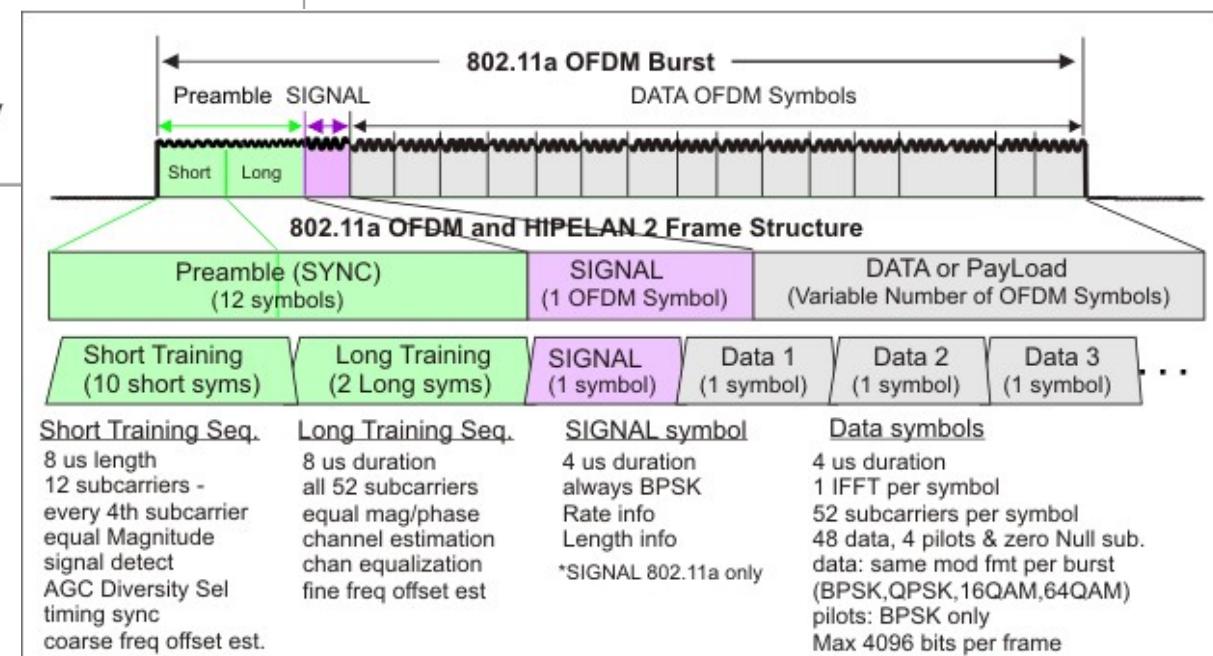
Diverse grade de QAM

Tipuri de modulații

Orthogonal Frequency Division Multiplexing - OFDM



Spațierea semnalelor
“purtătoare” OFDM



Cadru de transmisie WiFi pe
standardul 802.11a

Sursa:

https://rfmw.em.keysight.com/wireless/helpfiles/89600B/WebHelp/Subsystems/wlan-ofdm/Content/ofdm_basicprinciplesoverview.htm

17 © 2020 Nokia

NOKIA



That's all for today, see you next time!

Nu uitați de tema de casă
termen de predare până în data de 31.10.2022

Rețele de Calculatoare

Adresarea în rețelele de calculatoare

1

Încapsularea datelor
Antetul Network Access
Antetul Internet

2

Adresare fizică
Adrese MAC

3

Adresare logică
Adrese IPv4
Adrese IPv6
EUI-64

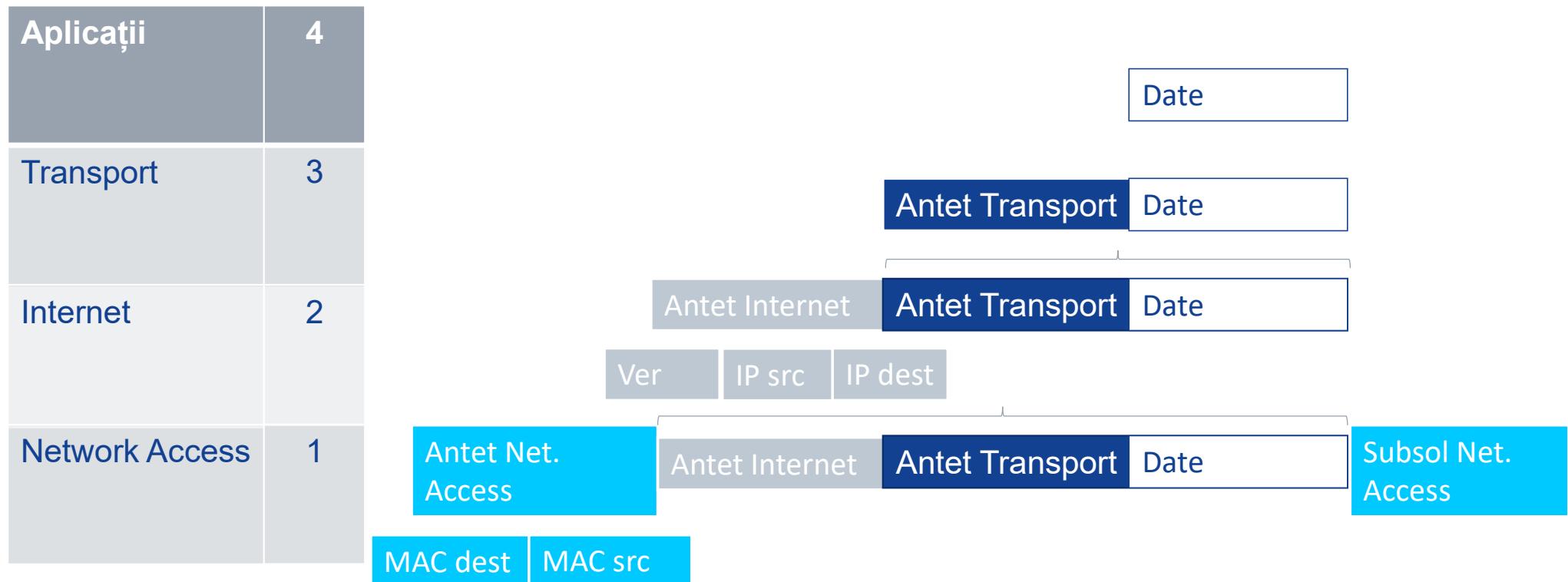
4

Protocolul ARP
Ce face
Cum funcționează



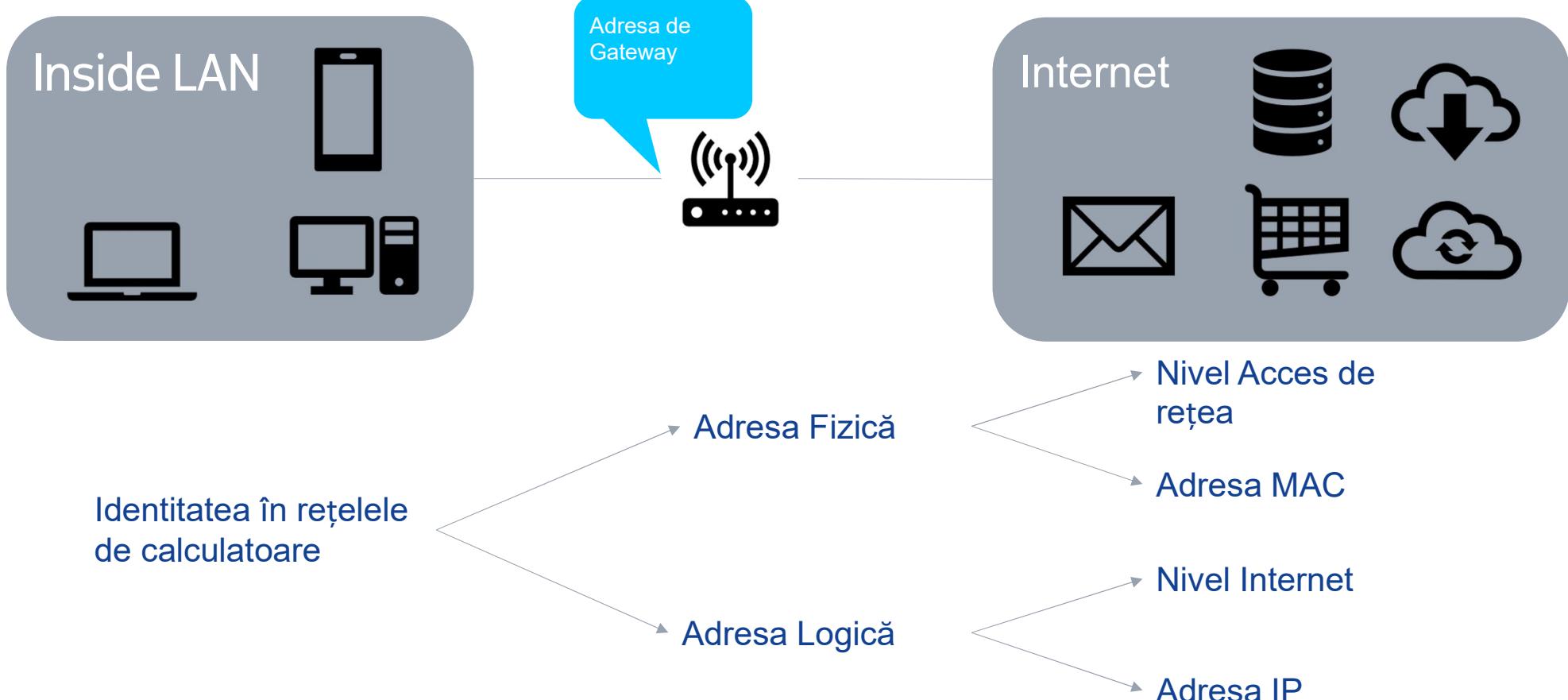
Procesul de încapsulare

Recapitulare



Identitatea În Rețelele de Calculatoare

Cum ne identificăm



Încapsularea datelor

Antetul Network Access



Câmp	Dimensiune	Explicații
Preambul	8 octeți	Sincronizează dispozitivele de emisie și receptie
MAC dest.	6 octeți	Adresa fizică a destinatarului. Unicast, multicast sau broadcast
MAC sursă	6 octeți	Adresa fizică a interfeței de emisie. Tot timpul unicast
EtherType	2 octeți	Reprezintă protocolul încapsulat în cadrul Ethernet
Date	46 – 1500 octeți	Reprezintă pachetul IPv4
FCS	4 octeți	Utilizat pentru detectia erorilor într-un cadru

Adresare Fizică

Adrese MAC

Adresă Unică

La nivel global

Reprezentată pe

12 cifre hexazecimală

Alcatuită din

48 Biti

Exemple:

Windows: 54-E1-AD-BF-E2-AB
Linux: 00:50:56:ac:0a:e9

Formată
din

OUI
3 octeți

Valoare
Unică
3 octeți

NOKIA

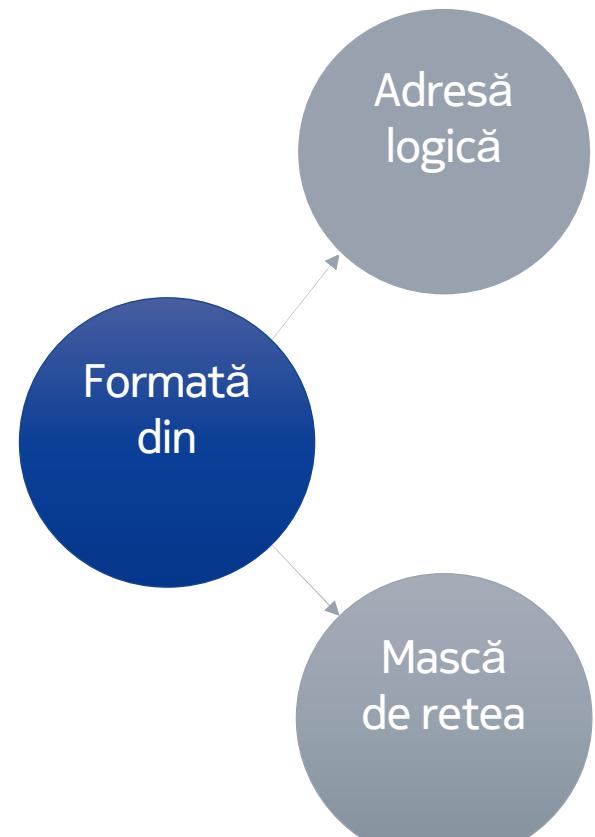
Adresare Logică

Adrese IP

Deși adresarea MAC este unică la nivel global, ar prezenta o adevarată problemă transferul de date de la un host la altul dacă ne-am baza doar pe aceasta.

Se introduce conceptul de adresare logică – adresa IP

În rețelele de calculatoare, informația este transportată dintr-o rețea în alta utilizând rutere, iar acestea funcționează pe baza adreselor de rețea



NOKIA

Adresare Logică

Adresa IPv4

- Adresele IPv4:
 - Formate dintr-un sir de 32 de biți și un separator “.”
 - O parte dintre biți identifică rețeaua, iar cealaltă parte definește zona de gazde (host-uri)

	Network Portion			Host Portion
IPv4 Address	192	168	10	21
Binary	11000000	10101000	00001010	00010101

- Masca de rețea (sau prefixul)
 - Formată din 32 biti cu 2 modalități de prezentare:
 - Binar/zecimal similar cu adresele de IP
 - “/xx” unde XX reprezintă numărul de biți de “1” din mască
 - Identifică rețeaua din care un dispozitiv face parte, prin comparare bit cu bit cu adresa IP
 - Biții de 1 din masca de rețea identifică rețeaua, în timp ce biții de 0 identifică host-urile

Adresare Logică

Adrese IPv6 – de ce?

Adresele de 128 biți care sunt folosite în IPv6 permit un număr mai mare de adrese



Am ajunge
la
 10^{15}
adrese

IPv6 are următoarele avantaje în comparație IPv4:

- Managementul și delegarea adreselor devine mai ușoară;
- IPsec încorporat – Securitate ridicată;
- Rutare optimizată;
- Depistarea adreselor duble.

Adresare Logică

Adrese IPv6 – reguli de abreviere

În acest exemplu pornim de la adresa:

2031:0000:130F:0000:0000:09C0:876A:130B

Identificăm grupuri compacte de “0”:

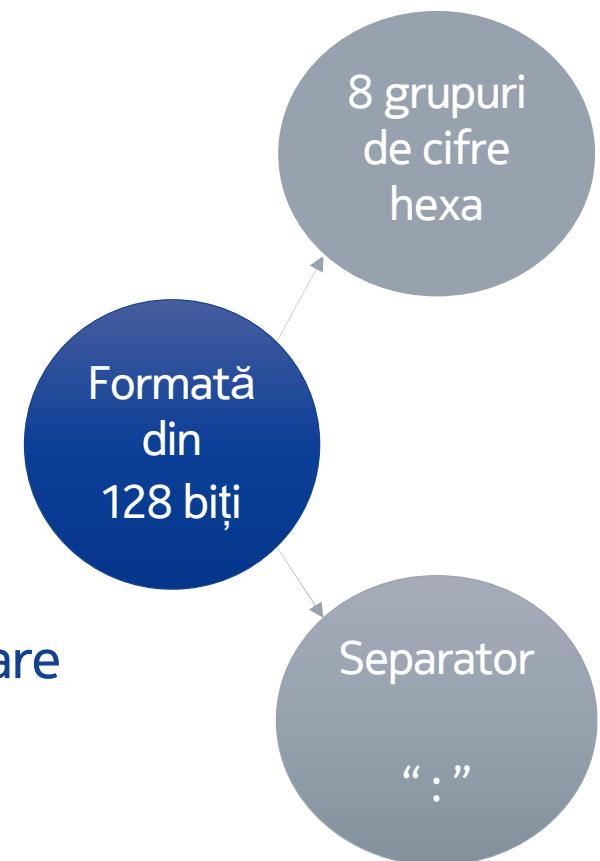
2031:0000**:130F:**0000**:**0000**:**09C0**:876A:130B**

Etapa 1 – înlocuim grupurile de “0” cu un singur simbol:

2031:0**:130F:**0**:**0**:**9C0**:876A:130B**

Etapa 2 – domeniile succesive de “0” sunt înlocuite de separatoare consecutive:

2031:0**:130F::**9C0**:876A:130B**



Adresare Logică

Tipuri de adrese - 1

Tip adresă	Adresare MAC	Adresare IP
Unicast	00-07-E9-42-AC-28	192.168.1.200/24
Multicast	01-00-5E-00-00-C8	224.0.0.200
Broadcast	FF-FF-FF-FF-FF-FF	192.168.1.255/24

Adrese IP private:

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

Adresă MAC Broadcast: FF-FF-FF-FF-FF-FF

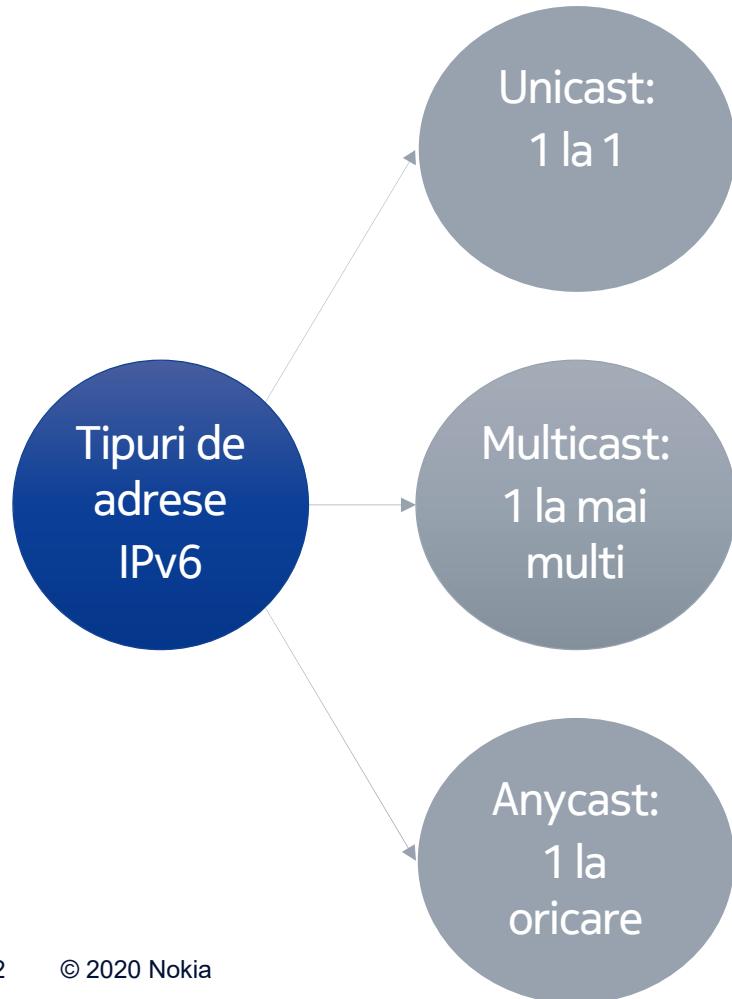
Adrese IP Broadcast: bitii din partea de host sunt setați pe 1

Adrese MAC Multicast: 01-00-5E-xx-xx-xx

Adrese IP Multicast: 224.0.0.0 – 239.255.255.255

Adresare Logică

Tipuri de adrese - 3



Adresa echivalentă public IPv4

Adresa valabilă la nivelul rețelei locale
(firme cu mai multe sedii)

Adresa echivalentă privată
in IPv4

Adresare Logică

Tipuri de adrese unicast IPv6

Adrese unicast Globale – 2000::/3 (cele care incep 2...)

2003:4581:A7C1:EFDB::1327:1

2017:ACAD:1234:9999:FFFF:0010:51CD:AAAF

2001:db8:a0b:12f0::1 (sau poate fi scris cu litere mici)

Adrese unicast Link Local (Locale) – FE80::/10

FE80::C001:37FF:FE6C:0

FE80::203:FFFF:FEE1:2a74

Adrese unicast site-local:

Primii 10 biți pot lua valori astfel încât primul câmp este între FEC0 și FEFF.

Adresare logică

Adrese IPv6 – cum le obținem: EUI-64 sau generare aleatoare

1

Se ia adresa MAC și se împarte în 2 componente

2

Se introduc octeții FF și FE în mijlocul adresei

3

Al 7-lea bit din primul octet își modifică valoarea



Încapsularea datelor

Antetele protocoalelor IP

Version	Lungime antet	Tipul serviciului	Lungime totală	
Identificare		Flags	Offset	
Time to live	protocol	Header checksum		
Ip Sursă (32 biti)				
Ip Destinatie (32 biti)				
Optiuni				

IPv4

Antetele protocoalelor

IPv6

În cadrul laboratorului
ne interesează doar
campurile ROȘII

De câmpul TTL/nr de hop-uri ne
vom lega în cadrul altui lab.

Version	Trafic Class	Flow label
Lungimea pachetului	Următorul antet	Numar de hop-uri
Adresa Ip Sursă (128 biti)		
Adresa IP Destinatie (128 biti)		
Date		

Protocolul ARP

Funcție

```
C:\> C:\WINDOWS\system32\cmd.exe
C:\Users\cmisici>arp -a

Interface: 135.243.230.21 --- 0xb
Internet Address      Physical Address      Type
135.243.230.1          2c-fa-a2-49-b2-6e  dynamic
135.243.230.14         54-e1-ad-bf-e5-af  dynamic
135.243.230.20         54-e1-ad-bf-e6-9d  dynamic
135.243.230.28         f4-30-b9-19-49-b9  dynamic
135.243.230.33         00-50-b6-a1-f4-e7  dynamic
135.243.230.34         54-e1-ad-bf-e1-d0  dynamic
135.243.230.46         28-80-23-00-f6-aa  dynamic
135.243.230.48         8c-16-45-5f-24-54  dynamic
135.243.230.50         a0-2b-b8-3a-85-67  dynamic
135.243.230.77         8c-16-45-31-9a-55  dynamic
135.243.230.106        8c-16-45-78-d8-56  dynamic
135.243.230.120        fc-3f-db-ff-64-4b  dynamic
135.243.230.160        8c-16-45-59-18-5d  dynamic
135.243.230.207        54-e1-ad-bf-e5-b1  dynamic
135.243.230.209        54-e1-ad-bf-e1-26  dynamic
135.243.230.225        98-e7-f4-f1-46-00  static
135.243.230.234        54-e1-ad-bf-c3-ac  dynamic
135.243.230.253        3c-18-a0-b0-c1-03  dynamic
135.243.231.255        ff-ff-ff-ff-ff-ff  static
169.254.181.255        8c-16-45-78-d6-9a  dynamic
224.0.0.22              01-00-5e-00-00-16  static
224.0.0.251             01-00-5e-00-00-fb  static
224.0.0.252             01-00-5e-00-00-fc  static
239.255.255.250        01-00-5e-7f-ff-fa  static
255.255.255.255        ff-ff-ff-ff-ff-ff  static

C:\Users\cmisici>
```

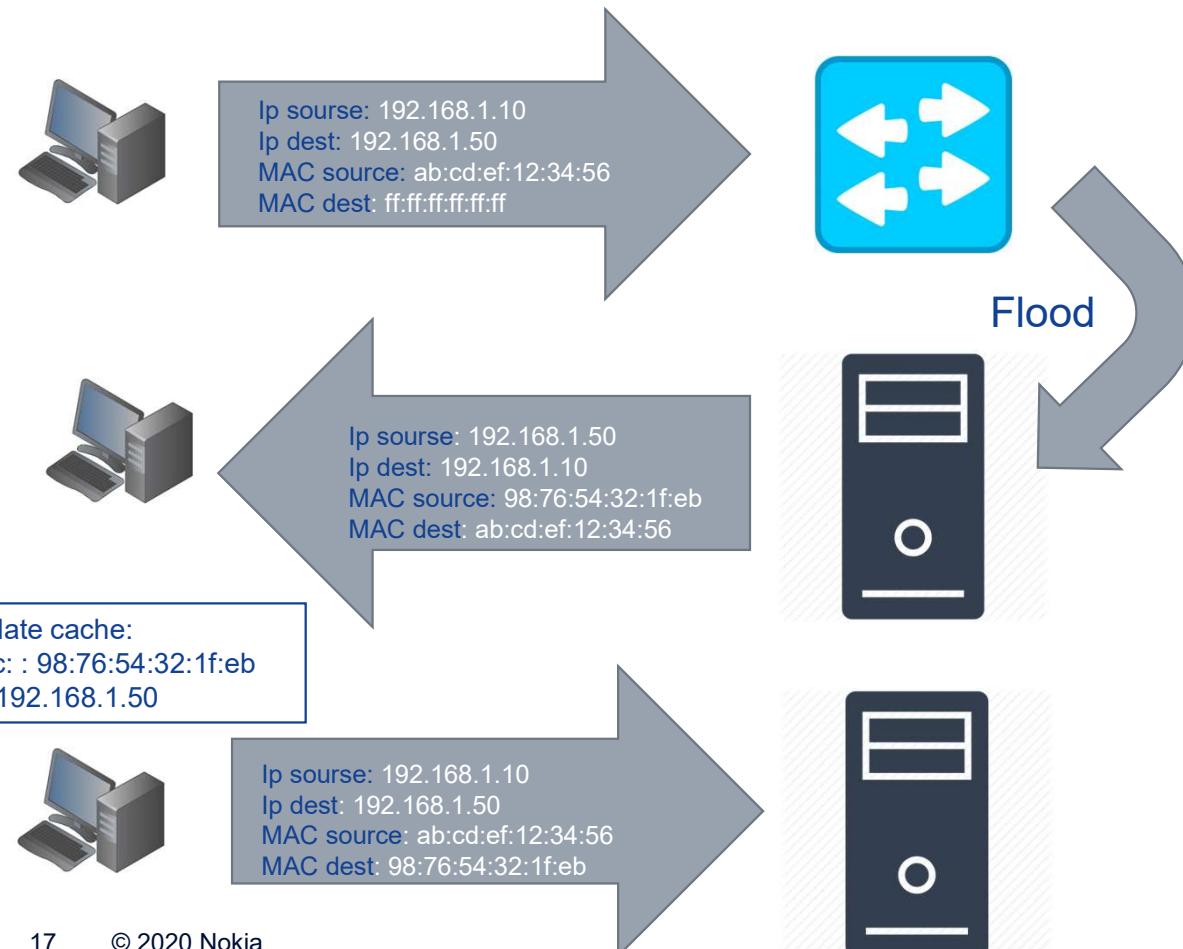
ARP- Address Resolution Protocol

Funcție: mapează o adresă fizică pe o adresă logică

Din fereastra cmd.exe din windows putem vedea întreaga tabelă ARP a unui host

Protocolul ARP

Funcționare



1. Înainte de a trimite pachete în mod unicast este necesară cunoașterea MAC-ului destinație
2. Emițătorul (E.) trimite o cerere ARP de tip broadcast, utilizând adresa de IP a Receptorului (R.)
3. Switch-ul trimite broadcast pe toate porturile sale, R. își recunoaște adresa IP și trimite un răspuns de tip unicast cu MAC-ul și IP-ul său
4. E. primește un răspuns ARP, își face update a tabelui ARP și va transmite mai departe unicat.



That's all for today, see you next time!

Nu uitați de tema de casă
termen de predare până în data de 14.11.2020

Rețele de Calculatoare

Rețele Wireless - WiFi

Sumarul laboratorului

1

Rețele Wireless

Rețele Mobile
Bluetooth
IRDA
WiFi

2

Analizoare WiFi

Acrylic WiFi Home
Analizoare Linux

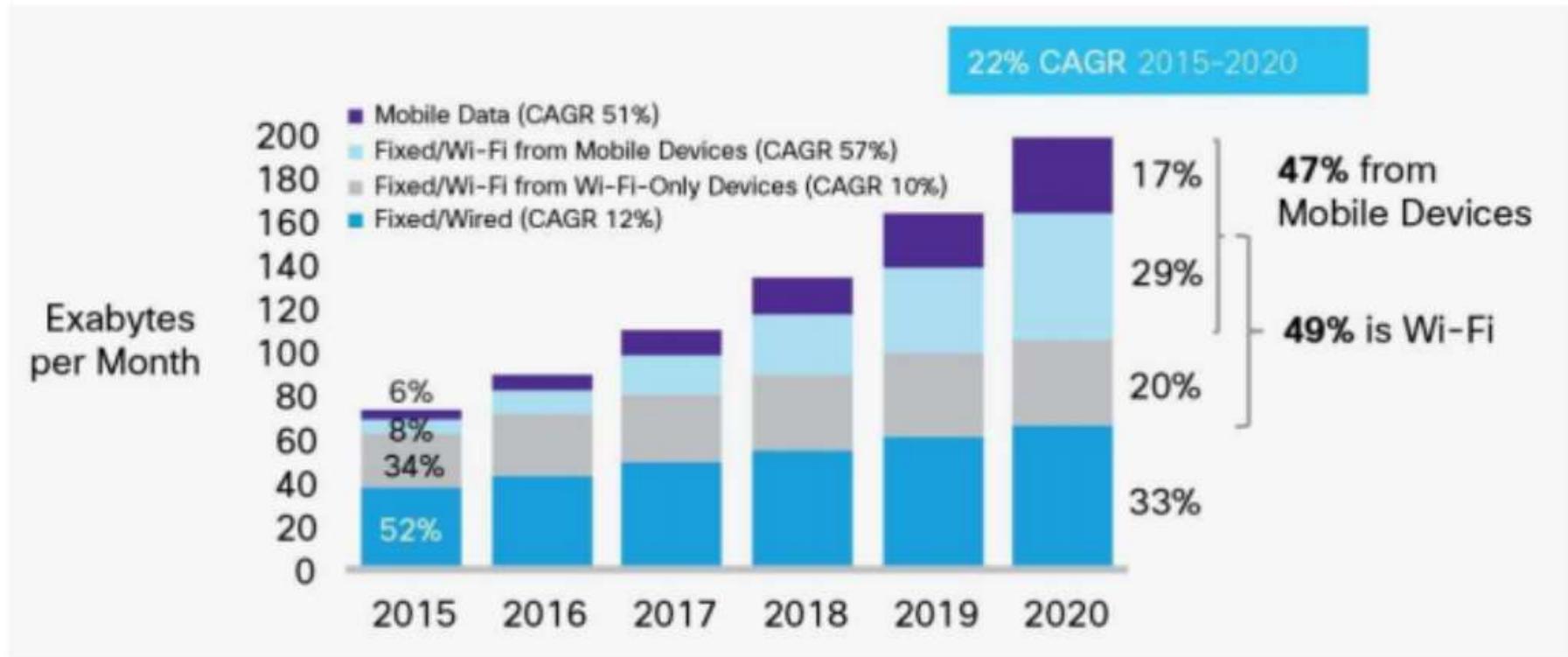
3

Cum configurăm un
ruter WiFi



Rețele Wireless

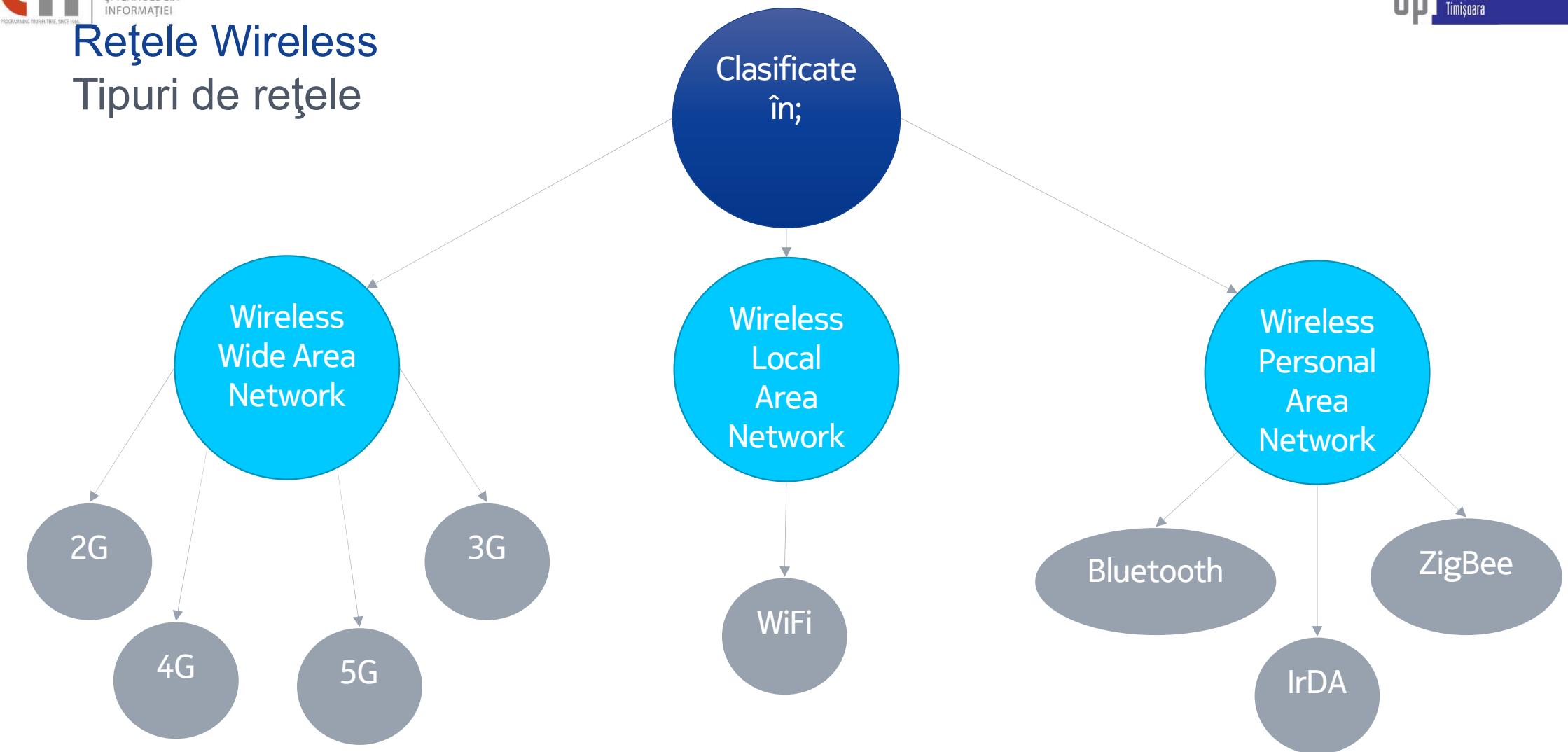
De ce le discutăm separat?



*Figure 1: IP Traffic By Access Technology
Cisco Visual Networking Index*

Rețele Wireless

Tipuri de rețele

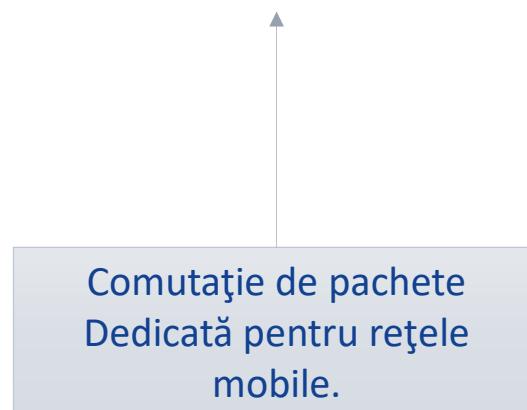
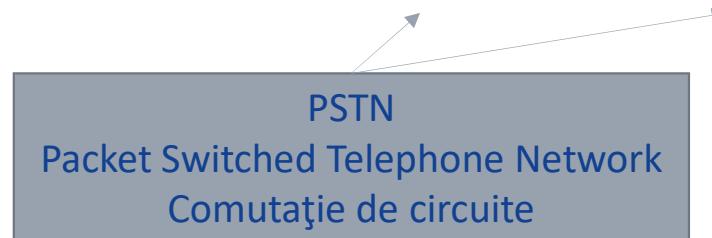


Sursa:

https://en.wikipedia.org/wiki/Comparison_of_wireless_data_standards

Scurtă comparație între generațiile mobile

Specificații	1G	2G	3G	4G	5G
An de lansare	1970/1984	1980/1999	1990/2002	2000/2010	2010->
Tehnologie de bază	---	GSM	WCDMA	LTE/WiMAX	MIMO/mmWaves
Frecvență	30KHz	1.8GHz	1.6-2GHz	2-8GHz	3-30GHz
Debit	2kbps	14.4-64kbps	2Mbps	200Mbps-1Gbps	1Gbps an up
Rețeaua de core	PTSN	PTSN	Switching de pachete	Internet	Internet

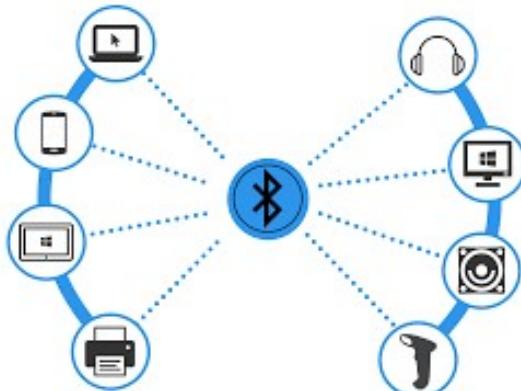


Categorii

Bluetooth

Caracteristici

- Debit: 768kbps (v1) -> 24Mbps (Bt Smart)
- Distanță: 10 – 100m
- Frecvență: 2,4GHz



Probleme:
Interferență de canal

IrDA

Caracteristici

- Debit: 9.6kbps (SIR) -> 1Gbps (GigaIR)
- Distanță: 2m
- Unghi maxim: 15°



Probleme: Întreruperea Line of sight (LoS)

Caracteristici

→ Standard: 802.11

Distanță: 46m
(indoor) – 92m
(outdoor)

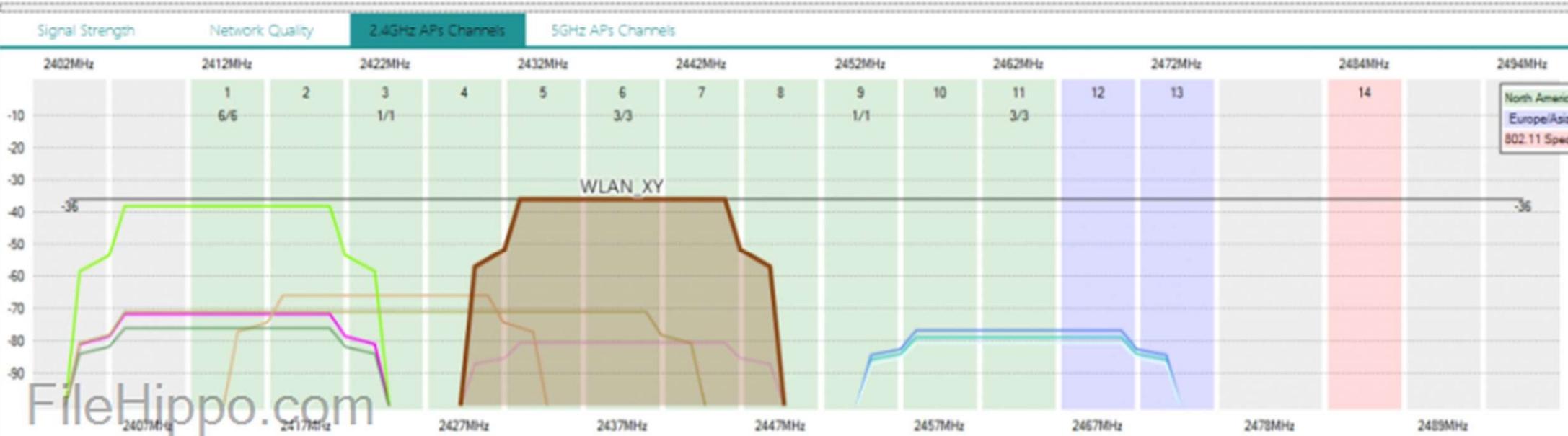
→ Versiuni: 18



Versiunea	Debit maxim [Mbps]	Frecvență [GHz]	Backwards compatibility
802.11a	54	5	Nu
802.11b	11	2.4	Nu
802.11g	54	2.4	802.11b
802.11n	600	2.4 & 5	802.11b/g
802.11ac	1300	2.4 & 5	802.11b/g/n
802.11ad	7000	2.4 , 5 & 60	802.11b/g/n/ac



SSID	MAC Address	RSSI	Chan	802.11	Max Speed	WEP	WPA	WPA2	WPS	Vendor	First	Last	Type
WLAN_XY	60:A4:4C:69:D2:48	-36	6	b, g, n	216.7 Mbps		MGT-CCMP			ASUSTek COMPUTER IN	18:57:14	now	Infrastructure
5b4d2d	AP 5b4d2d	-79	11	b, g, n	144.4 Mbps	PSK-(TKIP CCMP)	PSK-(TKIP CCMP)	1.0	PEGATRON CORPORATI	18:57:14	now	Infrastructure	
famalsu	E2:41:36:0C:4C:A0	-77	11	b, g, n	144.4 Mbps		PSK-CCMP				18:57:14	now	Infrastructure
WebSTAR	AP WebSTAR	-72	1	b, g	54 Mbps SharedKey					ASUSTek COMPUTER IN	18:57:14	now	Infrastructure
SERGIO	AP SERGIO	-66	3	b, g, n	130 Mbps	PSK-(TKIP CCMP)	PSK-(TKIP CCMP)	1.0	HITRON TECHNOLOGIES INC	18:57:14	now	Infrastructure	
HACKERS AHEAD	00:1E:E6:6B:A5:B3	-38	1	b, g	54 Mbps		PSK-CCMP			CISCO-LINKSYS, LLC	18:57:14	now	Infrastructure
RodMos	00:26:24:CD:D4:D4	-76	1	b, g	54 Mbps	PSK-(TKIP CCMP)	PSK-(TKIP CCMP)			Thomson Inc.	18:57:12	00:00:10 ago	Infrastructure
WIRE6969	E8:DE:27:C0:61:8E	-71	1+5	b, g, n	300 Mbps	PSK-(TKIP CCMP)	PSK-(TKIP CCMP)	1.0	TP-LINK TECHNOLOGIE	18:57:14	now	Infrastructure	
wifidientesR	02:71:C2:14:7D:CF	-81	11	b, g, n	144.4 Mbps	MGT-(TKIP CCMP)	MGT-(TKIP CCMP)				18:57:12	now	Infrastructure
BURLINGTON	F8:63:94:9A:16:B3	-81	6	b, g, n	144.4 Mbps	PSK-(TKIP CCMP)					18:57:40	00:00:10 ago	Infrastructure
CBS-27BA	14:B9:68:FD:27:C0	-83	1	b, g, n	270 Mbps	PSK-(TKIP CCMP)		1.0	HUAWEI TECHNOLOGIE	18:58:01	00:00:39 ago	Infrastructure	
iPhone de Javier	2A:70:45:D4:61:EF	-81	1	b, g	54 Mbps		PSK-CCMP				19:00:04	00:16:58 ago	Infrastructure
JAZZTEL_uhvf	54:22:F8:E4:85:70	-83	9	b, g, n	130 Mbps	PSK-CCMP	PSK-CCMP			zte corporation	19:01:33	00:15:30 ago	Infrastructure
wVF695	96:D8:59:10:EC:A2	-80	6	b, g, n	72.2 Mbps	PSK-(TKIP CCMP)					19:11:11	00:01:44 ago	Infrastructure



Analizoare WiFi

Acrylic WiFi Home

SSID	MAC Address	RSSI	Chan	Max Speed	WEP	WPA	WPA2	WPS	Vendor
UPC Wi-Free	AE:22:05:C2:4F:36	-21	36+40+44+48	1300.05 Mbps			MGT-CCMP		Compal Broadband Networks. In
Orange-HrN6	40:EE:DD:67:54:D8	-79	1+5	300 Mbps			PSK-CCMP		HUAWEI TECHNOLOGIES CO.LT
UPC Wi-Free	46:32:C8:9D:72:F1	-79	11	144.4 Mbps			MGT-(TKIP CCMP)		Technicolor CH USA Inc.
FBI Surveillance	54:67:51:41:99:F5	-79	36+40+44+48	1300.05 Mbps	PSK-TKIP		PSK-CCMP	1.0	Compal Broadband Networks. In
UPC Wi-Free	3A:43:1D:8C:57:31	-82	6	300 Mbps			MGT-(TKIP CCMP)		
HUAWEI-Q6Gy	90:17:AC:72:17:5C	-83	7	144.4 Mbps	PSK-(TKIP CCMP)		PSK-(TKIP CCMP)		HUAWEI TECHNOLOGIES CO.LT
UPCED7277	AC:22:05:C2:50:35	-10	36+40+44+48	1300.05 Mbps	PSK-TKIP		PSK-CCMP	1.0	Compal Broadband Networks. In
UPC Wi-Free	AE:22:15:C2:50:42	-17	11	144.4 Mbps			MGT-(TKIP CCMP)		
UPCED7277	AC:22:05:C2:50:42	-16	11	144.4 Mbps	PSK-(TKIP CCMP)		PSK-(TKIP CCMP)	1.0	Compal Broadband Networks. In
FBI Surveillance	54:67:51:41:99:C5	-76	1	144.4 Mbps	PSK-(TKIP CCMP)		PSK-(TKIP CCMP)	1.0	Compal Broadband Networks. In
UPC Wi-Free	56:67:11:A1:90:BC	-83	1	300 Mbps			MGT-(TKIP CCMP)		
Tenda	58:D9:D5:7F:25:91	-54	1	144.4 Mbps	PSK-CCMP		PSK-CCMP		Tenda Technology Co.Ltd.Dongg
Orange-hD4P-2.4G	28:41:C6:B5:7D:C8	-68	11	144.4 Mbps			PSK-CCMP		HUAWEI TECHNOLOGIES CO.LT
Orange-hD4P-5G	28:41:C6:B5:7D:CC	-79	36+40+44+48	1300.05 Mbps			PSK-CCMP	1.0	HUAWEI TECHNOLOGIES CO.LT

- Parametri disponibili
- SSID pentru fiecare rețea
 - Puterea semnalului in dB
 - Numărul canalului și frecvența centrală
 - Setările de securitate
 - Versiunea 802.11
 - Diverse grafice

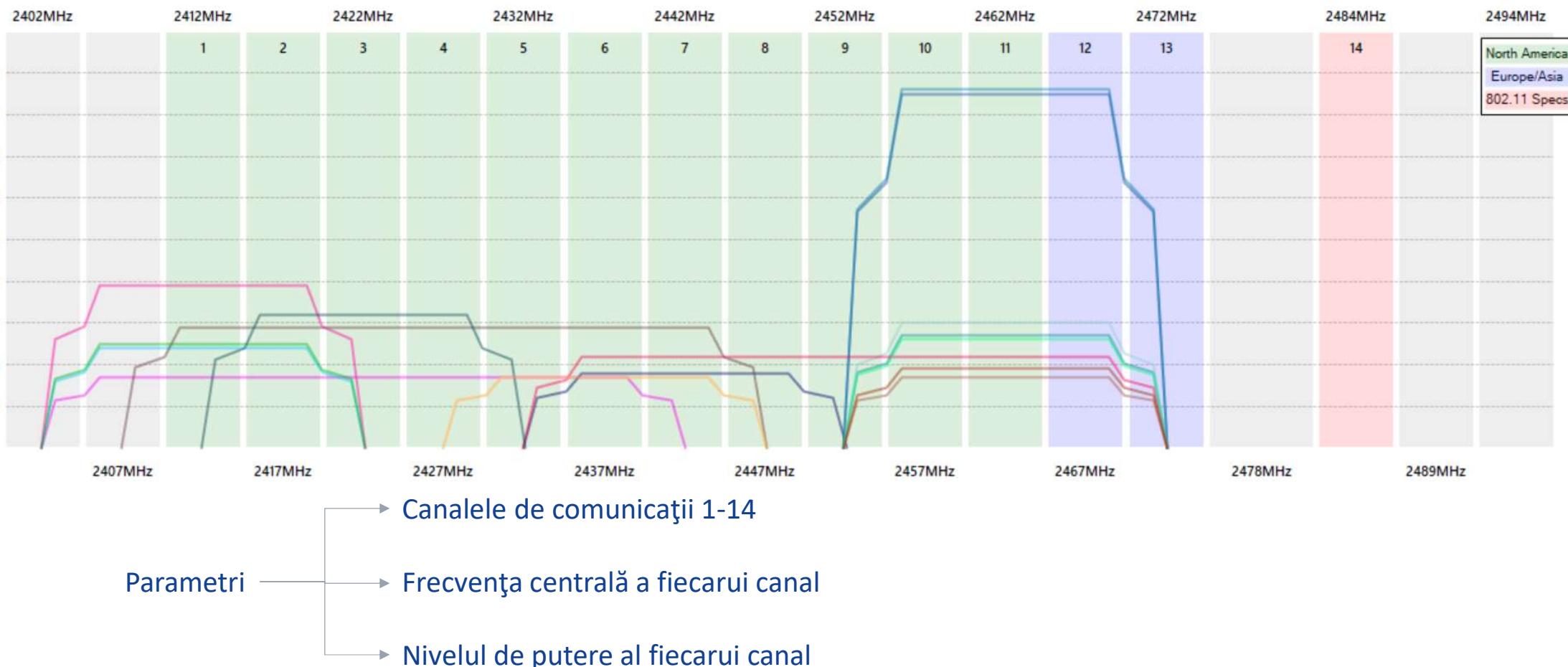
Pentru Android & IOS
verificați store-ul

Windows:
Acrylic WiFi home,
PRTG network monitor

NOKIA

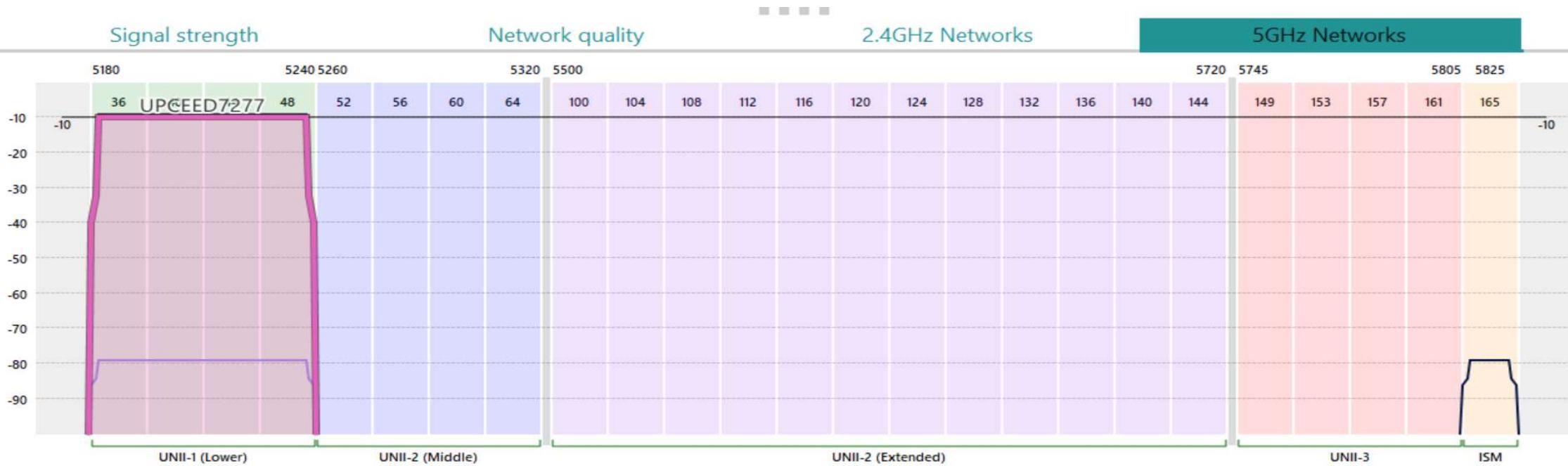
Analizoare WiFi

Acrylic WiFi Home

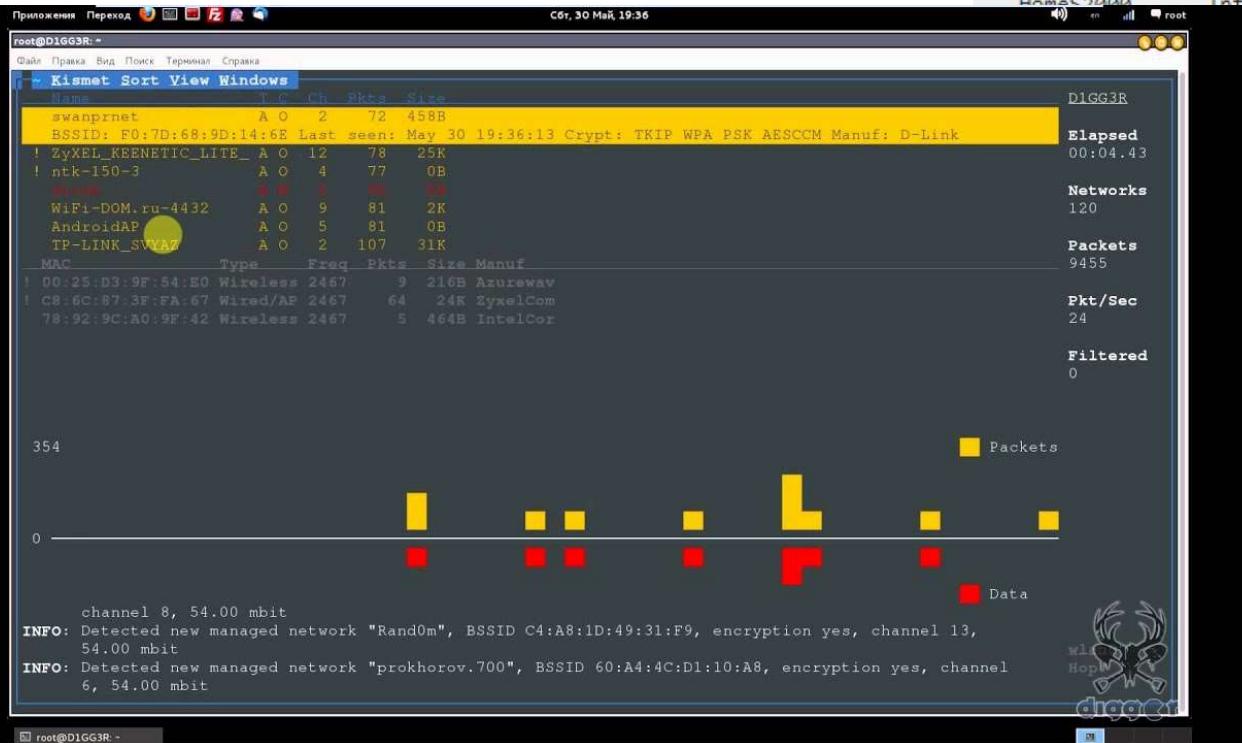


Analizoare WiFi

Acrylic WiFi Home



For Linux & IOS



Network manager command line interface nmcli

```
nmcli d wifi
```

This is an output of command:

*	SSID	MODE	CHAN	RATE	SIGNAL	BARS	SECURITY
	151022	Infra	4	54 Mbit/s	74	████	WPA2
	mary	Infra	4	54 Mbit/s	74	████	WPA2
	151022	Infra	40	54 Mbit/s	70	████	WPA2
	mary5	Infra	40	54 Mbit/s	60	████	WPA2
	Don Ceci	Infra	6	54 Mbit/s	34	███	WPA1 WPA2
	epg72	Infra	11	54 Mbit/s	34	███	WPA1 WPA2
	Mitio Paynera	Infra	2	54 Mbit/s	24	███	WPA1 WPA2
	Nina	Infra	1	54 Mbit/s	17	███	WPA1 WPA2
	Filka	Infra	10	54 Mbit/s	17	███	WPA1
	Mihaylov	Infra	11	54 Mbit/s	14	███	WPA1 WPA2
	HomeS2000	Infra	6	54 Mbit/s	7	███	WPA1 WPA2
	a	Infra	7	54 Mbit/s	7	███	WPA1 WPA2
	a	Infra	11	54 Mbit/s	7	███	WPA1 WPA2
	a	Infra	11	54 Mbit/s	7	███	WPA1 WPA2
	a	Infra	10	54 Mbit/s	4	██	WPA2

Linux :
Terminal: nmcli d wifi
Kismet,
PRTG network monitor

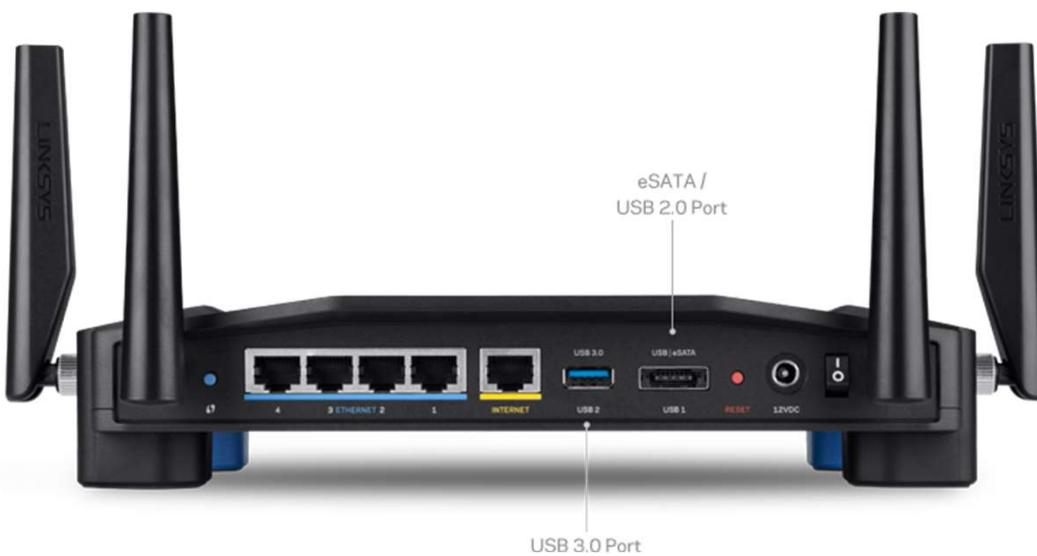
NOKIA

Rutere WiFi

Funcții

Funcții:

- Punct de acces;
- Switch pentru nivelul fizic
- Filtrare de pachete
- Access parental



SSID – numele fiecărei rețele

Benzi de frecvență:

- 2.4 GHz
- 5 GHz
- 60GHz

Securiate:

- None – mereu o idee proastă;
- AES
- WPA2

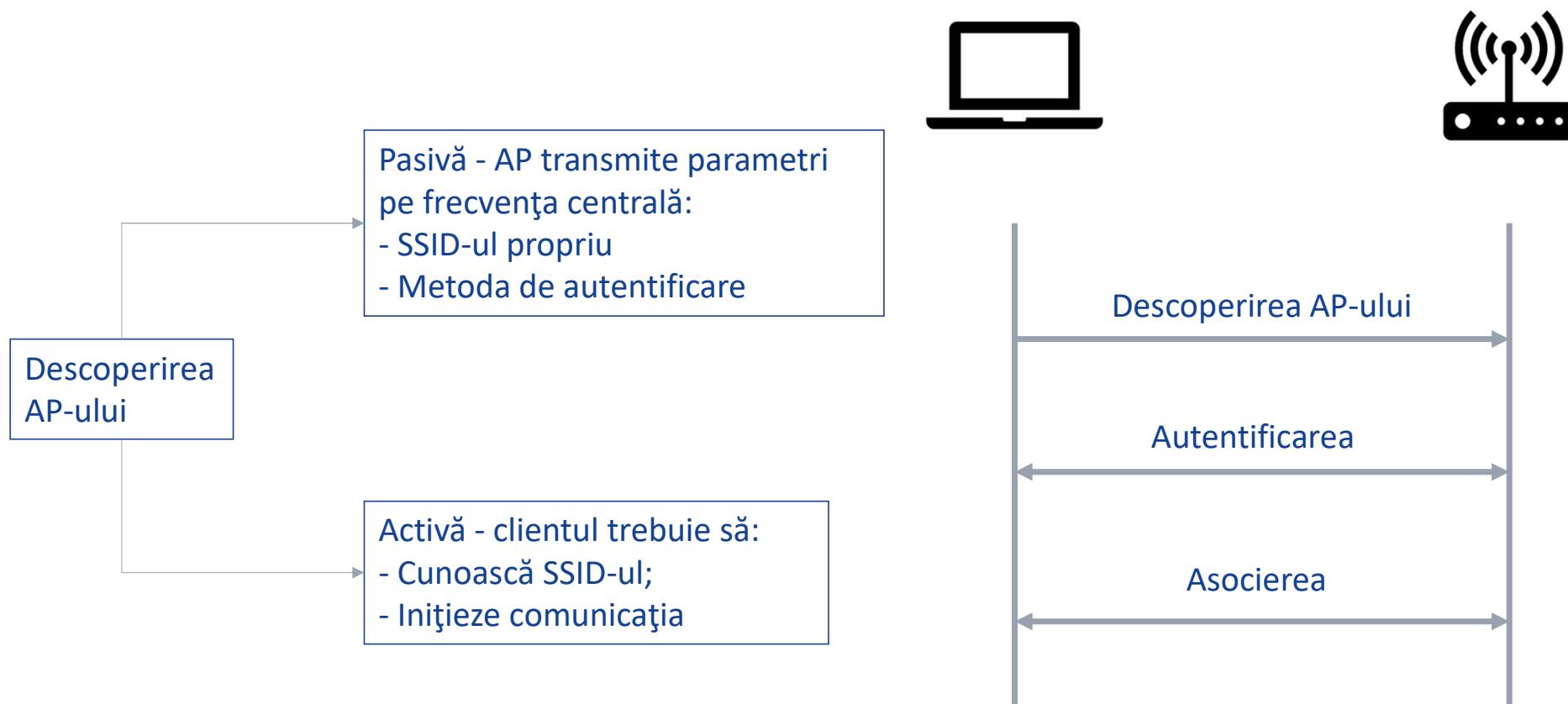
În laborator ne jucăm cu ruterele:

- Mercursys MW301R;
- Tenda F3

Manualele disponibile pe CV.

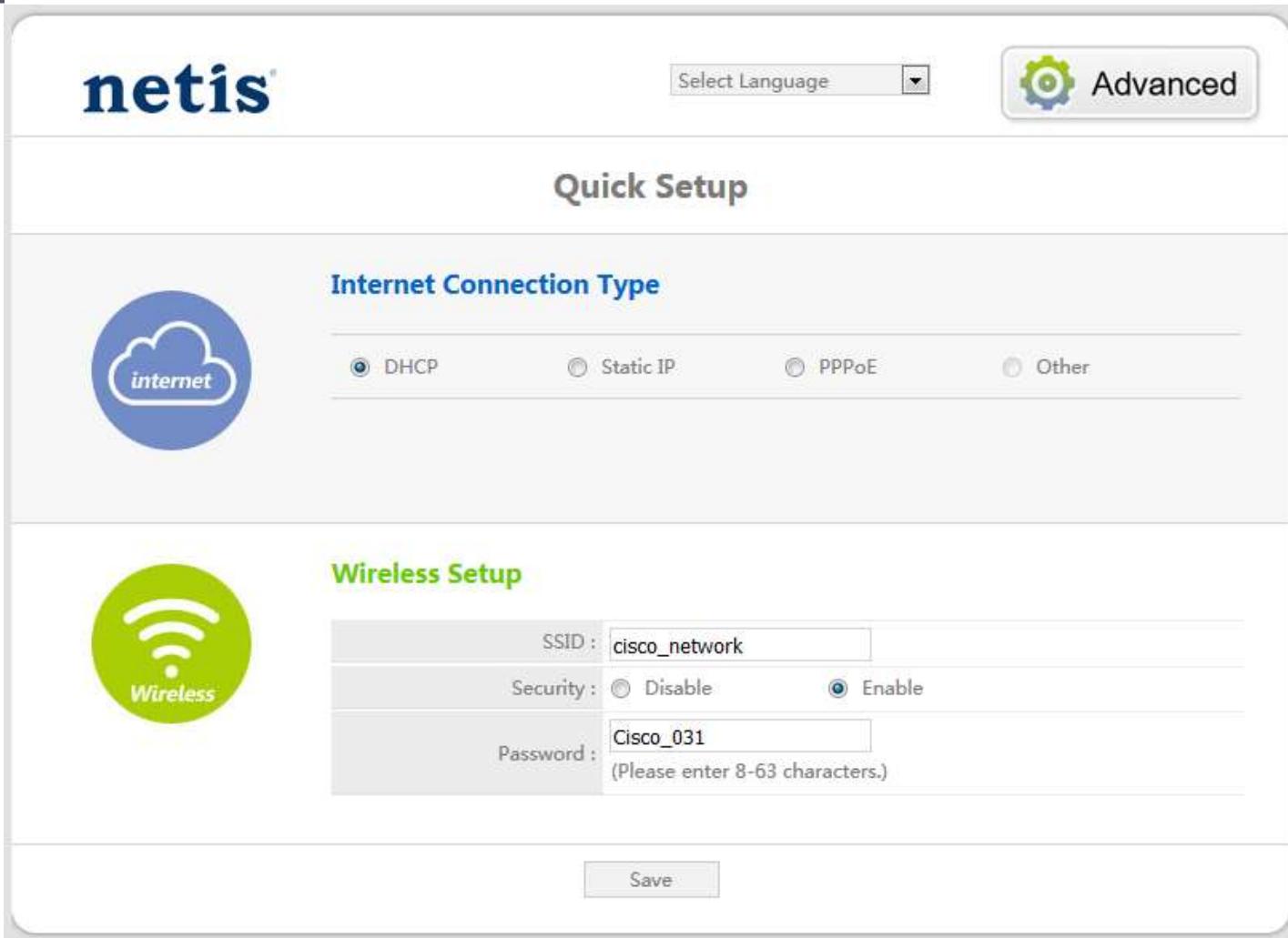
Rutere WiFi

Atașarea la un ruter



Rutere WiFi

Quick setup



The screenshot shows the Netis Quick Setup interface. At the top, there is a "Select Language" dropdown and an "Advanced" button with a gear icon. The main title is "Quick Setup".
Internet Connection Type: This section contains four radio buttons: "DHCP" (selected), "Static IP", "PPPoE", and "Other".
Wireless Setup: This section contains fields for SSID, Security, and Password.

- SSID: cisco_network
- Security: Disable (radio button)
- Enable (radio button) (selected)
- Password: Cisco_031
(Please enter 8-63 characters.)

At the bottom right of the interface is a "Save" button.

Rutere WiFi WAN Config

As detected, your connection type is: **PPPoE**

 Internet Settings

Connection Type PPPoE Dynamic IP Static IP
Select PPPoE if your Internet connection asks for the user name and password.

User Name

Password

 Wireless Settings

WiFi Name

WiFi Password

OK

WAN – ceea ce vine de la operator:

- PPPoE – necesită user și parolă
- Dynamic IP – MAC-urile sunt înregistrate la nivelul operatorului
- Static IP – necesită parametri Ip

Rutere WiFi

PPPoE - Nokia

PPPoE – necesită user și parolă

NOKIA
Ethernet Gateway
Logout
English | Español

Network>WAN

- [Status](#)
- [Network](#)
- [LAN](#)
- [LAN_IPv6](#)
- [WAN](#)
- [WAN DHCP](#)
- [Wireless \(2.4GHz\)](#)
- [Wireless \(5GHz\)](#)
- [Wireless Schedule](#)
- [IP Routing](#)
- [DNS](#)
- [TR-069](#)
- [QoS Config](#)
- [Security](#)
- [Application](#)
- [Maintenance](#)
- [RG Troubleshooting](#)

WAN Connection List	<input type="text" value="1_TR069_INTERNET_OTHER_R_VID_881"/>
Connection Type	<input checked="" type="radio"/> IPoE <input checked="" type="radio"/> PPPoE
IP mode	<input type="text" value="IPv4"/>
Enable/Disable	<input checked="" type="checkbox"/>
NAT	<input checked="" type="checkbox"/>
Service	<input checked="" type="checkbox"/> TR-069 <input checked="" type="checkbox"/> INTERNET <input checked="" type="checkbox"/> IPTV
Enable VLAN	<input checked="" type="checkbox"/>
VLAN ID	<input type="text" value="881"/>
VLAN PRI	<input type="text" value="7"/>
WAN IP Mode	<input type="text" value="PPPoE"/>
Connection Trigger	<input type="text" value="AlwaysOn"/>
Username	<input type="text" value="user"/>
Password	<input type="password" value="*****"/>
Keep Alive Time	<input type="text" value="60"/> (5~60)seconds
Keep Alive Retry	<input type="text" value="3"/> (1~10)times
Echo Value	<input type="text" value="180"/>
Manual DNS	<input type="text"/>

Save
Delete

Rutere WiFi

WAN –IP static

As detected, your connection type is: **Static IP**

Internet Settings

Connection Type: PPPoE Dynamic IP Static IP

Select Static IP if your Internet connection asks for static IP info.

IP Address	IP
Subnet Mask	Subnet Mask
Default Gateway	Default Gateway
Preferred DNS	Preferred DNS Server
Alternative DNS	Alternative DNS Server

Wireless Settings

WiFi Name: Tenda_1E5FE0

WiFi Password

OK

Pentru Tenda & Mercursys

Trebuie să știm TOȚI parametrii Ip

WAN Settings

Connection Type	Static IP	Auto Detect
IP Address	0.0.0.0	
Subnet Mask	0.0.0.0	
Gateway	0.0.0.0	
Primary DNS	0.0.0.0	
Secondary DNS	0.0.0.0	Optional
MTU Size (in bytes)	1500	The default value is 1500. Do NOT change unless necessary.
WAN Rate Settings	Auto Negotiation	

Save

Rutere WiFi

Configurația Wireless

The screenshot shows a user interface for configuring a WiFi router. On the left sidebar, there are several menu items: Status, Internet Settings, Wireless Settings (highlighted in orange), Bandwidth Control, Wireless Repeating, Parental Controls, Advanced, and Administration.

WIFI ON/OFF: A toggle switch is set to "ON".

WiFi Name and Password:

- WiFi Name: Tenda_XXXXXX
- Hide WiFi:
- Security Mode: WPA/WPA2-PSK Mixed(Recommend)
- WiFi Password: *****

WiFi Schedule:

- WiFi Schedule: Enable Disable

WPS:

- WPS: Enable Disable

Parametri WiFi:

- SSID
- Securitatea
- Parola

Ce face funcția
“Hide WiFi”?

UPC9063546	38:43:7D:8C:57:31	-83	6	300 Mbps	PSK-(TKIP CCMP) PSK-(TKIP CCMP)	1.0	Compal Broadband Network
[Hidden]	62:45:B0:7A:32:CD	-79	165+165	N/A SharedKey			
HUAWEI-H64y	7C:A2:3E:97:63:2C	-83	1	144.4 Mbps	PSK-(TKIP CCMP) PSK-(TKIP CCMP)		HUAWEI TECHNOLOGIES

Rutere WiFi

Configurația Wireless

Parametri WiFi:

- Setarea benzii (2.4 sau 5GHz)
- Meniurile sunt similare pentru cele 2
- **Atenție: anumite rutere permit selectarea manuală a canalelor**
- Numărul maxim de utilizatori

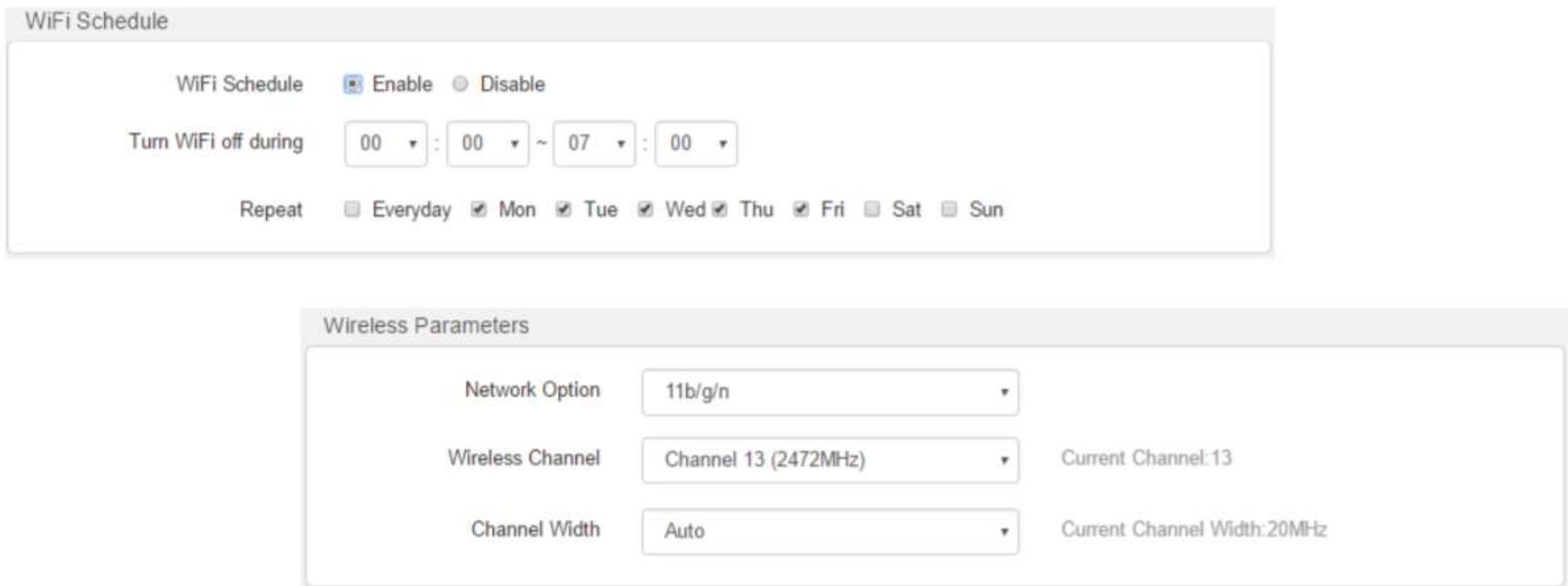
NOKIA
Ethernet Gateway
[Logout](#)
[English](#) | [Español](#)

Network>Wireless (2.4GHz)

<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input checked="" type="checkbox"/> Status </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <input checked="" type="checkbox"/> Network </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> LAN </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> LAN_IPv6 </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> WAN </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> WAN DHCP </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Wireless (2.4GHz) </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Wireless (5GHz) </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Wireless Schedule </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> IP Routing </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> DNS </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> TR-069 </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> QoS Config </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Security </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Application </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Maintenance </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> RG Troubleshooting </div>	<table border="0"> <tr> <td>Enable</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Mode</td> <td>auto(b/g/n)</td> </tr> <tr> <td>Bandwidth</td> <td>20MHz</td> </tr> <tr> <td>Channel</td> <td>Auto</td> </tr> <tr> <td>Transmitting Power</td> <td>100%</td> </tr> <tr> <td>WMM</td> <td>Enable</td> </tr> <tr> <td>Total MAX Users</td> <td>32</td> </tr> </table> <p>SSID Configuration</p> <table border="0"> <tr> <td>SSID Select</td> <td>SSID1</td> </tr> <tr> <td>SSID Name</td> <td>NOKIA-0580</td> </tr> <tr> <td>Enable SSID</td> <td>Enable</td> </tr> <tr> <td>SSID Broadcast</td> <td>Enable</td> </tr> <tr> <td>MAX Users</td> <td>32</td> </tr> <tr> <td>Encryption Mode</td> <td>WPA/WPA2 Personal</td> </tr> <tr> <td>WPA Version</td> <td>WPA2</td> </tr> <tr> <td>WPA Encryption Mode</td> <td>AES</td> </tr> <tr> <td>WPA Key</td> <td>*****</td> </tr> <tr> <td>Show password</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Enable WPS</td> <td>Enable</td> </tr> <tr> <td>WPS Mode</td> <td>PBC</td> </tr> </table> <p style="text-align: center;">WPS Connect</p> <p style="text-align: right; margin-top: 10px;"> Save Refresh </p>	Enable	<input checked="" type="checkbox"/>	Mode	auto(b/g/n)	Bandwidth	20MHz	Channel	Auto	Transmitting Power	100%	WMM	Enable	Total MAX Users	32	SSID Select	SSID1	SSID Name	NOKIA-0580	Enable SSID	Enable	SSID Broadcast	Enable	MAX Users	32	Encryption Mode	WPA/WPA2 Personal	WPA Version	WPA2	WPA Encryption Mode	AES	WPA Key	*****	Show password	<input type="checkbox"/>	Enable WPS	Enable	WPS Mode	PBC
Enable	<input checked="" type="checkbox"/>																																						
Mode	auto(b/g/n)																																						
Bandwidth	20MHz																																						
Channel	Auto																																						
Transmitting Power	100%																																						
WMM	Enable																																						
Total MAX Users	32																																						
SSID Select	SSID1																																						
SSID Name	NOKIA-0580																																						
Enable SSID	Enable																																						
SSID Broadcast	Enable																																						
MAX Users	32																																						
Encryption Mode	WPA/WPA2 Personal																																						
WPA Version	WPA2																																						
WPA Encryption Mode	AES																																						
WPA Key	*****																																						
Show password	<input type="checkbox"/>																																						
Enable WPS	Enable																																						
WPS Mode	PBC																																						

Rutere WiFi

Selectări Canalelor



The screenshot displays two configuration panels for a WiFi router.

WiFi Schedule

- WiFi Schedule: Enable Disable
- Turn WiFi off during: 00 : 00 ~ 07 : 00
- Repeat: Everyday Mon Tue Wed Thu Fri Sat Sun

Wireless Parameters

Network Option:	11b/g/n	Current Channel: 13
Wireless Channel:	Channel 13 (2472MHz)	Current Channel Width: 20MHz
Channel Width:	Auto	

Rutere WiFi

Controlul lățimii de bandă

The screenshot shows a web-based management interface for a WiFi router. On the left, a sidebar lists various settings: Status, Internet Settings, Wireless Settings, Bandwidth Control (which is selected and highlighted in orange), Wireless Repeating, Parental Controls, Advanced, and Administration. The main content area is titled "Attached Devices(2)". It lists two devices: "DESKTOP-M5MSVIJ" (IP: 192.168.1.100) and "Dudu" (IP: 192.168.1.101). For each device, there are fields for "Download Speed" (0KB/s), "Upload Speed" (0KB/s), "Download Limit" (set to "256KB/s"), and "Upload Limit" (set to "No Limit"). An "Internet Access" column indicates "Native Device" for the first and "Dudu" for the second. Below this section is "Blocked Devices(0)". A dropdown menu is open over the "Download Limit" field for the "Dudu" device, showing options: "No Limit", "128 KB/s(Web Browsing)", "256 KB/s(SD Videos)" (which is highlighted in orange), "512 KB/s(HD Videos)", and "Manual(unit: KB/s)".

Putem limita debitul pentru fiecare utilizator

Rutere WiFi

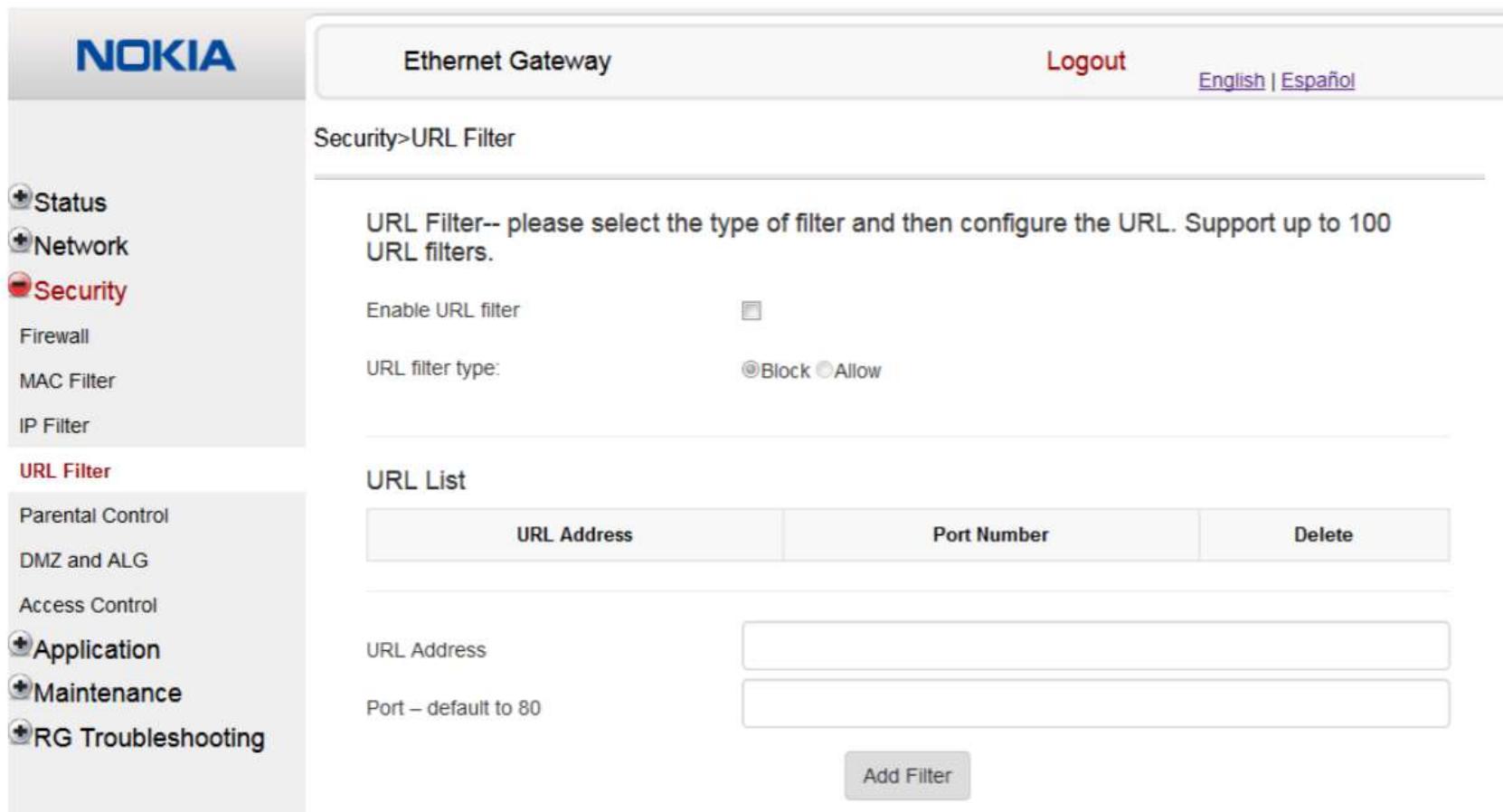
Control parental

The screenshot shows a web-based configuration interface for a WiFi router. On the left, a sidebar lists various settings: Status, Internet Settings, Wireless Settings, Bandwidth Control, Wireless Repeating, Parental Controls (which is currently selected), and Advanced. The main content area is titled "Attached Devices" and shows two devices: "Dudu" (IP 192.168.1.101) and "DESKTOP-M5MSVIJ" (IP 192.168.1.100). Below this is the "Access Restrictions" section. It includes a note that settings apply to all managed devices, a time range selector set from 20:00 to 22:00, and a weekly repeat selector where Monday, Tuesday, Wednesday, and Thursday are checked. Under "Website Restrictions", it says "Only Forbid". In the "Websites Specified" section, "youtube" is listed as a restricted site. At the bottom right are "OK" and "Cancel" buttons.

Ce se întamplă
aici?

Rutere WiFi

Control parental - Nokia



The screenshot shows the Nokia Ethernet Gateway interface. On the left, a sidebar lists various security and management options. The main content area is titled "Security > URL Filter". It contains instructions to select a filter type and URL. A "URL List" section is shown with columns for URL Address, Port Number, and Delete. Below this is a form for adding new filters with fields for URL Address and Port (default to 80), and a "Add Filter" button.

NOKIA

Ethernet Gateway

Logout English | Español

Status

Network

Security

Firewall

MAC Filter

IP Filter

URL Filter

Parental Control

DMZ and ALG

Access Control

Application

Maintenance

RG Troubleshooting

Security>URL Filter

URL Filter-- please select the type of filter and then configure the URL. Support up to 100 URL filters.

Enable URL filter

URL filter type: Block Allow

URL List

URL Address	Port Number	Delete
-------------	-------------	--------

URL Address

Port – default to 80

Add Filter

Aici ce se
întamplă?

Rutere WiFi

Control parental - Nokia

NOKIA

Ethernet Gateway Logout English | Español

Security>Parental Control

Block access of LAN devices at given times, according to their MAC or IPv4 addresses

Aici ce se întamplă?

Status
Network
Security
Firewall
MAC Filter
IP Filter
URL Filter

Parental Control
DMZ and ALG
Access Control
Application
Maintenance
RG Troubleshooting

Access Control □

Policy Name	Device	IP	Days Of Week	From	To	Delete	Edit	Enable
-------------	--------	----	--------------	------	----	--------	------	--------

+



That's all for today, see you next time!

Rețele de Calculatoare

Nivelul Transport

Sumar al laboratorului

1

Încapsularea datelor
Antetul Nivelului Transport

2

Nivelul Transport
Generalități

3

Numere de porturi

4

Protocolele
TCP
UDP

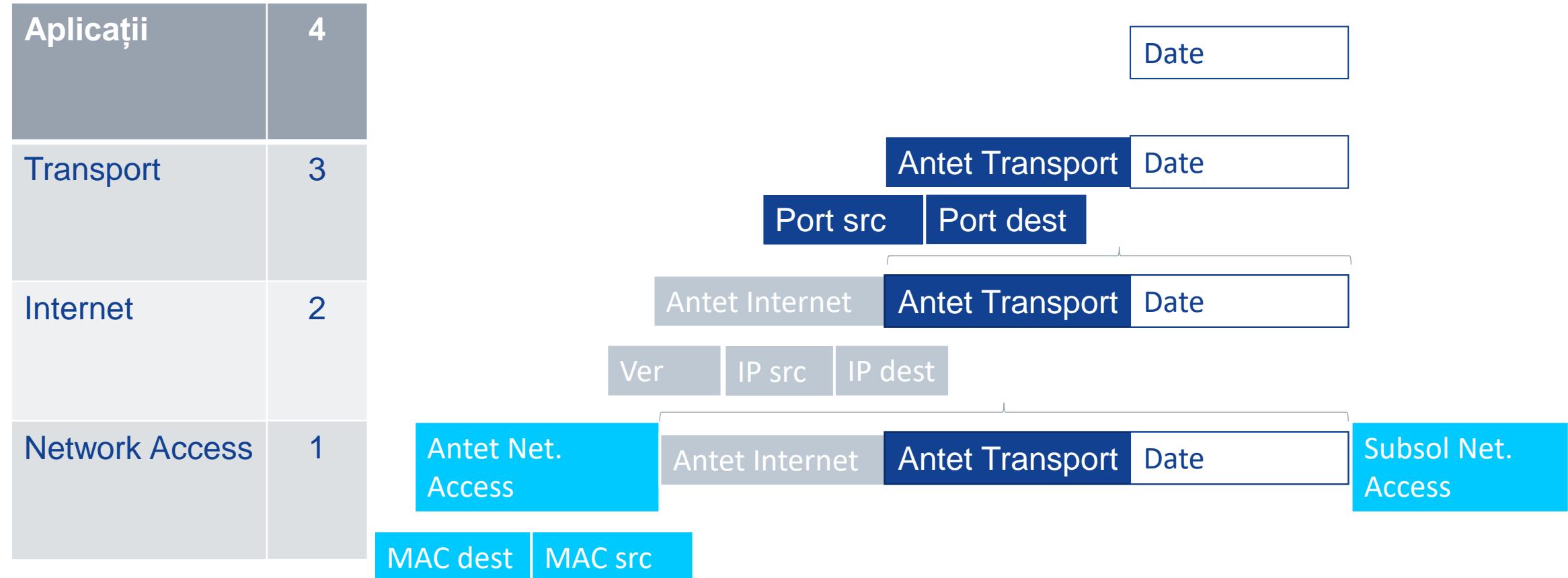
5

Sesiunile de comunicare
Procesul de windowing
Asamblare/Reasamblare
Unde folosim protocolele



Procesul de încapsulare

Recapitulare



Nivelul Transport

La ce este folositor?

Rolul nivelului Transport:

- Stabilirea comunicațiilor temporare între 2 aplicații
- Transportul datelor între ele
- Oferă servicii de multiplexare, fiabilitate, controlul fluxurilor

Multiplexarea conversațiilor:

- Împarte datele în segmente mai scurte;
- Etichetează segmentele în concordanță cu conversația

Protocolele nivelului transport

TCP

Transmission Control Protocol

UDP

User Datagram Protocol

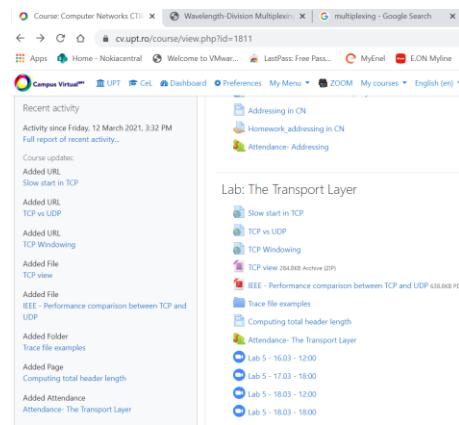
Nivelul Transport

Multiplexarea

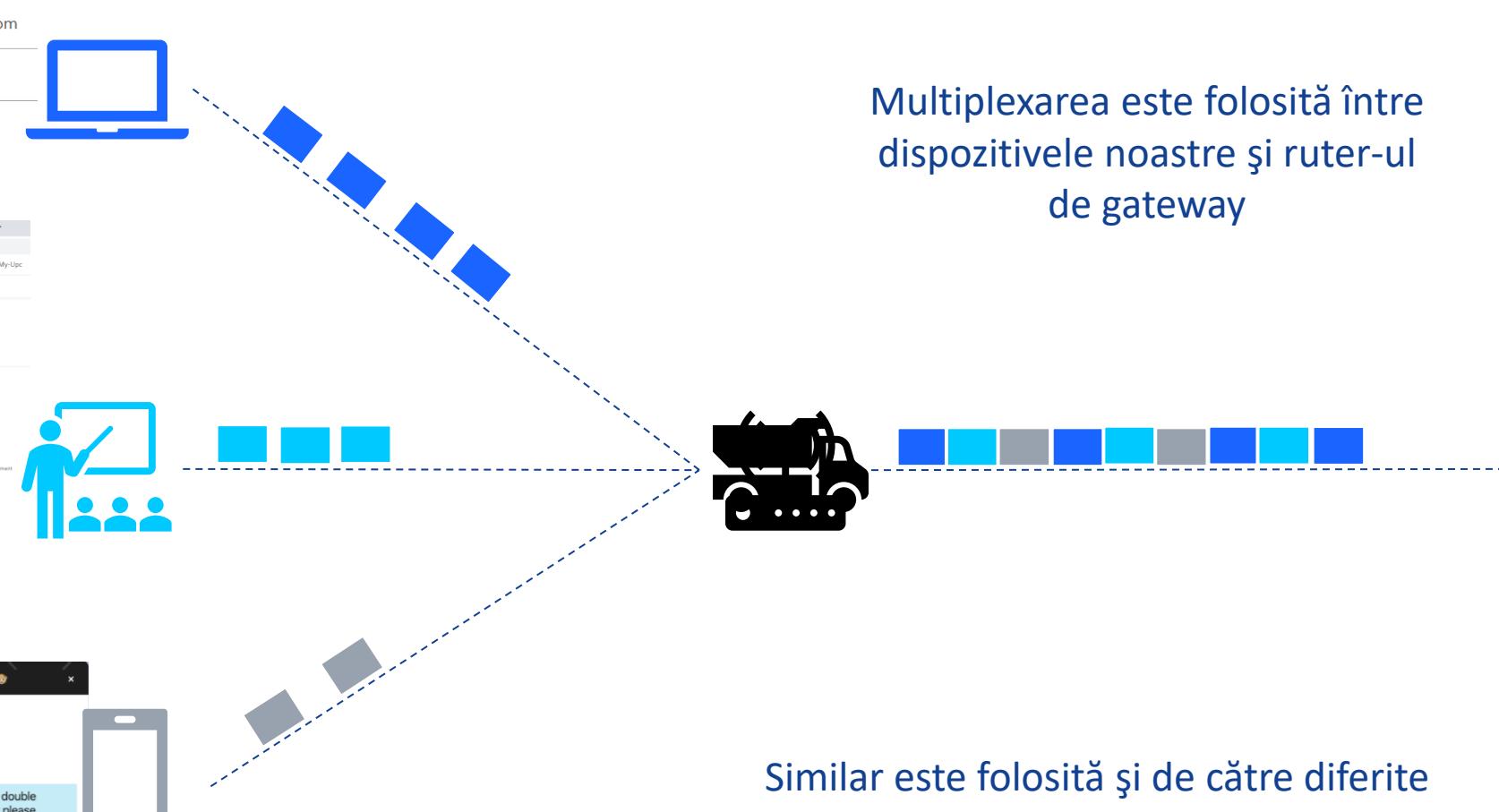
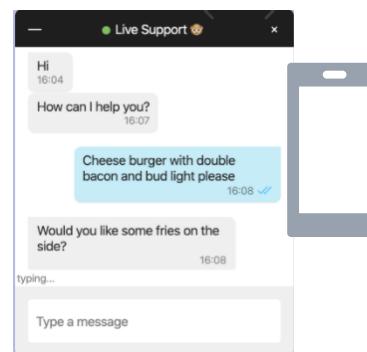
E-mail



HTML page



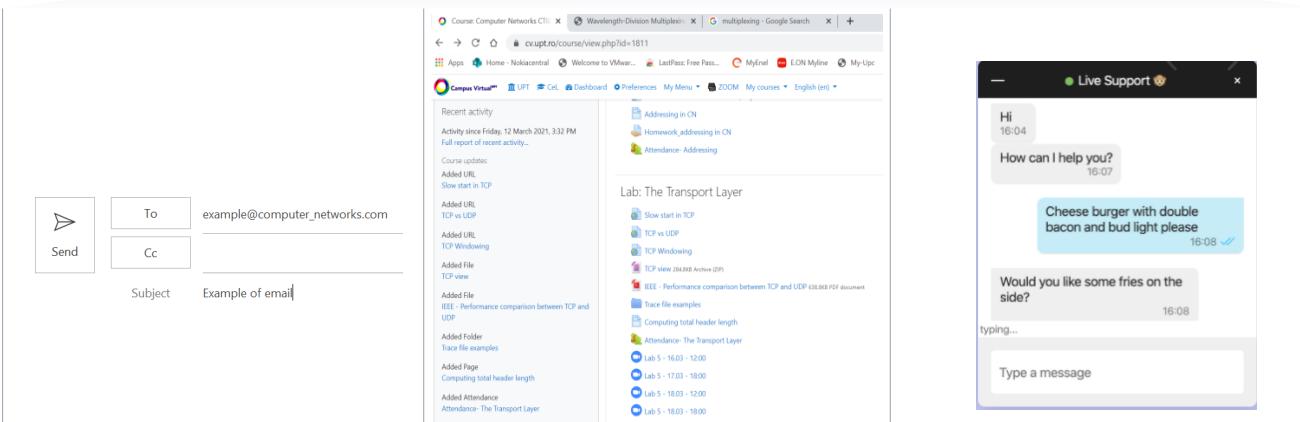
Web chat



Multiplexarea este folosită între dispozitivele noastre și ruter-ul de gateway

Similar este folosită și de către diferite aplicații și nivelul Transport

Numere de porturi



Diverse aplicații	Electronic mail		HTML pages		Internet Chat	
Protocolul	POP3		HTTP		IM	
Transport	App port	Data	App port	Data	App port	Data
Numere de porturi	110		80		531	

Grupuri de numere de porturi:

- Porturi bine cunoscute: 0-1023
- Porturi rezervate: 1024-49151
- Porturi dinamice: 49152-65535

În programare se folosește mult mai mult conceptul de **SOCKET**

ATENȚIE:

un socket = adresa IP+număr de port

Numere de porturi

Cum le vedem

Identifier unic
pentru fiecare
aplicație și sesiune
de comunicare

Portul destinație:
folosit pentru
identificarea
aplicației la
recepție

Portul sursă:
ales aleator de
către inițiator

Văzute ca și
pereche sursă-
destinație

Local Port	Remote Address	Remote Port
14766	localhost	52780
14766	localhost	52793
52563	131.228.2.175	https
52594	52.85.10.88	https
52598	172.217.20.226	https
52600	172.217.22.225	https
52601	172.217.23.1	https
52641	34.98.67.61	https

Protocolul UDP

Antetul protocolului

Adaos scăzut de biți la date

Lungimea mesajului este formată din:

- Antet
- Date

Nu oferă fiabilitate prin structura sa

Octet 1	Octet 2	Octet 3	Octet 4
Port Sursă			Port Destinație
Lungimea mesajului			Sumă de control
	Date		

User Datagram Protocol, Src Port: 50209, Dst Port: 53

→ Source Port: 50209

Destination Port: 53

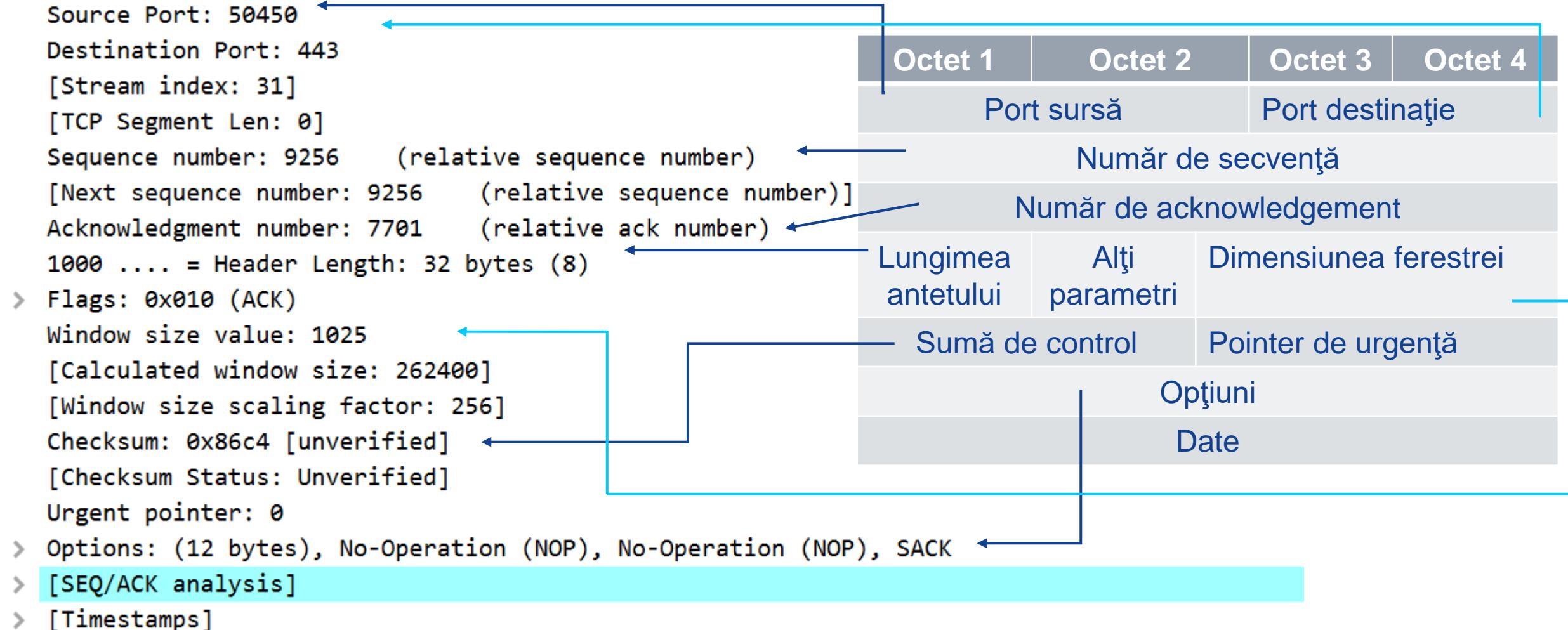
Length: 52

Checksum: 0x16fd [unverified]

Protocolul TCP

Antetul protocolului

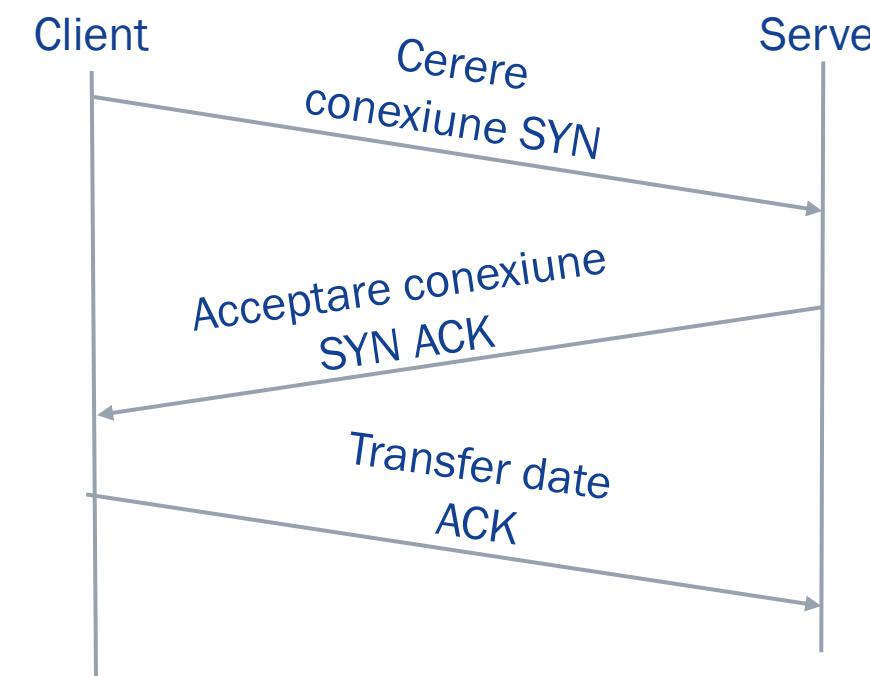
Transmission Control Protocol, Src Port: 50450, Dst Port: 443, Seq: 9256, Ack: 7701, Len: 0



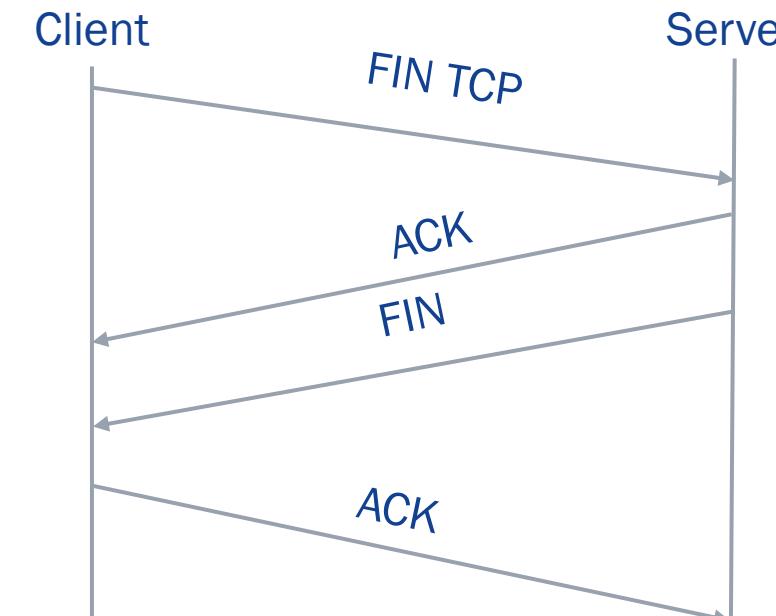
Sesiunile de comunicare

Cum le deschidem și cum le închidem

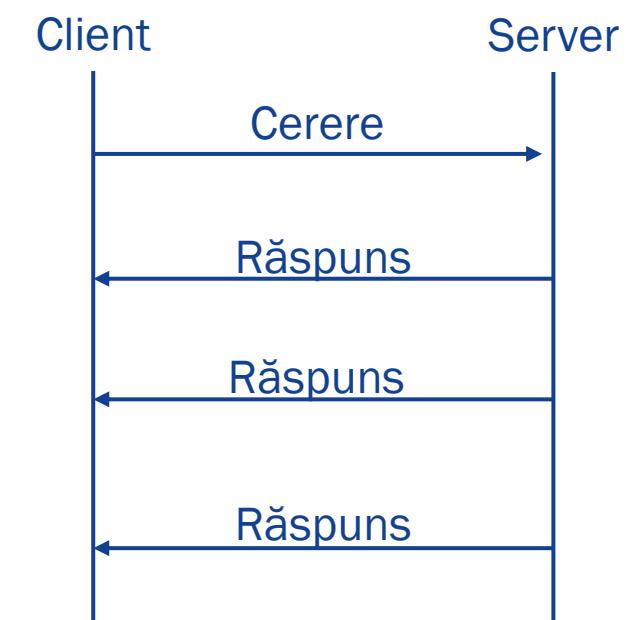
Stabilirea conexiunii



Încheierea conexiunii



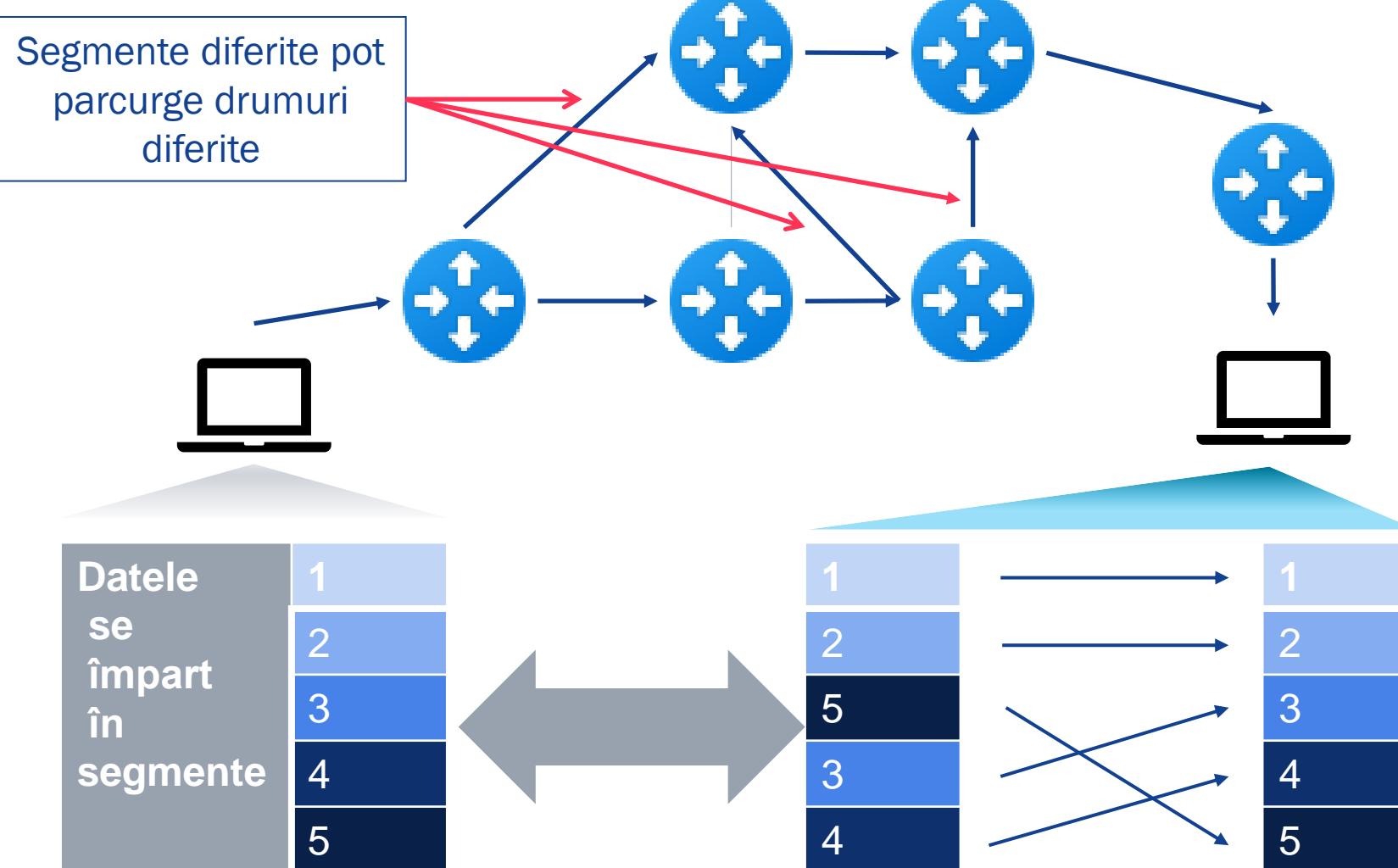
TCP



UDP

Sesiunile de comunicare

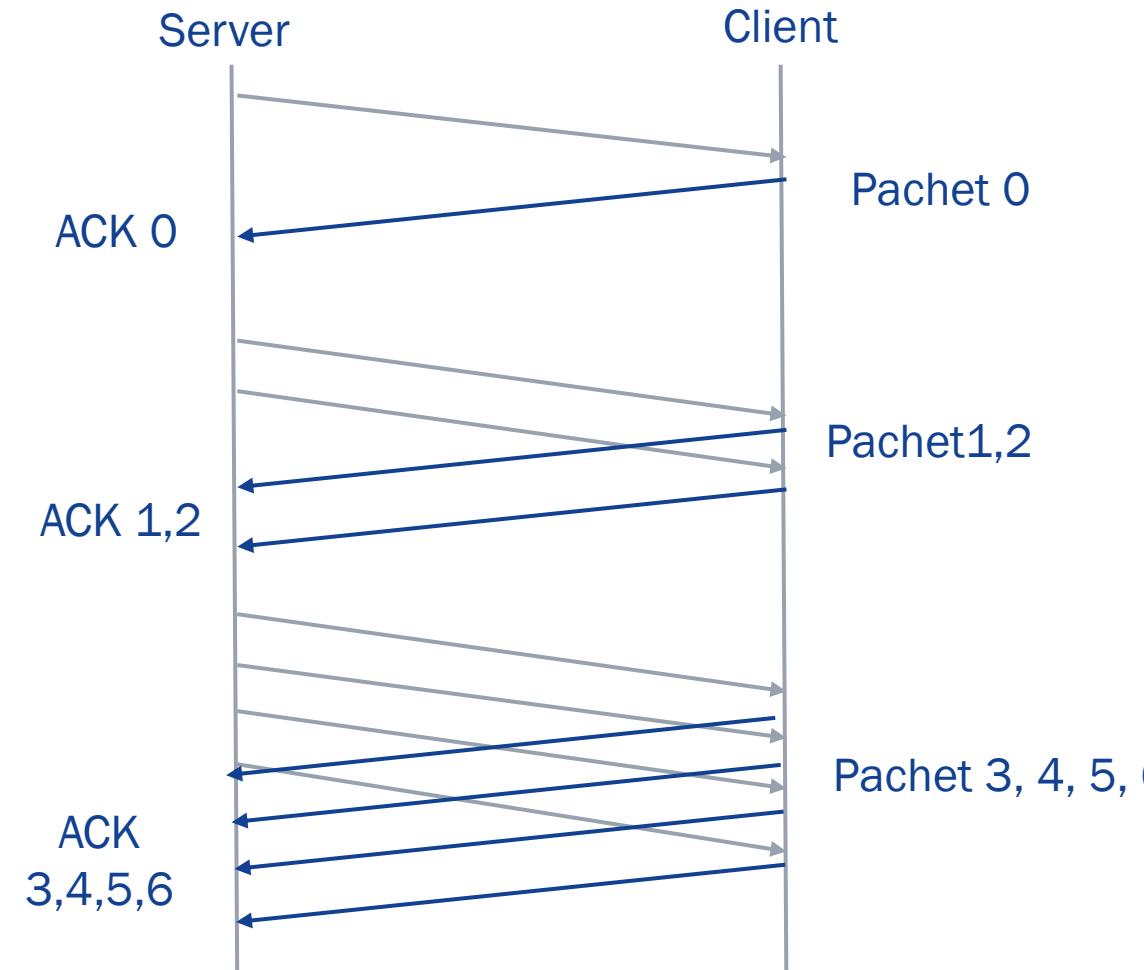
Asamblarea și reasamblarea datelor



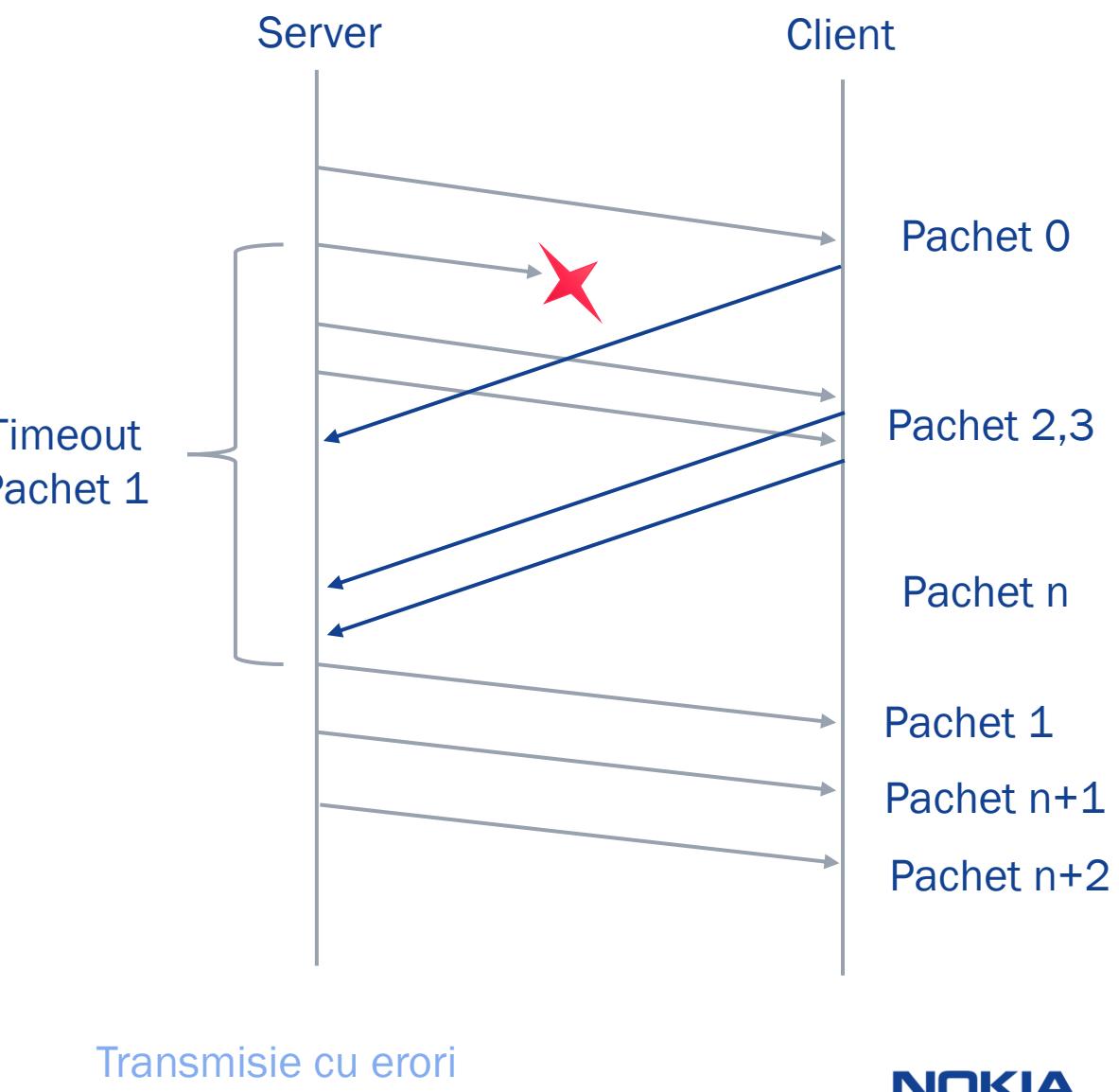
1. La începerea transmisiunii primul segment de date primește un număr de secvență (ISN)
2. Acest ISN se incrementează cu numărul de octeți transmiși
3. TCP-ul pune într-un buffer datele până ce toate segmentele au fost recepționate
4. La nivelul Aplicații se transmit datele complet recepționate și asamblate în mod corect

Sesiunile de comunicare

Procesul de windowing

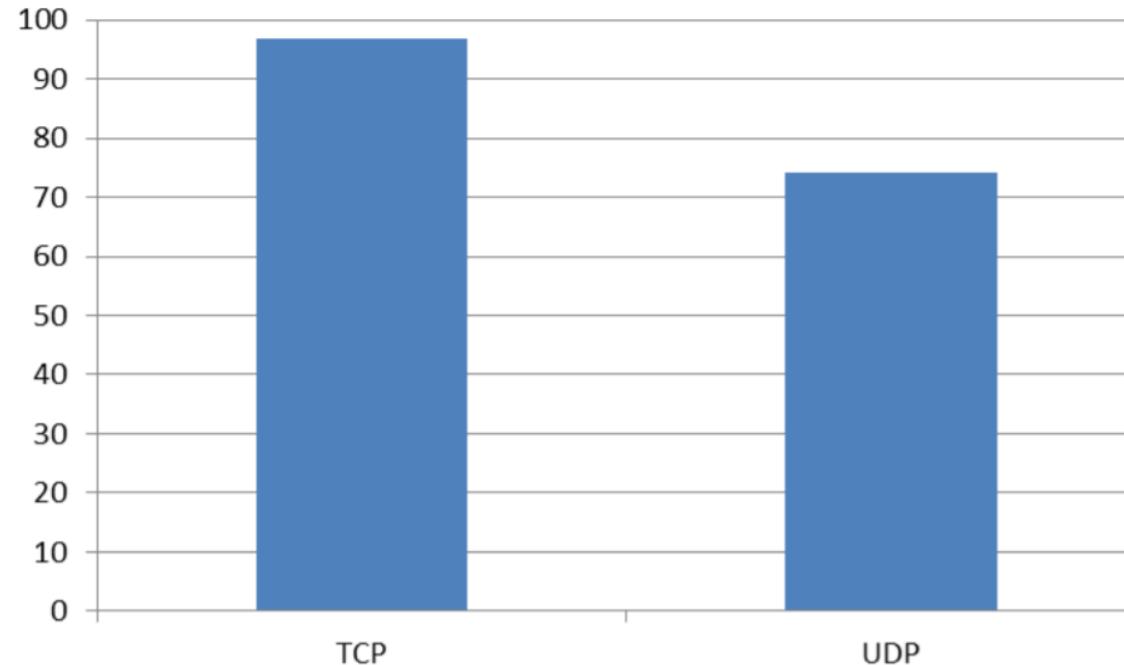


Transmisie fără erori



Cum alegem între ele?

TCP sau UDP



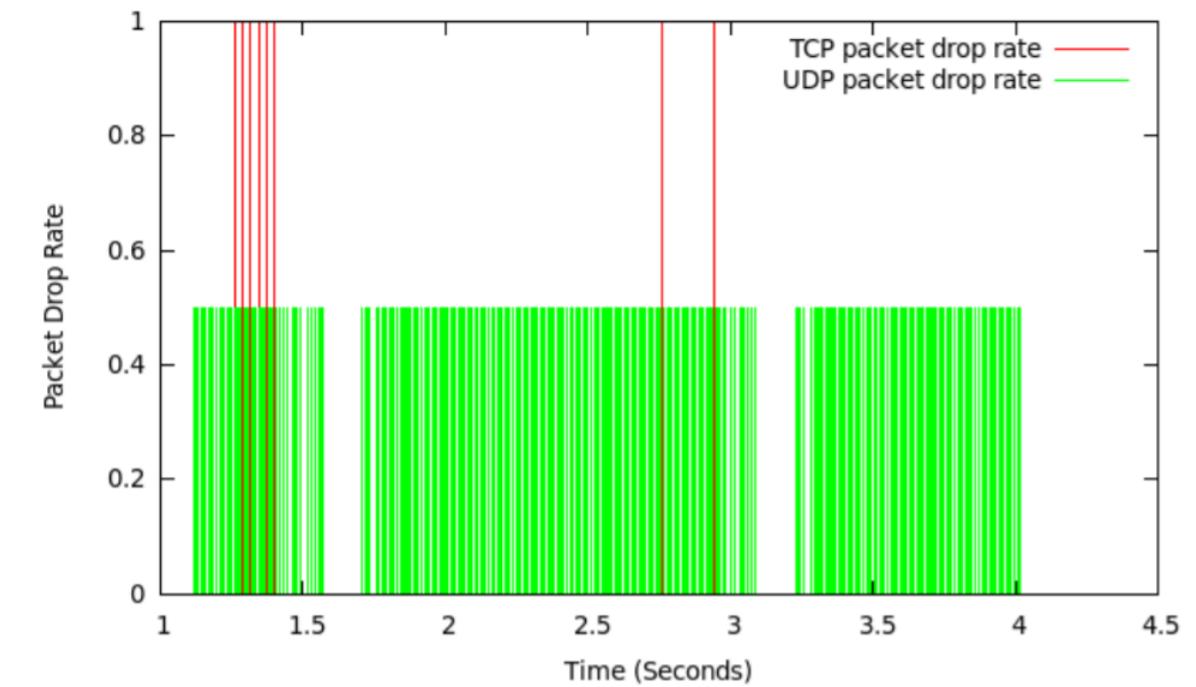
Conform articolului:

[Performance Comparison of Transport Layer Protocols](#)

By: Ali Hussein Wheeb University of Baghdad,
Iraq

Rata de livrare a pachetelor

Rata de drop a pachetelor



Protocolul UDP

Unde îl folosim?

QUERY-RESPONSE PROTOCOLS

**DNS****NTP**

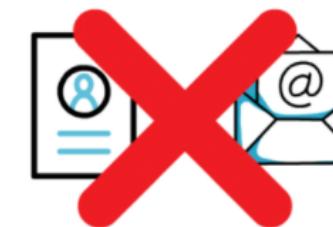
BROADCASTING SERVICES

**MULTICAST**

SERVICE DISCOVERY SYSTEM

ROUTING PROTOCOL

TIME-SENSITIVE APPLICATIONS

**IPTV****VOIP****ONLINE GAMES****Sursa:**<https://finematics.com/udp-vs-tcp/>**Simplu****Caracteristici:****Rapid**

Nu este orientat pe conexiune – nu oferă servicii de retransmisie complicate și resecvențiere

Fiabilitatea se asigură la nivel de aplicație

Protocolul TCP

Unde îl folosim?

ONE OF THE MOST COMMONLY USED PROTOCOLS



ON THE INTERNET

HTTP



FTP



SSH



SMTP



Sursa:

<https://finematics.com/udp-vs-tcp/>

COMPLEX

CONNECTION-ORIENTED

PROTOCOL

Caracteristici:



→ Stabilește o sesiune de comunicare

→ Asigură livrarea datelor

→ Control al livrării

→ Adaos crescut de biți la date

Cum vedem porturile?

Windows

```
C:\Users\cmisici>netstat
```

```
Active Connections
```

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:7700	N-20L6PF1K8TWG:54606	ESTABLISHED
TCP	127.0.0.1:14766	N-20L6PF1K8TWG:49451	ESTABLISHED
TCP	127.0.0.1:14766	N-20L6PF1K8TWG:52409	TIME_WAIT
TCP	127.0.0.1:14766	N-20L6PF1K8TWG:52412	TIME_WAIT
TCP	127.0.0.1:14766	N-20L6PF1K8TWG:52413	TIME_WAIT
TCP	127.0.0.1:14766	N-20L6PF1K8TWG:52421	FIN_WAIT_2
TCP	127.0.0.1:49451	N-20L6PF1K8TWG:14766	ESTABLISHED
TCP	127.0.0.1:52421	N-20L6PF1K8TWG:14766	CLOSE_WAIT
TCP	127.0.0.1:54606	N-20L6PF1K8TWG:7700	ESTABLISHED
TCP	127.0.0.1:54921	N-20L6PF1K8TWG:54922	ESTABLISHED
TCP	127.0.0.1:54922	N-20L6PF1K8TWG:54921	ESTABLISHED
TCP	127.0.0.1:54923	N-20L6PF1K8TWG:54924	ESTABLISHED

Utilizând comanda netstat din cmd

Functie: afisează conexiunile active prin:

1. Protocolul ce le folosește
2. Socket-ul folosit (aici vedem și portul)
3. Statusul conexiunii

Dacă dorim să vedem și PID-ul procesului adaugăm opțiunea -aon

Sursă și inspirație: <https://www.howtogeek.com/howto/28609/how-can-i-tell-what-is-listening-on-a-tcpip-port-in-windows/>

Cum vedem porturile?

Windows

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State
chrome.exe	11924	TCP	n-20l6pf1k8twg.nsn-intra.net	50381	wl-in-f188.1e100.net	5228	ESTABLISHED
chrome.exe	11924	TCP	n-20l6pf1k8twg.nsn-intra.net	52426	199.232.18.49	https	ESTABLISHED
chrome.exe	11924	TCP	n-20l6pf1k8twg.nsn-intra.net	52429	ec2-52-203-83-16...	https	ESTABLISHED
chrome.exe	11924	TCP	n-20l6pf1k8twg.nsn-intra.net	52431	ec2-52-203-83-16...	https	ESTABLISHED

Utilizând utilitarul TCPview – atașat în CV

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State	Sent Packets	Sent Bytes
Teams.exe	24632	TCP	n-20l6pf1k8twg.nsn-intra.net	50383	52.114.104.98	https	ESTABLISHED	5	893
Teams.exe	24500	TCP	n-20l6pf1k8twg.nsn-intra.net	65426	52.113.205.20	https	ESTABLISHED	8	1,478
Teams.exe	24500	UDP	N-20L6PF1K8TwG	53414	*	*			
Teams.exe	24500	UDP	N-20L6PF1K8TwG	60227	*	*			

Cum vedem porturile?

Linux

```
admin@tecmint ~ $ sudo netstat -lntup
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*            LISTEN     1423/nginx -g daemon
tcp        0      0 127.0.1.1:53             0.0.0.0:*            LISTEN     2992/dnsmasq
tcp        0      0 0.0.0.0:22              0.0.0.0:*            LISTEN     1409/sshd
tcp        0      0 127.0.0.1:631             0.0.0.0:*            LISTEN     2738/cupsd
tcp        0      0 0.0.0.0:443             0.0.0.0:*            LISTEN     1423/nginx -g daemon
tcp6       0      0 :::80                 :::*                  LISTEN     1423/nginx -g daemon
tcp6       0      0 :::22                 :::*                  LISTEN     1409/sshd
tcp6       0      0 ::1:631                :::*                  LISTEN     2738/cupsd
tcp6       0      0 :::443                :::*                  LISTEN     1423/nginx -g daemon
udp        0      0 0.0.0.0:631             0.0.0.0:*            LISTEN     2740/cups-browsed
udp        0      0 0.0.0.0:5353            0.0.0.0:*            LISTEN     1022/avahi-daemon:
udp        0      0 0.0.0.0:36390            0.0.0.0:*            LISTEN     2992/dnsmasq
udp        0      0 0.0.0.0:59072            0.0.0.0:*            LISTEN     1022/avahi-daemon:
udp        0      0 127.0.1.1:53              0.0.0.0:*            LISTEN     2992/dnsmasq
udp        0      0 0.0.0.0:68               0.0.0.0:*            LISTEN     2982/dhcclient
udp        0      0 192.168.43.31:123            0.0.0.0:*            LISTEN     1465/ntp
udp        0      0 127.0.0.1:123             0.0.0.0:*            LISTEN     1465/ntp
udp        0      0 0.0.0.0:123              0.0.0.0:*            LISTEN     1465/ntp
udp6       0      0 :::43740               :::*                  LISTEN     1022/avahi-daemon:
udp6       0      0 :::5353                :::*                  LISTEN     1022/avahi-daemon:
udp6       0      0 fe80::dd8c:3d40:817:123  ::*:*
udp6       0      0 ::1:123                :::*                  LISTEN     1465/ntp
udp6       0      0 ::::123                :::*                  LISTEN     1465/ntp
admin@tecmint ~ $
```

Utilizând comanda netstat din terminal

Funcție: afișează conexiunile active prin:

1. Protocolul ce le folosește
2. Socket-ul folosit (aici vedem și portul)
3. Statusul conexiunii

Sursă și inspirație: <https://www.tecmint.com/find-listening-ports-linux/>

Lab Work



That's all for today, see you next time!

Nu uitați de tema de casă
termen de predare până în data de 28.11.2020

Rețele de Calculatoare

Aplicații uzuale: DHCP, FTP, HTTP

Sumar al laboratorului

1

DHCP

Caracteristici generale
Mesajele transmise
Configurare DHCP

2

FTP

Ce transferăm
Cum transferăm

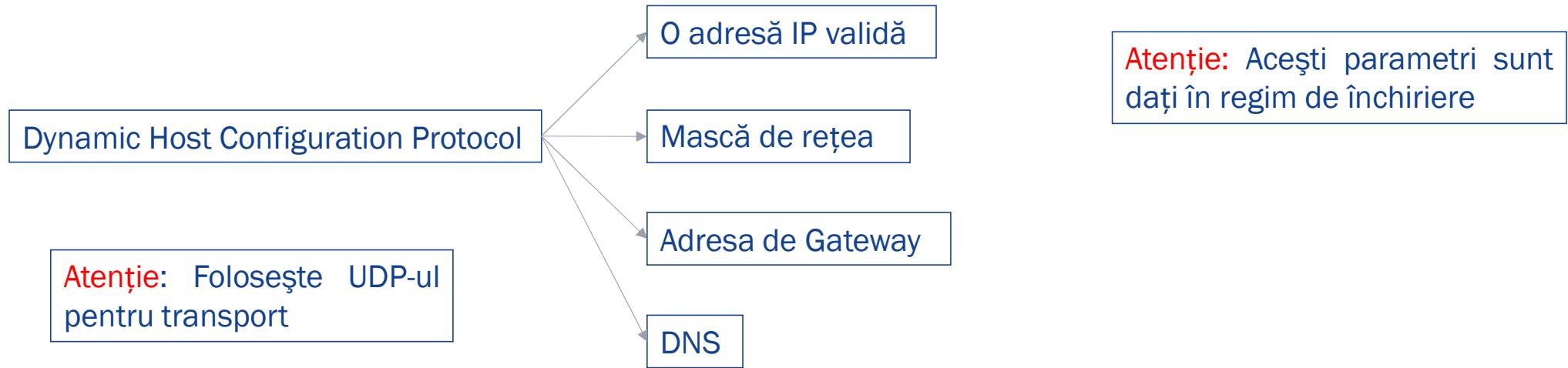
3

HTTP

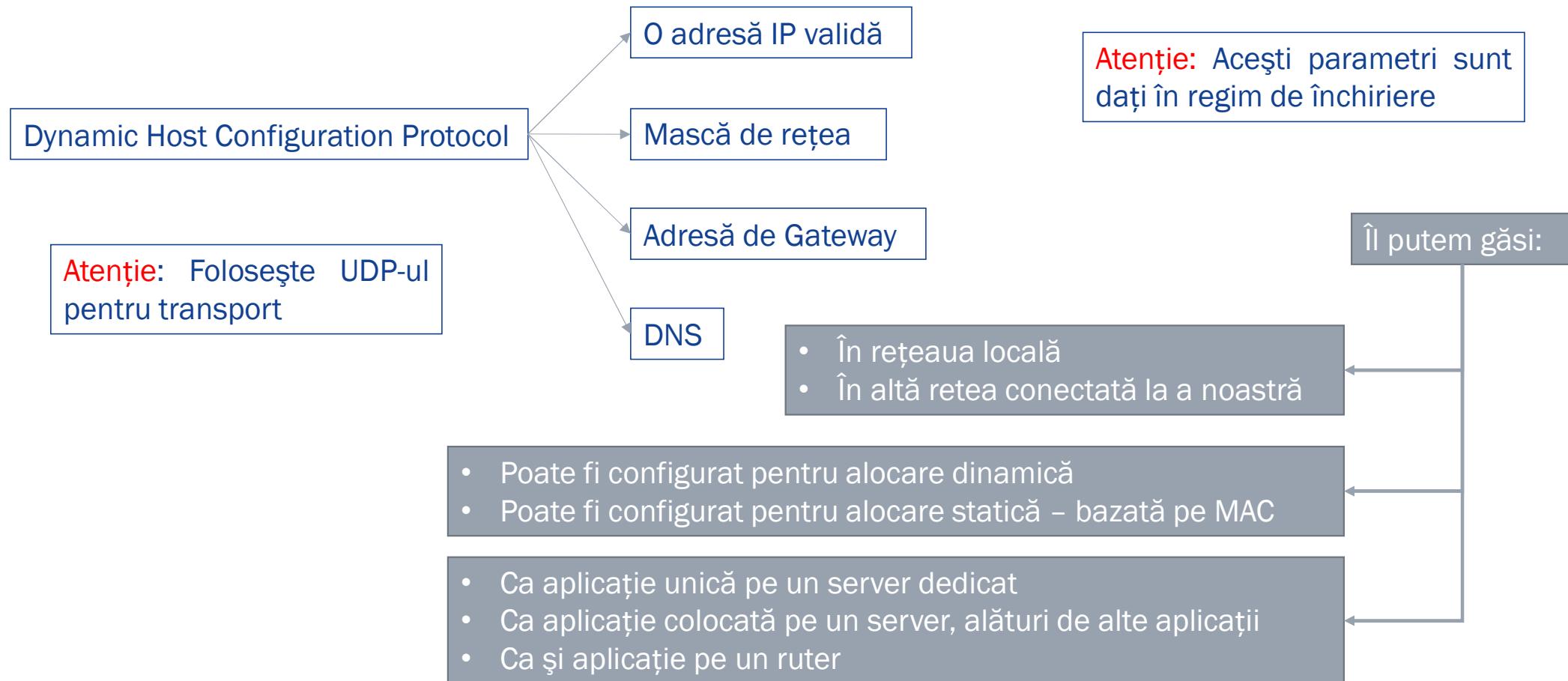
Cadrele HTTP
Cum se încarcă o pagină web



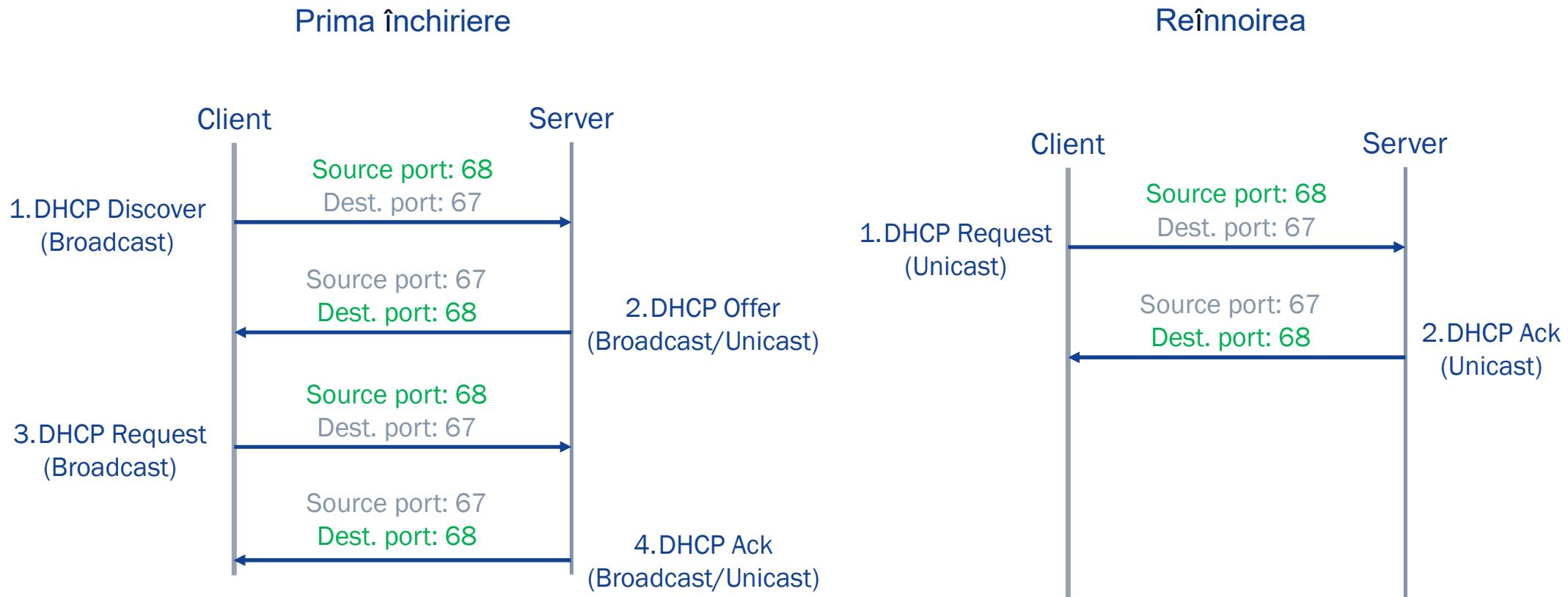
Caracteristici generale



Caracteristici generale



Mesajele transmise



Antetul protocolului

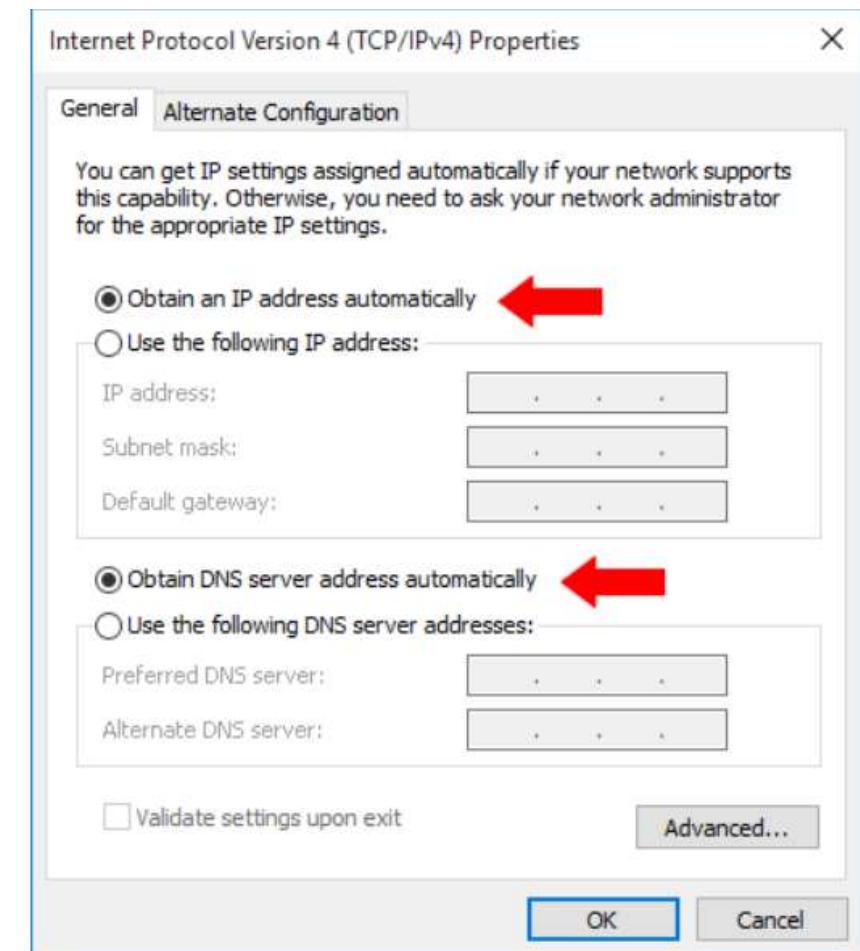
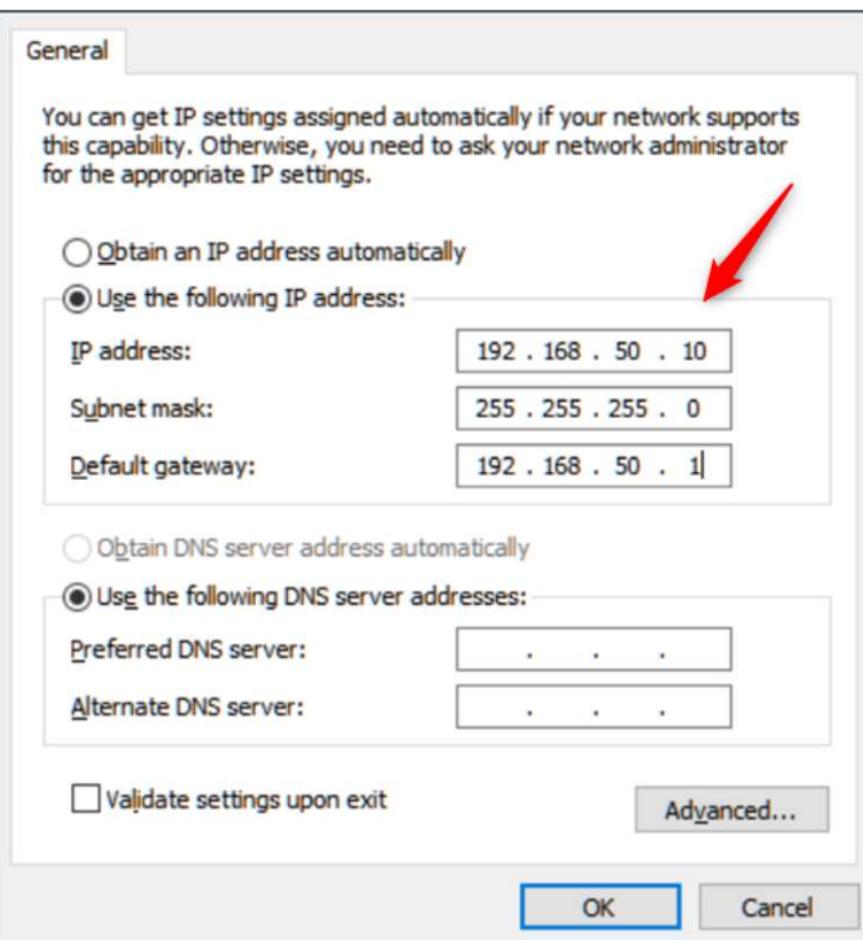
8	16	24	32
OP code (1)	Hardware Type (1)	Hw address length (1)	Hops (1)
Transaction Identifier (4)			
Seconds (2)		Flags (2)	
Client Ip Address (4)			
Your Ip Address (4)			
Server Ip Address (4)			
Gateway Ip Address (4)			
Client Hardware Address (16)			
Server Name (64)			
Boot Filename (128)			
DHCP Options (variabil)			

Transaction Identifier –
Câmp unic pentru fiecare schimb de mesaje cu un host

Boot Filename –
Fișier ce poate folosi la pornirea unor dispozitive

DHCP options –
Aici vedem tipul de mesaj transmis

Alocare adresă IP - Windows

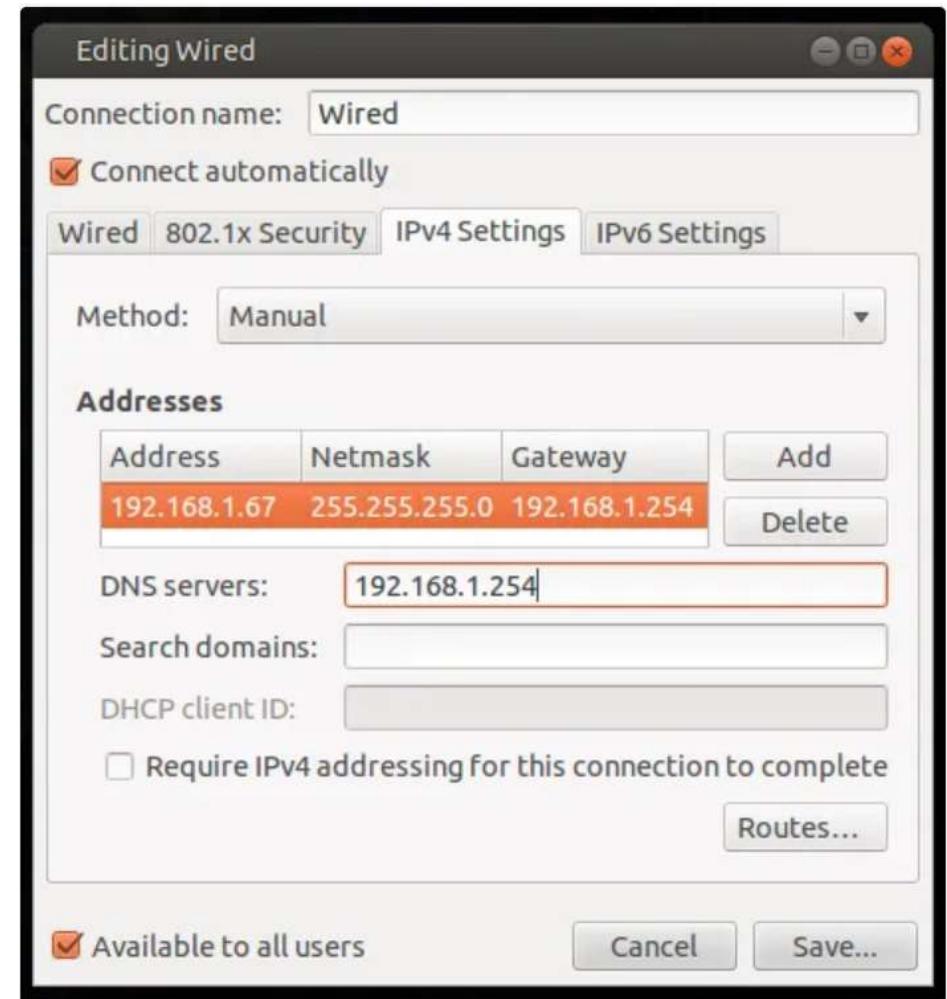


Alocare adresă IP - Linux

```
network:  
version: 2  
renderer: networkd  
ethernets:  
ens33:  
    dhcp4: no  
    dhcp6: no  
    addresses: [192.168.1.100/24]  
    gateway4: 192.168.1.1  
    nameservers:  
        addresses: [8.8.8.8,8.8.4.4]  
root@server1:/#
```

Prin editarea fișierului de networking – de ex cu vi, nano, etc.

Prin interfață grafică



Alocare statică – bazată pe MAC

Devices

List Tree RouterOS Types Mac Mappings

Address	MAC	/	Source Device	Source Type
10.5.8.82	00:1D:7D:90:75:08	/	10.5.8.1	snmp dhcp lease
10.5.8.82	00:1D:7D:90:75:08	/	10.5.8.1	snmp arp
10.5.8.92	00:1D:92:B0:C7:56	/	10.5.8.1	snmp dhcp lease
10.5.8.4	00:1F:29:26:4F:30	/	10.5.8.1	snmp dhcp lease
10.5.6.233	00:1F:29:26:4F:30	/	10.5.8.1	snmp dhcp lease
10.5.8.105	00:1F:5B:E7:23:F7	/	10.5.8.1	snmp dhcp lease
192.168.88.254	00:1F:D0:5F:58:A1	/	192.168.88.1	snmp arp
10.5.8.177	00:1F:D0:5F:58:A1	/	10.5.8.1	snmp dhcp lease
192.168.88.254	00:1F:D0:5F:58:A1	/	192.168.88.1	ros arp
10.5.8.3	00:21:5A:E6:E0:32	/	10.5.8.1	snmp dhcp lease
10.5.8.3	00:21:5A:E6:E0:32	/	10.5.8.1	snmp arp
10.5.8.3	00:21:5A:E6:E0:32	/	192.168.88.1	ros arp

Ruter Mikrotik

Maparea statică pe
Win2016 Server

New Reservation

Provide information for a reserved client.

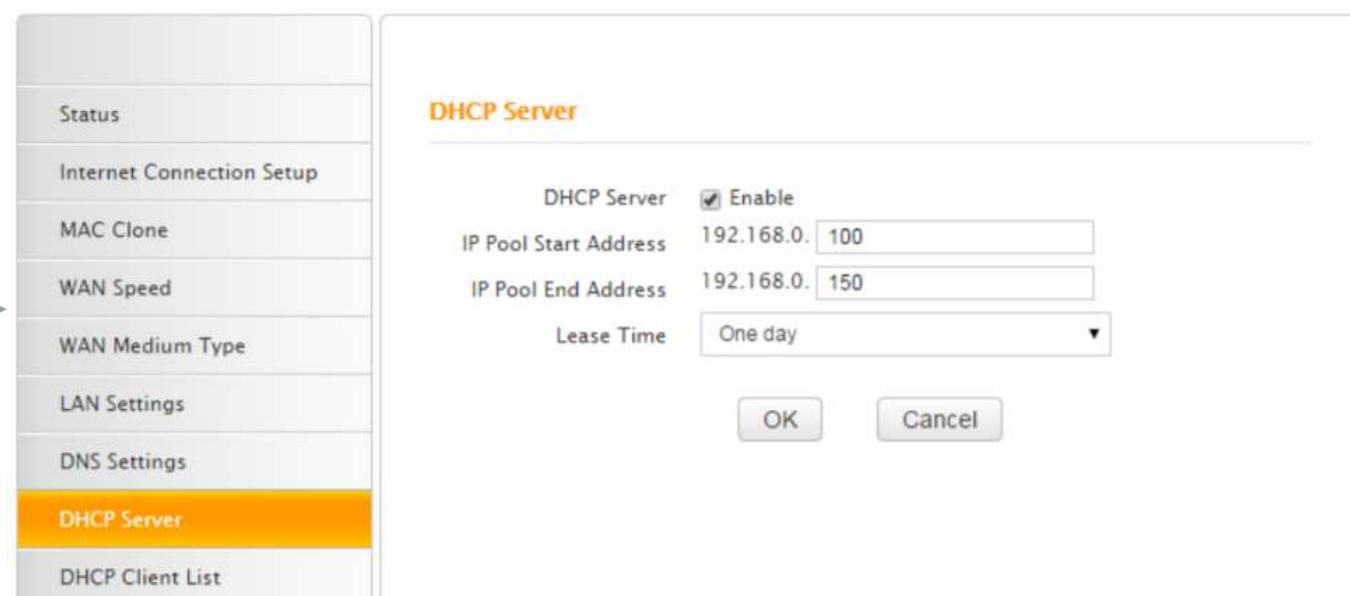
Reservation name:	hr printer
IP address:	10 . 0 . 0 . 150
MAC address:	00-11-22-33-44-55
Description:	Printer for HR
Supported types:	<input type="radio"/> Both <input checked="" type="radio"/> DHCP <input type="radio"/> BOOTP
<input type="button" value="Add"/> <input type="button" value="Close"/>	

Configurare pe un ruter

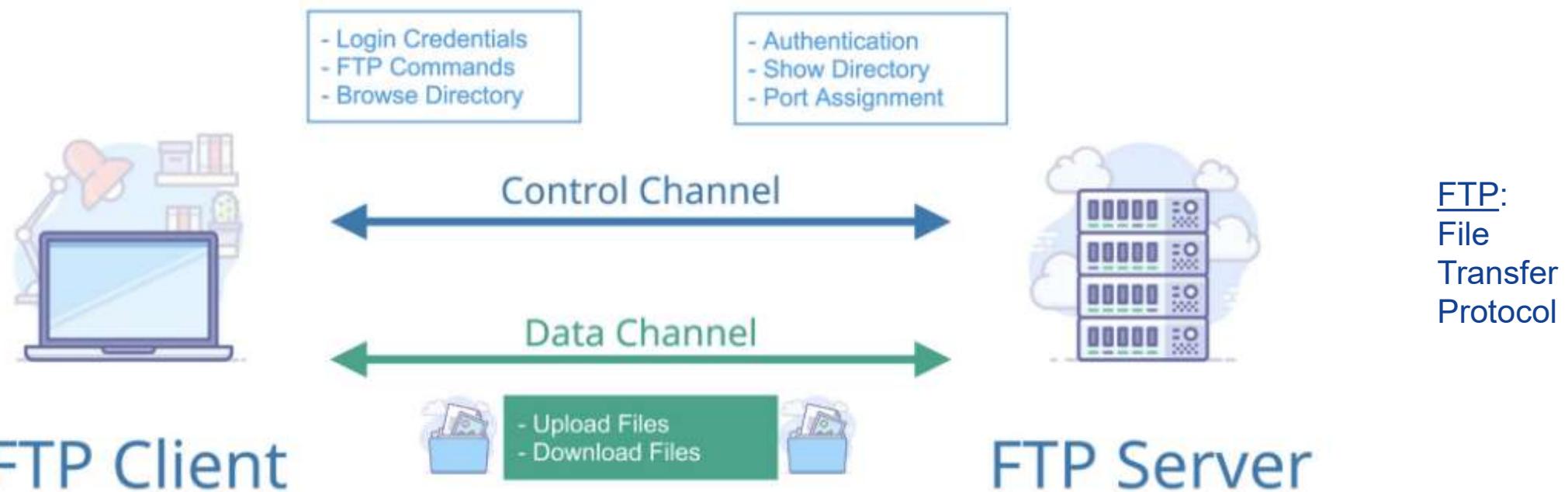
```
R1(config)#service dhcp          > Enables DHCP
R1(config)#ip dhcp pool mydhcp    > Create a DHCP pool
R1(dhcp-config)#network 192.168.100.0 255.255.255.0      > The network used to get IP address from Router
R1(dhcp-config)#default-router 192.168.100.254        > Default gateway of Hosts
R1(dhcp-config)#dns-server 192.168.100.250        > DNS Server of Hosts
R1(dhcp-config)#exit
R1(config)#
R1(config)#ip dhcp excluded-address 192.168.100.1 192.168.100.50      > The address that we want to exclude from DHCP pool. These address
never been used by Router to lease.
```

Configurarea DHCP pe un ruter
Cisco

Configurarea DHCP pe un ruter Tenda



Ce transferăm?



Cum transferăm



Cyberduck



FileZilla



WinSCP



SmartFTP



FREE FTP



Folosim clienti.

Vom utiliza 2 porturi



Portul 21 - control

Portul 20 - datele



Ambele țin de TCP



NOKIA

Cadrul cerere

5	0.060942	192.168.0.87	148.251.41.80	HTTP	529 GET /black-cat-images.html HTTP/1.1
12	0.127709	148.251.41.80	192.168.0.87	HTTP	895 HTTP/1.1 200 OK (text/html)
14	0.412051	192.168.0.87	148.251.41.80	HTTP	472 GET /images/6TpLybM7c.jpg HTTP/1.1
15	0.415979	192.168.0.87	148.251.41.80	HTTP	474 GET /image_gallery/3256.png HTTP/1.1
37	0.479489	192.168.0.87	148.251.41.80	HTTP	472 GET /images/5iRrpzKxT.jpg HTTP/1.1
39	0.479965	192.168.0.87	148.251.41.80	HTTP	482 GET /new_gallery/bus-clipart-49.jpg HTTP/1.1
56	0.485528	192.168.0.87	148.251.41.80	HTTP	476 GET /image_gallery/389987.jpg HTTP/1.1
84	0.540391	148.251.41.80	192.168.0.87	HTTP	542 HTTP/1.1 200 OK (PNG)

> Transmission Control Protocol, Src Port: 61807, Dst Port: 80, Seq: 1, Ack: 1, Len: 475

▼ Hypertext Transfer Protocol

> GET /black-cat-images.html HTTP/1.1\r\nHost: clipart-library.com\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.111 Safari/537.36\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\nAccept-Encoding: gzip, deflate\r\nAccept-Language: ro-RO,ro;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n[Full request URI: http://clipart-library.com/black-cat-images.html] ← URL-ul pe care îl căutăm

[HTTP request 1/3]

[Response in frame: 12] ← Cadrul în care vine răspunsul

[Next request in frame: 14]

Elementul căutat

Site-ul originar al informației

Varianta minimă de browser

Tipul de informație pe care îl acceptăm

Cadrul răspuns

5	0.060942	192.168.0.87	148.251.41.80	HTTP	529 GET /black-cat-images.html HTTP/1.1	
12	0.127709	148.251.41.80	192.168.0.87	HTTP	895 HTTP/1.1 200 OK (text/html)	
•	14	0.412051	192.168.0.87	148.251.41.80	HTTP	472 GET /images/6TpLybM7c.jpg HTTP/1.1
	15	0.415979	192.168.0.87	148.251.41.80	HTTP	474 GET /image_gallery/3256.png HTTP/1.1
	37	0.479489	192.168.0.87	148.251.41.80	HTTP	472 GET /images/5iRrpzKxT.jpg HTTP/1.1
	39	0.479965	192.168.0.87	148.251.41.80	HTTP	482 GET /new_gallery/bus-clipart-49.jpg HTTP/1.1
	56	0.485528	192.168.0.87	148.251.41.80	HTTP	476 GET /image_gallery/389987.jpg HTTP/1.1
	84	0.540391	148.251.41.80	192.168.0.87	HTTP	542 HTTP/1.1 200 OK (PNG)

> HTTP/1.1 200 OK\r\n

Server: nginx/1.10.2\r\nDate: Wed, 04 Nov 2020 14:55:13 GMT\r\nContent-Type: text/html; charset=UTF-8\r\nTransfer-Encoding: chunked\r\nConnection: keep-alive\r\nVary: Accept-Encoding\r\nContent-Encoding: gzip\r\n\r\n

[HTTP response 1/3]

[Time since request: 0.066767000 seconds]

[Request in frame: 5]

[Next request in frame: 14]

[Next response in frame: 270]

[Request URI: http://clipart-library.com/images/6TpLybM7c.jpg]

Data și ora la care am primit răspunsul

Care e tipul de conținut: text, imagine, etc

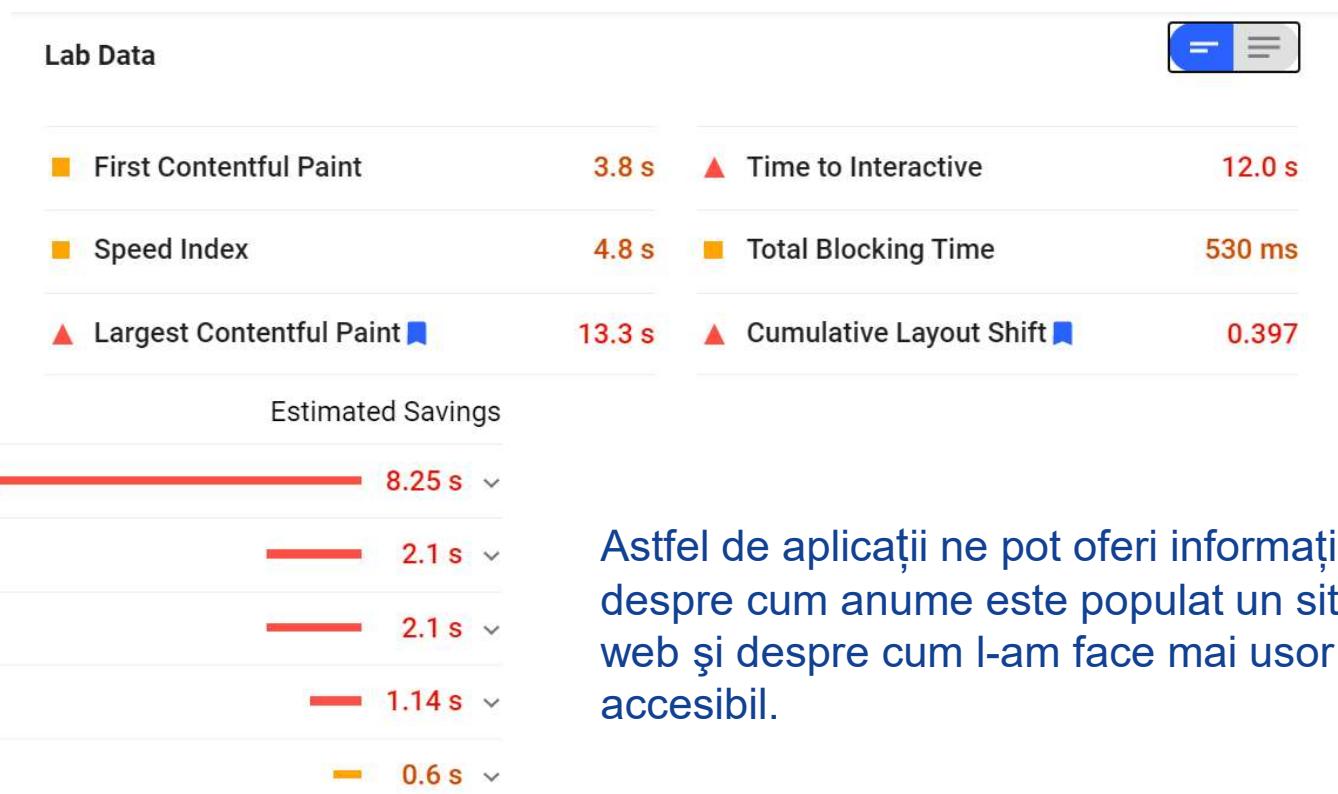
Cadrul în care a fost făcută cererea

HTTP

Cum se încarcă o pagină web

<https://developers.google.com/speed/pagespeed/insights/>

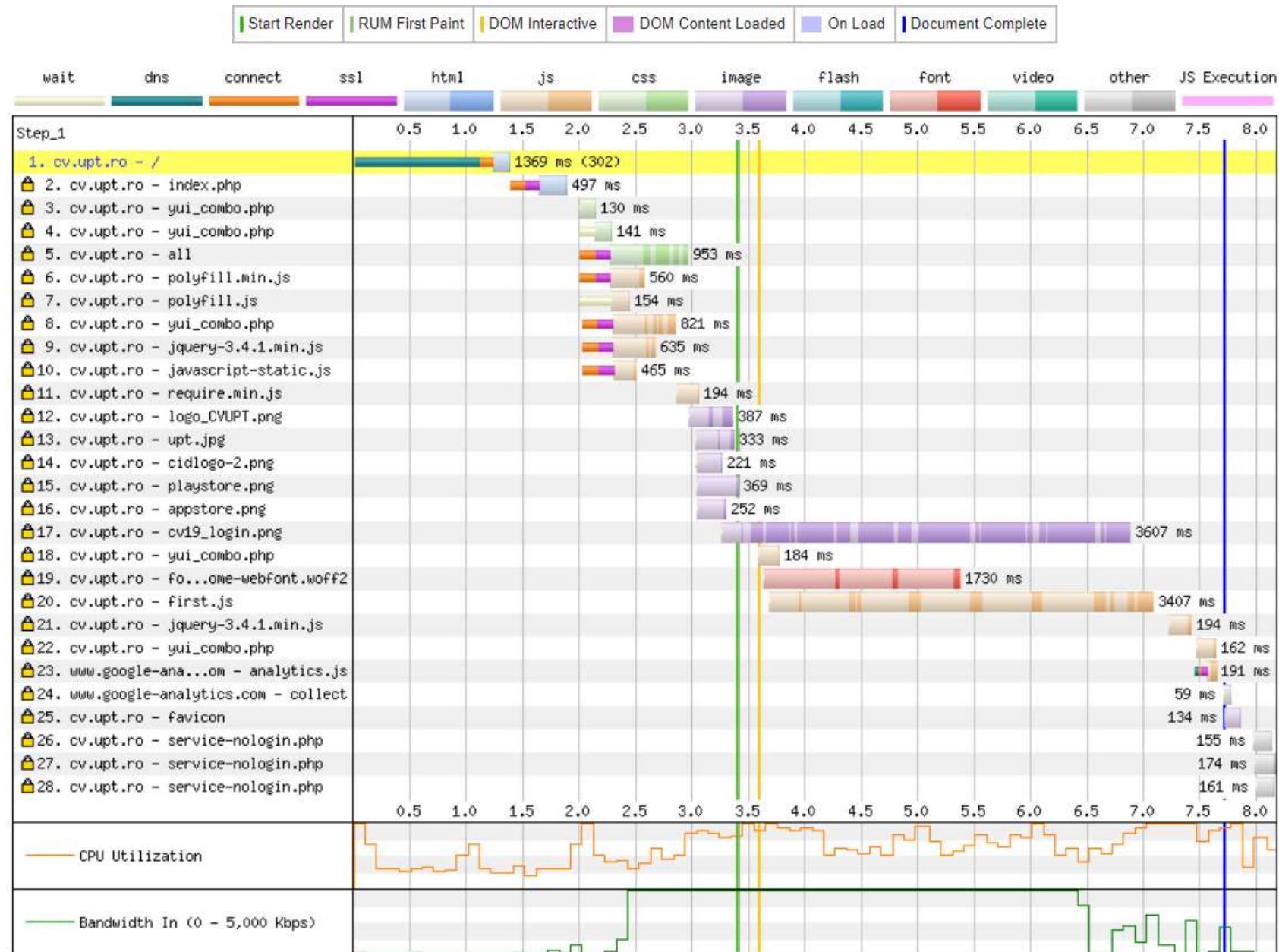
<https://webpagetest.org/>



Astfel de aplicații ne pot oferi informații despre cum anume este populat un site de web și despre cum l-am face mai usor accesibil.

HTTP Analiză

În această imagine puteți vedea o analiză a site-ului cv.upt.ro





That's all for today, see you next time!

Rețele de Calculatoare

Subnetări

Sumar al laboratorului

1

Necesitatea subnetării

2

Subnetare statică

Exemplu

3

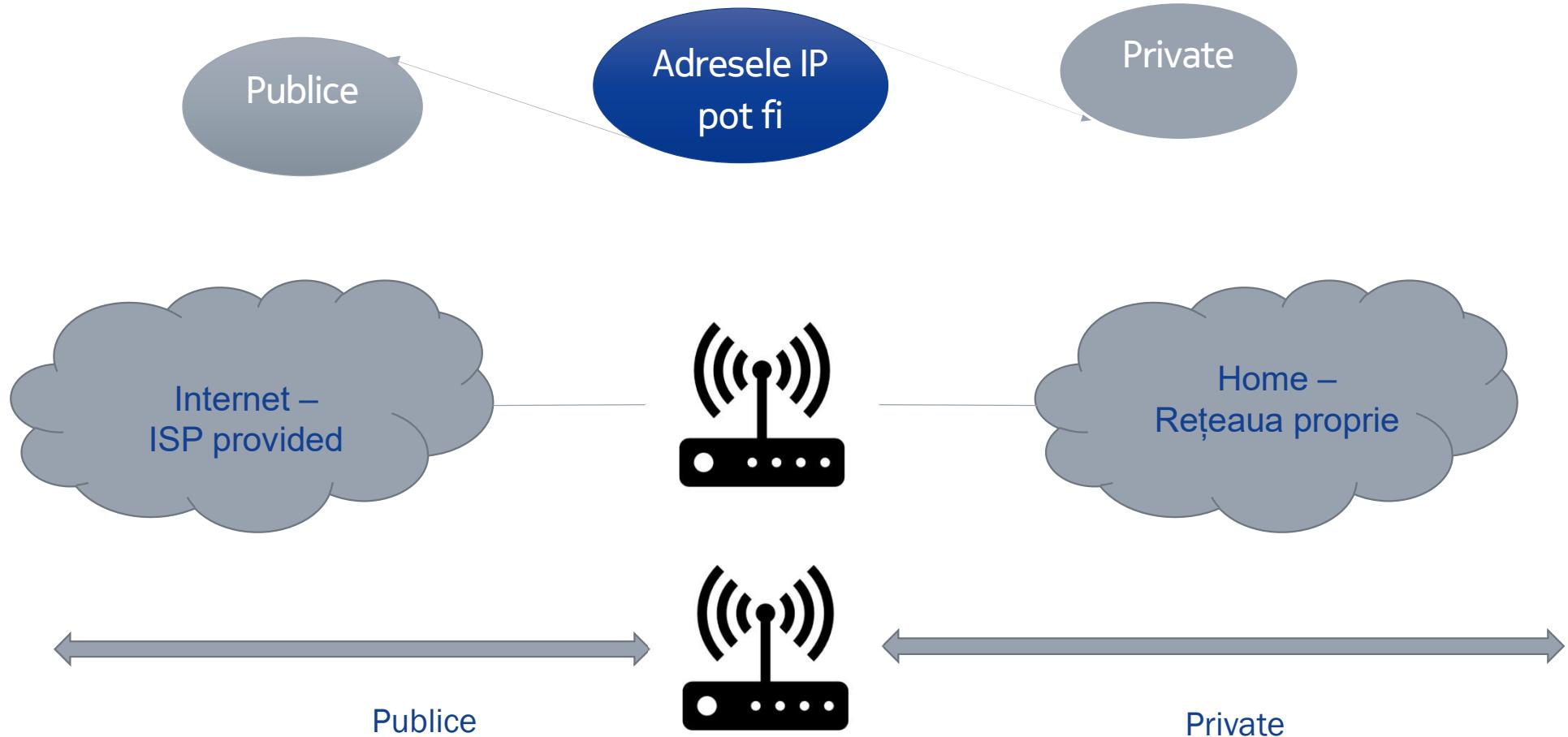
Subnetare dinamică

VLSM

Exemplu



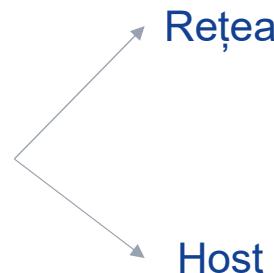
Necesitatea subnetării



Dacă în domeniul public trebuie să respectăm diverse reguli, în cel privat, singurii care fac regulile suntem noi – administratorii de rețea

În continuare vom parcurge un exemplu din cele 2 perspective ale subnetării, cea statică și cea dinamică, urmând să constatăm diferențele dintre ele.

Ne amintim că o adresă de IPv4 este formată din 32 biți împărțiți în biți de



Diferența dintre cele 2 categorii făcându-se prin masca de rețea

Elementele esentiale

Pentru a defini o retea avem nevoie de:

1. Adresă de retea;
2. Adresă de broadcast;
3. Adrese de host;
4. Adresă de gateway.

Acestea sunt elementele pe care le vom căuta în exemplul ce urmează, cu câteva particularitați.

Exemplu

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Pornim de la o adresă ce va fi dată, acompaniată de o mască de rețea: 192.168.25.172/24

Etapele 1 și 2

1. Transformăm în binar adresa de IP și masca date:

192.168.25.172/24

192.168.25.172	11000000.10101000.00011001.10101100
255.255.255.0	11111111.11111111.11111111.00000000

2. Aflăm adresa de rețea efectuând un ȘI logic între cele 2

192.168.25.0/24 11000000.10101000.00011001.00000000

ATENȚIE!! Nu uită să adăugăm masca de rețea oricărei adrese, altfel rezultatul nu este cel corect

11000000.10101000.00011001.00000000

Biți de rețea din adresa de rețea

Biți de host
din adresa de
rețea

3. Aflăm adresa de broadcast, transformând biți de HOST din adresa de rețea din "0" în "1"

11000000.10101000.00011001.11111111

Biți de rețea din adresa de rețea

Biți de host
din adresa de
rețea

4. Determinăm adresele de host între limitele calculate anterior

Adresa de rețea < adresele de host < adresa de broadcast

Host 1 = adresa de rețea +1

Host 2 = adresa de rețea +2

.

.

Host n = adresa de rețea +n

.

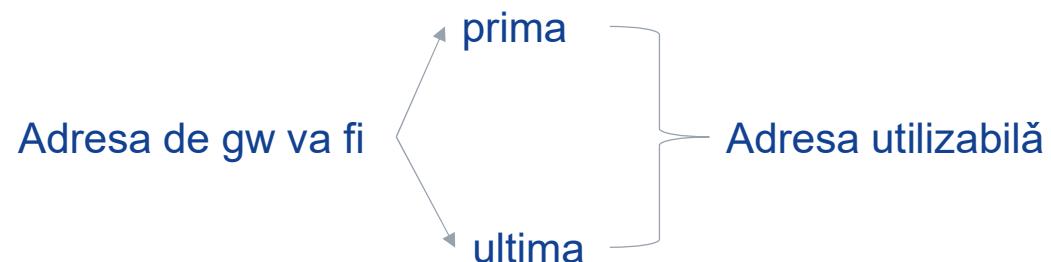
.

Penultimul host= adresa de broadcast -2

Ultimul host = adresa de broadcast -1

5. Determinarea unei adrese de gateway

Nu există nici o regulă pentru determinarea adresei de gateway, aceasta putând fi aleasă ca orice adresă de host.



Subnetare statică

Implicită împărțirea unei rețele, în rețele mai mici având caracteristica unei măști de rețea de dimensiune constantă pe tot parcursul subnetării

Dacă masca de rețea va fi constantă =>

Numărul de rețele va fi mereu puterile lui 2

Numărul de host-uri va fi același pentru toate rețelele rezultate

Pornim de la aceeași adresă de rețea ca și în exemplul anterior și o vom împărți în 3 rețele, conform unei mici firme de dezvoltare de software, având nevoie de : dezvoltatori, testori, și contabili

Subnetare statică

Exemplu – partea 1

Adresa de rețea de la care vom porni: 192.168.25.0/24

Având nevoie de 3 rețele: dev, test și contabili vom “fura” 2 biți din cei de host și îi vom împrumuta celor de rețea

Rețeaua 1: dezvoltatori



Transformarea în zecimal:

1. Adresa de rețea

192.168.25.0/**26**

2. Adresa de broadcast

192.168.25.63/**26**

3. Adresele de host:

192.168.25.1/**26** -> 192.168.25.62/**26**

Subnetare statică

Exemplu – partea 2

! Pentru a obține adresa de broadcast, transformăm biți de host din “0” în “1” !

Rețeaua 2: testori

11000000.10101000.00011001.01000000

Biți de rețea originari

biți de host

Biți de rețea pentru noua rețea

Transformarea în zecimal:

1. Adresa de rețea
192.168.25.64/26

2. Adresa de broadcast
192.168.25.127/26

3. Adresele de host:
192.168.25.65/26 -> 192.168.25.126/26

Subnetare statică

Exemplu – partea 3

! Pentru a obține adresa de broadcast,
transformăm biți de host din “0” în “1” !

Rețeaua 3: contabili



Transformarea în zecimal:

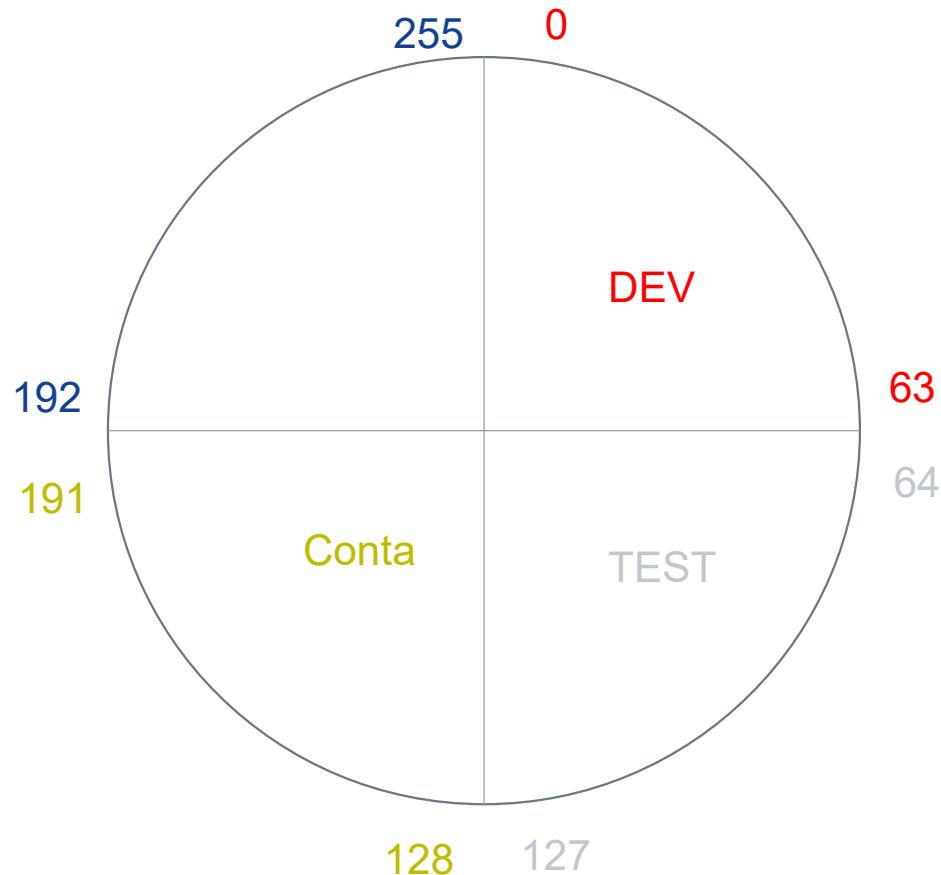
1. Adresa de rețea
192.168.25.128/26

2. Adresa de broadcast
192.168.25.191/26

3. Adresele de host:
192.168.25.129/26 -> 192.168.25.190/26

Subnetare statică

Distributia de host-uri



După cum se poate observa în diagrama alăturată rețeaua a fost împărțită în 4 rețele egale, din care vom avea nevoie doar de 3, toate rețelele fiind egale ca și număr de host-uri.

Subnetare dinamică - VLSM

Implicită împărțirea unei rețele, în rețele mai mici având caracteristica unei măști de rețea de dimensiune variabilă pe parcursul subnetării. Această subnetare ține cont de numarul de host-uri in loc de cel de rețele

Dacă masca de rețea va fi variabilă

Nu avem constrângeri legate de numărul de rețele pe care le folosim

Numărul de host-uri va fi diferit, în funcție de necesități

Pornim de la aceeași adresa de rețea ca și în exemplul anterior și o vom împărți în 3 rețele, conform unei mici firme de dezvoltare de software, având nevoie de :

- 100 dezvoltatori;
- 50 de testori;
- 18 manageri si contabili

Exemplu – partea 1

Adresa de rețea de la care vom porni: 192.168.25.0/24

a) Rețeaua dezvoltatorilor

100 dezvoltatori => 100 adrese de host + 1 adresa de rețea + 1 adresa de broadcast=102 adrese necesare
102 adrese se pot reprezenta pe 7 biți de host => vom avea 24(originari)+1 bit de rețea imprumutat

11000000.10101000.00011001.00000000

Biți de rețea originari

Biți de rețea pentru noua rețea

biți de host

! Pentru a obține
adresa de broadcast,
transformam biții de
host din “0” în “1” !

Transformarea în zecimal:

1. Adresa de rețea
192.168.25.0/25

2. Adresa de broadcast
192.168.25.127/25

3. Adresele de host:
192.168.25.1/25 -> 192.168.25.126/25

Exemplu – partea 2

b) Rețeaua testorilor

50 testori => 50 adrese de host + 1 adresa de rețea + 1 adresa de broadcast=52 adrese necesare

52 adrese se pot reprezenta pe 6 biți de host => vom avea 25 biți rețea (24 orig+1 anterior) + 1 bit rețea=26

11000000.10101000.00011001.10000000

Biți de rețea originari

biți de host

Biți de rețea pentru noua rețea

3. Adresele de host:

192.168.25.129/26 -> 192.168.25.190/26

! Pentru a obține
adresa de broadcast,
transformam biții de
host din "0" în "1" !

Transformarea în zecimal:

1. Adresa de rețea

192.168.25.128/26

2. Adresa de broadcast

192.168.25.191/26

Exemplu – partea 3

c) rețeaua Mng si contabili

18 mng=> 18 adrese de host + 1 adresa de rețea + 1 adresa de broadcast=20 adrese necesare

20 adrese se pot reprezenta pe 5 biți de host => vom avea 26 biți rețea (24 orig+2 anterior) + 1 bit rețea=27

11000000.10101000.00011001.11000000

Biți de rețea originari

biți de host

Biți de rețea pentru noua rețea

! Pentru a obține
adresa de broadcast,
transformam biții de
host din “0” în “1” !

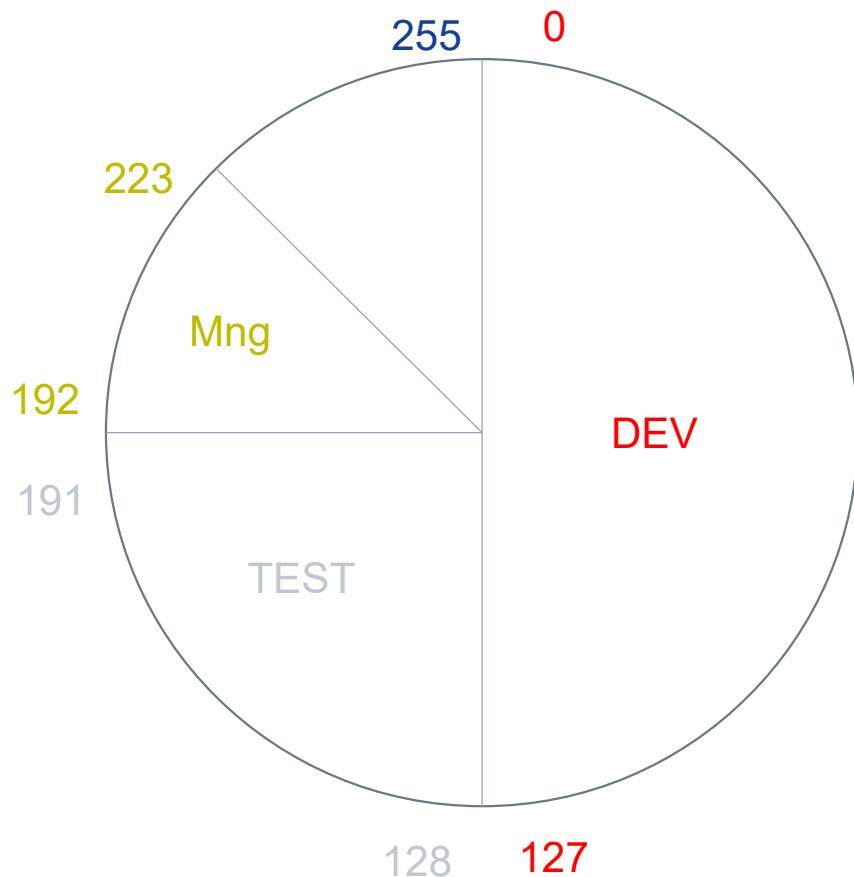
Transformarea în zecimal:

1. Adresa de rețea
192.168.25.192/27

2. Adresa de broadcast
192.168.25.223/27

3. Adresele de host:
192.168.25.193/27 -> 192.168.25.222/27

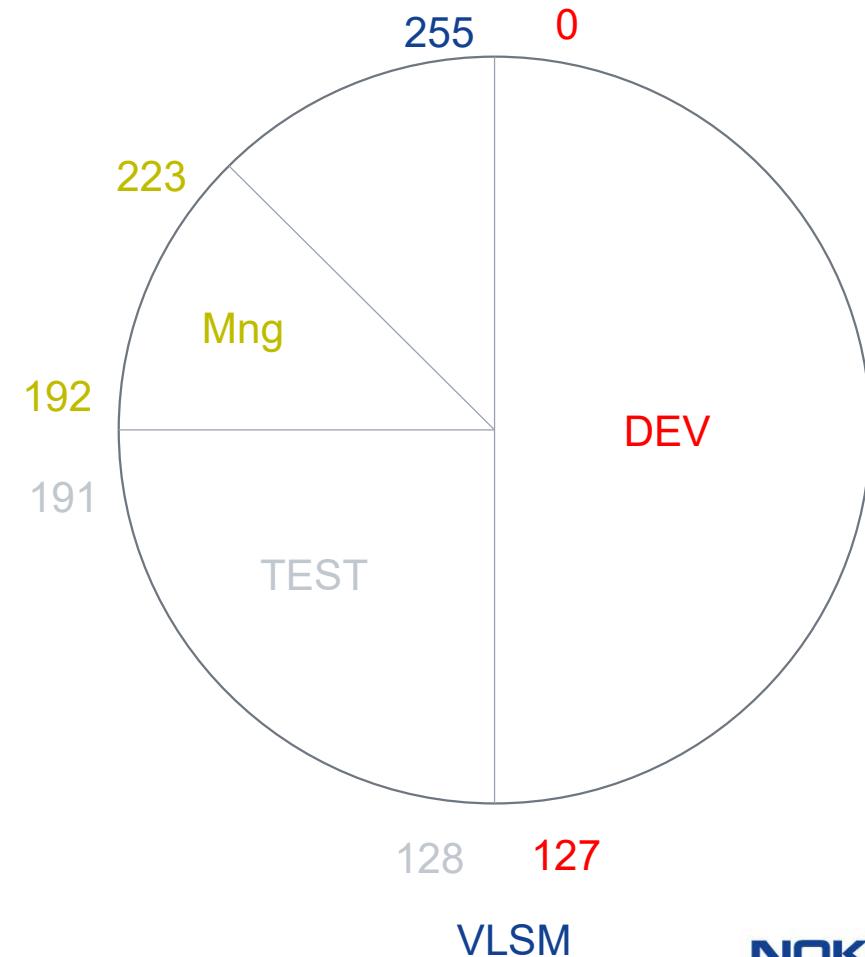
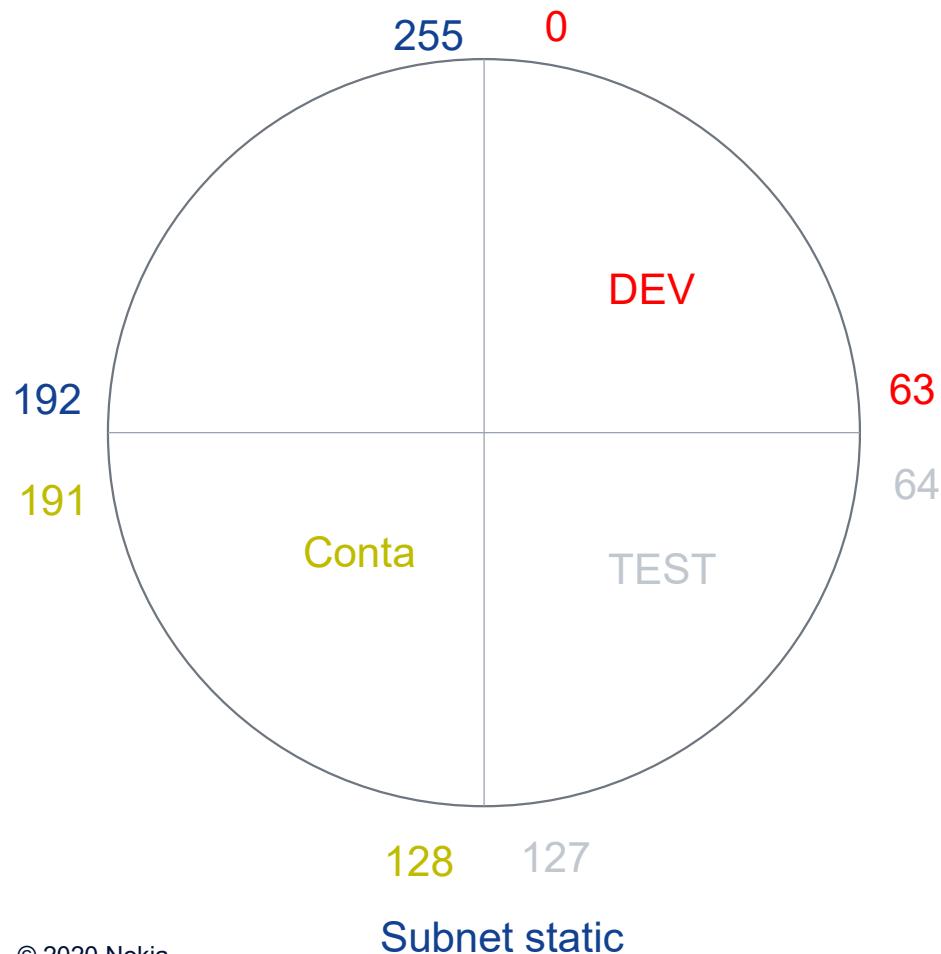
Distributia host-urilor



Se observă că deși numărul de rețele folosit este același ca în cazul anterior, distribuția host-urilor în rețele este “mai realistă” pentru cazul unei firme de dezvoltare soft.

Distributia host-urilor - comparatie

Static vs VLSM



- În cadrul subnetării atât statice cât și cu VLSM, succesiunea bițiilor împrumutați va ține cont de succesiunea normală a acestora “0” apoi “1”.
- Dacă trebuie împrumutați mai multi biți, succesiunea se păstrează:
“00”; “01”; “10”; “11”; etc.
- Pentru VLSM ordinea host-urilor va fi mereu considerată în ordine descrescătoare, indiferent de rețele.
- Tot timpul țineți cont de adresa și masca de rețea primită din start pentru a răspunde la orice întrebare.

Exemple de întrebări

- Împărțiți rețeaua dată în X subrețele egale. Cât este masca obținută?
- Dacă împărțim o rețea în R subrețele egale, care este numărul maxim de host-uri pe care îl putem avea în fiecare rețea?
- Care este host-ul Y din rețeaua Z?
- Care este adresa de gateway din rețeaua M, dacă se consideră că este prima/ultima din rețea?
- Care este ultimul host din rețeaua A?
- Din care rețea face parte adresa a.b.c.d?
- Orice alte variațiuni pe temă.



That's all for today, see you next time!

Rețele de Calculatoare

Echipamente folosite în RC

Sumarul laboratorului

1

Unde putem trimite date

În interiorul unei rețele

Între rețele

Domenii de coliziune și de broadcast

3

Packet tracer

Cofigurațiile de bază ale
unui echipament

2

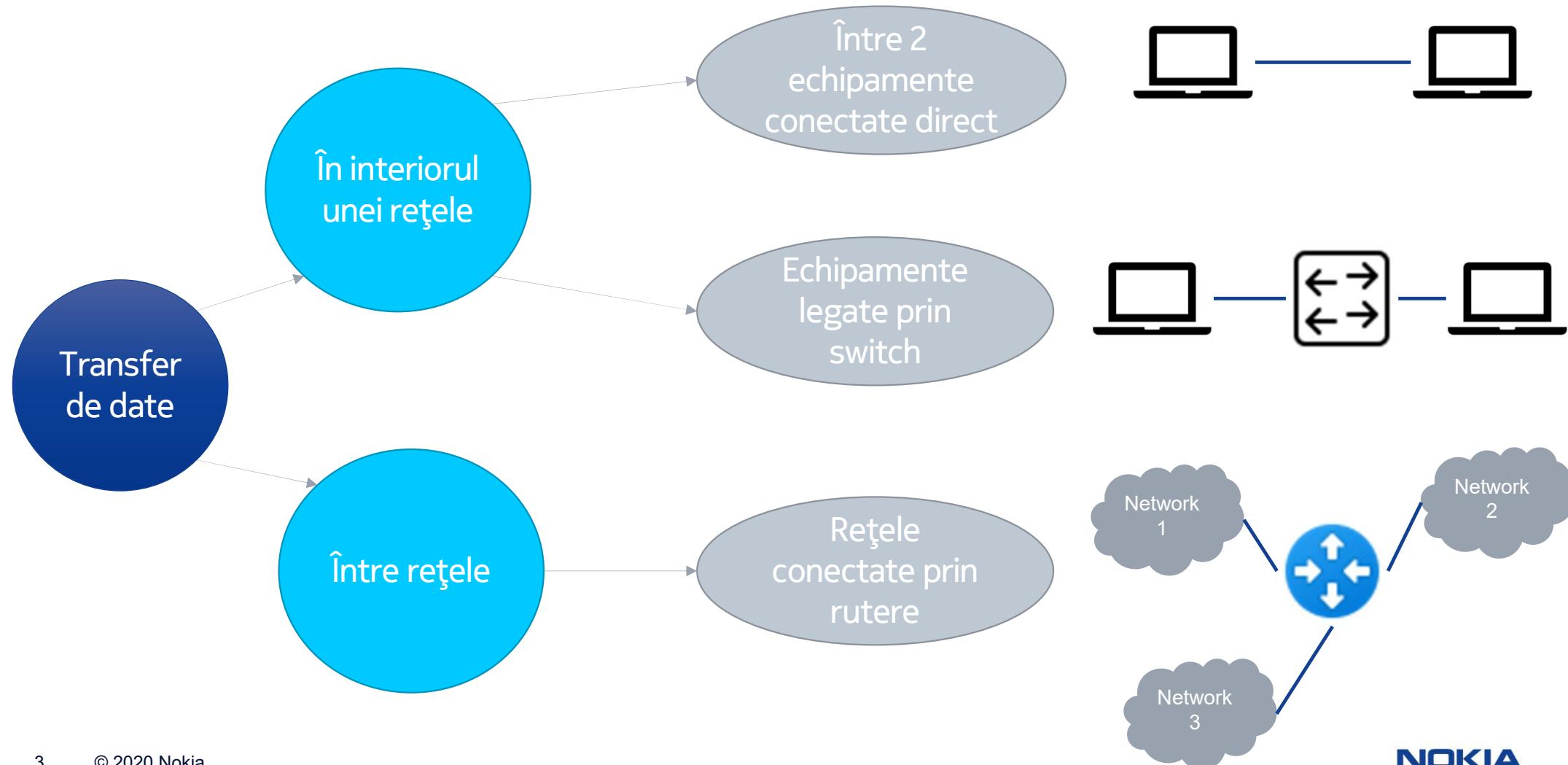
Echipamente de transport

Switch-uri

Rutere

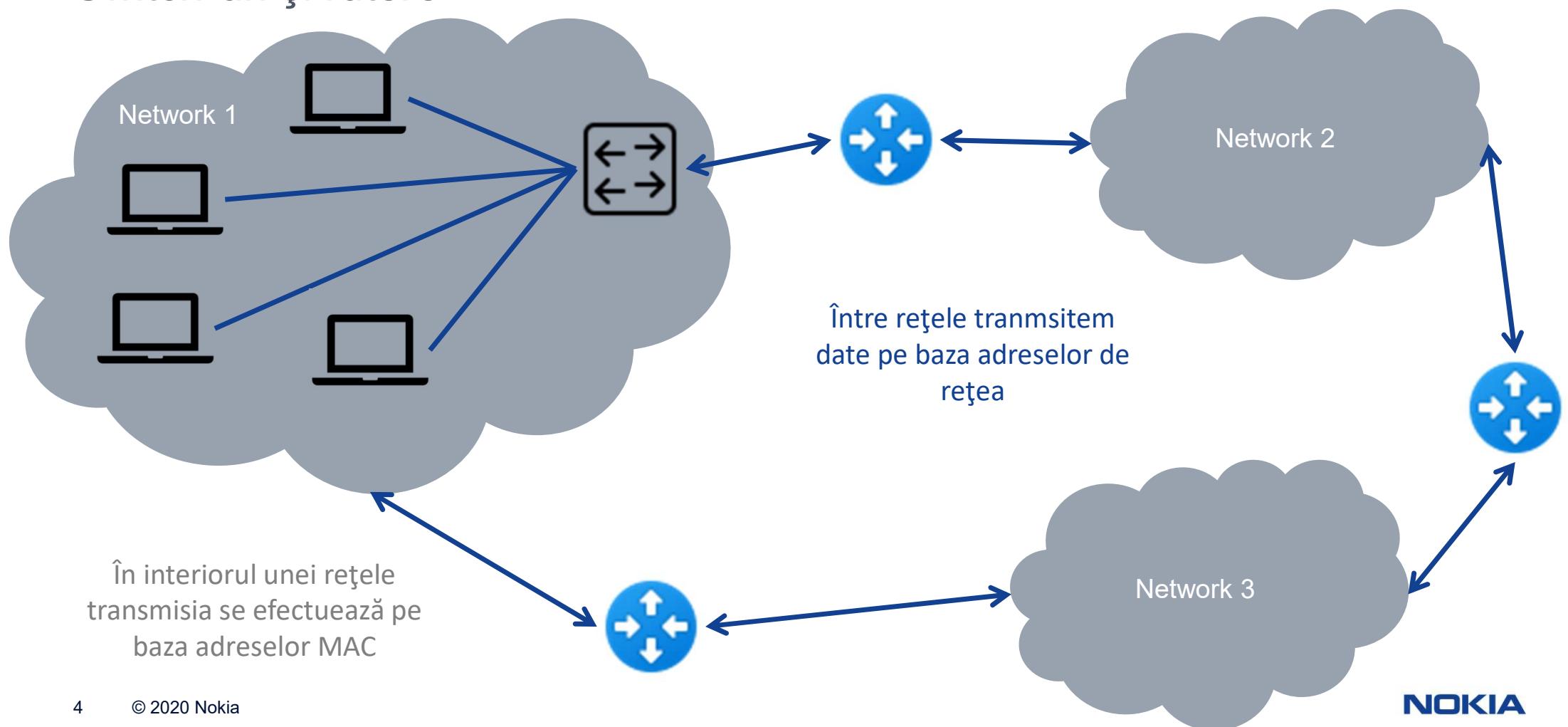


Unde putem trimite datele

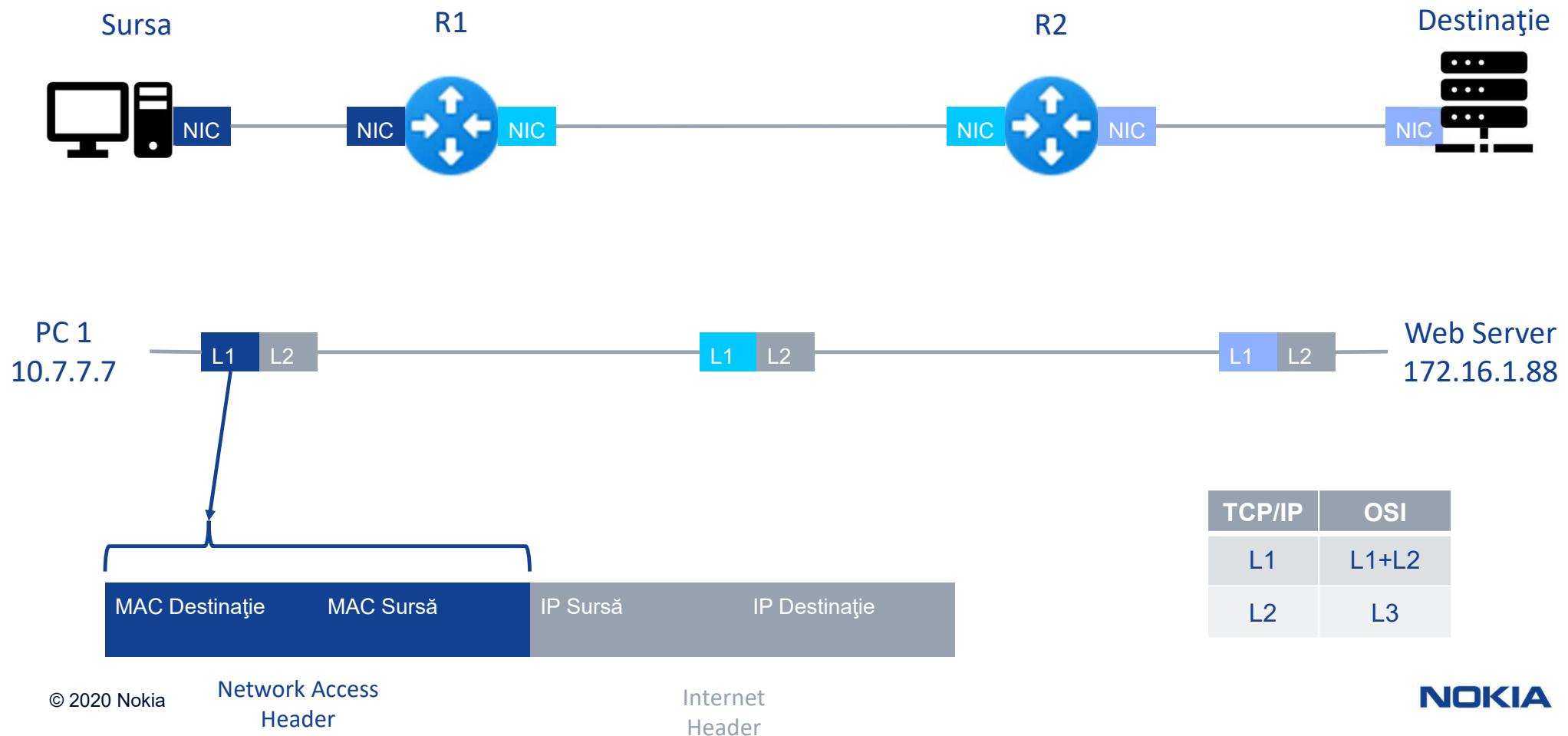


Cum transmitem datele

Switch-uri și rutere

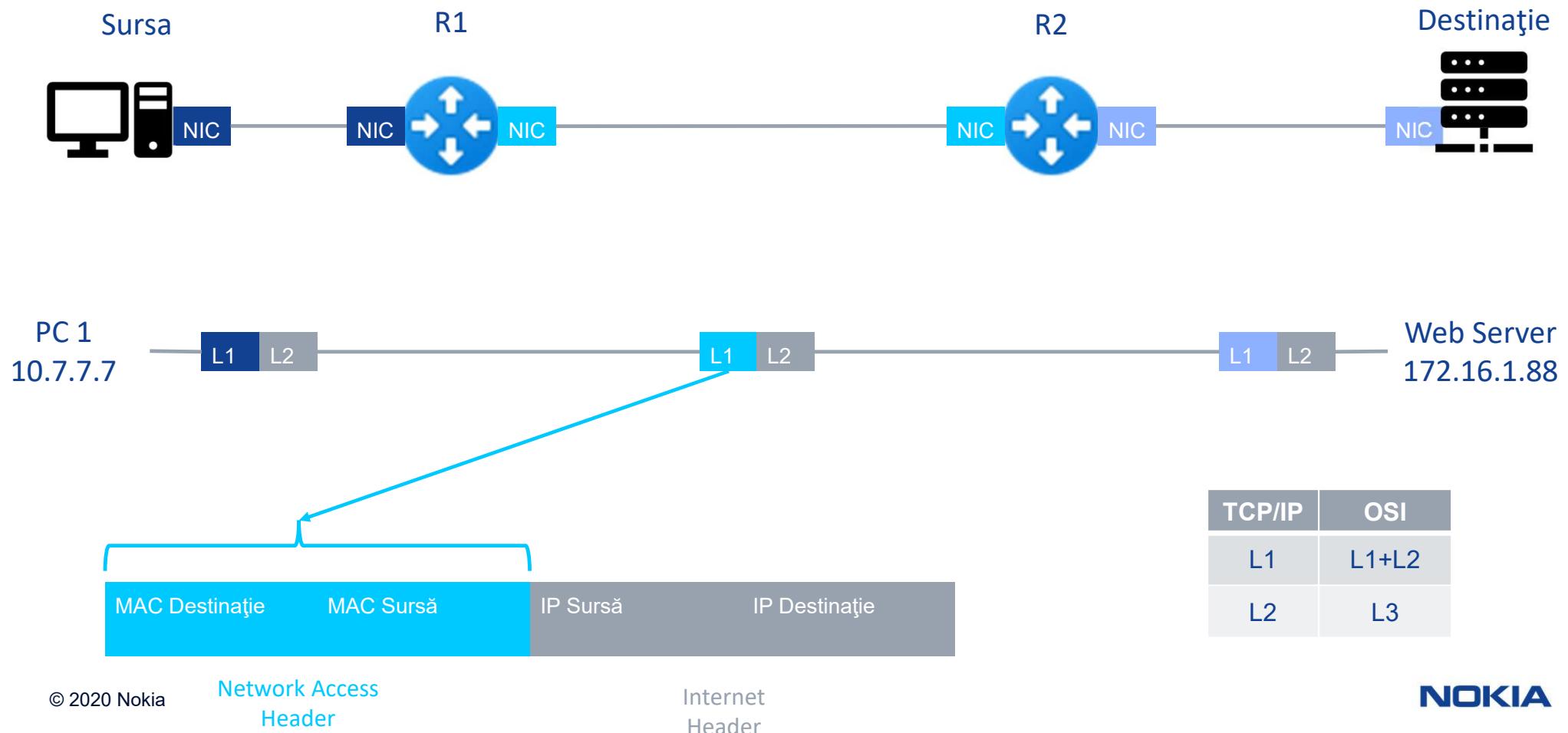


Transferul de date Prin Internet



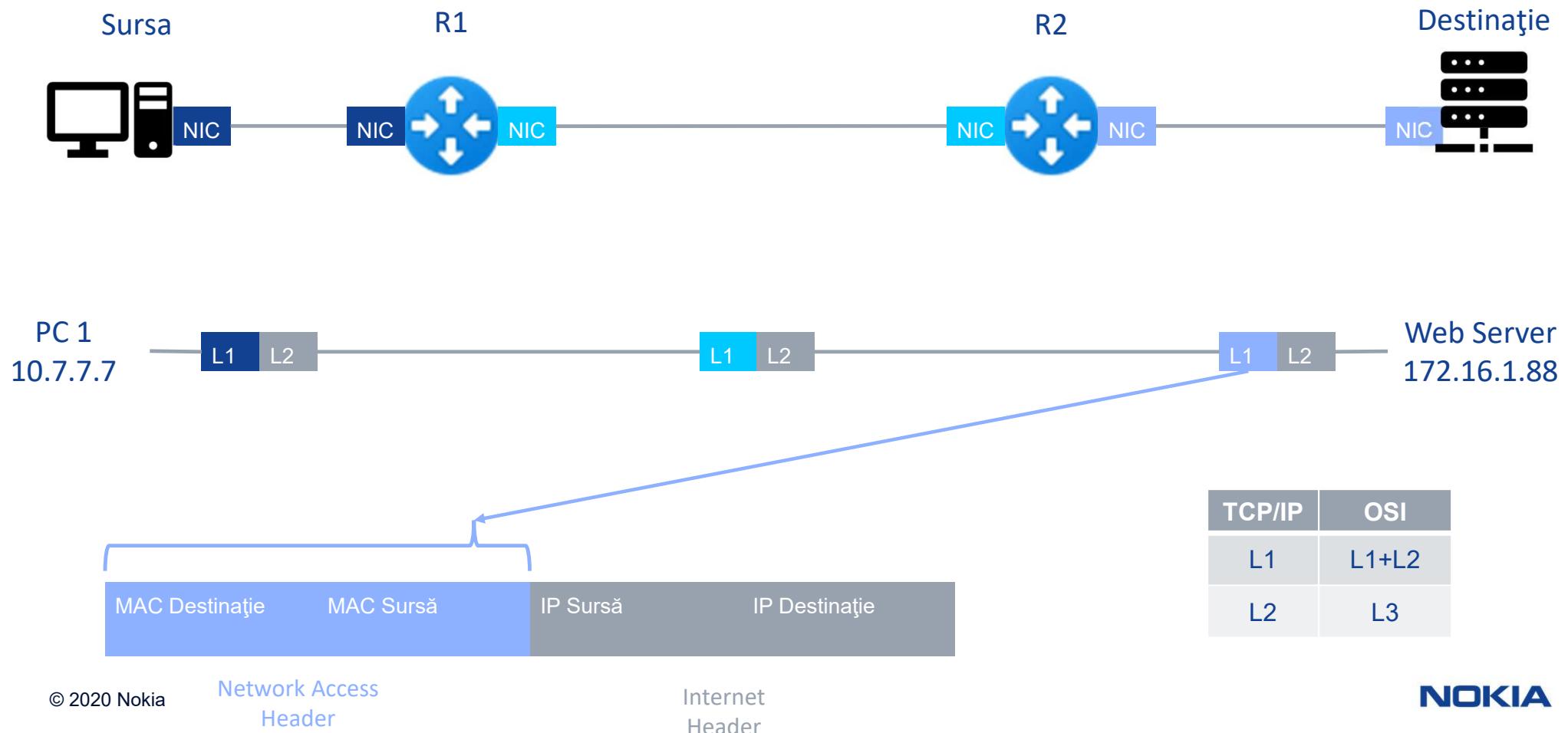
Transferul de date

Prin Internet



Transferul de date

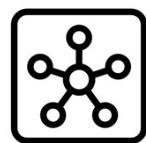
Prin Internet



Echipamente de transport

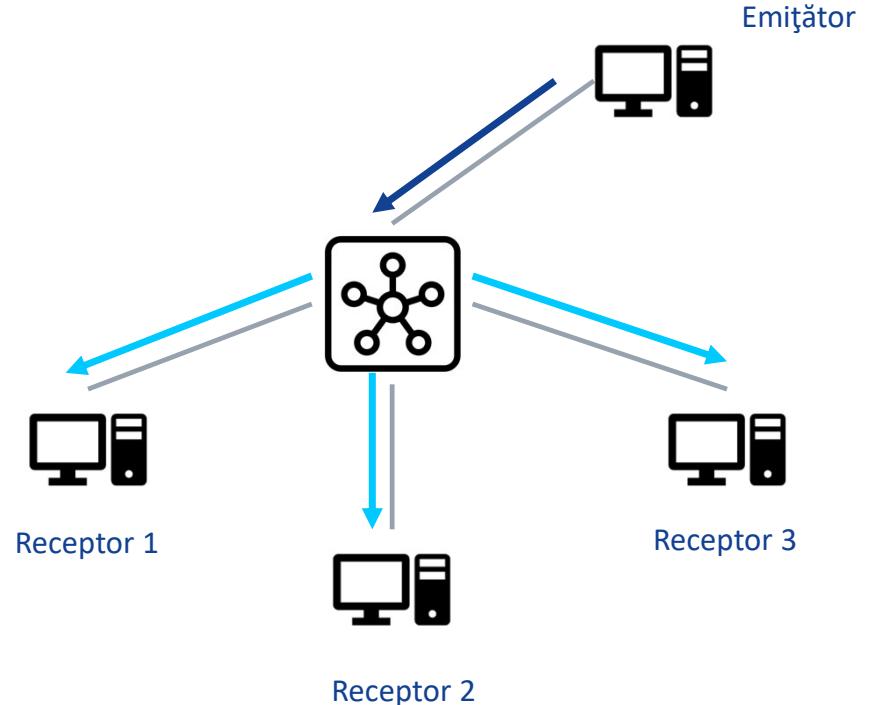
Hub-uri

Reprezentare în diagrame



Funcțiile unui Hub

- Conectează diferite echipamente via cabluri Ethernet
- Are rol de repotor și transmite doar în regim broadcast



Echipamente de transport

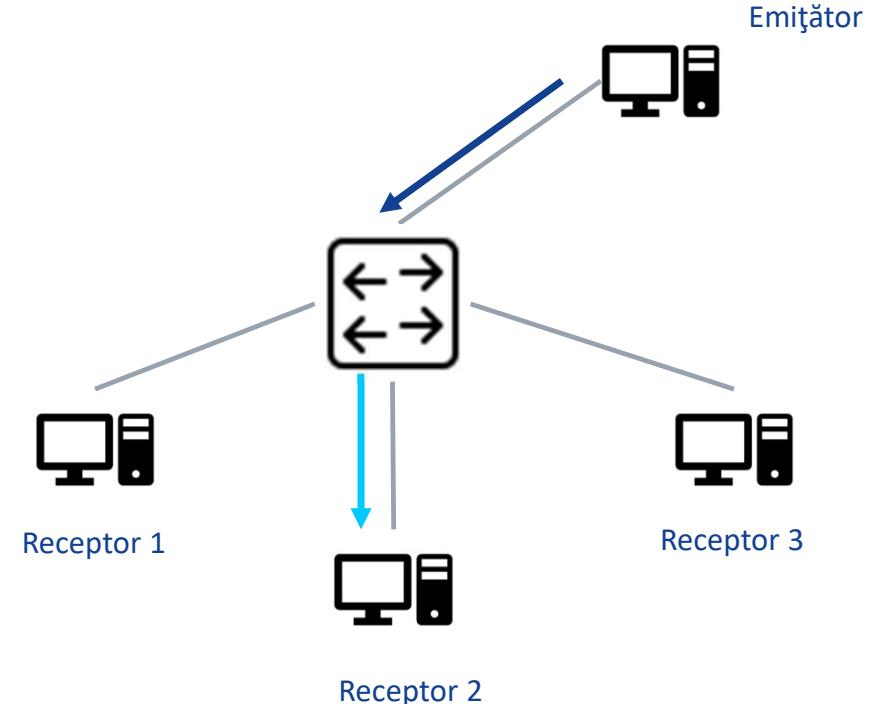
Switch-uri

Reprezentare în diagrame



Funcțiile switch-ului

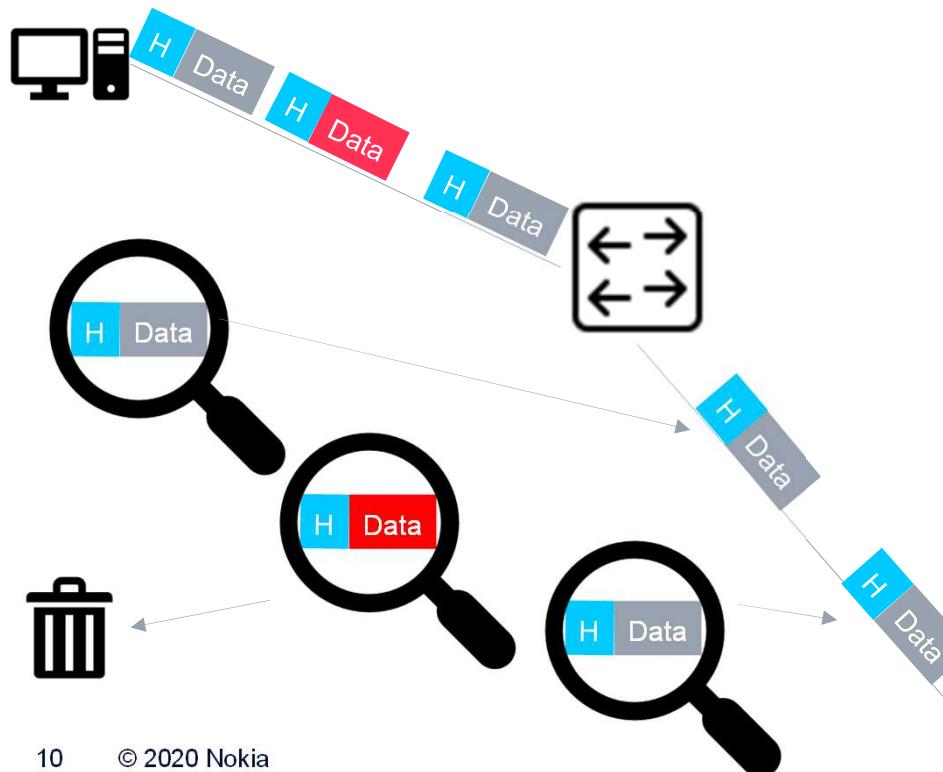
- Conectează echipamente prin Ethernet și fibră optică
- Permite separarea domeniilor de coliziune
- Permite transmisiuni unicast, multicast și broadcast



Echipamente de transport

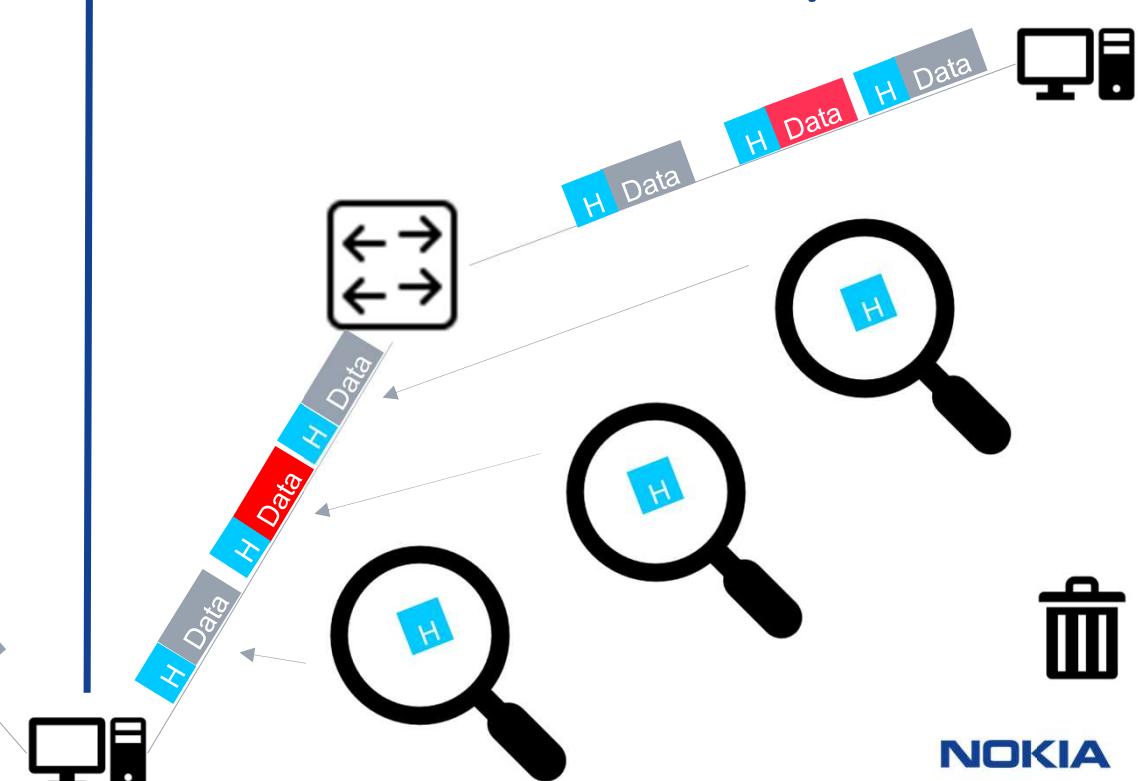
Switch-uri

Store and forward
Examinăm întreg frame-ul



Tipuri de switch-ing

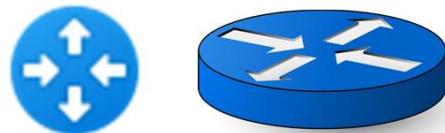
Cut-through
Examinăm header-ul, până la MAC-ul destinație



Echipamente de transport

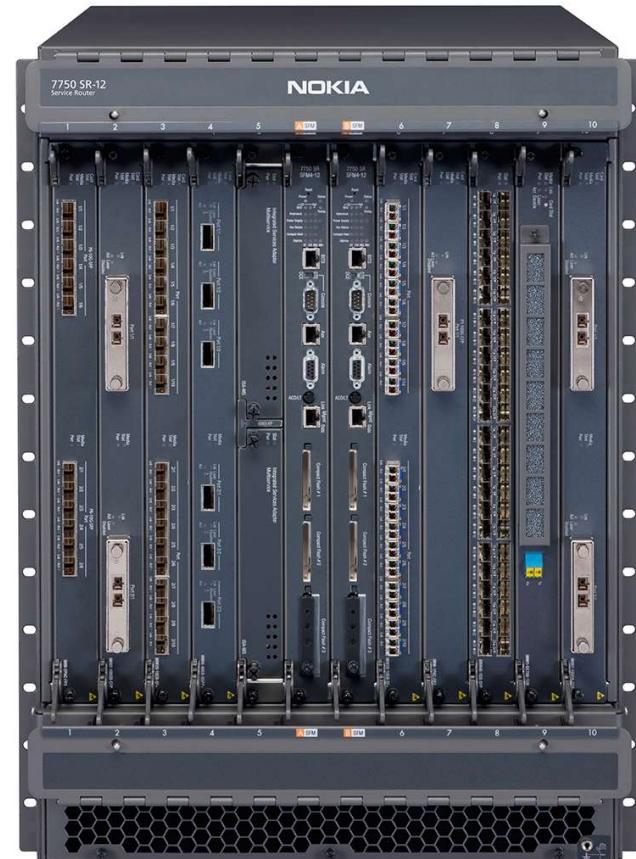
Rutere

Reprezentare în diagrame:



Funcțiile ruter-ului

- Conectează rețele prin diverse medii de transmisie: Eth, FO, Serial, ATM, etc.
- Limitează domeniile de broadcast
- Permite crearea listelor de acces (ACL) ce funcționează ca un firewall pentru rețea
- Permite funcția de Network Address Translation – transformă o adresă privată într-o publică



Domenii în RC

Coliziune și broadcast

O coliziune între 2 pachete poate avea loc când pachetele sunt pe același tronson de rețea.

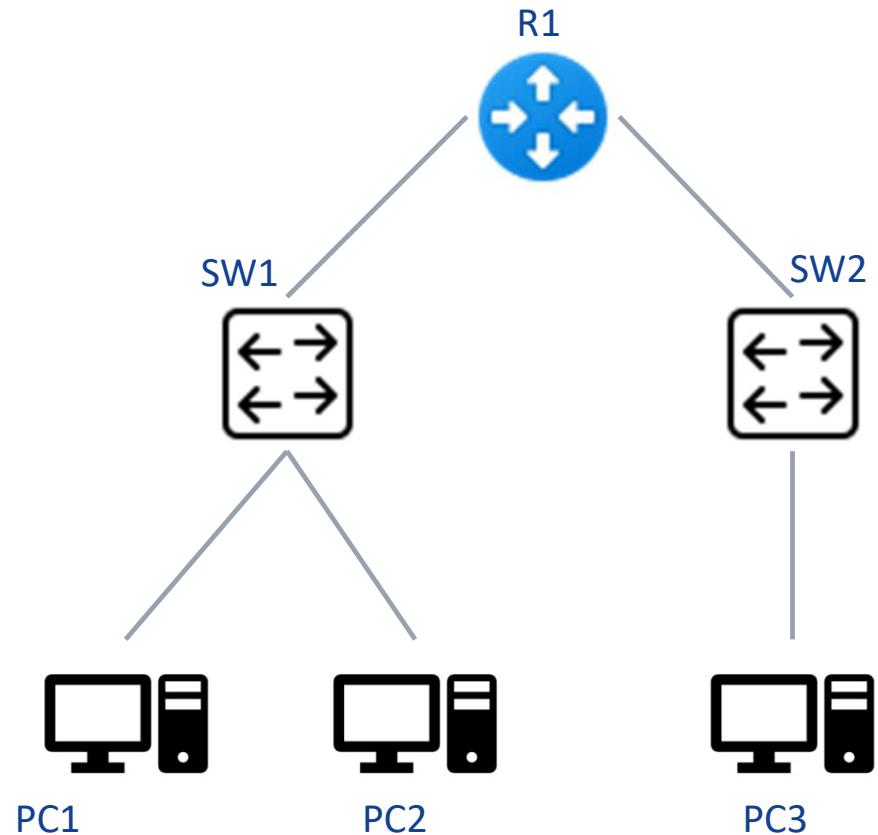
În RC Coliziunile sunt elemente distructive.

Există 2 mecanisme de tratare a coliziunilor:

- Deteccie (CSMA – CD → folosit pe Ethernet)
- Evitare (CSMA – CA → folosit pe wireless)

Domeniile de broadcast sunt zonele în care se transmit mesaje de broadcast

Broadcast-ul poate reprezenta o problemă în rețea când rețeaua este încarcată de mesaje și limitează lățimea de bandă (debitul) utilă.



Domenii în Rc

Coliziune și Broadcast

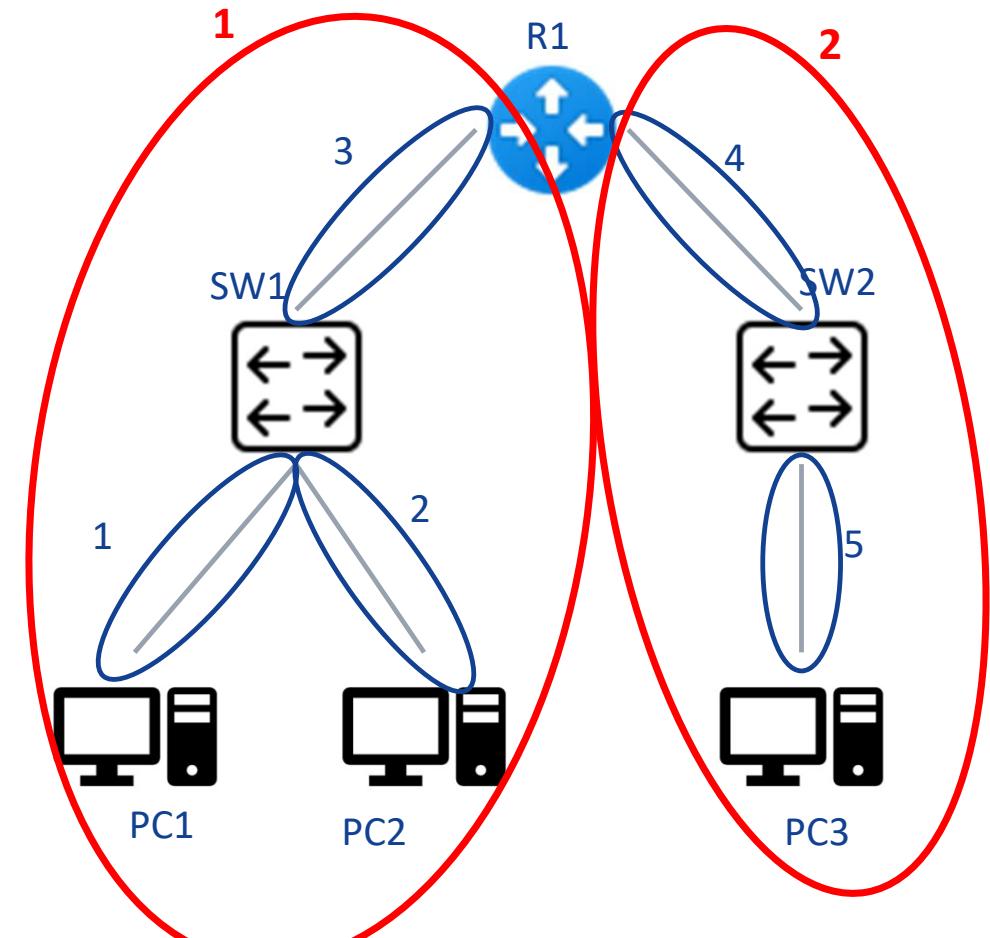
Să identificăm câte domenii de coliziune și de broadcast avem în imaginea alăturată.

Fiecare conexiune utilizată a unui switch reprezintă un domeniu de coliziune.

Astfel, putem identifica 5 domenii de coliziune

Fiecare conexiune a unui ruter duc la un domeniu de broadcast

Astfel, putem identifica 2 domenii de broadcast



Moduri de configurare

User exec commands:

Ping	Show (Limited)
Enable	etc.

Privileged Exec commands:

ALL user exec commands

Debug commands

Reload

Configure

Etc.

Global configuration commands:

Hostname
Enable secret
Ip route
Interface

Interface commands
Ip address
Encapsulation
Shutdown / no shutdown

ethernet

serial

Router

rip
ospf
eigrp

vty
console
etc

Router engine commands

network
version
Auto-summary

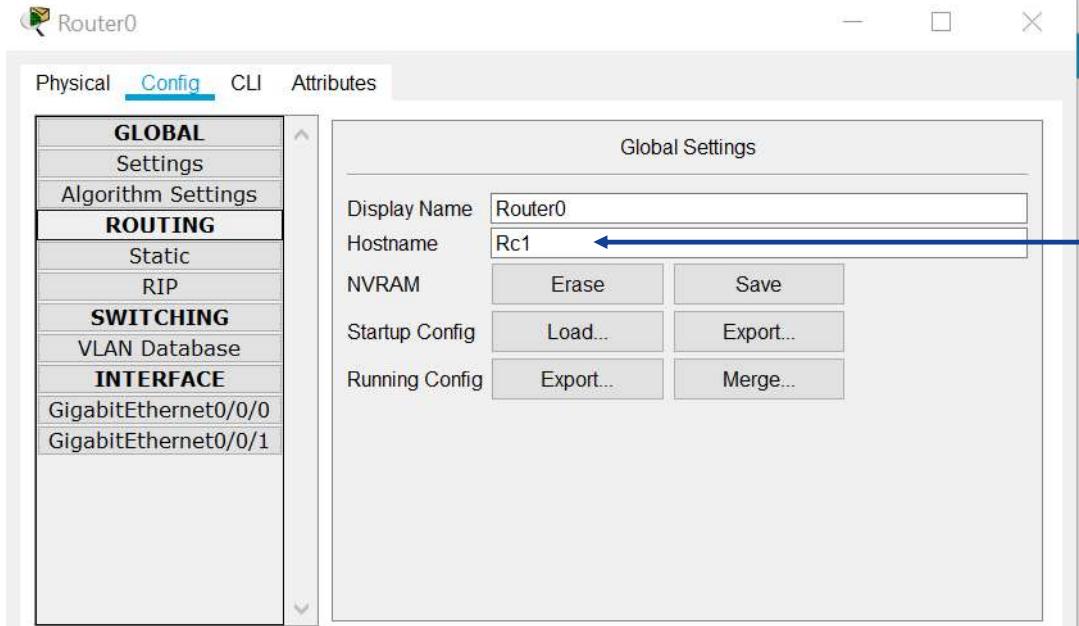
Line commands

password
login

Command mode	Descriere	Promptul
User Exec Mode	<ul style="list-style-type: none"> Permit un număr limitat de comenzi Defină ca și mod view-only. Accesăm nivelul următor prin comanda "enable" 	Switch> Router>
Privileged Exec Mode	<ul style="list-style-type: none"> Permite suita completă de monitorizare Accesăm nivelul următor cu comanda: "configure terminal" 	Switch# Router#
Global Configuration Mode	<ul style="list-style-type: none"> Permite modul de configurare globală 	Router(config)#

Packet tracer

Configurarea de bază – numele device-ului



H Hostname : Rc1

Configurând în modul graphic vedem în partea de jos comenzi corespunzătoare

Equivalent IOS Commands

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Rc1
Rc1(config)#
Rc1(config)#
Rc1(config)#
```

Comenzile corespunzătoare:

Enable -> trecem în modul privilegat
Configure terminal -> trecem în modul de configurare
Hostname Rc1 -> dăm numele device-ului

Configurarea de bază - interfețele

Router0

Physical Config CLI Attributes

GLOBAL	
Settings	
Algorithm Settings	
ROUTING	
Static	
RIP	
SWITCHING	
VLAN Database	
INTERFACE	
GigabitEthernet0/0/0	
GigabitEthernet0/0/1	
GigabitEthernet0/0/0	
Port Status	<input checked="" type="checkbox"/> On
Bandwidth	<input checked="" type="radio"/> 1000 Mbps <input type="radio"/> 100 Mbps <input type="radio"/> 10 Mbps <input checked="" type="checkbox"/> Auto
Duplex	<input checked="" type="radio"/> Half Duplex <input type="radio"/> Full Duplex <input checked="" type="checkbox"/> Auto
MAC Address	000B.BE60.1801
IP Configuration	
IP Address	192.168.10.1
Subnet Mask	255.255.255.0
Tx Ring Limit	10

Equivalent IOS Commands

```
Rc1(config)#
Rc1(config)#
Rc1(config)#interface GigabitEthernet0/0/0
Rc1(config-if)#ip address 192.168.10.1 255.255.255.0
Rc1(config-if)#ip address 192.168.10.1 255.255.255.0
Rc1(config-if)#no shutdown
Rc1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up
```

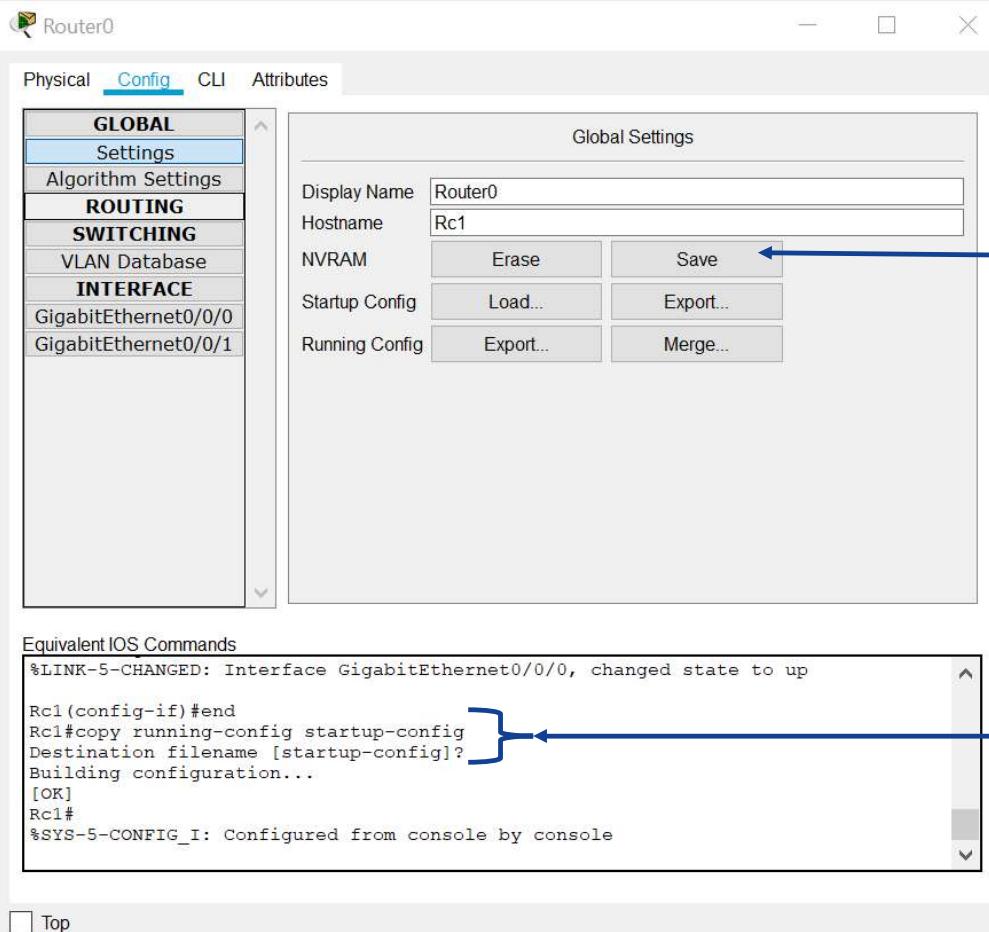
Interface : on, IP and netmask

Configurând în modul graphic vedem în partea de jos comenzi corespunzătoare

Comenzi corespunzătoare:
 Interface GigabitEthernet0/0/0 -> selecția interfeței
 Ip address 192.168.10.1 255.255.255.0 -> ip & netmask
 No shutdown -> face enable la interfață

Packet Tracer

Configurarea de bază - salvarea



save : NVRAM -> save

Configurând în modul graphic vedem în partea de jos comenzi corespunzătoare

Comenzi corespunzătoare:
End -> ieșim din modul de configurare
Copy running-config startup-config -> salvarea



That's all for today, see you next time!

Rețele de Calculatoare

Transportul datelor în rețelele de calculatoare

Sumar al laboratorului

1

Rutarea
Statică
Dinamică

2

Rutarea dinamică
RIP
OSPF

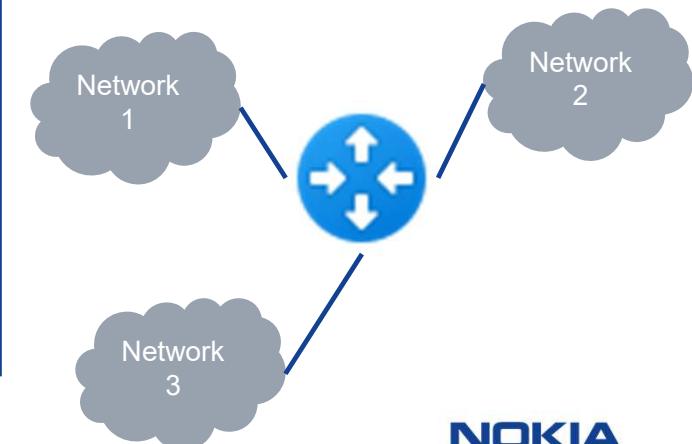
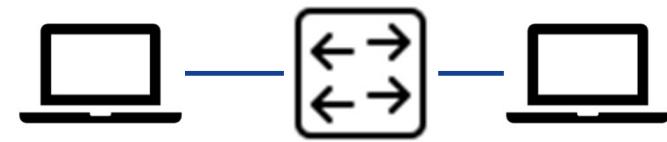
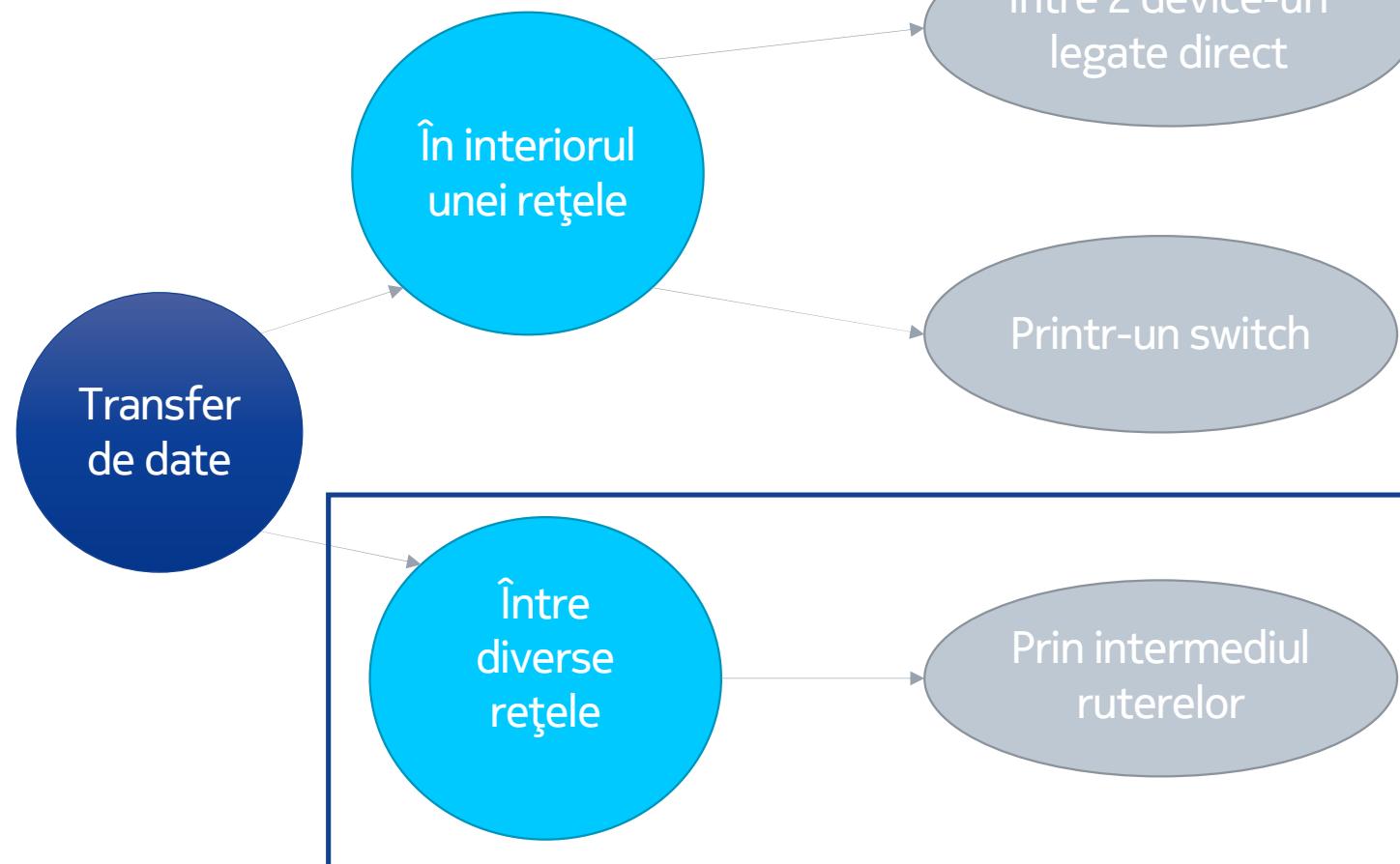
3

Verificarea conexiunii
Ping
Traceroute



Unde transportăm datele

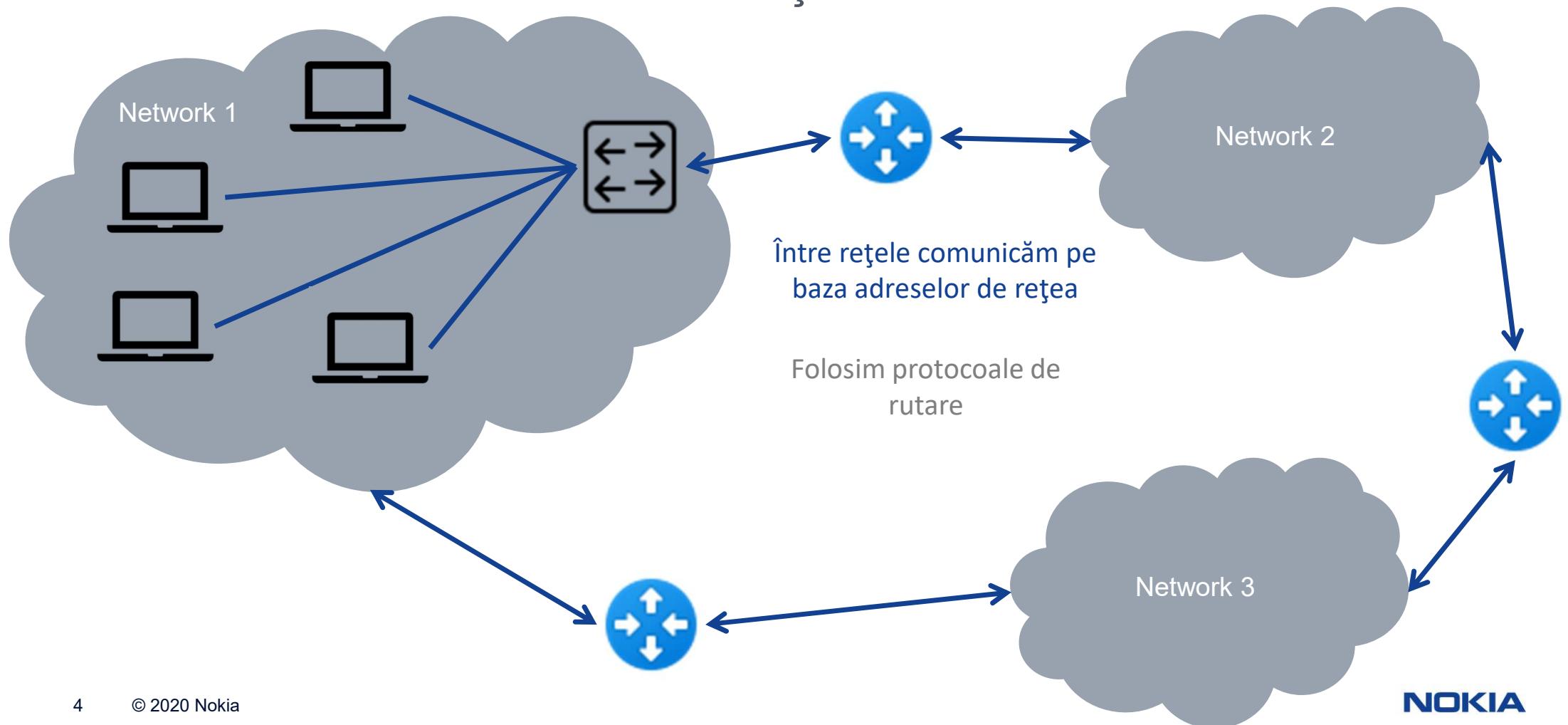
Rutere->rutare-> protocole



NOKIA

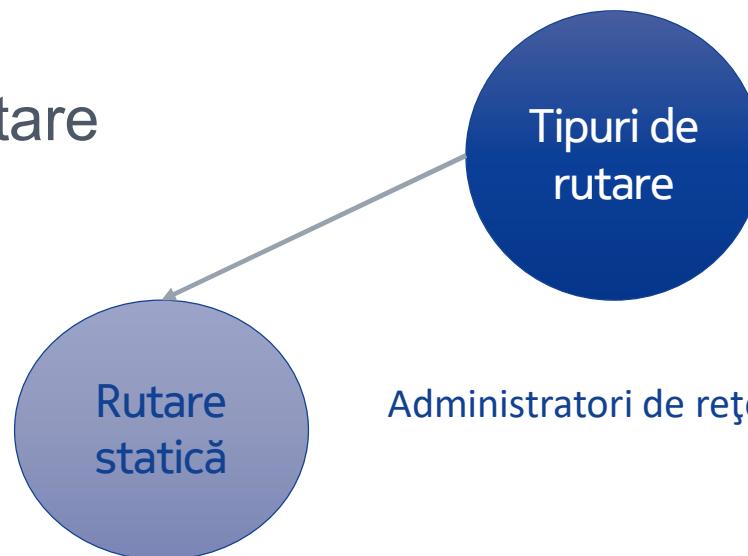
Cum transportăm datele

Protocole de rutare -> adrese de rețea



Rutarea

Tipuri de rutare



Administratori de rețea adaugă rutele manual

Avantaje:

- Putere de procesare scazută;
- Securitate crescută: doar administratorul poate adăuga echipamente în rețea
- Nu folosește lătime de bandă suplimentară

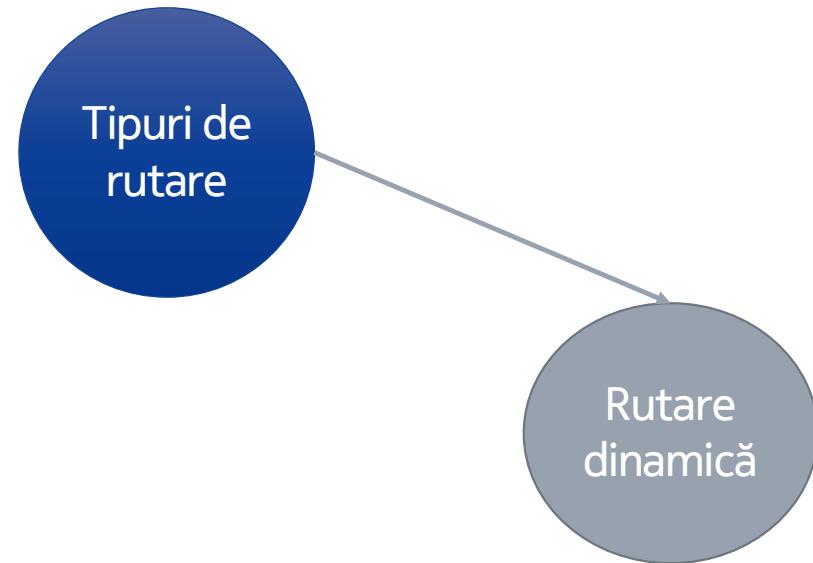
Dezavantaje:

- Pentru rețele mari, administratorul trebuie să creeze un model ierarhic pentru a menține controlul
- Necesită cunoasterea în prealabil a rețelei

Rutarea

Tipuri de rutare

Ruterele își transmit singure tabelele de rutare



Avantaje:

- Ușor de configurat
- Eficiență crescută în selectarea drumului cel mai bun

Dezavantaje:

- Consum ridicat de lătime de bandă
- Securitate scazută

Tipuri de rutare

Algoritmi
dinamică

în rutarea

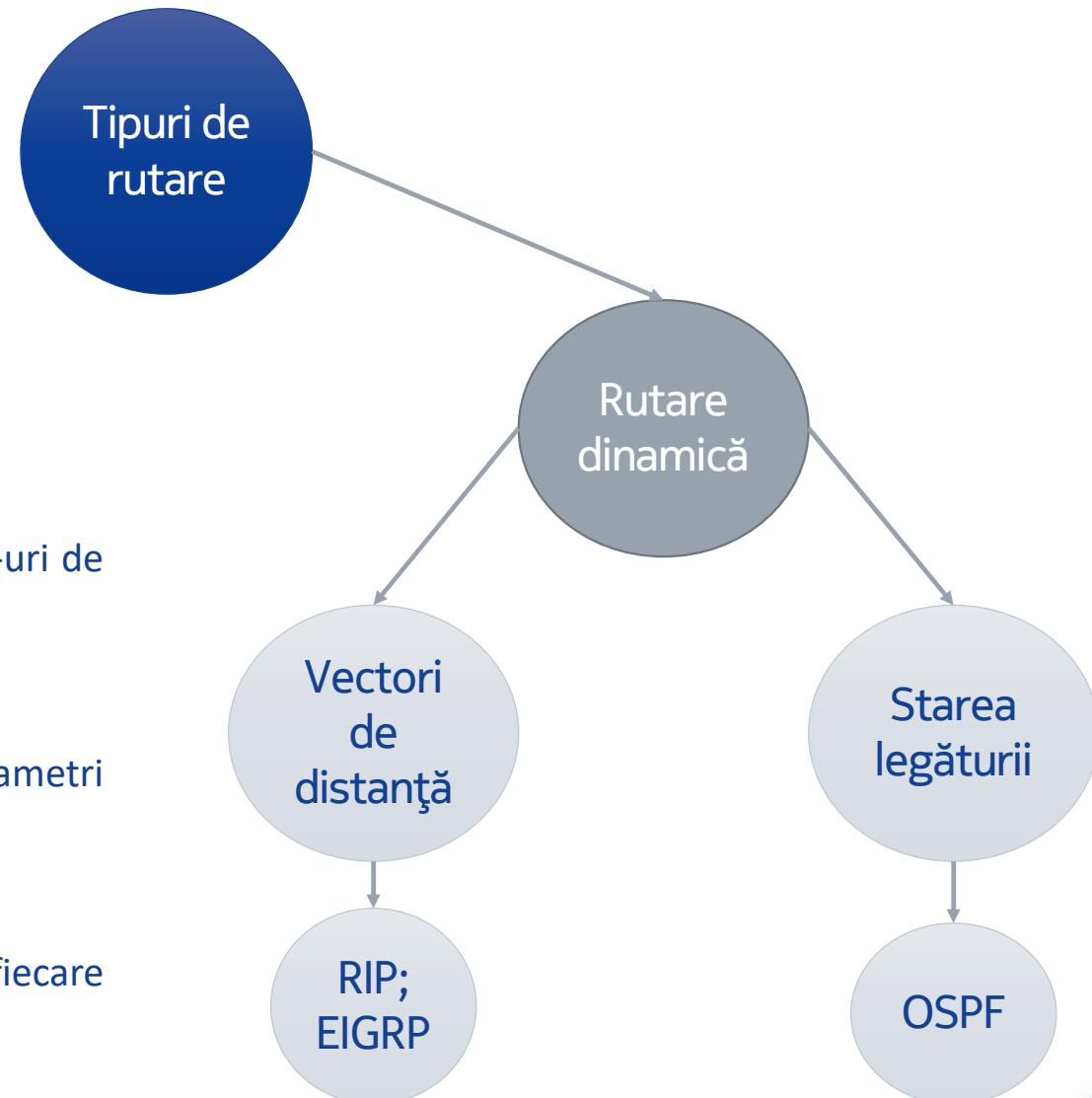
Vectori de distanță:

Ține cont de numărul de hop-uri de la sursă la destinație

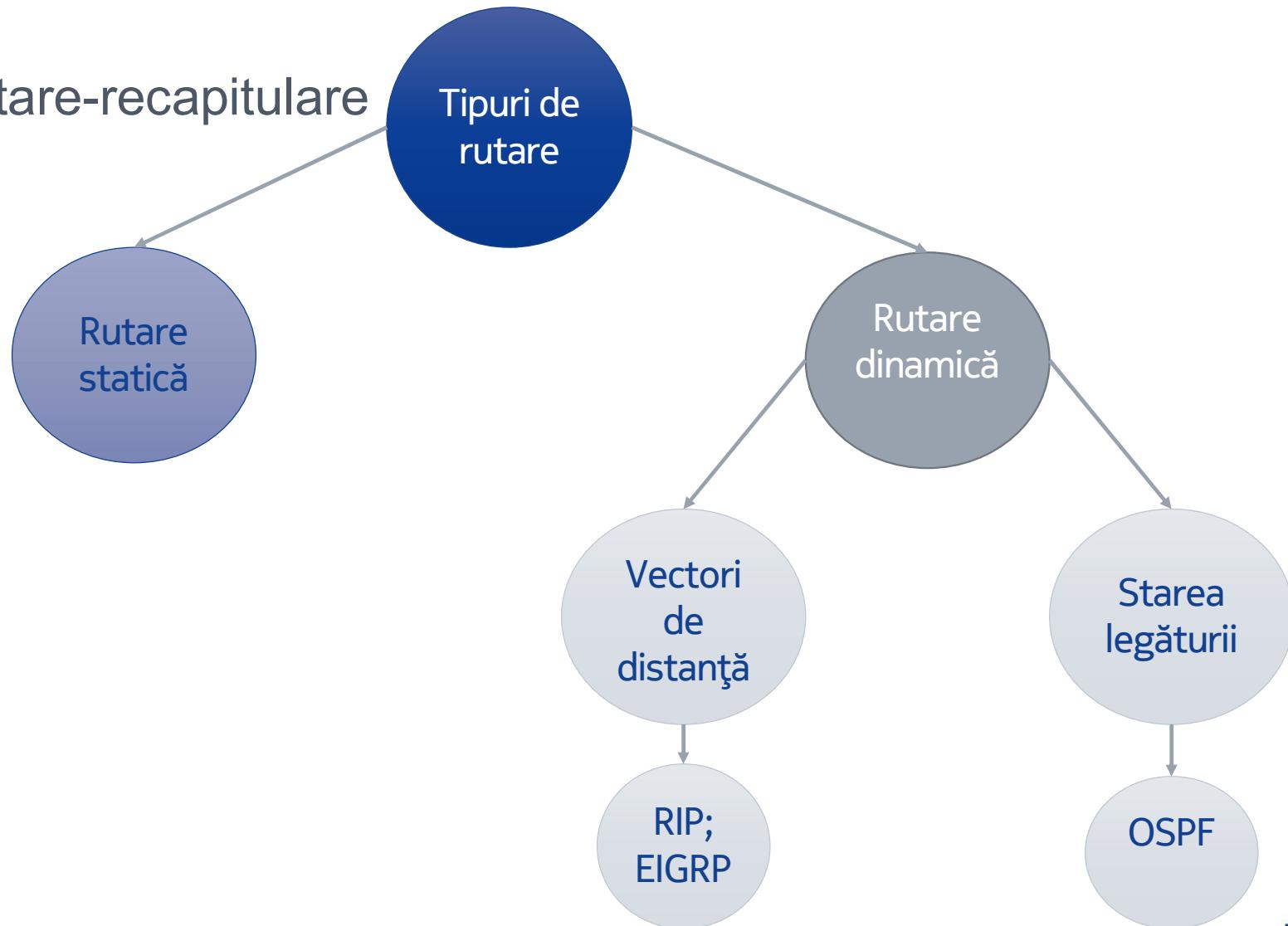
Starea legăturii (Link-state):

Ține cont de o serie de parametri printre care:

- Numărul de hop-uri;
- Încărcarea rețelei;
- Viteza de transfer pe fiecare tronson;
- Etc.

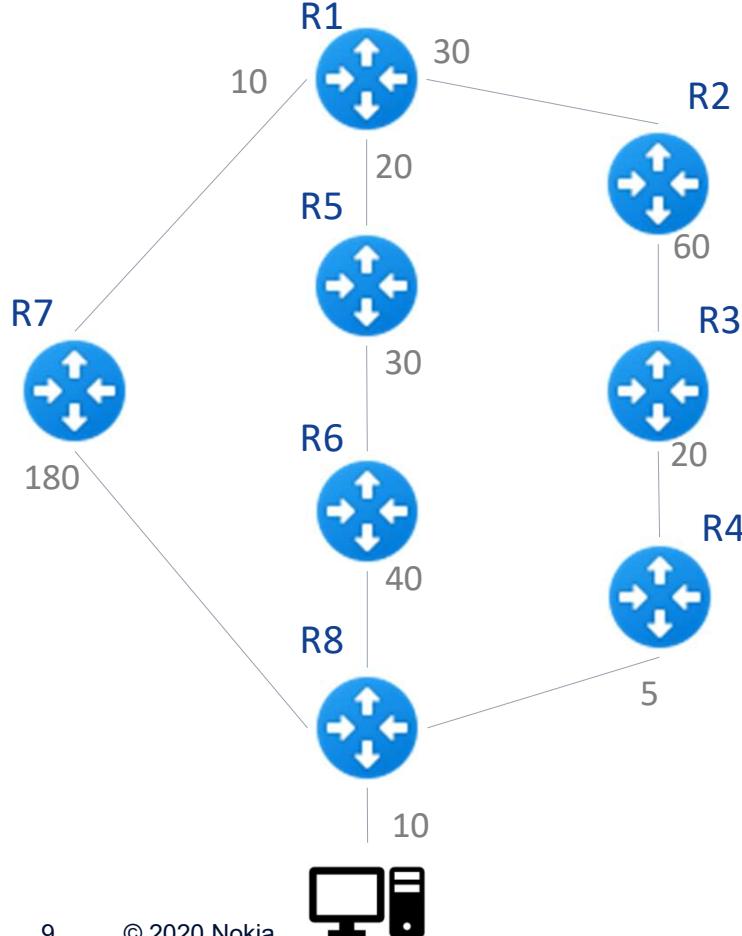


Tipuri de rutare-recapitulare



Rutare dinamică

Rip vs OSPF



RIP – Routing Information Protocol

OSPF – Open Shortest Path First

RIP – ia în calcul doar numărul de hop-uri de la sursă la destinație

OSPF – ia în calcul un “cost” al legăturii ce ține cont de mai mulți parametri (trecute cu gri în figură)

Ruta	Localizare	Cost cumulat OSPF
R1-R7-R8	Stanga	$10+180+10=200$
R1-R5-R6-R8	Mijloc	$20+30+40+10=100$
R1-R2-R3-R4-R8	Dreapta	$30+60+20+5+10=125$

Astfel:

- Dacă am folosi protocolul RIP am parcurge calea din stânga
- Dacă folosim OSPF folosim calea din mijloc

Sursa:

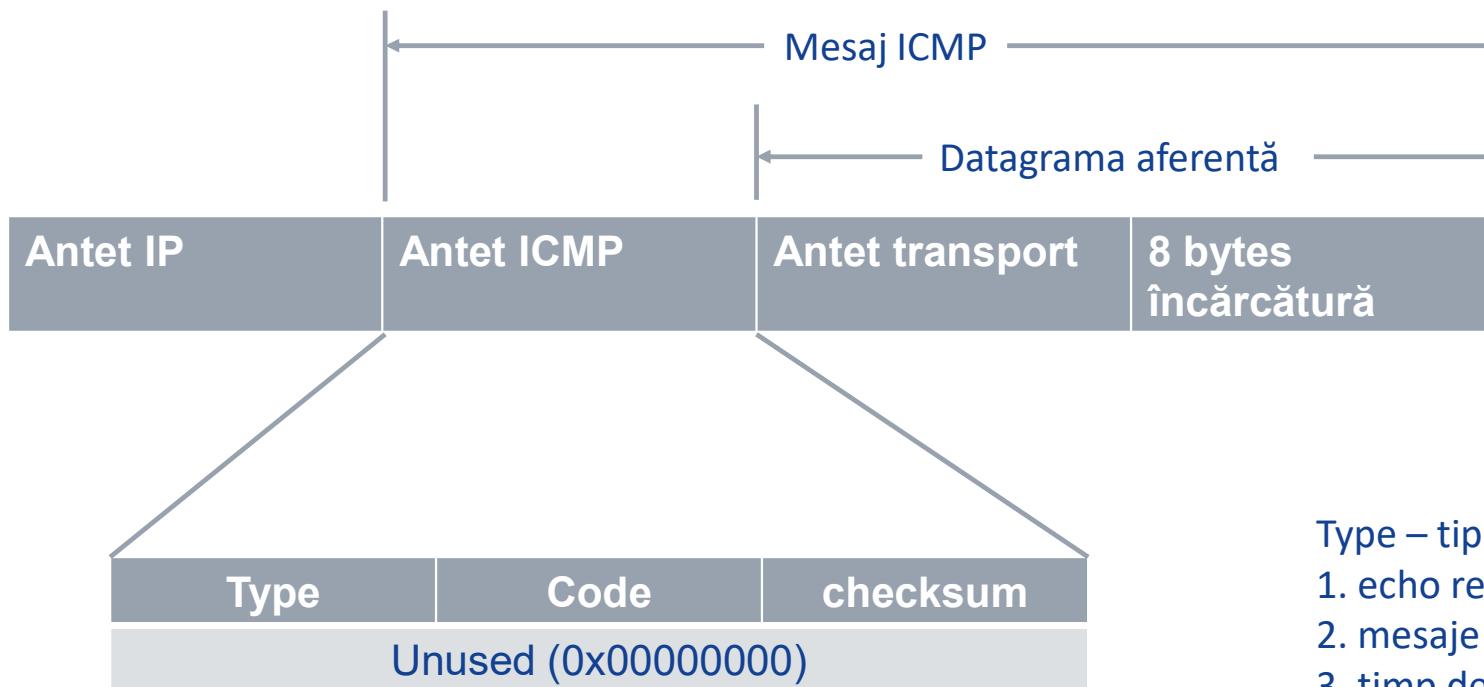
<https://www.ciscopress.com/articles/article.asp?p=2262897&seqNum=5>

Verificarea conexiunii

Protocolul ICMP

ICMP – Internet Control Message Protocol

Protocol în subordinea stratului
Internet din stivă TCP/IP



Type – tipul mesajului transmis:
1. echo request sau reply (exemplu ping)
2. mesaje între rutere,
3. timp depășit (TTL-ul depășit în tranzitii)

Verificarea conexiunii

Comanda Ping

ping *ip-address* - rolul acestei comenzi este de a determina dacă o adresă IP indicată este accesibilă sau nu.

Comanda ping trimite un pachet (echo request packet) către adresa IP specificată și așteaptă un răspuns (echo reply).

```
C:\Users\cmisici>ping 216.58.207.132

Pinging 216.58.207.132 with 32 bytes of data:
Reply from 216.58.207.132: bytes=32 time=43ms TTL=53
Reply from 216.58.207.132: bytes=32 time=42ms TTL=53
Reply from 216.58.207.132: bytes=32 time=43ms TTL=53
Reply from 216.58.207.132: bytes=32 time=48ms TTL=53

Ping statistics for 216.58.207.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 42ms, Maximum = 48ms, Average = 44ms
```

```
C:\Users\cmisici>ping www.google.com

Pinging www.google.com [216.58.207.132] with 32 bytes of data:
Reply from 216.58.207.132: bytes=32 time=43ms TTL=53
Reply from 216.58.207.132: bytes=32 time=43ms TTL=53
Reply from 216.58.207.132: bytes=32 time=38ms TTL=53
Reply from 216.58.207.132: bytes=32 time=40ms TTL=53

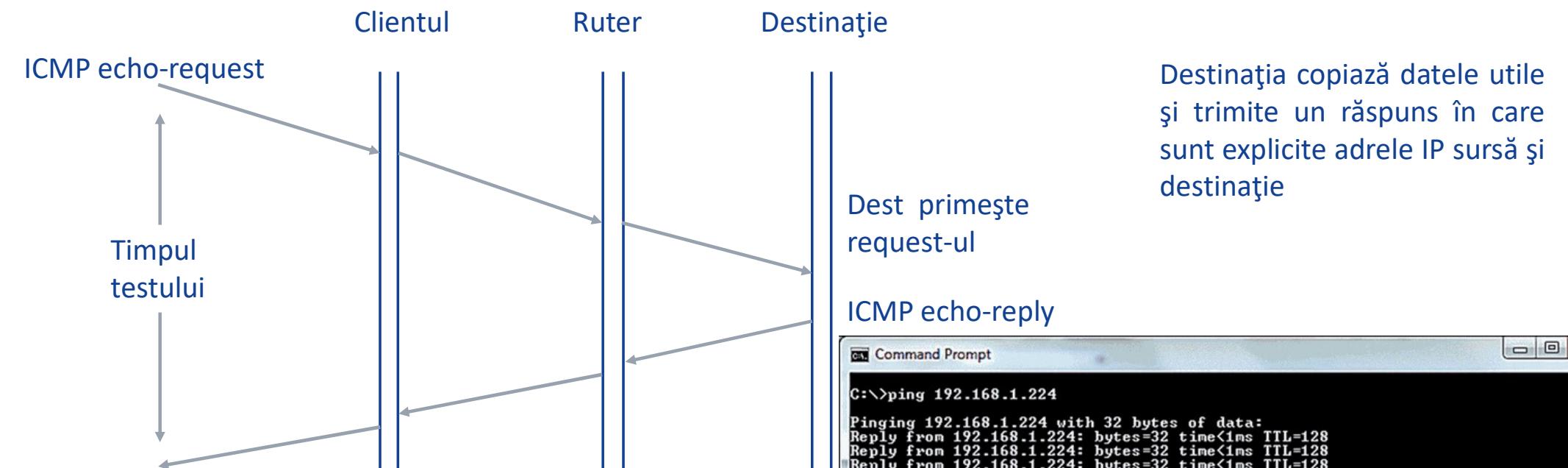
Ping statistics for 216.58.207.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 38ms, Maximum = 43ms, Average = 41ms
```

```
C:\>ping 10.10.10.10

Pinging 10.10.10.10 with 32 bytes of data:
Request timed out.
Request timed out.
```

Verificarea conexiunii

Comanda Ping



```
C:\>ping 192.168.1.224

Pinging 192.168.1.224 with 32 bytes of data:
Reply from 192.168.1.224: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.224:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Verificarea conexiunii

Comanda Ping

No	Day	Time	Source	Destination	Length	Protocol	Info
53	6.692	192.168.1.6		www.google.com	74	ICMP	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 54)
54	6.704		www.google.com	192.168.1.6	74	ICMP	Echo (ping) reply id=0x0001, seq=6/1536, ttl=55 (request in 53)
70	7.700	192.168.1.6		www.google.com	74	ICMP	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 71)
71	7.721		www.google.com	192.168.1.6	74	ICMP	Echo (ping) reply id=0x0001, seq=7/1792, ttl=55 (request in 70)
84	8.719	192.168.1.6		www.google.com	74	ICMP	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 85)
85	8.731		www.google.com	192.168.1.6	74	ICMP	Echo (ping) reply id=0x0001, seq=8/2048, ttl=55 (request in 84)
89	9.750	192.168.1.6		www.google.com	74	ICMP	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 90)
90	9.768		www.google.com	192.168.1.6	74	ICMP	Echo (ping) reply id=0x0001, seq=9/2304, ttl=55 (request in 89)

Comanda ping
văzută în Wireshark

Detalierea primului
cadru ping

```
> Frame 53: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: IntelCor_8c:ce:77 (5c:e0:c5:8c:ce:77), Dst: BestItWo_56:14:c0 (00:1e:a6:56:14:c0)
> Internet Protocol Version 4, Src: 192.168.1.6 (192.168.1.6), Dst: www.google.com (172.217.167.132)
  Internet Control Message Protocol
    Type: 8 (Echo (ping) request)  8 means ICMP request
    Code: 0  Always 0 for ICMP request and reply
    Checksum: 0x4d55 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)  We will match this identifier number with ICMP reply.1/256
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 6 (0x0006)  We will match this sequence number with ICMP
    Sequence number (LE): 1536 (0x0600)  reply for this ICMP request.
    [Response frame: 54]
    > Data (32 bytes)  Data 32 bytes
```

Verificarea conexiunii

Comanda Traceroute

- ***traceroute ip-address*** – returnează calea parcursă de pachetul transmis de la dispozitivul nostru până la destinație.
- **Exemplu:**

```
C:\>tracert google.com

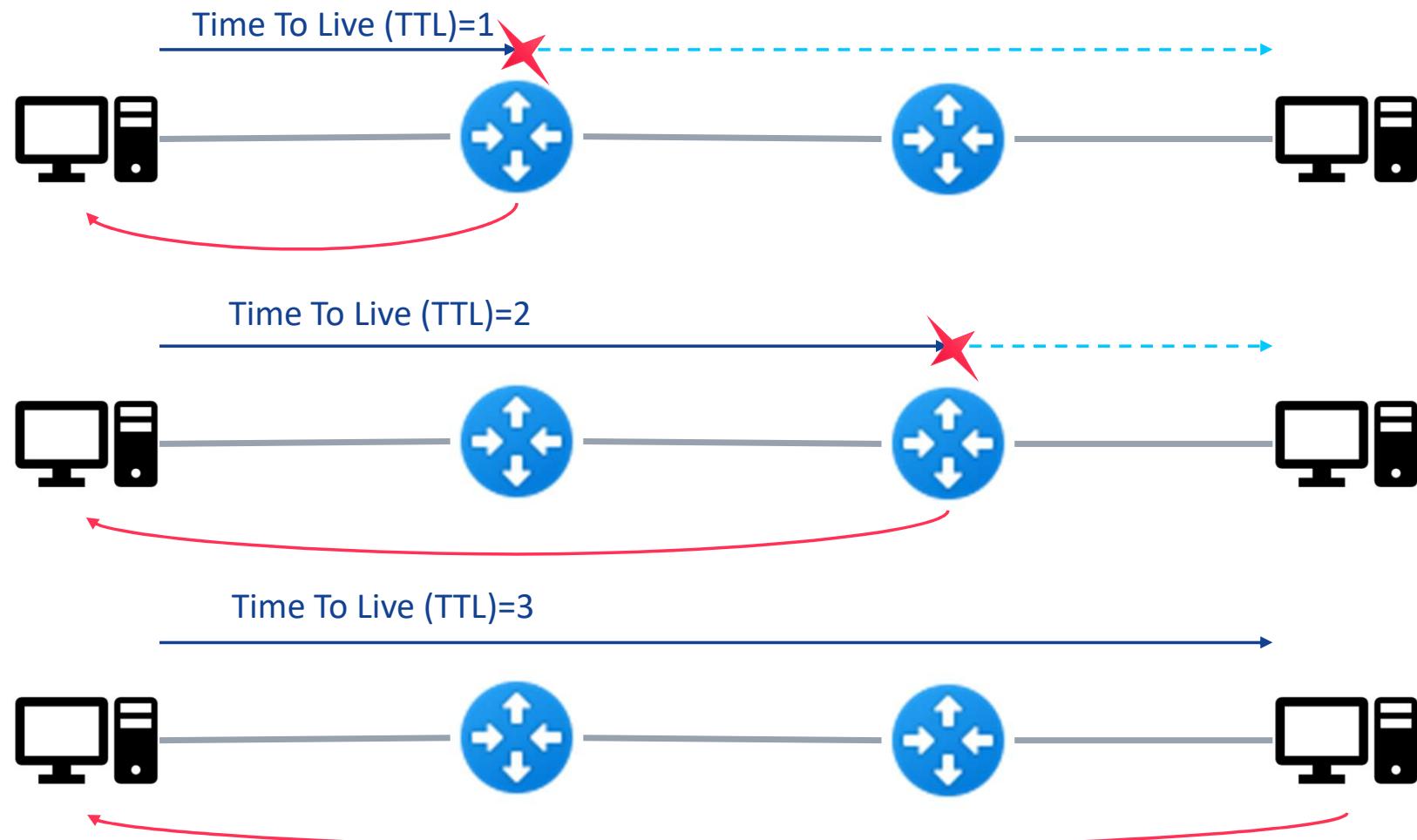
Tracing route to google.com [216.58.209.206]
over a maximum of 30 hops:

 1      4 ms      3 ms      3 ms  192.168.5.1 [192.168.5.1]
 2      *          *          *      Request timed out.
 3      6 ms      5 ms      8 ms  10.0.0.1 [10.0.0.1]
 4      5 ms      8 ms      6 ms  10.128.5.1 [10.128.5.1]
 5     16 ms     13 ms     13 ms  10.220.128.52 [10.220.128.52]
 6     13 ms     14 ms     13 ms  213-154-130-234.rdsnet.ro [213.154.130.234]
 7     12 ms     13 ms     12 ms  74.125.242.225
 8     12 ms     10 ms     11 ms  72.14.236.121
 9     12 ms     13 ms     13 ms  bud02s22-in-f206.1e100.net [216.58.209.206]

Trace complete.
```

Verificarea conexiunii

Comanda Traceroute





That's all for today, see you at the exam!

Rețele de calculatoare

1. Noțiuni introductive

Sebastian Fuicu

.Definiția unei rețele

.Ce se așteaptă de la o rețea?

.Moduri de transmisie a datelor

.Legături și noduri

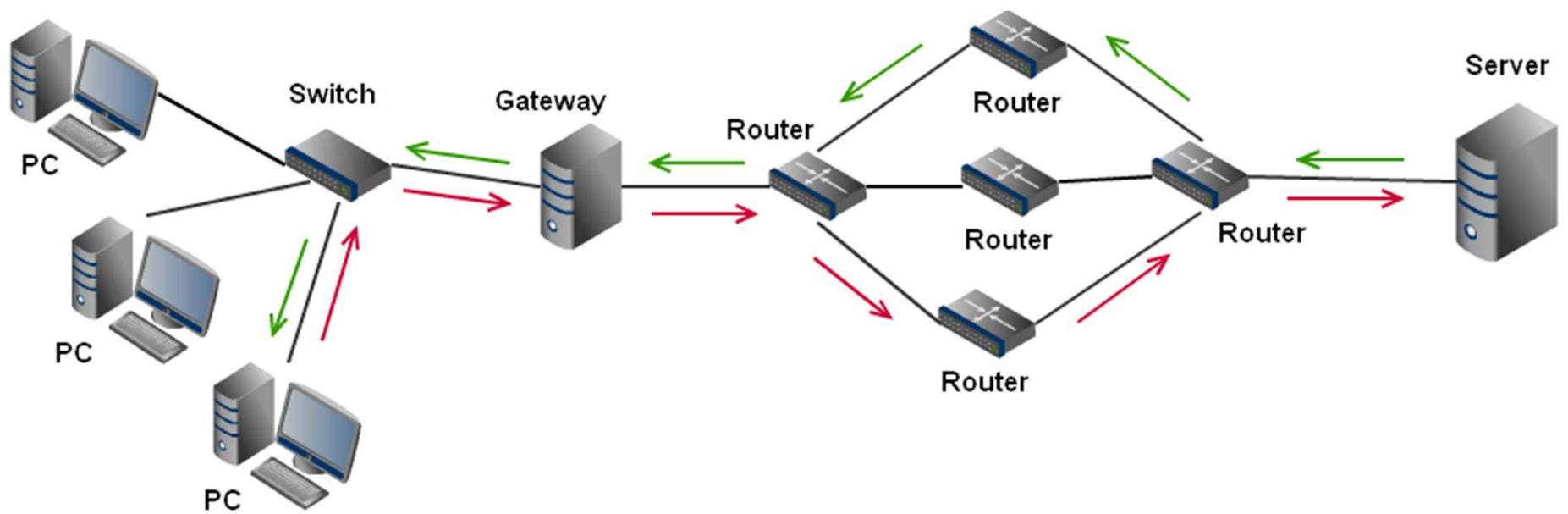
.Clasificări ale rețelelor

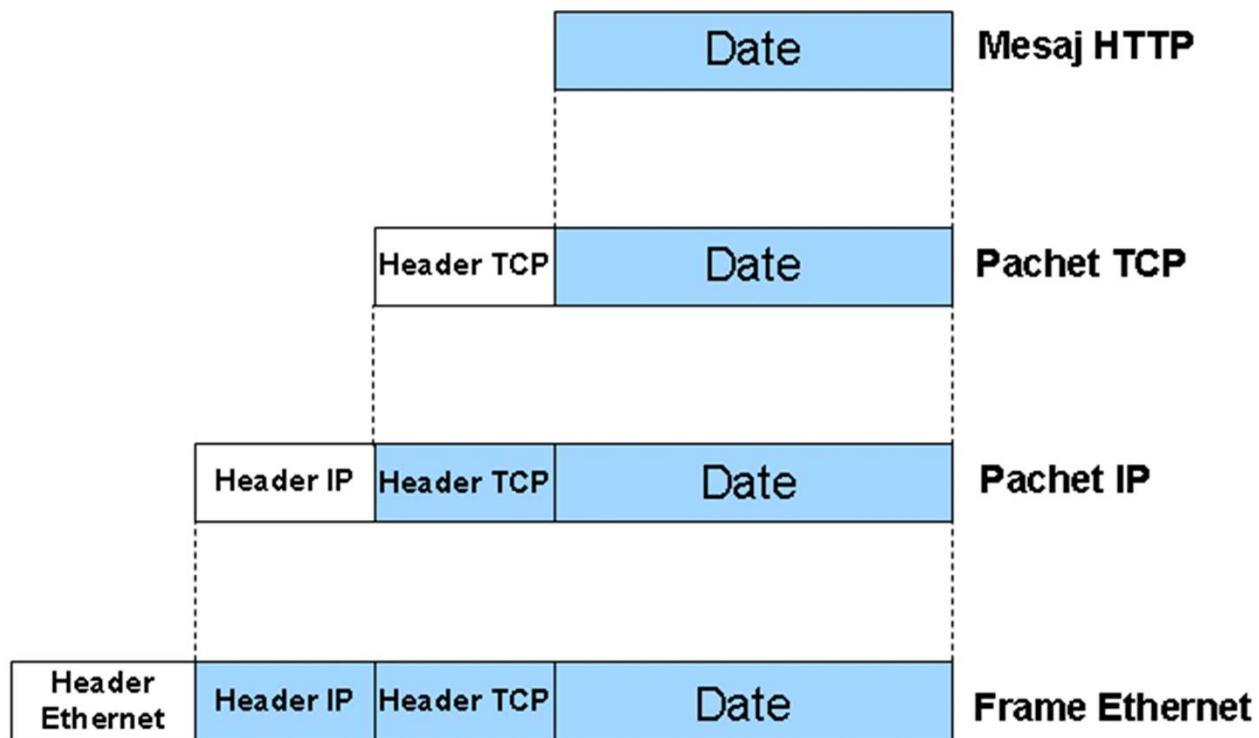
.Clasificarea transmisiilor

.Partajarea eficientă a resurselor

.Fiabilitatea transmiterii datelor

.Performanța unei rețele





Definiția unei rețele

O rețea este un sistem de linii interconectate:

- rețele pentru transportul persoanelor și al mărfurilor
 - transport rutier
 - transport feroviar
 - transport aerian
- rețele telefonice
- rețele de calculatoare

Rețea de calculatoare: grup de calculatoare și circuite de interconectare care funcționează într-un mod specific în scopul partajării resurselor și al schimbului de informații.

Definiția unei rețele

O rețea de calculatoare este compusă din două categorii de componente:

- componenta fizică**: reprezintă infrastructura rețelei.
- componenta logică**: reprezintă informația transportată de la sursă la destinație. Informația transportată poartă numele de “date”.

Ce se aşteaptă de la o rețea?

Există diverse perspective:

- cea a utilizatorul de rețea (pone în evidență serviciile necesare unei anume aplicații).
- cea a proiectantului (aspecte legate de costuri, utilizare eficientă).
- cea a furnizorului de servicii (se dorește un sistem ușor de administrat și gestionat).

Moduri de transmisie a datelor

Există 3 moduri de transmisie a datelor:

- **transmisie simplex**: este o transmisie într-un singur sens, de la transmițător spre receptor.
- **transmisie semiduplex**: se poate desfășura fie într-un sens fie în altul, dar nu simultan.
- **transmisie fulduplex**: permite transmisia simultană în ambele sensuri.

Legături și noduri

Legătură (link)

- mediul fizic (cablu cuaxial, fire torsadate, fibră optică, unde radio) care este folosit pentru conectarea a două calculatoare.

Noduri

- este denumirea folosită pentru calculatoarele care intră în componența unei rețele.

Clasificări ale rețelelor

După modul de interconectare al nodurilor din cadrul rețelei putem avea:

- Rețele cu difuzare
- Rețele punct la punct

Clasificări ale rețelelor

Rețele cu difuzare

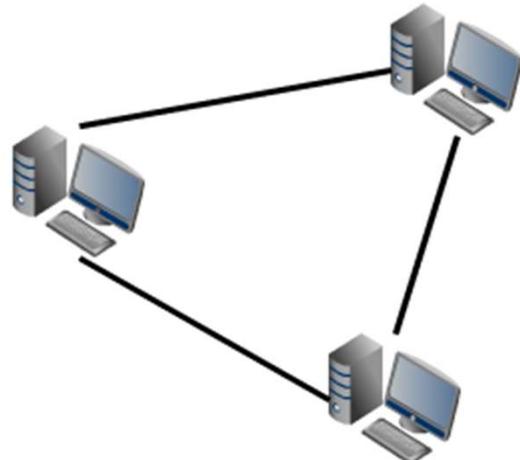
- au un singur canal de comunicații partajat de către toate nodurile din rețea.
- mesajele sunt primite de toate nodurile.
- destinatarul este specificat prin intermediul unui câmp de adresă.
- mașinile cărora nu le este destinat mesajul, îl ignoră.
- pe lângă adresarea individuală (unicast) permit adresarea unui pachet către toate mașinile (broadcast) sau către un grup de mașini (multicast) din rețea.



Clasificări ale rețelelor

Rețele punct la punct

- prezintă o multitudine de conexiuni între perechi de mașini individuale.
- pentru a ajunge la destinație, un mesaj poate tranzita prin mașini intermediare.
- sunt posibile trasee multiple.
- se impune folosirea algoritmilor de dirijare.



Clasificări ale rețelelor

O rețea de calculatoare funcționează ca rețea de comutare (switched network).

Rețelele cu comutare sunt de două tipuri

- Comutare de circuite (circuit-switched)
 - întâlnită în cazul sistemului telefonic.
- Comutare de pachete (packet-switched)
 - rețelele de calculatoare funcționează după acest principiu.

Clasificări ale rețelelor

Comutare de circuite

- Rețeaua alocă resurse hardware pentru desfășurarea comunicației.
- Se stabilește un circuit fizic între sursă și destinație format dintr-o secvență de legături.
- Resursele ramân alocate pe întreaga durată a desfășurării comunicației și nu pot fi partajate.

Clasificări ale rețelelor

Pașii parcursi la o comutare de circuite

- Stabilirea circuitului
- Transmisia datelor
- Deconectarea circuitului

Clasificări ale rețelelor

Comutare de circuite

Avantaje:

- Parametrii comunicației sunt garanți.

Dezavantaje:

- Scalabilitate redusă: necesită număr mare de conexiuni fizice.
- Folosirea ineficientă a resurselor rețelei.

Clasificări ale rețelelor

Comutare de pachete

- Transferul de date se realizează printr-o succesiune de mesaje.
- Mesajele sunt complet independente unele de altele.
- Nu se alocă resurse fizice pentru un anumit canal de comunicație.
- Resursele rețelei sunt partajate în comun de către toate nodurile care comunică.
- Rețelele cu comutare de pachete folosesc o strategie numită **store and forward**. Aceasta presupune că mesajele sunt stocate în nodurile intermediare înainte de a fi transmise mai departe către destinație. Nodurile intermediare fac verificări asupra integrității datelor, înainte de a le trimite mai departe. Fiecare nod intermediar trebuie să stabilească care este următorul nod intermediar spre care va fi transmis mesajul în drumul său către destinație.

Clasificări ale rețelelor

Comutare de pachete

Avantaje:

- Alocarea eficientă a resurselor
- Posibilitatea prioritizării mesajelor
- Scalabilitate crescută

Dezavantaje:

- Întârzieri mai mari la transmiterea mesajelor

Clasificări ale rețelelor

După dimensiunea rețelei și aria geografică ocupată rețelele pot fi clasificate în felul următor:

- Rețele locale (Local Area Networks - LAN)
- Rețele metropolitane (Metropolitan Area Network - MAN)
- Rețele extinse (Wide Area Network – WAN)

Dimensiunea rețelei are implicații asupra tehnologiei de bază folosite, un factor important fiind timpul de propagare al datelor de la un capăt în altul al rețelei.

Clasificarea transmisiilor

După modul de adresare al destinatarului putem avea transmisi:

- unicast (mesajul este destinat unui singur nod din rețea)
- multicast (mesajul este destinat unui grup de noduri)
- broadcast (mesajul este destinat tuturor nodurilor din rețea)

Partajarea eficientă a resurselor

Cum partajează nodurile rețelei același canal fizic, atunci când doresc să comunice simultan?

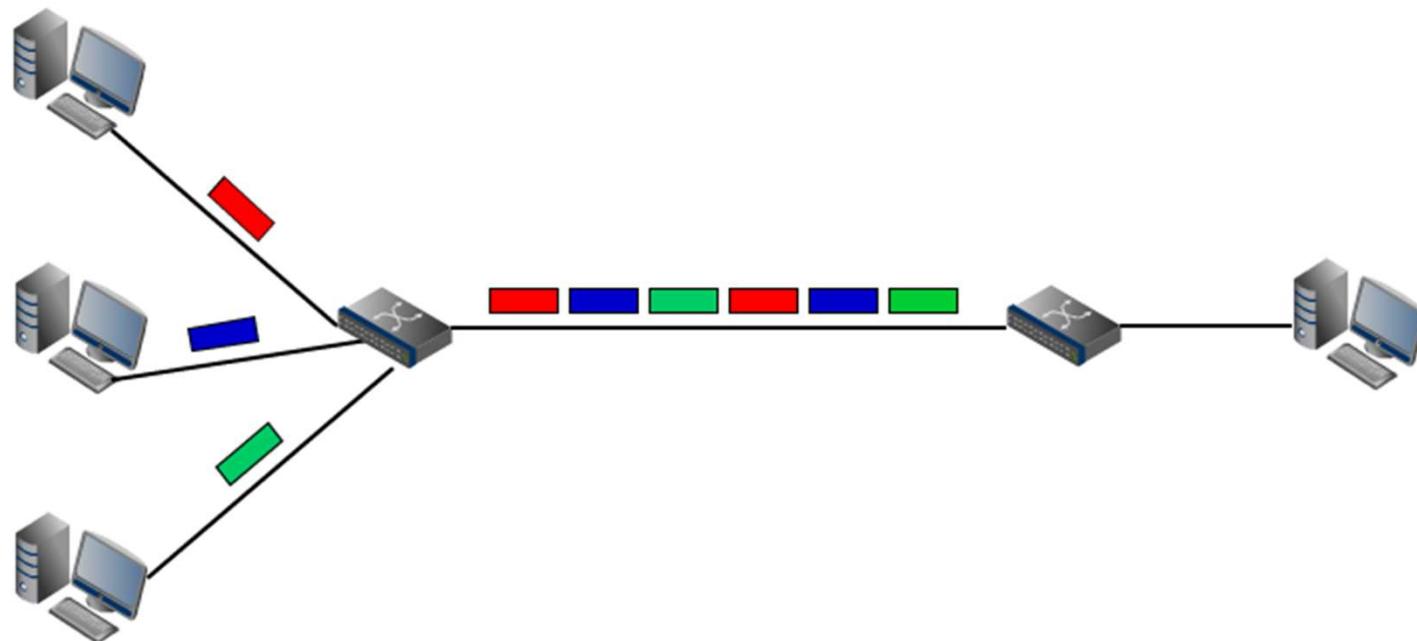
- Se recurge la multiplexare.

Tehnici de multiplexare:

- multiplexare în timp (Time Division Multiplexing – TDM)
- multiplexare în frecvență (Frequency Division Multiplexing - FDM)

Partajarea eficientă a resurselor

În cazul rețelelor de calculatoare este folosită multiplexarea în timp.



Fiabilitatea transmiterii datelor

Există trei clase de erori apărute la transmisea datelor:

- Erori apărute la nivelul conexiunii fizice - afectează unul sau mai mulți biți (burst error).
- Incidente care se produc la nivel de pachet – rețeaua pierde un întreg pachet.
- Întreruperea unei conexiuni sau blocarea unui nod.

Performanța unei rețele

Există doi parametrii importanți care exprimă performanța unei rețele

- Rată de transfer (debit)
- Latență (întârzierea)

Performanța unei rețele

Rata de transfer

- reprezintă numărul de biți care pot fi transmiși pe un canal în unitate de timp
- se măsoară în bps (bits per second)

Performanța unei rețele

Latență

- reprezintă intervalul de timp necesar unui bit pentru a se propaga de la sursă la destinație
- se măsoară în secunde

Performanța unei rețele

Latența are 3 componente

Latență = Propagare + Transmitere + Coadă

- Propagare = Distanță / Viteza luminii**

Distanță: reprezintă lungimea totală a mediului de transmisie.

Viteza de propagare a luminii în diverse medii este diferită:

$c = 3 \cdot 10^8 \text{ m/s}$ (viteza de propagare prin vid)

$c = 2.3 \cdot 10^8 \text{ m/s}$ (viteza de propagare prin cablu)

$c = 2 \cdot 10^8 \text{ m/s}$ (viteza de propagare prin fibra optică)

- Transmitere = Dimensiune / Rată de transfer**

Dimensiune: reprezintă cantitatea de date care trebuie transmisă.

- Coadă:** reprezintă întârzierea introdusă de nodurile intermediare (routerele). Un router nu poate procesa simultan toate pachetele care ajung la el și astfel le introduce într-o coadă de așteptare.

Performanța unei rețele

Un canal de comunicație se comportă ca un element de memorare

- Canalele stochează temporar un anumit număr de biți de informație, ce au fost deja transmiși de către sursă, dar nu au ajuns încă la destinație.
- Produsul dintre latență și rata de transfer corespunde numărului de biți ce vor fi transmiși înainte ca primul bit să ajungă la destinatar.

Latență * Rata de transfer

Performanța unei rețele

Cât durează transmisia unui bloc de 1 Ko la o rată de transfer de 10 Mbps?

Transmitere = Dimensiune / Rată de transfer

$$t = 1 \text{ ko} / 10 \text{ Mbps}$$

$$= 1 \times 1024 \times 8 \text{ biti} / 10 \times 10^6 \text{ bps}$$

$$= 819,2 \times 10^{-6} \text{ s}$$

$$= 0,82 \text{ ms}$$

Performanța unei rețele

Cât durează transmisia unui bit pe o legătură cu rata de transfer 100 Mbps?

Durata = 1 / Rată de transfer

$$t = 1 / 100 \text{ Mbps}$$

$$= 0,01 \times 10^{-6} \text{ s}$$

$$= 0,01 \mu\text{s}$$

Performanța unei rețele

Care este timpul de propagare necesar parcurgerii de către un bit a unei legături din fibră optică având lungimea de 2 km?

Propagare = Distanță / Viteza luminii

$$t = 2 \text{ km} / 2 \times 10^8 \text{ m/s}$$

$$= 10^{-5} \text{ s}$$

$$= 10 \text{ } \mu\text{s}$$

Care este latența unui pachet de 1 ko transmis prin cablu de cupru la distanță de 100 km cu o rată de transfer de 10 Mbps?

Latenta = Propagare + Transmitere

Propagare = Distanța / Viteza luminii

$$tp = 100 \text{ km} / 2,3 \times 10^8 \text{ m/s}$$

$$= 1/2,3 \times 10^{-3} \text{ s}$$

$$= 0,43 \text{ ms}$$

Transmitere = Dimensiune / Rată de transfer

$$tt = 1 \text{ ko} / 10 \text{ Mbps}$$

$$= 1024 \times 8 \times 10^{-7} \text{ s}$$

$$= 0,82 \text{ ms}$$

Latenta = 1,25 ms

Performanța unei rețele

Ce cantitate de informații poate conține un canal cu o latență de 50 ms și o rată de transfer de 45 Mbps?

Cantitate = Latență x Rată de transfer

$$n = 50 \text{ ms} \times 45 \text{ Mbps}$$

$$= 50 \times 10^{-3} \times 45 \times 10^6 \text{ biti}$$

$$= 275 \text{ kBytes}$$

Rețele de calculatoare

Partea a 2-a

Sebastian Fuicu

- Arhitecturi de rețea
- Straturi și protocoale
- Încapsularea
- Modelul OSI (*Open System Interconnection*)
- Modelul Internet (Stiva TCP/IP)

Arhitecturi de rețea

- **Cerințe pentru proiectarea unei rețele**
 - să asigure o conectivitate generală.
 - să fie eficientă sub aspectul costurilor.
 - să fie corectă.
 - să fie robustă.
 - să asigure performanțe ridicate între un număr mare de calculatoare.
 - să răspunde la cerințele care vin din partea programelor de aplicații.
 - trebuie să evolueze pentru a se adapta la modificările care apar în tehnologia de bază.

Arhitecturi de rețea

- Pentru a simplifica problema proiectării s-au creat modele generale denumite ***arhitecturi de rețea***.
- ***Arhitecturile de rețea*** ghidează proiectarea și implementarea rețelelor de calculatoare.
- Există două tipuri importante de arhitecturi de rețea:
 - modelul OSI
 - modelul Internet

Straturi și protocoale

- Când complexitatea sistemului (a rețelei) este ridicată se preferă impărțirea acestuia în mai multe **nivele de abstractizare**.
- Prin **abstractizare** se urmărește:
 - definirea unui **model** care să poată sintetiza un anumit aspect important al sistemului.
 - înglobarea modelului într-un **obiect** care să furnizeze o **interfață** manipulabilă de către alte componente din sistem.
 - ascunderea detaliilor de implementare ale obiectului față de utilizatorii acestuia.

Straturi și protocoale

Layer $k+1$



Layer k



Layer $k-1$

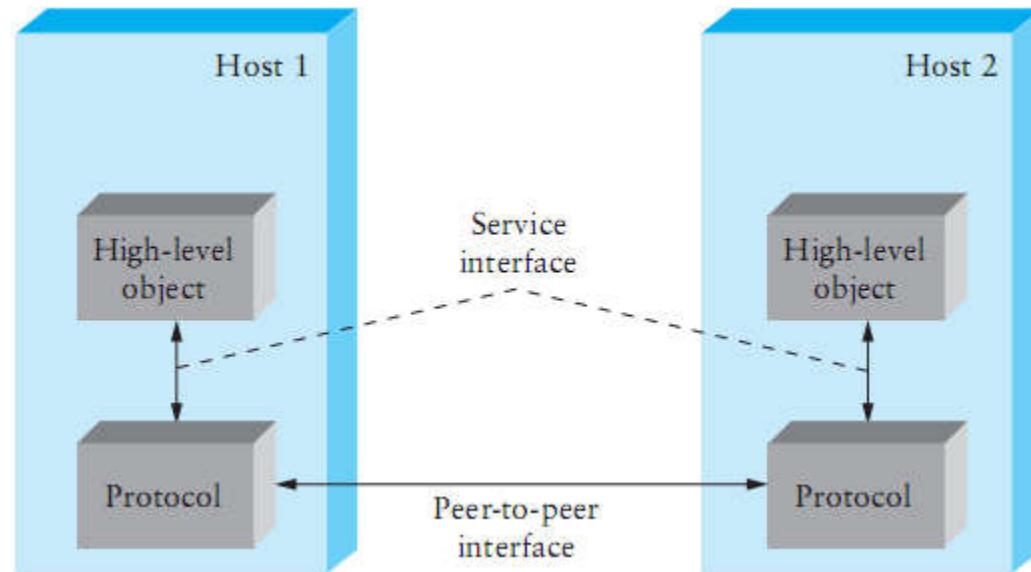
- **Abstractizările** conduc în mod natural la **stratificare** în cazul rețelelor de calculatoare
- Se pornește de la **serviciile** oferite de echipamentele de bază și se adaugă o **secvență de straturi**.
- Fiecare strat oferă un set de servicii mai ridicat și implicit mai abstract.
- Serviciile asigurate de straturile superioare sunt implementate în funcție de serviciile furnizate de straturile inferioare.

Straturi și protocoale

- Stratificarea oferă două facilități importante:
 - dezvoltarea unei rețele se descompune în componente mai ușor de controlat.
 - se asigură o proiectare modulară.
- Obiectele abstracte care alcătuiesc straturile unei rețele se numesc **protocole**.

Straturi și protocoale

- Un **protocol** asigură un **serviciu de comunicație** pe care obiectele de nivel mai ridicat le folosesc pentru a realiza schimbul de mesaje.
- Fiecare protocol definește două **interfețe** distincte:
 - **Service interface**: reprezintă interfața cu celelalte obiecte din cadrul stivei.
 - **Peer interface**: reprezintă interfața către omologul său de pe același nivel dar de pe un alt sistem.



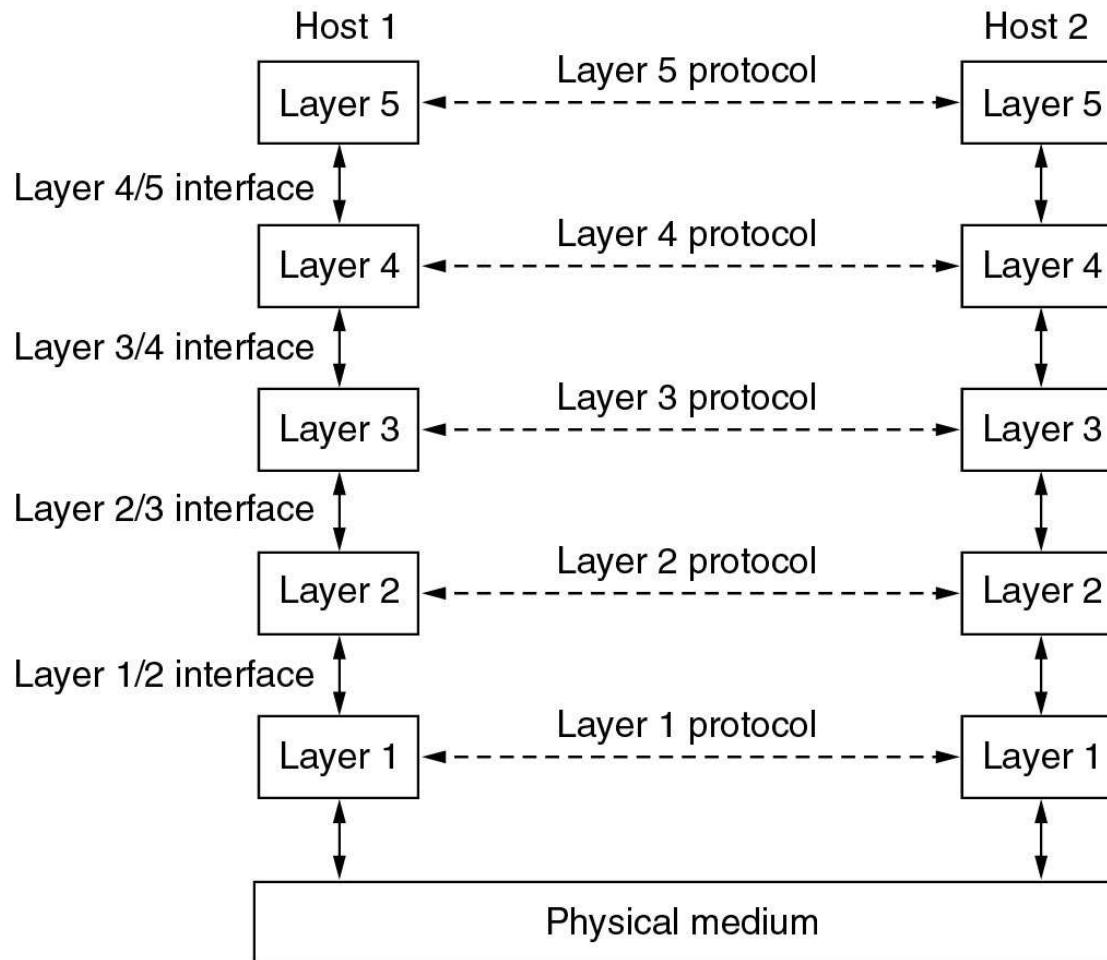
Straturi și protocoale

- Un protocol definește:
 - un serviciu de comunicații pe care îl exportă local.
 - un set de reguli în vederea implementării serviciului.
- Protocolele pot oferi două tipuri de servicii:
 - serviciu orientat pe conexiune (**connection oriented**): mesajele circulă folosind un circuit virtual (**virtual connection**)
 - serviciu fără conexiune: mesajele circulă independent unele față de altele
- Un circuit virtual poate oferi o conexiune sigură sau nesigură.

Straturi și protocoale

- Doar la nivelul hardware, echipamentele omoloage comunică direct unele cu altele.
- Comunicația peer-to-peer este indirectă:
 - fiecare protocol comunică indirect cu perechea sa de pe același nivel.
 - pentru a transmite un mesaj, acesta este transferat protocolului de pe nivelul inferior, care încearcă să livreze mesajul către perechea sa.

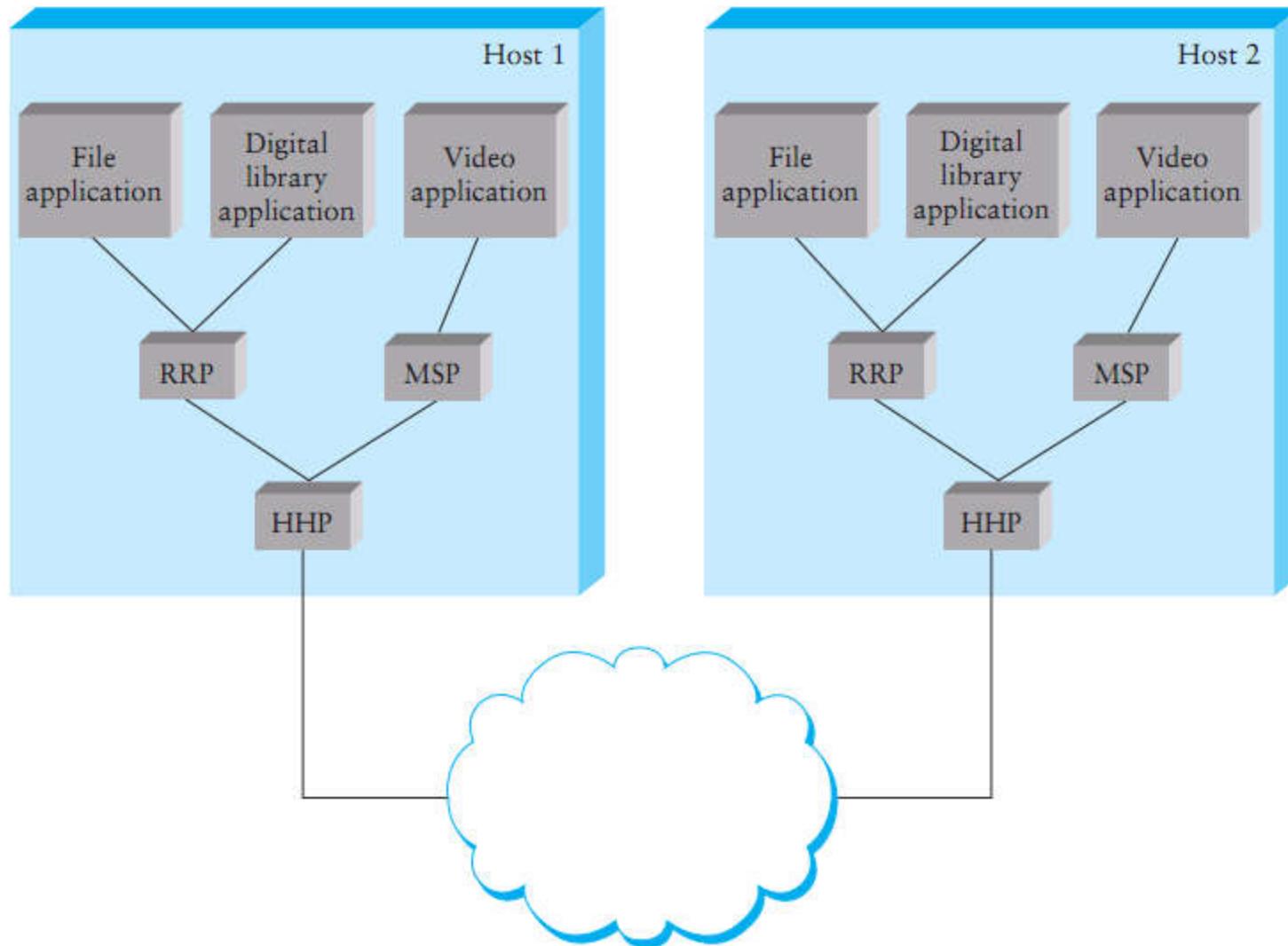
Straturi și protocoale



Straturi și protocoale

- Potențial, pot exista mai multe protocoale pe un anumit nivel, fiecare furnizând un serviciu de comunicații diferit.
- Suta de protocoale care alcătuiesc un sistem de rețele poate fi reprezentată sub forma unui graf de protocoale.
- În reprezentarea următoare este redat un graf ipotetic care conține următoarele protocoale:
 - RRP (Request/Reply Protocol)
 - MSP (Message Stream Protocol)
 - HHP (Host-to-Host Protocol)

Straturi și protocoale



Încapsularea

- Să luăm ca exemplu o aplicație care vrea să transmită un mesaj
 - Această aplicație face apel la serviciile oferite de protocolul RRP.
 - Protocolul care va transporta mesajul nu trebuie să cunoască natura informațiilor transportate.

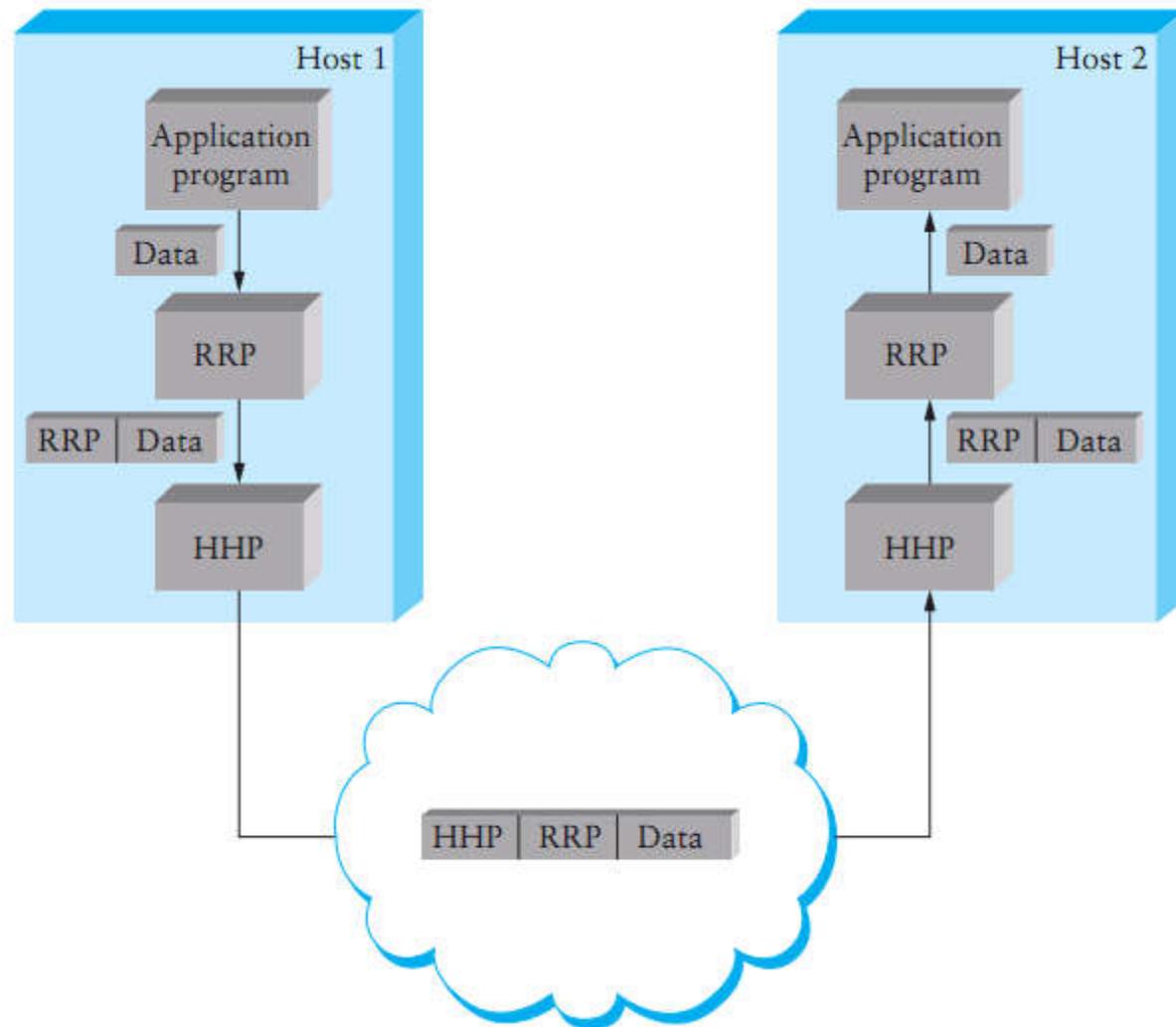
Încapsularea

- Când un protocol de pe un nivel, comunică cu perechea sa, trebuie să specifice într-un fel anumite informații de control.
 - Acest lucru se realizează prin anexarea unui **antet (header)** la mesajul care trebuie transmis.
 - Antetul este o structură de date de maxim câțiva zeci de octeți.
 - Se formează astfel un nou mesaj care conține antetul și corpul mesajului.
 - Noul mesaj poartă denumirea de pachet de date (**data packet**) sau cadru de date (**data frame**)

Încapsularea

- Spunem că mesajul pe care aplicația dorește să îl transmită a fost **încapsulat** într-un nou mesaj creat de protocolul RRP.
- Acest proces al încapsulării este repetat la fiecare nivel al grafului de protocoale.
- Protocolele implementează un canal logic de comunicație care poate fi folosit simultan de mai multe aplicații.
- Deci apare un proces de multiplexare la un capăt și de demultiplexare la celălalt capăt.

Încapsularea

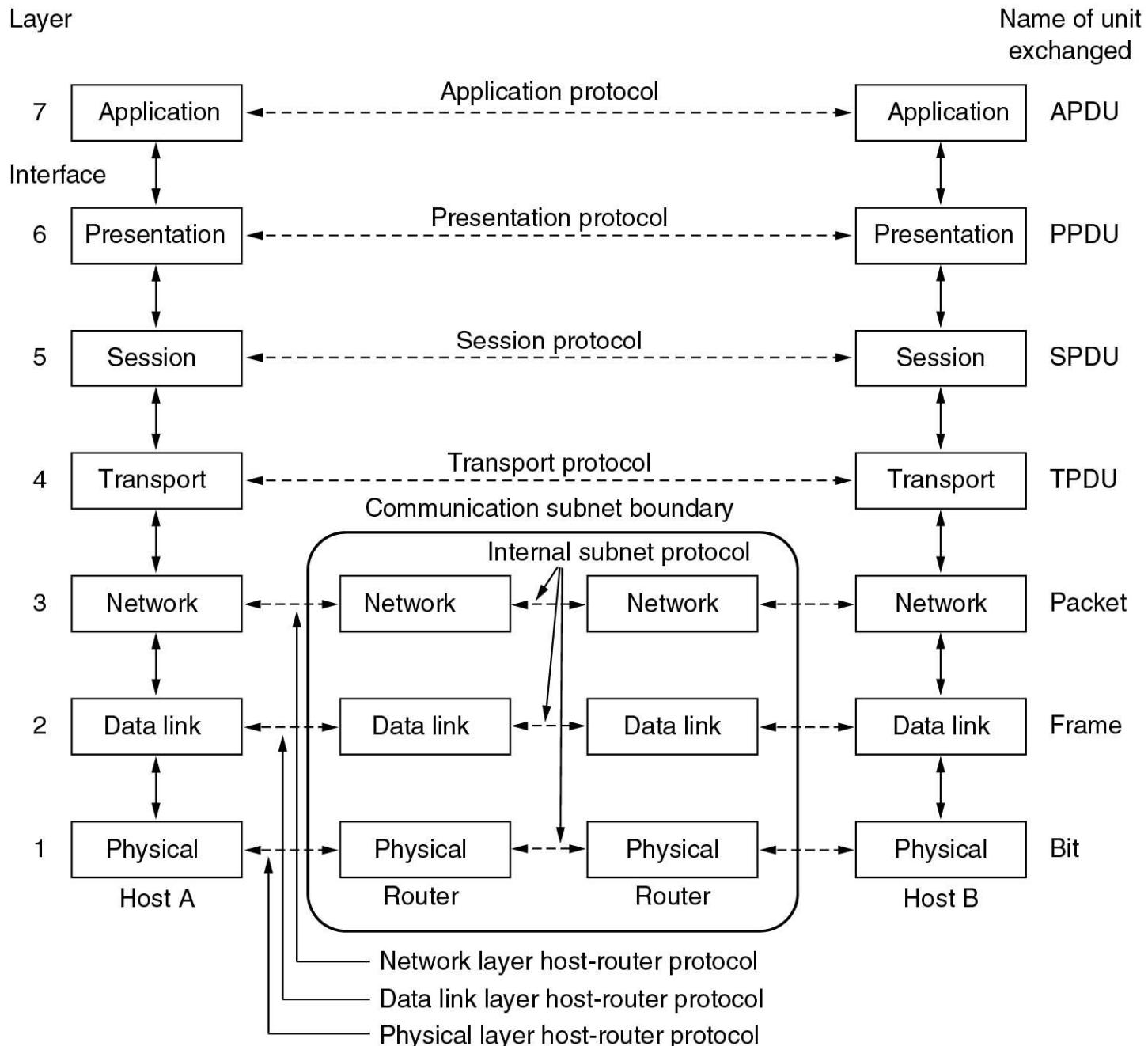


Modelul OSI (*Open System Interconnection*)

- Modelul a fost propus de către International Organization for Standardization.
- Scopul acestui set de standarde era să definească o modalitate uniformă de conectare a unor sisteme cu caracteristici diferite.
- Modelul OSI prezintă la nivel de principiu serviciile care trebuie asociate fiecărui nivel neimpunând însă soluții concrete de implementare a acestora.
- Modelul OSI prezintă 7 nivele.



Modelul OSI (*Open System Interconnection*)



Modelul OSI (*Open System Interconnection*)

- Nivelul Fizic
 - Serviciul pus la dispoziție este acela de a transporta un sir de biți de la un capăt la celălalt al unei legături fizice.
 - Legătura fizică poate fi realizată prin fire metalice, fibre optice sau canale radio.

Modelul OSI (*Open System Interconnection*)

- **Nivelul Legătură de Date**
 - Când Nivelul Fizic transportă date, acestea pot fi afectate de erori.
 - Pentru a realiza o comunicație sigură între două puncte a fost necesar să se introducă Nivelul Legătură de Date.
 - Aceasta va fi responsabil cu detecția și eventual corecția erorilor care pot apărea pe Nivelul Fizic.
 - Nivelul legătură de date organizează datele care trebuie trimise sub forma unor cadre.
 - La acest nivel trebuie să se practice și un control al fluxului de date.

Modelul OSI (*Open System Interconnection*)

- Nivelul Rețea
 - Trebuie să îndeplinească sarcina mai complexă de a transporta date între două noduri neadiacente, adică informația va trebui să tranziteze noduri intermediare.
 - Nivelul Rețea organizează datele care trebuie trimise sub forma unor pachete.
 - Cu alte cuvinte, Nivelul Rețea este responsabil de dirijarea pachetelor de la sursă la destinație trecând prin noduri intermediare.

Modelul OSI (*Open System Interconnection*)

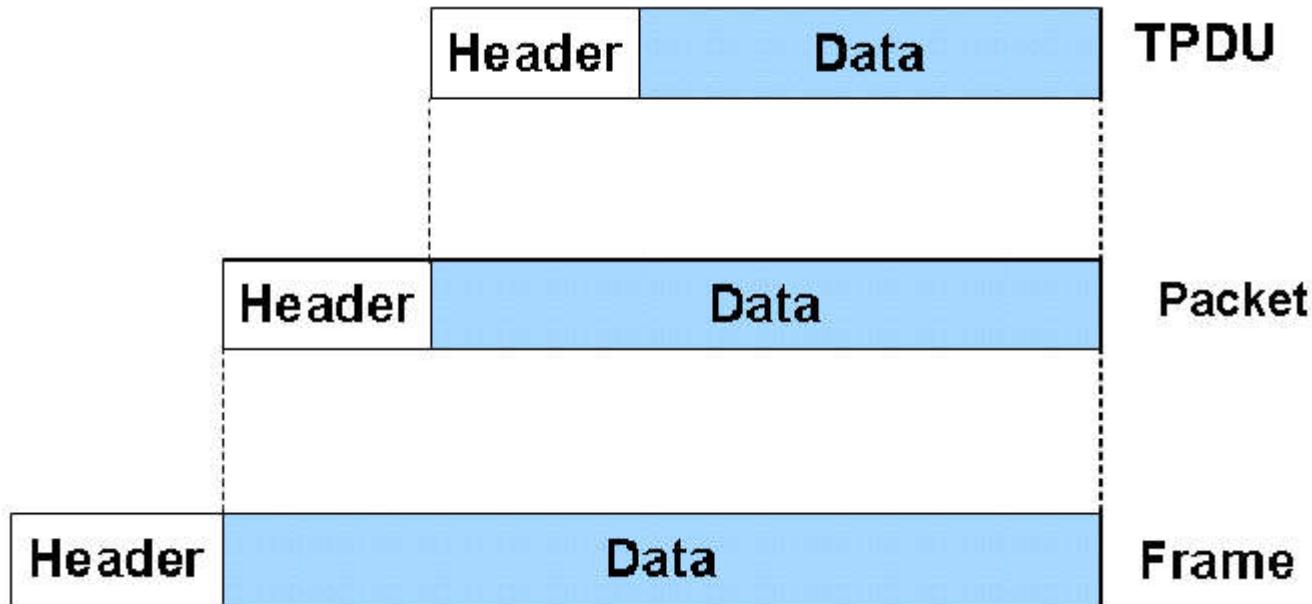
- Nivelul Rețea
 - Pachetul folosit de Nivelul Rețea este prevăzut cu un câmp în partea de Header, care reprezintă adresa nodului destinație.
 - Când datele tranzitează nodurile intermediare, este nevoie ca de fiecare dată să se verifice valoarea acestui câmp de adresă.

Modelul OSI (*Open System Interconnection*)

- Nivelul Transport
 - Nivelul Transport spunem că este de tipul *capăt la capăt* deoarece o instanță a protocolelor de pe acest nivel trebuie să existe doar la nivelul nodurilor care comunică între ele.
 - Realizează fragmentarea mesajelor prea lungi.
 - Asigură ca datele să ajungă în aceiași ordine în care au fost transmise.
 - Asigură un control al fluxului.
 - Transformă Nivelul Rețea dintr-unul nesigur, în unul sigur.

Modelul OSI (*Open System Interconnection*)

- Nivelul Transport organizează datele sub forma unor pachete numite TPDU (Transport Protocol Data Units).



Modelul OSI (*Open System Interconnection*)

- **Nivelul Sesiune**
 - Nivelul sesiune a fost gândit pentru a permite utilizatorilor să stabilească sesiuni, adică o modalitate de sincronizare și de control al dialogului între două procese care comunică la distanță.
- **Nivelul Prezentare**
 - Acest nivel procesează informațiile pentru a le face compatibile între două aplicații diferite, asigurând o independență între aplicații și Nivelul Transport
 - Operațiile tipice pe care acest nivel le realizează sunt de conversie, formatare, criptare și compresie.
- **Nivelul Aplicație**
 - Conține toate protocoalele și aplicațiile care interacționează direct cu utilizatorul oferind o interfață pentru accesul acestuia la rețea.

Modelul Internet (Stiva TCP/IP)



Modelul Internet (Stiva TCP/IP)

- Nivelul de Acces la Mediu
 - La acest nivel stiva TCP/IP nu definește un anume protocol.
 - Ideea este de a suporta toate standardele de pe acest nivel (ex. Ethernet, Frame Relay, ATM, rețele bazate pe fibră optică, rețele fără fir, etc.)

Modelul Internet (Stiva TCP/IP)

- Nivelul Internet
 - Protocolul care funcționează pe acest nivel este protocolul IP.
 - Tipul de serviciu oferit de acest protocol este de tipul comutare de pachete.
 - Datele care urmează a fi transmise vor fi încapsulate în pachete.
 - Pachetele vor fi direjate spre destinație în mod independent unele față de altele

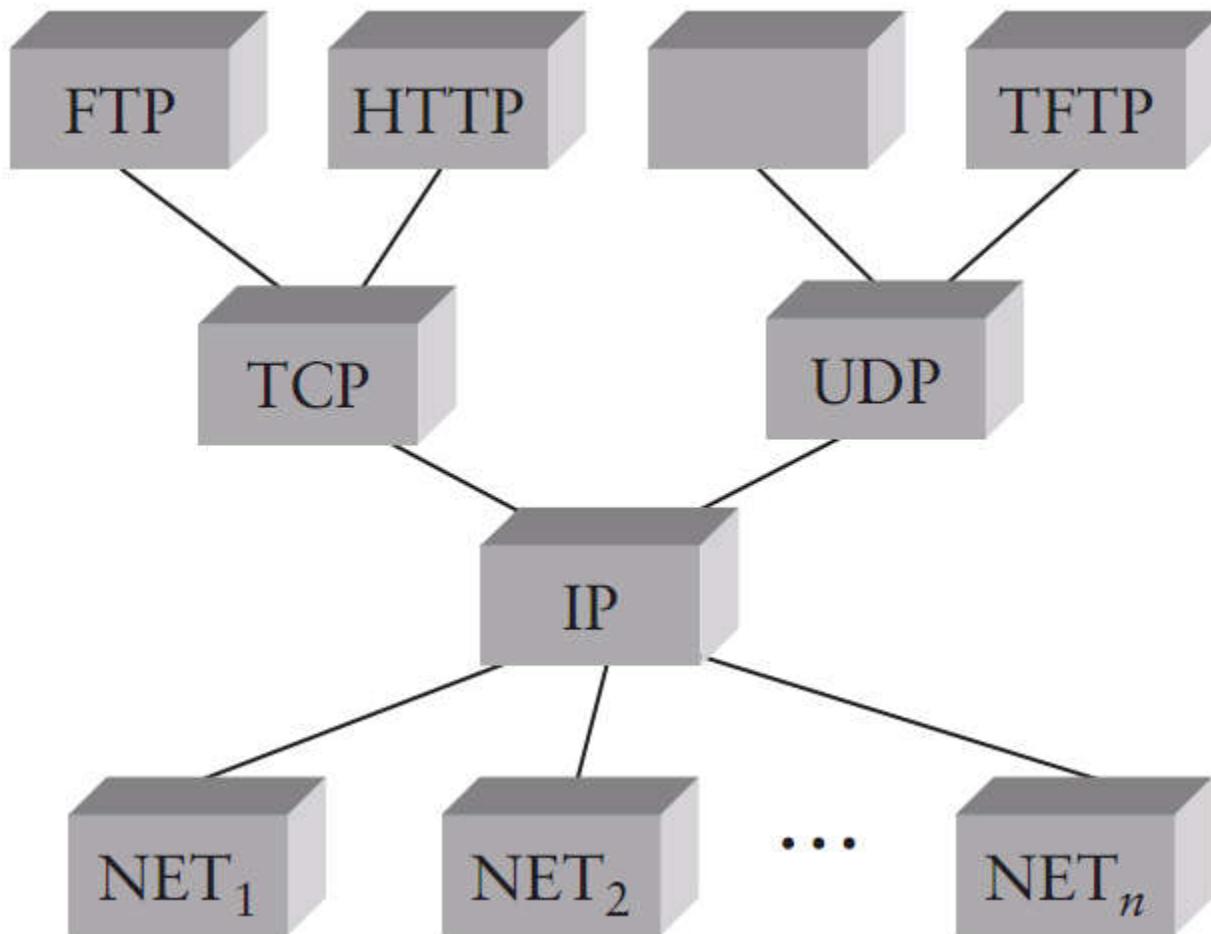
Modelul Internet (Stiva TCP/IP)

- Nivelul Transport
 - Există două tipuri de servicii pe care Nivelul Transport le poate oferi:
 - serviciu orientat pe conexiune, fără erori, care furnizează octetii în ordinea în care au fost trimiși (protocolul TCP).
 - celălalt serviciu nu oferă nici o garanție asupra ordinii în care vor fi recepționate datele (protocolul UDP).

Modelul Internet (Stiva TCP/IP)

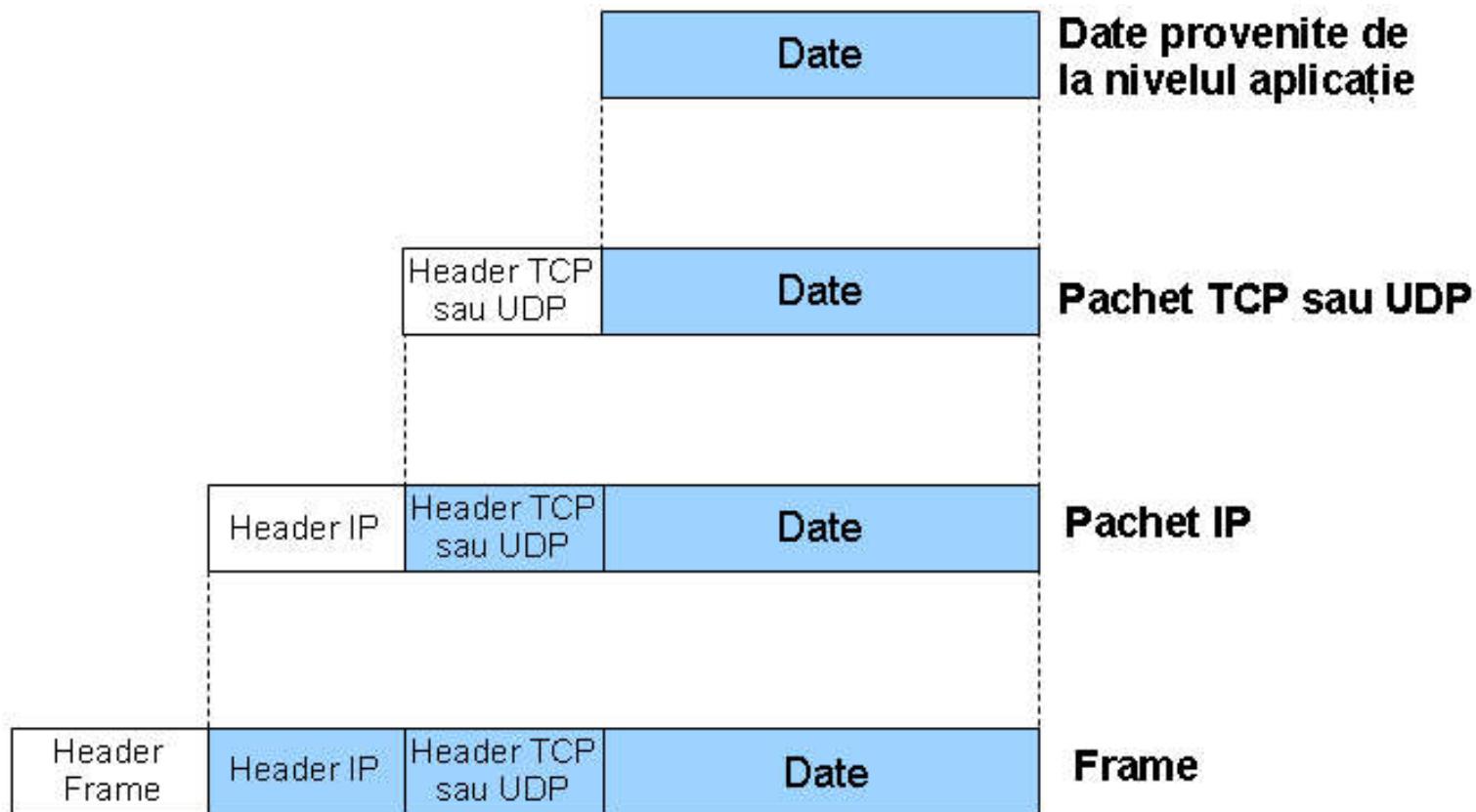
- Nivelul Aplicație
 - La acest nivel se găsesc toate aplicațiile și protocoalele care asigură accesul utilizatorului la resursele rețelei.

Modelul Internet (Stiva TCP/IP)



Modelul Internet (Stiva TCP/IP)

- Încapsularea datelor practicată în interiorul stivei TCP/IP.



Modelul Internet (Stiva TCP/IP)

- Caracteristici importante
 - Stiva TCP/IP nu impune o stratificare strictă: o aplicație poate ignora Nivelul Trasport și să facă apel direct la Nivelul Internet sau la una din tehnologiile de pe Nivelul de Acces la Mediu.
 - IP servește ca punct focal al intregii arhitecturi Internet: definește o metodă comună de schimb de pachete pentru un număr foarte mare de rețele.

Rețele de calculatoare

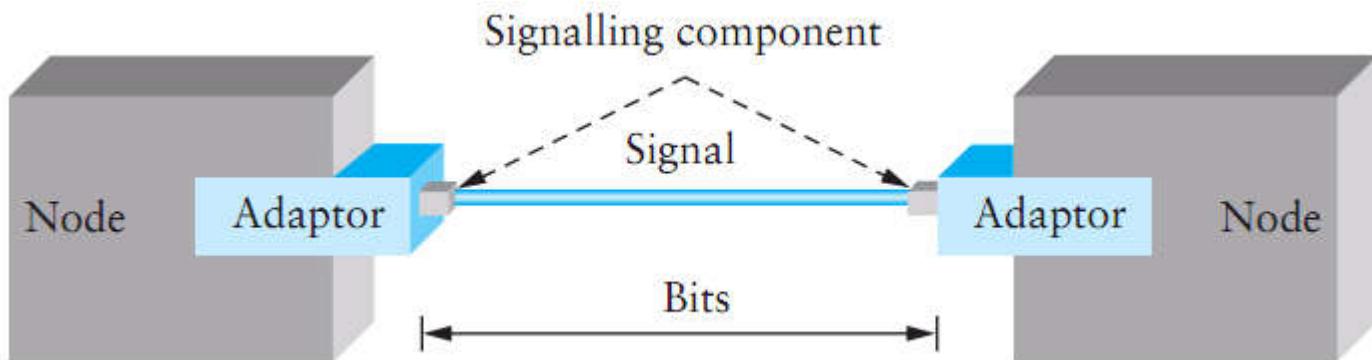
Partea a 3-a

Sebastian Fuicu

- Tehnici de transmisie a datelor
- Transmisia digitală a datelor digitale
- Transmisia digitală a datelor analogice
- Transmisia analogică a datelor digitale
- Boud rate vs. Bit rate

Tehnici de transmisie a datelor

- Datele se propagă prin legături fizice.
- Sunt necesare metode de codificare pentru datele care vor fi transmise.



Tehnici de transmisie a datelor

- **Date:** entități care conțin informație.
- **Semnale:** privite ca purtătoare de date.
- **Transmisia:** definită prin comunicarea datelor folosind propagarea și procesarea semnalelor.

Tehnici de transmisie a datelor

- **Date analogice**: iau valori continue intr-un anumit interval.
- **Date digitale**: iau valori discrete dintr-o anumită mulțime finită.
- **Semnal analogic**: definit ca o undă electromagnetică continuă.
- **Semnal digital**: definit ca o secvență de impulsuri de tensiune sau curent cu valori dintr-o multime finita.

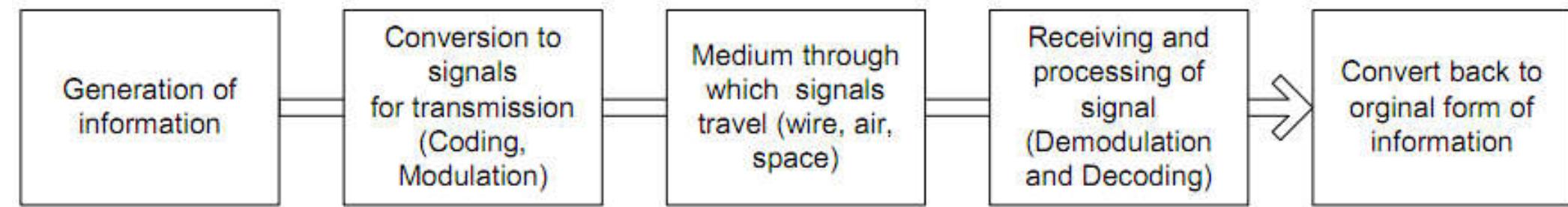
Tehnici de transmisie a datelor

- **Semnal discret în timp vs. semnal digital**
- **Semnal discret în timp**: ia valori doar la momente discrete de timp, între aceste momente el nefiind definit.
- **Semnal digital**: poate lua doar anumite valori dintr-o anumită mulțime finită.
- Un semnal digital care poate lua doar două valori se numește **semnal binar**.
- Un semnal binar este un caz particular al unui semnal digital.
- Un semnal digital este un caz particular al unui semnal discret în timp.

Tehnici de transmisie a datelor

- Moduri de transmisie a informației
 - 1) Transmisie digitală a datelor digitale
 - se folosesc tehnici de codificare a datelor.
 - 2) Transmisia digitală a datelor analogice
 - se realizează mai întâi conversia A/D a datelor.
 - 3) Transmisia analogică a datelor digitale sau analogice
 - se folosesc procedee de modulare și demodulare a semnalului transmис.

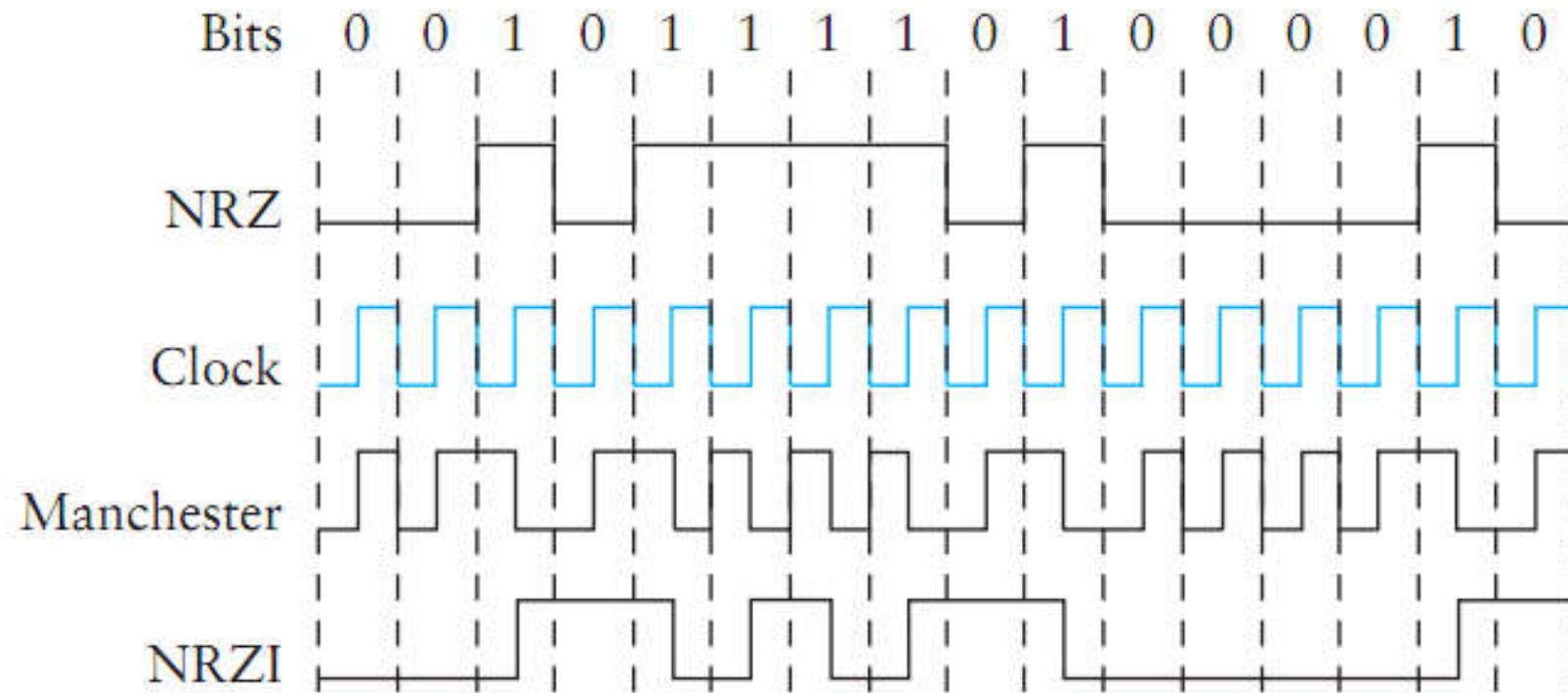
Tehnici de transmisie a datelor



1) Transmisiă digitală a datelor digitale

- Există diverse metode de codificare a datelor:
 - NRZ (Non-Return to Zero)
 - NRZI (Non-Return to Zero Inverted)
 - Manchester
 - 4B/5B

1) Transmisiă digitală a datelor digitale



1) Transmisia digitală a datelor digitale

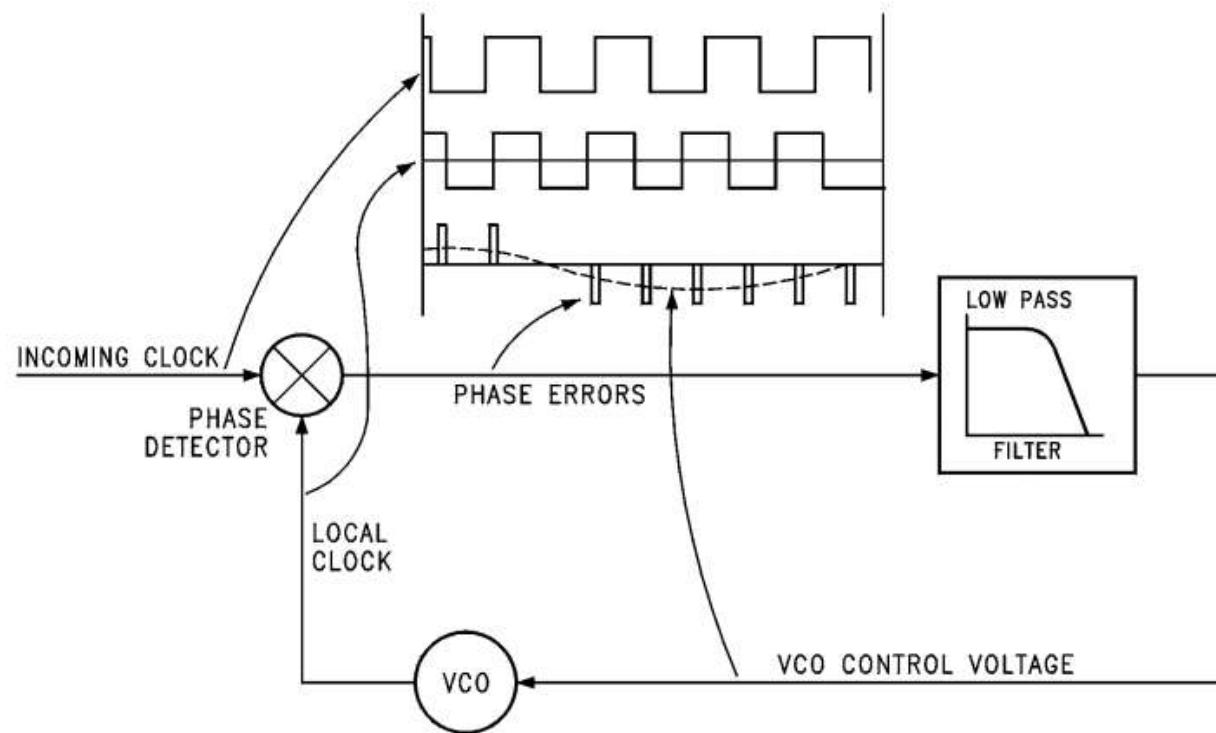
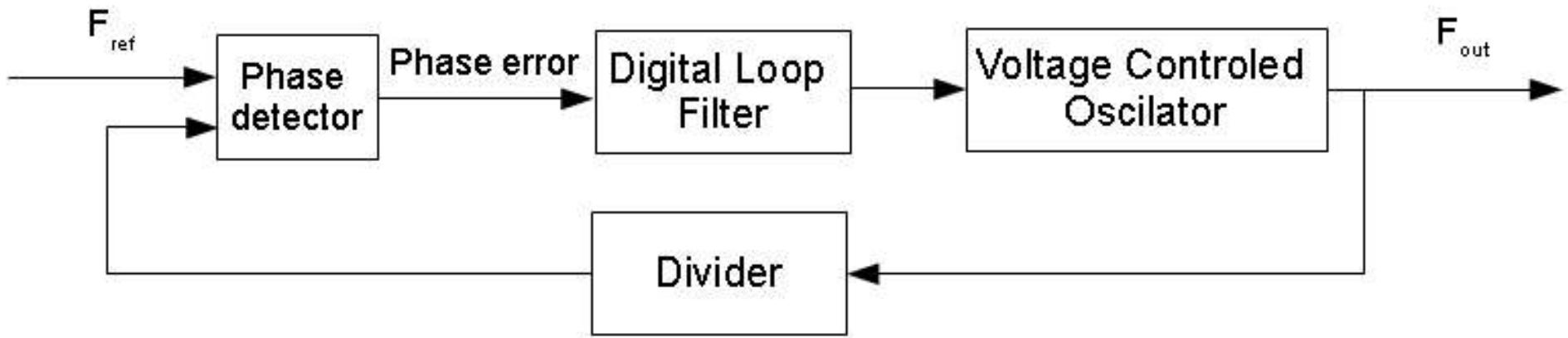
- Codificarea NRZ

- Dezavantajele codificării NRZ

- O secvență prelungită de valori 1 sau 0, va determina rămânerea semnalului pe un anumit nivel de tensiune pentru un interval lung de timp.
- Un nivel scăzut al tensiunii pe o durată mai lungă de timp poate să corespundă și absenței semnalului.
- Lipsa tranzițiile repetitive ale semnalului determină imposibilitatea refacerii semnalului de tact la receptor.

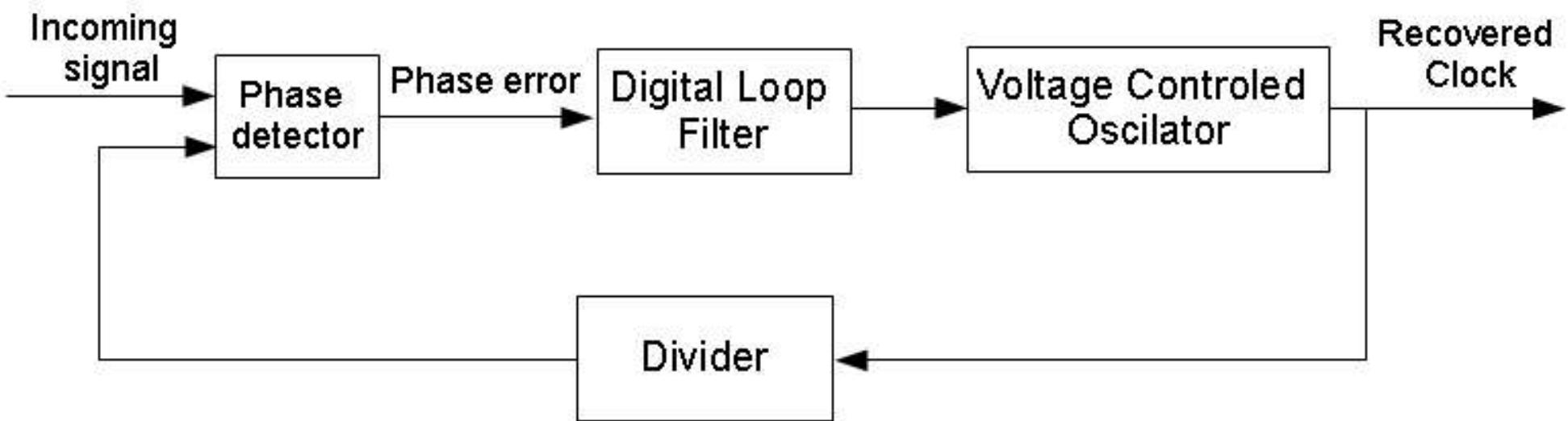
1) Transmisia digitală a datelor digitale

PLL (Phase Locked Loop)



1) Transmisiă digitală a datelor digitale

– Clock recovery



PLL – Clock recovery

1) Transmisia digitală a datelor digitale

- Codificarea Manchester

- Dezavantajele codificării Manchester
 - Codificarea Manchester duce la o creștere a numărului de tranziții.
 - În medie, numărul tranzițiilor se dublează față de codificarea NRZ.
 - Aceeași cantitate de informația va necesita un număr dublu de tranziții.
 - Spunem că eficiența codificării Manchester este de 50% .

1) Transmisiă digitală a datelor digitale

- Codificarea 4B/5B

- Încearcă să rezolve ineficiența metodei Manchester
- Ideea este de a insera biți de 1 într-o secvență mai lungă de biți de 0.
- Fiecare 4 biți de date sunt codați într-o secvență de 5 biți, de aici numele de 4B/5B.
- În interiorul unei secvențe de 5 biți nu trebuie să existe mai mult de două zerorui consecutive.
- Codurile nou formate sunt transmise folosind metoda NRZI, care a rezolvat deja problema biților de 1 consecutivi.
- Eficiența metodei este de 80%

1) Transmisiă digitală a datelor digitale

- Codificarea 4B/5B

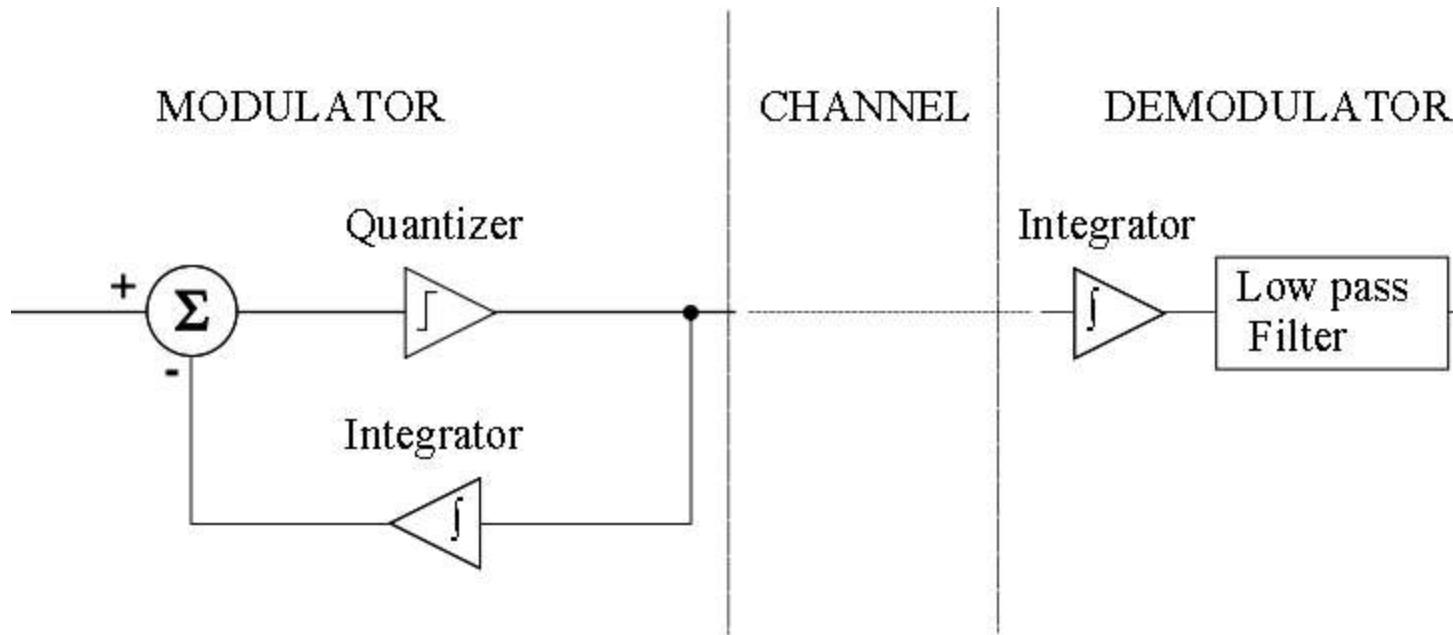
4-Bit Data Symbol	5-Bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

2) Transmisiua digitală a datelor analogice

- Semnalul analogic trebuie convertit mai intai in semnal digital.
- **Teorema esantionarii:** Un semnal de bandă limitată poate fi recuperat din esantioanele sale daca acestea sunt luate cu o frecvență mai mare sau egală cu dublul celei mai mari frecvențe din spectrul semnalului initial.
- Procesele prin care trece semnalul analogic sunt: aşantionarea și cuantizarea.

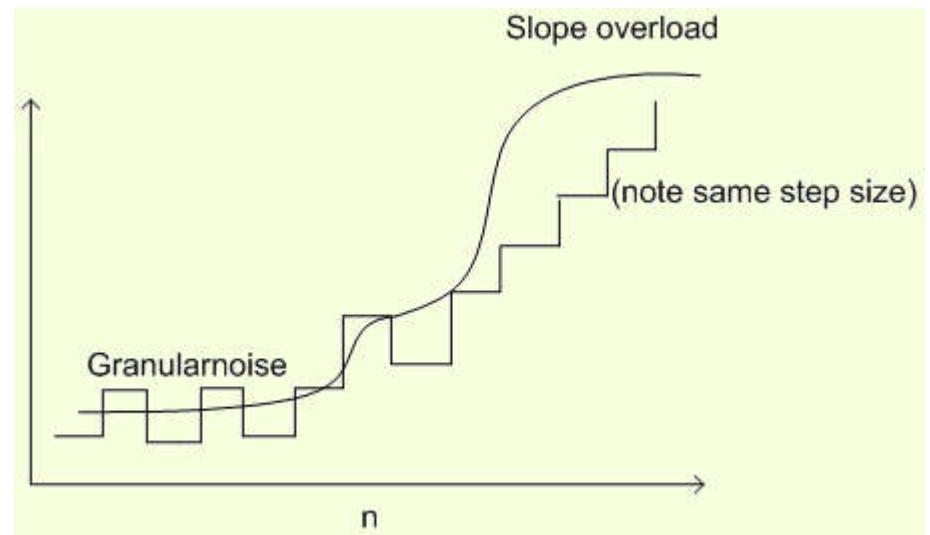
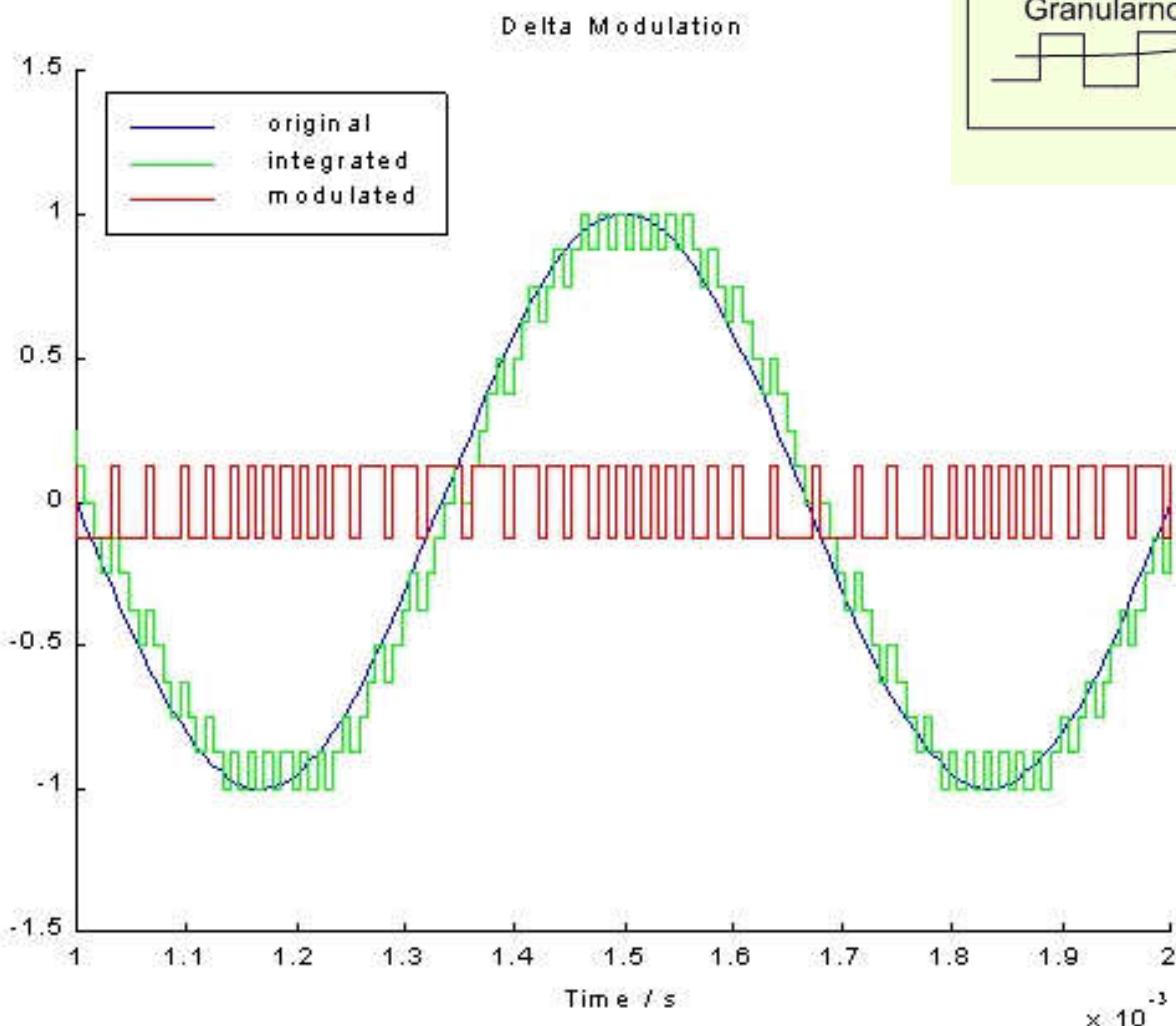
2) Transmisiă digitală a datelor analogice

- Tehnica cea mai folosită pentru transmiterea semnalului vocal este PCM (Pluse Code Modulation).
- O altă metodă este DM (Delta Modulation)



2) Transmisiă digitală a datelor analogice

DM (Delta Modulation)



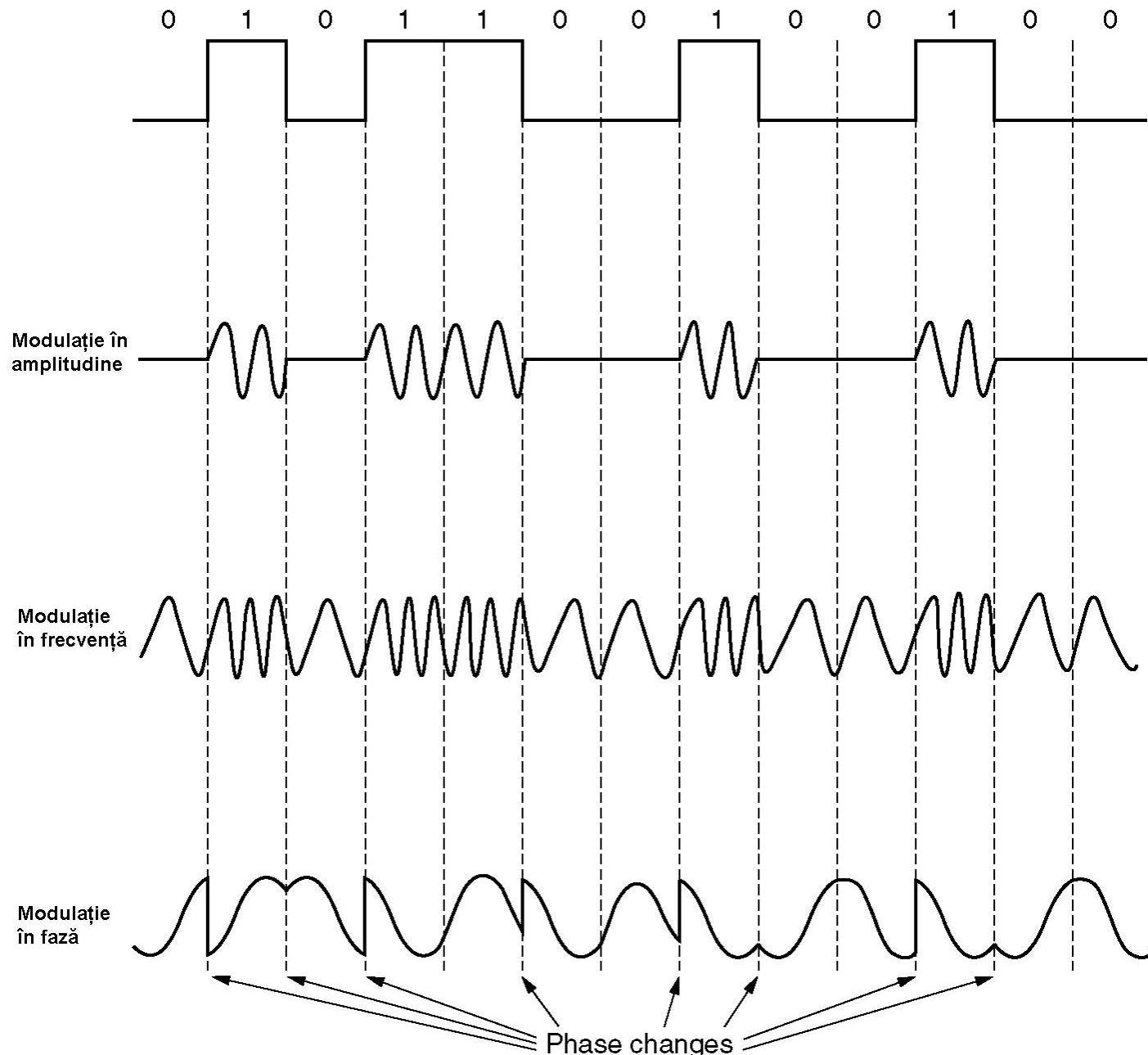
3) Transmisia analogică a datelor digitale

- Există 3 tipuri fundamentale de modulație:
 - Modulație în amplitudine
 - Modulație în frecvență
 - Modulație în fază
- Spunem că semnalul util, cel care conține informația, modulează un alt doilea semnal, pe care îl vom numi semnal purtător.

3) Transmisia analogică a datelor digitale

- Pentru transmiterea datelor digitale cele trei tipuri de modulații au primit următoarele denumiri:
 - ASK (Amplitude Shift Keying)
 - FSK (Frequency Shift Keying)
 - PSK (Phase Shif Keying)

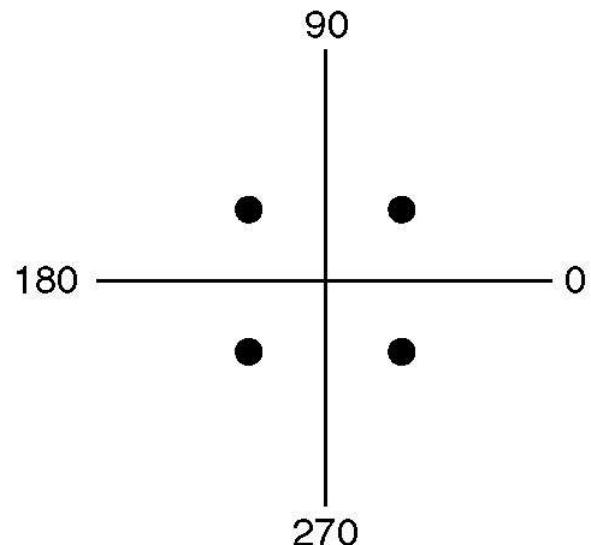
3) Transmisia analogică a datelor digitale



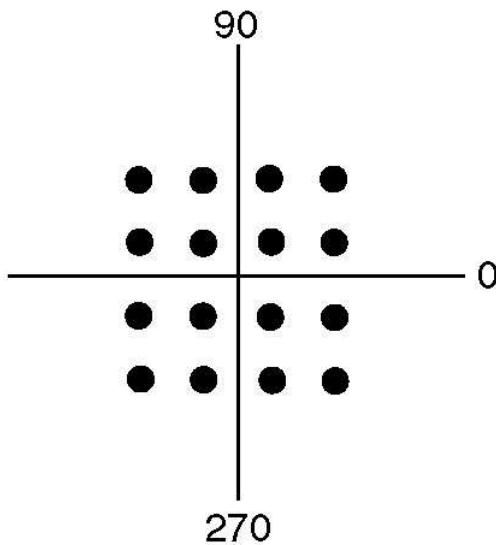
Boud rate vs. Bit rate

- **Boud rate:** frecvența cu care un semnal își schimbă starea pe un canal de comunicație.
- **Bit rate:** numărul de biți care sunt transmiși în unitate de timp (bps) pe un canal de comunicație.
- Să luăm ca exemplu niște tipuri mai speciale de modulații, obținute combinând modulația în amplitudine cu modulația în fază. Acestea sunt redate pe următoarele două slide-uri. Reprezentarea semnalului analogic se face sub forma unor “constelații” de puncte. Distanța punctului față de origine arată amplitudinea semnalului, iar poziția față de cele două axe reprezintă valoarea în grade a defazajului.
- Fiecare punct reprezintă o stare în care se poate afla semnalul analogic. Fiecărei stări i se poate asocia un cod binar numit și simbol.

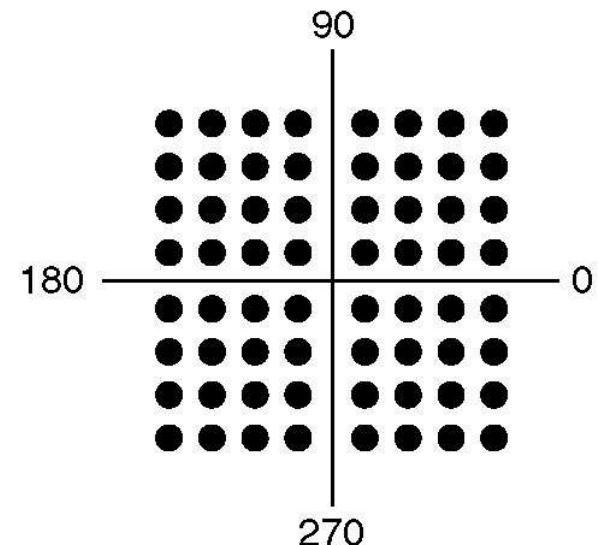
Boud rate vs. Bit rate



(a)



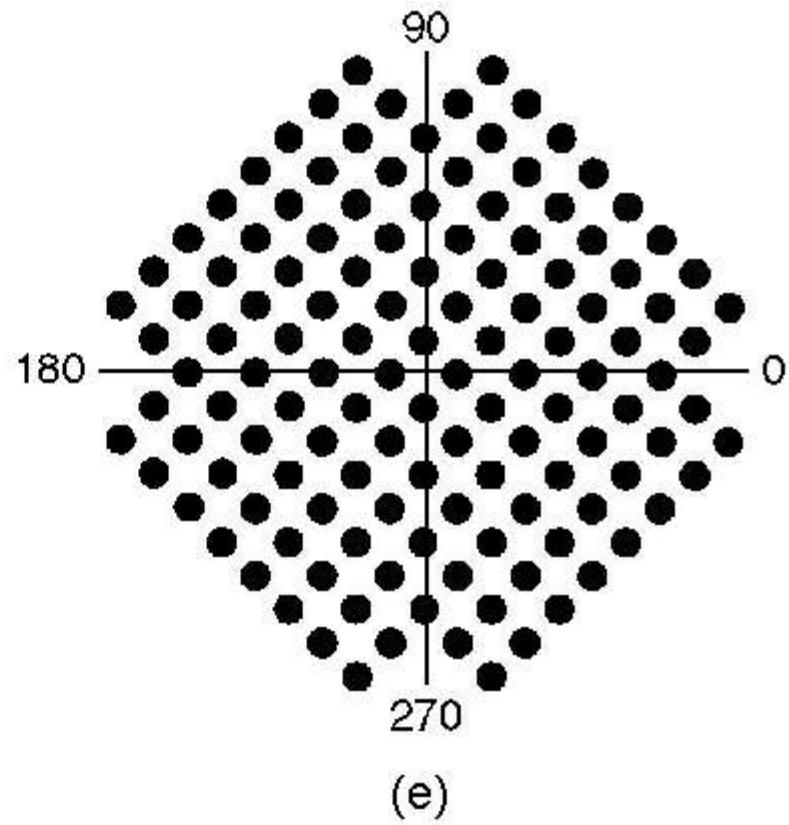
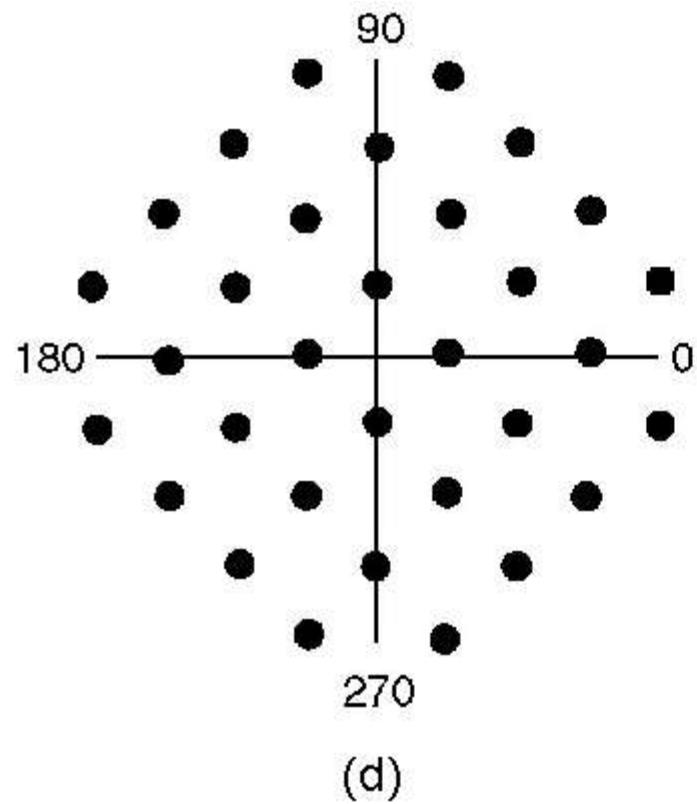
(b)



(c)

- a) QPSK (Quadrature Phase Shif Keying)
- b) QAM-16 (Quadrature Amplitude Modulation)
- c) QAM-64 (Quadrature Amplitude Modulation)

Boud rate vs. Bit rate



(d) V.32 for 9600 bps.

(e) V32 bis for 14,400 bps.

Boud rate vs. Bit rate

- Pentru modulația QPSK, semnalul analogic se poate găsi în 4 stări (cele patru puncte).
- Cele patru stări pot fi codificate folosind 2 biți. Deci celor 4 stări li se vor asocia 4 simboluri binare. Astfel boud-ul mai poate fi definit ca numărul de simboluri transmise în unitate de timp.
- Dacă semnalul analogic își schimbă starea cu o frecvență de 2 KHz, spunem că boud-ul pentru acel semnal este 2K.
- În acest caz, bit rate-ul, are valoare 4Kbps, pentru că la fiecare schimbare a stării semnalului vor fi transmiși 2 biți, deci se înmulțește valoarea boud-ului cu numărul de biți folosiți pentru codificarea stărilor semnalului analogic sau altfel spus, cu numărul de biți care intră în componența unui simbol binar.
- Un alt exemplu, în cazul lui QAM-16, dacă boud-ul este de 2 KHz, bit rate-ul are valoarea 8Kbps.

Rețele de calculatoare

Partea a 4-a

Sebastian Fuicu

- **Transmisii fiabile pe nivelul Legătură de Date**
- **Rețele locale (Standardul IEEE 802)**
- **Rețele LAN Ethernet (802.3)**
- **Rețele WLAN (802.11)**

Transmisii fiabile pe nivelul Legătură de Date

- Pentru a asigura o transmisie de date sigură se folosesc:
 - coduri detectoare și corectoare de erori.
 - mecanisme de tipul “automatic repeat request” (ARQ).
- Un protocol de nivel legătură de date care realizează livrarea sigură a datelor, trebuie să fie capabil să recupereze cadrele pierdute sau afectate de eroare.

Transmisii fiabile pe nivelul Legătură de Date

Automatic repeat request (ARQ)

- Presupune folosirea combinată a două mecanisme:
 - Confirmările (acknowledgements - ACK)
 - Temporizările (timeouts)
- Confirmarea este reprezentată de un mic cadru de control pe care un protocol îl trimită înapoi sursei pentru a semnaliza receptia corectă a cadrului.
- Există și varianta de **piggybacking** la transmiterea confirmării, adică atașarea confirmării la un pachet de date.
- Dacă nu se primește confirmarea după un anumit interval de timp (timeout), se retransmite cadrul original.

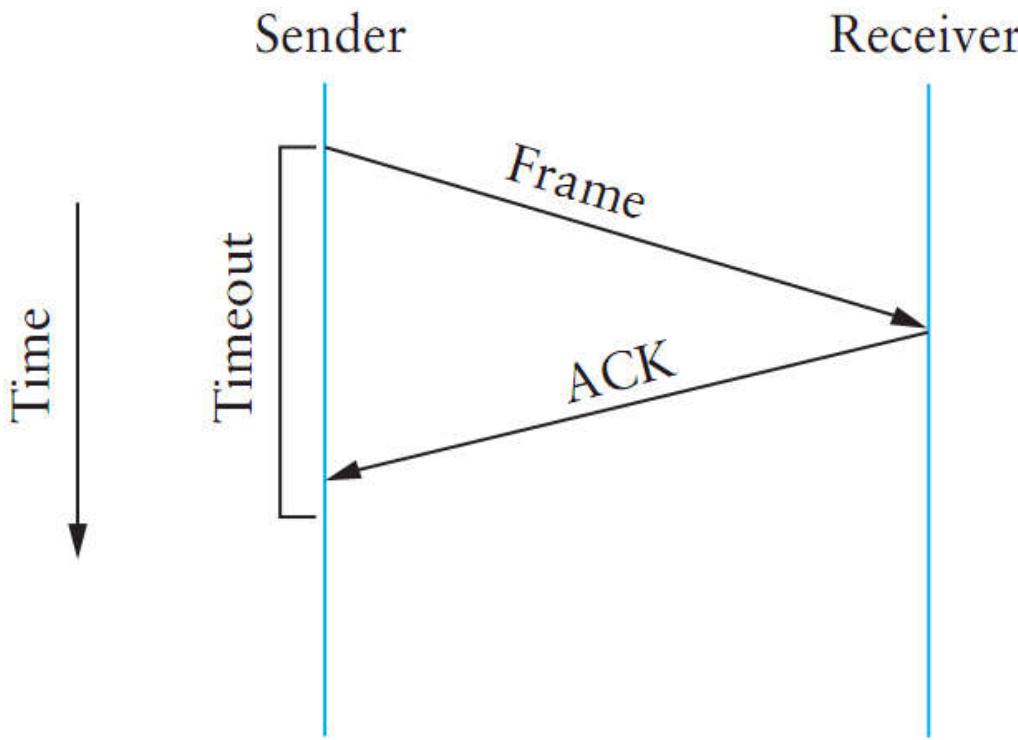
Transmisii fiabile pe nivelul Legătură de Date

Stop-and-Wait

- Cea mai simplă schemă ARQ este algoritmul stop-and-wait.
- După transmiterea unui cadru, emițătorul se oprește și așteaptă primirea confirmării.
- După primirea confirmării este trimis următorul cadru.
- Este posibil ca atât cadrul cât și confirmarea să fie afectate de erori sau să se piardă. În această situație, după scurgerea unui interval de timp, dat de un timer, cadrul de date este retransmis.
- Mai jos sunt redate situațiile în care se poate afla protocolul stop and wait.

Transmisii fiabile pe nivelul Legătură de Date

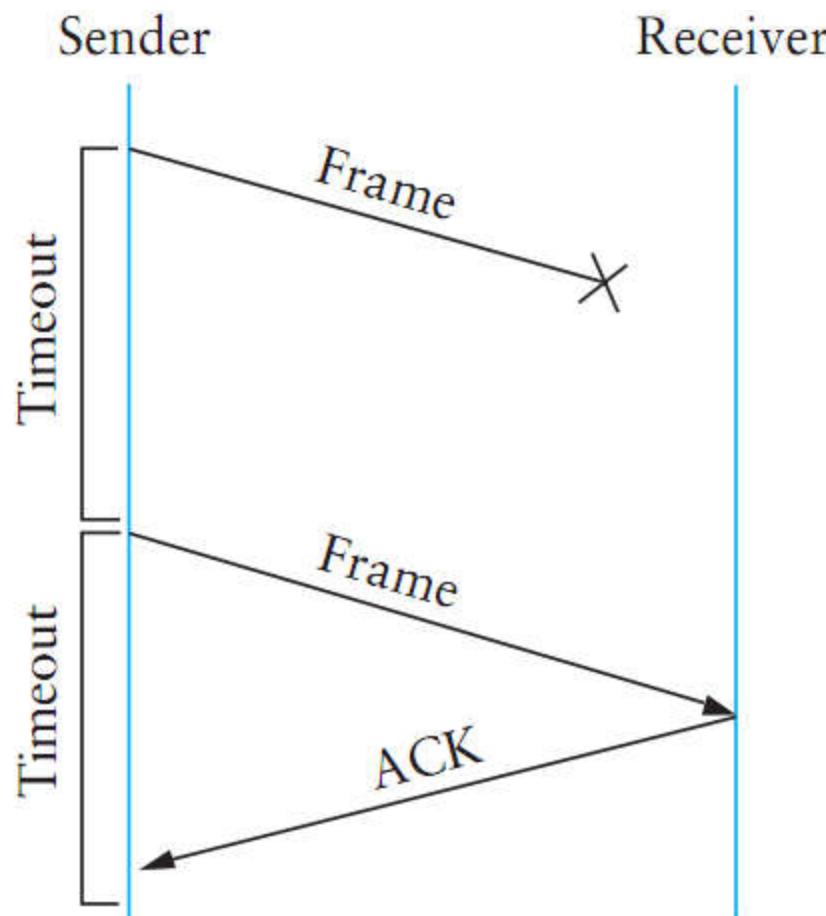
Stop-and-Wait



- a) Confirmarea este primită înainte de expirarea timpului

Transmisii fiabile pe nivelul Legătură de Date

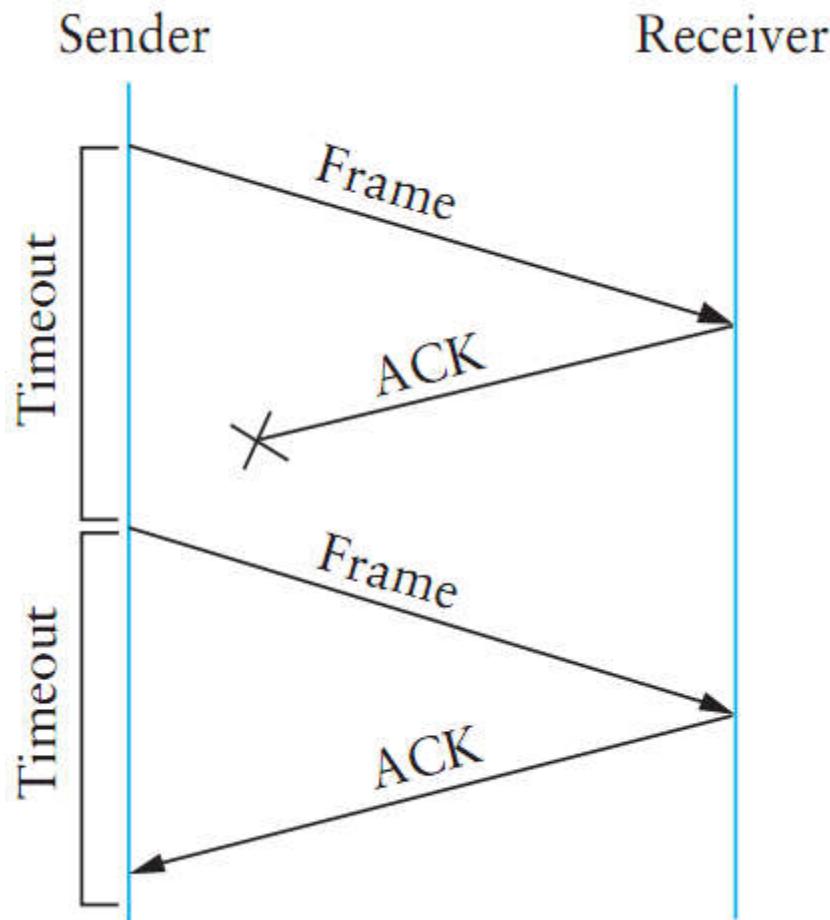
Stop-and-Wait



b) Cadrul de date original se pierde

Transmisii fiabile pe nivelul Legătură de Date

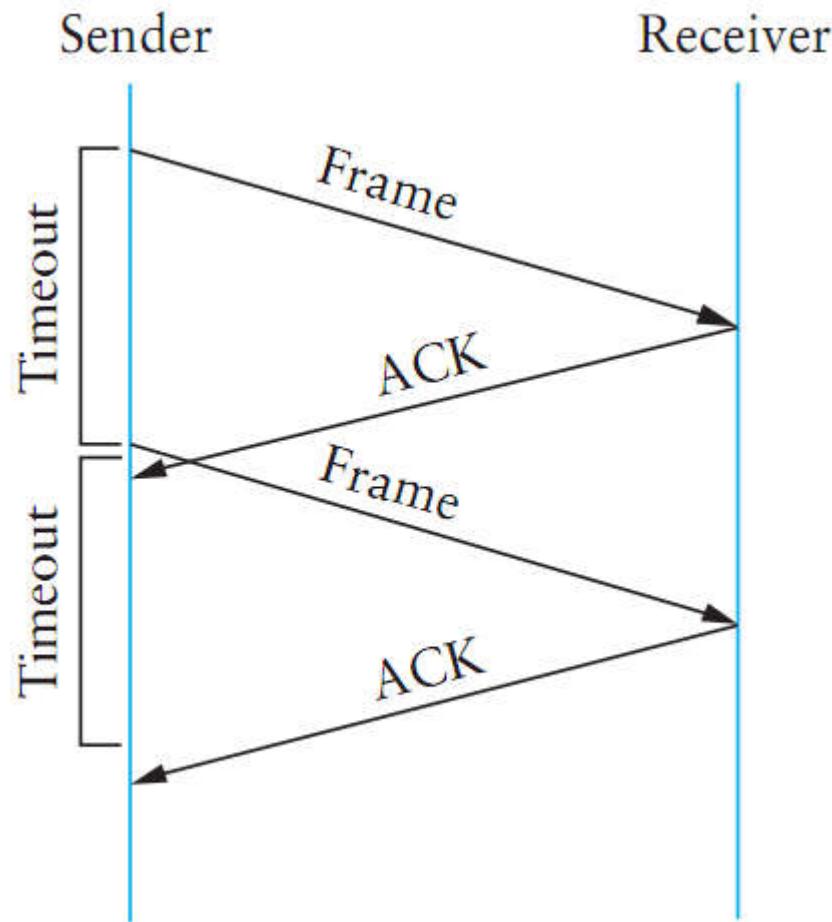
Stop-and-Wait



c) Cadrul de confirmare se pierde

Transmisii fiabile pe nivelul Legătură de Date

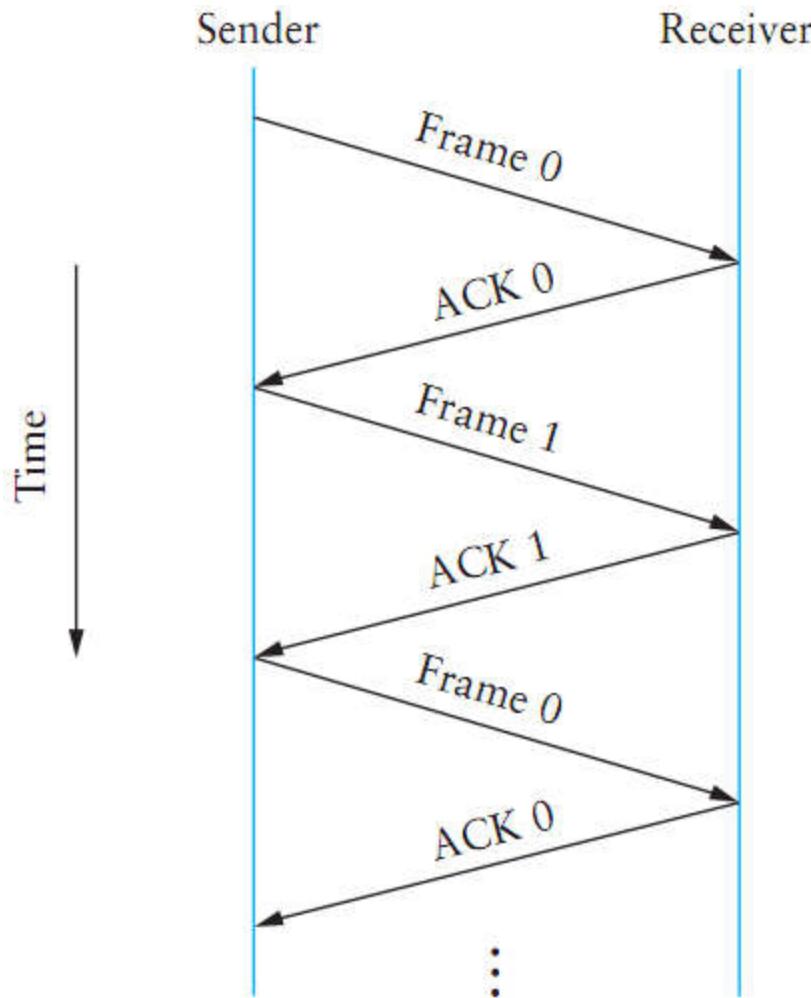
Stop-and-Wait



d) Timpul expiră prea repede

Transmisii fiabile pe nivelul Legătură de Date

Stop-and-Wait



- Pentru procolul stop and wait sunt suficiente doar două numere de secvență, în cazul acesta ele fiind 0 și 1.

Transmisii fiabile pe nivelul Legătură de Date

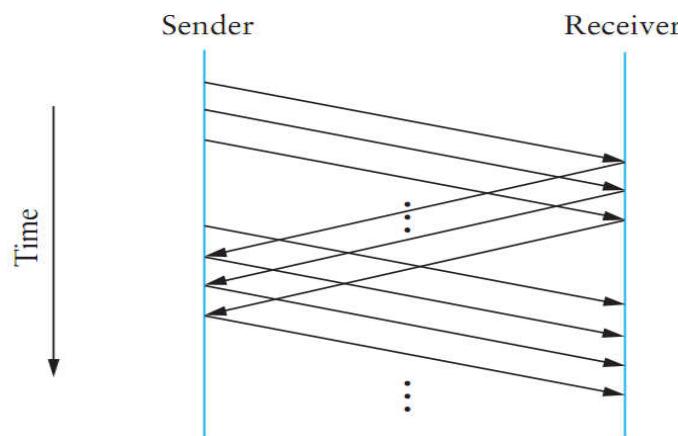
Stop-and-Wait

- Problema majoră a protocolului stop and wait este aceea că la un moment dat un singur frame se poate afla în rețea. După transmiterea unui frame, protocolul trebuie să se oprească și să așteapte primirea confirmării.
- În această manieră, capacitatea de transfer a liniei de comunicații nu este folosită la maxim.

Transmisii fiabile pe nivelul Legătură de Date

Protocole cu fereastră glisantă (Sliding Window Protocols)

- Problema protocolului stop and wait enunțată mai devreme este soluționată de către familia de protocolele numită cu “fereastră glisantă”. Acestea permit transmiterea în rețea a frame-urilor de date unul după altul, fără a fi necesară oprirea după fiecare frame în parte și așteptarea confirmării.
- Diagrama de timp pentru un protocol de tip “fereastră glisantă”.

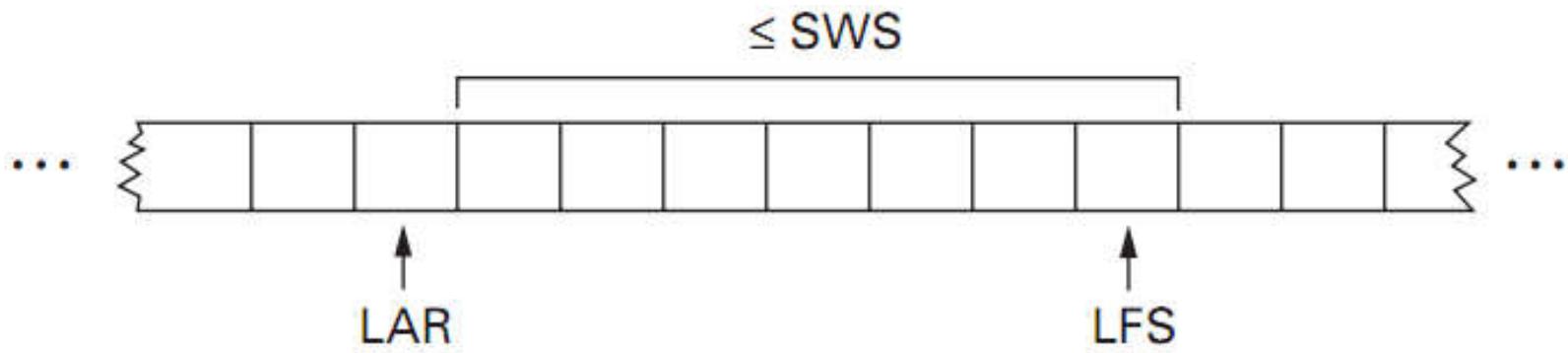


- Fiecare cadru i se atribuie un număr de secvență distinct (SequnceNumber).

Transmisii fiabile pe nivelul Legătură de Date

Protocole cu fereastră glisantă (Sliding Window Protocols)

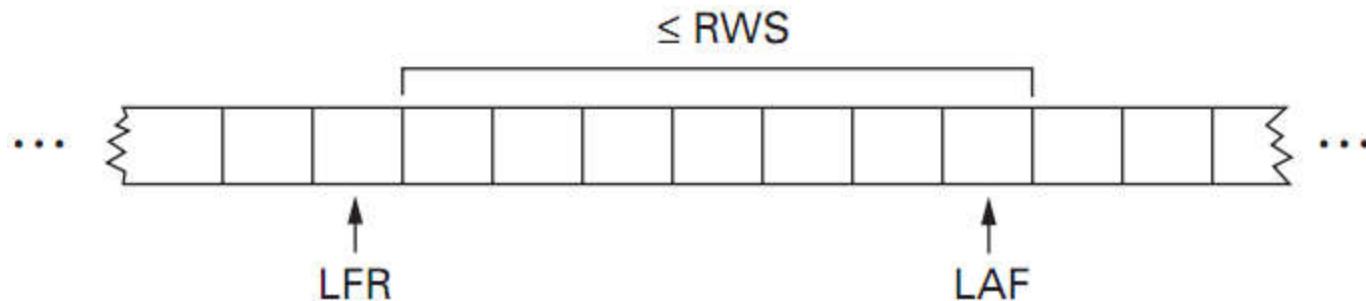
- Emițătorul folosește următoarele 3 variabile:
 - SWS (Sending Window Size)
 - LAR (Last Acknowledgement Received)
 - LFS (Last Frame Sent)
- Emițătorul păstrează următoarea inegalitate:
$$\text{LFS} - \text{LAR} \leq \text{SWS}$$



Transmisii fiabile pe nivelul Legătură de Date

Protocole cu fereastră glisantă (Sliding Window Protocols)

- Receptorul folosește următoarele 3 variabile:
 - RWS (Receiving Window Size)
 - LAF (Largest Acceptable Frame)
 - LFR (Last Frame Received)
- Emițătorul păstrează următoarea inegalitate:
$$\text{LAF} - \text{LFR} \leq \text{RWS}$$



Transmisii fiabile pe nivelul Legătură de Date

Protocolale cu fereastră glisantă

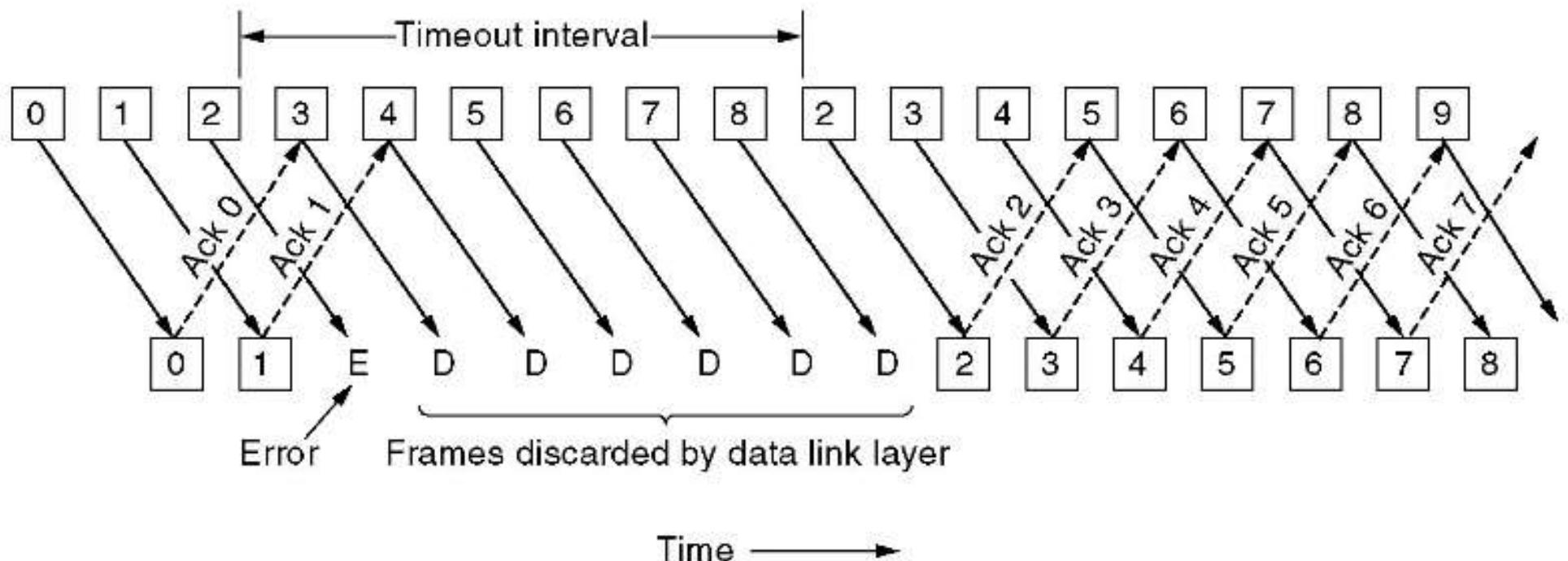
- Protocol “Go Back N” (RWS = 1)

- Protocolul Go Back N, este un caz particular de protocol cu fereastră glisantă, unde dimensiunea ferestrei receptorului are valoarea 1.
- Se observă ca dacă un cadru de date se pierde sau este afectat de eroare, toate cadrele de date care urmează după el sunt ignore, netrimițându-se confirmări pentru ele.
- Fiecare cadru de date trimis are asociat un timer. Dacă până la timeout, nu este recepționată confirmarea, atunci cadrul de date este retrimis.
- În cazul tuturor tipurilor de protocolale cu fereastră glisantă este necesar ca frame-urile de date care au fost trimise să fie salvate local într-un buffer, până la primirea confirmării din partea receptorului. În cazul în care confirmarea nu sosește, în momentul generării timeout-ului, frame-urile vor fi luate din acest buffer și retransmise.

Transmisii fiabile pe nivelul Legătură de Date

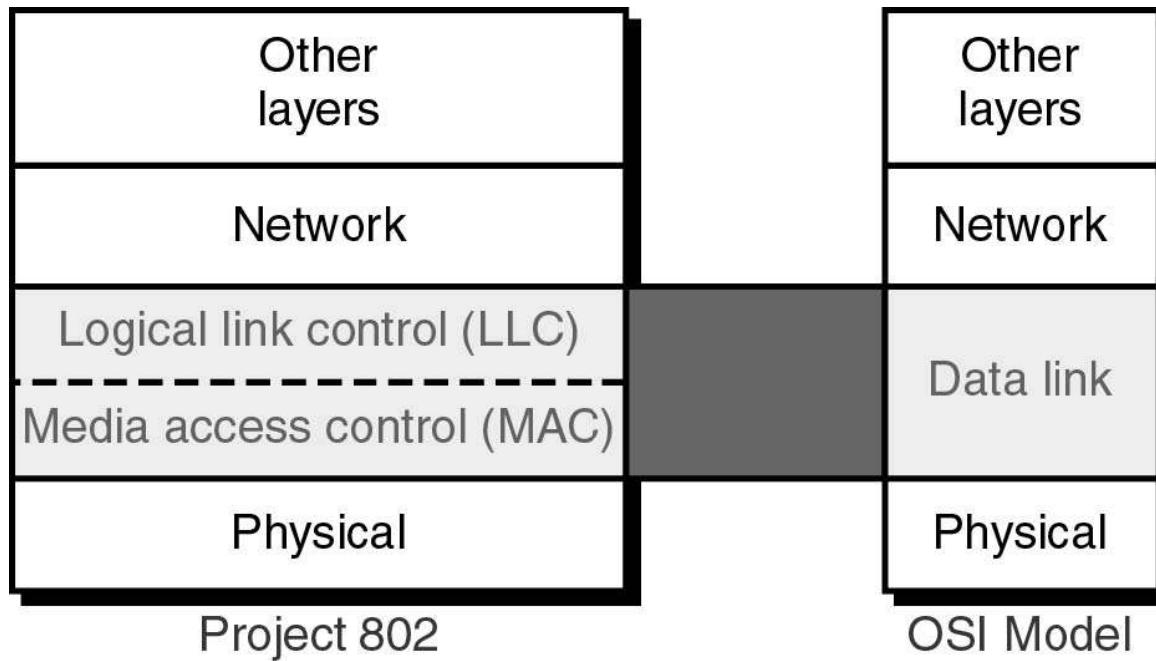
Protocole cu fereastră glisantă

- Protocol de tip “Go Back N” (RWS = 1)



Rețele locale (Standardul IEEE 802)

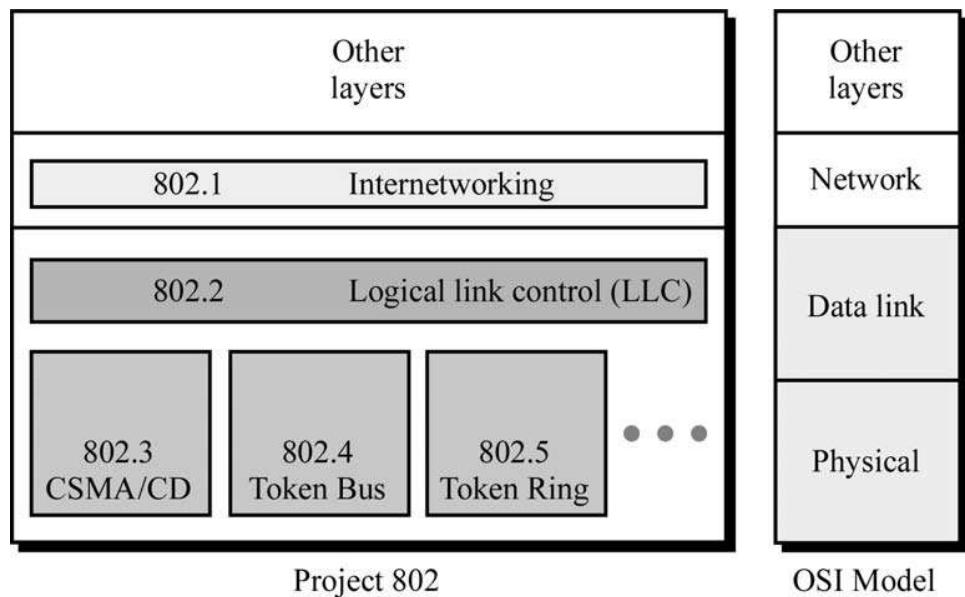
- Odată cu diversificarea rețelelor locale (Ethernet, Arcnet, Token-ring, etc.) s-a simțit nevoie unei standardizări.
- A fost demarat proiectul IEEE 802.
- A fost propus un model pentru rețelele locale.



Rețele locale (Standardul IEEE 802)

- Acest model specifică existența unui nivel Fizic:
 - acesta acoperă toate aspectele legate de comunicația pe un mediu fizic.
- Același model specifică două subniveluri ce implementează funcții asociate nivelului Legătură de Date:
 - Subnivelul de control al accesului la mediu (MAC – Medium Access Control): asigură accesul la mediul de transmisie.
 - Subnivelul pentru controlul legăturii logice (LLC – Logical Link Control): asigură interfața cu protocolele de pe nivelele superioare.

Rețele locale (Standardul IEEE 802)



IEEE 802 Standards	
802.1	Bridging & Management
802.2	Logical Link Control
802.3	Ethernet - CSMA/CD Access Method
802.4	Token Passing Bus Access Method
802.5	Token Ring Access Method
802.6	Distributed Queue Dual Bus Access Method
802.7	Broadband LAN
802.8	Fiber Optic
802.9	Integrated Services LAN
802.10	Security
802.11	Wireless LAN
802.12	Demand Priority Access
802.14	Medium Access Control
802.15	Wireless Personal Area Networks
802.16	Broadband Wireless Metro Area Networks
802.17	Resilient Packet Ring

Rețele locale (Standardul IEEE 802)

Subnivelul LLC (IEEE 802.2)

- Specifică tipurile de servicii oferite și protocolul care le implementează.
- Are ca scop oferirea unei interfețe unificate, indiferent de ceea ce se găsește dedesubtul său.
- Există 3 tipuri de servicii:
 - 1) serviciu nebazat pe conexiune și fără confirmare (*Unacknowledged Connectionless Service*)
 - 2) serviciu orientat pe conexiune și cu confirmare (*Connection Oriented Service*)
 - 3) serviciu neorientat pe conexiune, dar cu confirmare (*Semireliable Service*)

Rețele locale (Standardul IEEE 802)

Subnivelul MAC

- Este specific fiecărui tip de rețea LAN.
- Are ca principală funcție, aceea a partajării mediului fizic (*medium sharing*).
- Asigură modul de operare cu difuzare (*broadcast*) specific rețelelor locale. Prin acest mod de operare, o stație are acces la toate cadrele care circulă în rețea, indiferent care este emițătorul.
- Modul de operare cu difuzare implică două probleme:
 - *la transmsie* trebuie să determine dacă mediul este liber și apoi să detecteze eventualele conflicte.
 - *la receptie* fiecare stație trebuie să stabilească dacă mesajul îi este adresat ei.

Rețele LAN Ethernet (802.3)

- Cea mai de succes tehnologie pentru rețele locale din ultimii 20 de ani.
- Versiuni apărute de-a lungul timpului:
 - Ethernet v1.0
 - Ethernet v2.0
 - IEEE 802.3
- Primele două versiuni au fost generate de un consorțiu format din firmele (Digital, Intel și Xerox).
- Standardul IEEE 802.3 are la bază versiunea Ethernet v2.0
- Standardul IEEE 802.3 acoperă nivelul Fizic și subnivelul de acces la mediu (MAC).
- Pentru subnivel MAC, metoda de acces la mediu se numește CSMA/CD (*Carrier Sense Multiple Access / Collision Detection*).

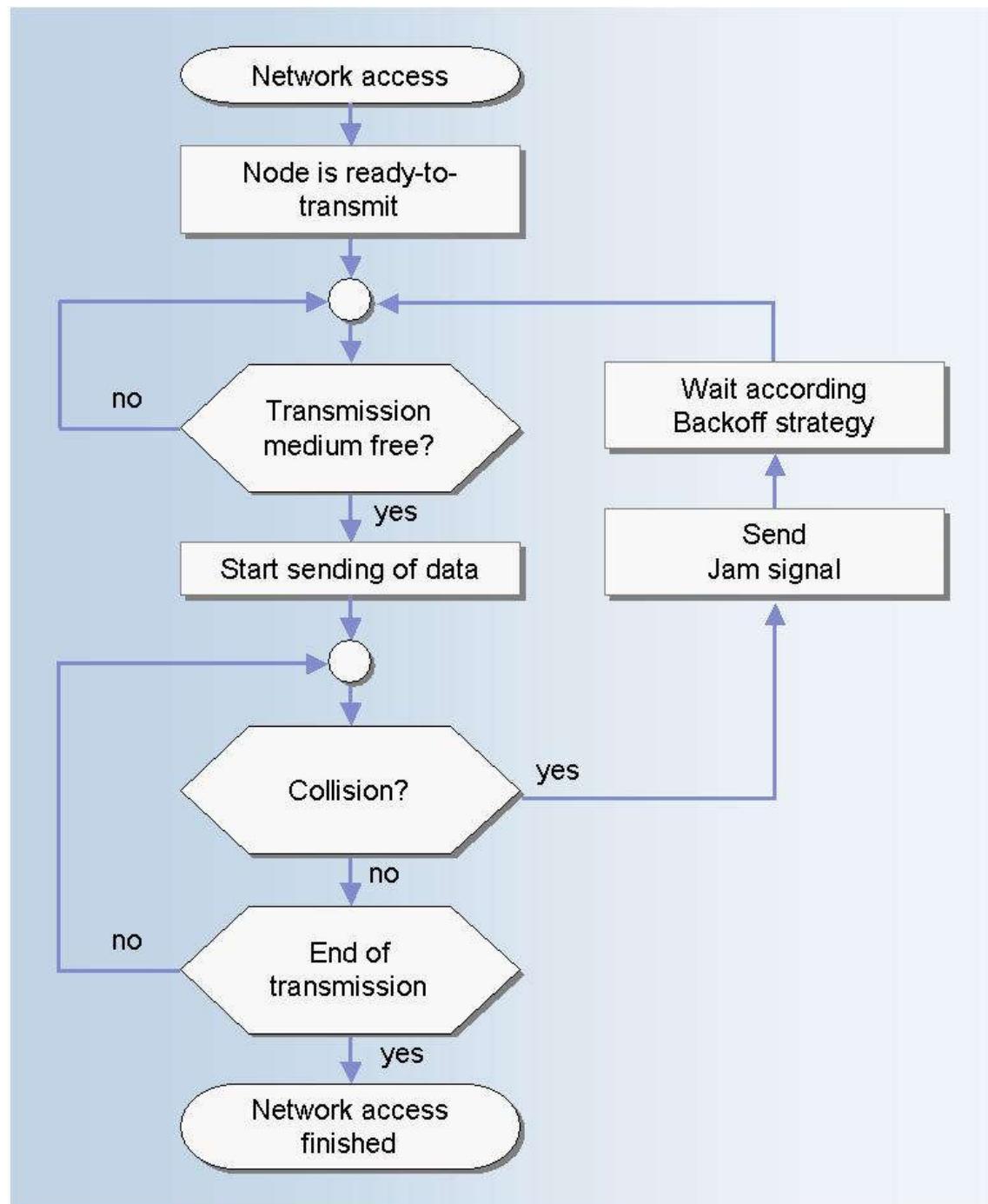
Rețele LAN Ethernet (802.3)

CSMA/CD

- Operează în 3 faze:
 1. sesizarea purtătoarei (*carrier sense*): fiecare stație trebuie să “asculte” dacă mediul este sau nu liber.
 2. accesul multiplu: posibilitatea ca oricare stație care a detectat mediul liber să poată transmite. Acesta poate duce la coliziuni.
 3. detectarea coliziunii (*collision detection*). În timp ce transmite, fiecare stație “ascultă” în continuare mediul pentru detectarea eventualelor coliziuni.
- La detectarea coliziunii este emis un semnal special (*jamming*), având lungimea echivalentă a 32 de biți. Acest semnal permite tuturor stațiilor să ia cunoștiință despre coliziune.
- Durata de așteptare până la reluarea pașilor pentru transmisie este variabilă, fiind dată de un algoritm de revenire (*back-off algorithm*).
- Prin dispozitivele de interconectare se pot crea domenii de coliziune diferite.

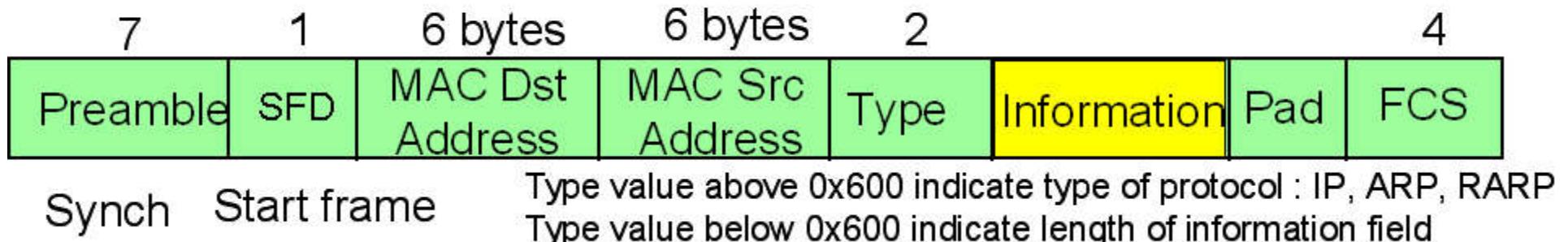
Rețele LAN Ethernet (802.3)

CSMA/CD



Rețele LAN Ethernet (802.3)

Ethernet Frame: 64 --1518 bytes



- Lungimea unui cadru Ethernet este cuprinsă între 64 și 1518 octeți
 - valoarea minimă este stabilită din considerente de detectare a coliziunii, iar cea maximă din considerente de timp legate de ocuparea mediului.
- **Preamble** este folosite pentru sincronizarea ceasului stației receptoare cu ceasul stație transmițătoare.
- Ultimul octet din **Preamble** se numește **SFD(Start Frame Delimiter)** și este folosit pentru a marca începutul cadrului.
- **FCS (Frame Control Sequence)** reprezintă valoarea sumei de control pentru câmpurile anterioare.

Rețele LAN Ethernet (802.3)

Versiuni ale standardului 802.3

- 10Base5, 10Base2, 10Base36 – toate sunt bazate pe cablu coaxial
- 10BaseT, 100BaseT, 10GBaseT, 100GBaseT – pentru cablu cu perechi de fire răsucite (cablu torsadat)
- 10BaseFP, 100BaseFX, 10GBaseR – pentru fibră optică

Rețele WLAN (802.11)

- Principala particularitate a rețelelor wireless este aceea că mediul fizic folosit în acest caz sunt undele radio.
- Acestea au proprietăți total diferite de ale celorlalte medii fizice folosite în comunicațiile de date:
 - este un mediu care nu are o delimitare clară în spațiu.
 - nu este protejat față de interferențele cu alte semnale.
 - are o topologie care se poate modifica ușor.
 - nu putem avea certitudinea că orice stație este „auzită” de către o altă stație.
 - modul de propagare a semnalelor poate varia în timp și poate prezenta asimetrii.

Rețele WLAN (802.11)

IEEE 802.2 Logical Link Control (LLC)		Data Link Layer
IEEE 802.11 Media Access Control (MAC)		
Radio	Infrared	Physical Layer

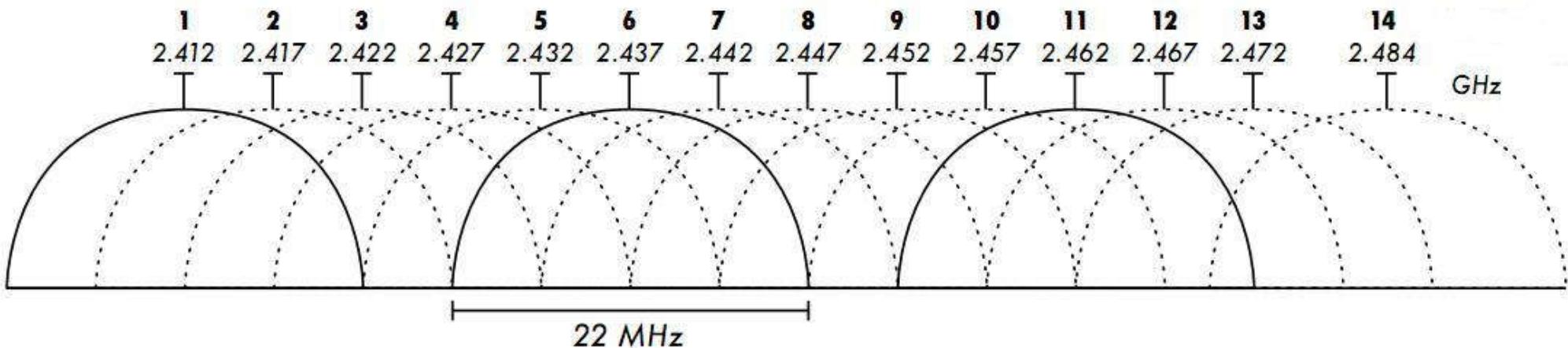
Rețele WLAN (802.11)

Wireless Transmission 802.11 Protocols					
Standards	Year Established	Band Frequency	Maximum Data Transfer	Channel Bandwidth	Antenna Configuration
802.11a	1999	5 GHz	54 Mbps	20 MHz	1 x1 SISO
802.11b	1999	2.4 GHz	11 Mbps	20 MHz	1 x1 SISO
802.11g	2003	2.4 GHz	54 Mbps	20 MHz	1 x1 SISO
802.11n	2009	2.4 & 5 GHz	600 Mbps	20 & 40 MHz	Up to 4x4 MIMO
802.11ac	2013	5 GHz	1.3 Gbps	20, 60, 80, 160 MHz	Up to 3x3 SU-MIMO
802.11ac Wave 2	2015	5 GHz	3.47 Gbps	20, 60, 80, 80+80, 160 MHz	Up to 4x4 SU-MIMO & MU-MIMO

Rețele WLAN (802.11)

802.11b

- Folosește ca metodă de acces la mediu DSSS (Direct Sequence Spread Spectrum) în banda de 2,4 GHz.
- Lățimea de bandă avută la dispoziție este de 97MHz, împărțită în 14 canale, cu doar 3 canale nesuprapuse.
- Lățimea fiecărui canal este de 22MHz, cu o distanță între purtătoare de doar 5MHz.
- Rata maximă de transfer este de 11Mbps, dar ca valoare efectivă se obține maxim 5Mbps.



Rețele WLAN (802.11)

802.11g

- Este o extensie a standardului 802.11b.
- Operează tot în banda de 2,4GHz, dar ca metodă de acces la mediul fizic este folosită tehnologia OFDM (*Orthogonal Frequency Division Multiplexing*).
- Lățimea de bandă oferită este la fel ca și în cazul lui 802.11b, adică de 97MHz, impărțită în 14 canale, cu 3 canale nesuprapuse.
- Rata maximă de transfer este de 54Mbps, dar ca valoare efectivă maximă se obține 22Mbps.
- Datorită compatibilității dintre cele două standarde, un dispozitiv 802.11g va putea comunica cu un dispozitiv 802.11b, dar la rate de transfer de maxim 11Mbps.

Rețele WLAN (802.11)

802.11a

- Operează în banda de 5GHz și de aceea compatibilitatea cu standardele 802.11b și 802.11g nu este posibilă.
- Metoda de acces la mediul fizic este tot OFDM, dar datorită lățimii de banda mai mari (300 MHz) s-au putut obține mai multe canale, existând 8 canale nesuprapuse, față de 3 în cazul benzii de 2,4GHz.
- Rata maximă de transfer este tot de 54Mbps, iar ca rată de transfer efectivă se obține un maxim de 27Mbps, mai mare decât în cazul lui 802.11g.

Rețele WLAN (802.11)

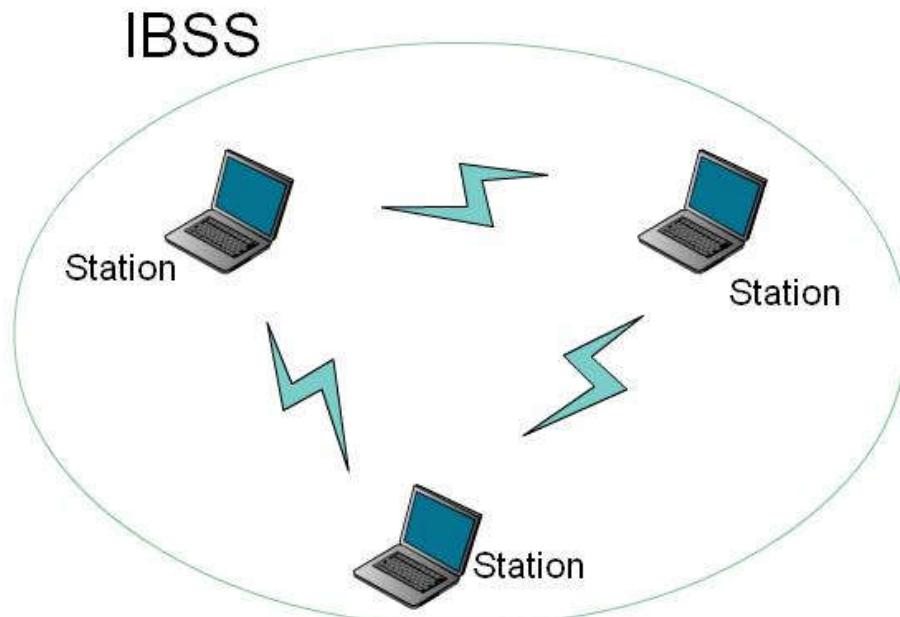
Topologii posibile pentru o rețea 802.11

- O rețea locală de tipul 802.11 se bazează pe o arhitectură de tip celular.
- O celulă poarte denumirea de BSS (*Basic Service Set*) și este controlată de către un AP (*Access Point*).
- AP-ul are un rol de releu pentru stațiile (STA în terminologie 802.11) din interiorul unui BSS, după cum se va vedea în continuare.
- Există trei tipuri de topologii pentru o rețea de tip WLAN:
 - Independent basic service set (IBSS)
 - Basic service set (BSS)
 - Extended service set (ESS)

Rețele WLAN (802.11)

Independend basic sevice set

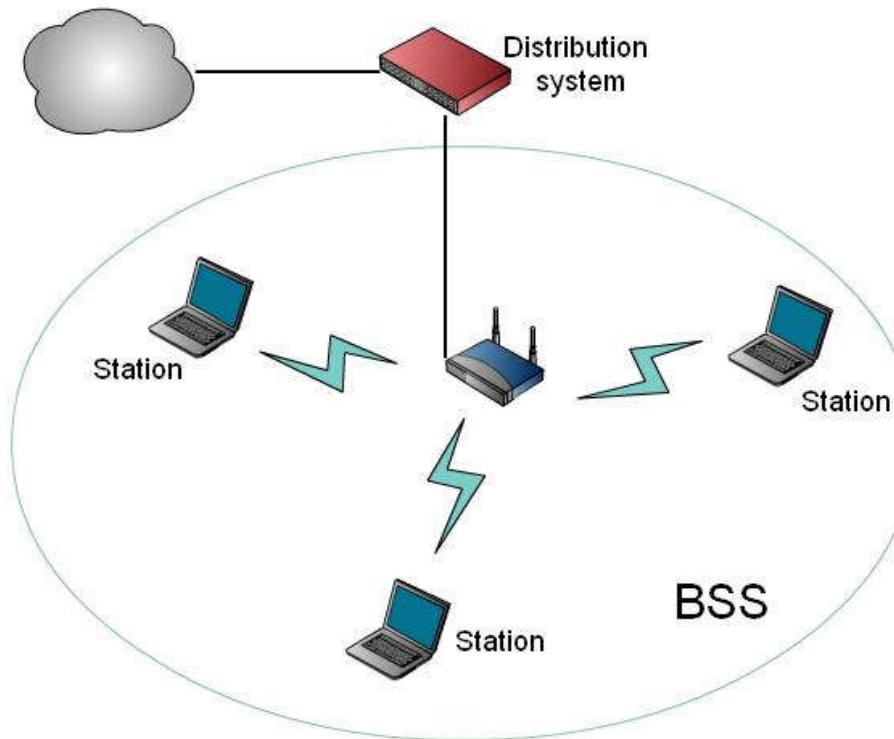
- În acest tip de topologie rețea WLAN este alcătuită dintr-un grup de stații care comunică direct unele cu altele și de aceea mai este numită și rețea ad-hoc.



Rețele WLAN (802.11)

Basic service set

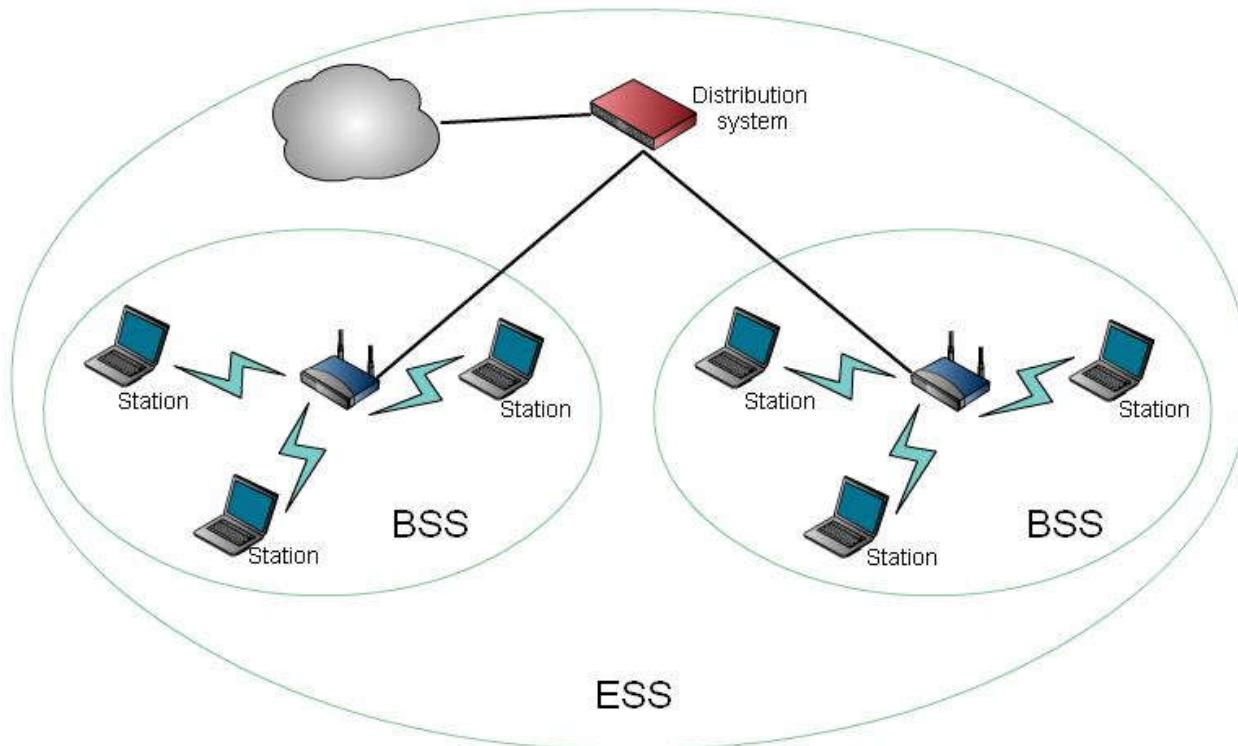
- Stațiile nu vor comunica direct între ele, ci doar cu un dispozitiv specializat, numit Access Point (AP).
- Se creează o topologie de tip celular, o celulă fiind alcătuită dintr-un AP și stațiile conectate la el.



Rețele WLAN (802.11)

Extended service set

- Mai multe AP-uri pot fi conectate între ele prin intermediul unei infrastructuri (ex: Ethernet)



Rețele WLAN (802.11)

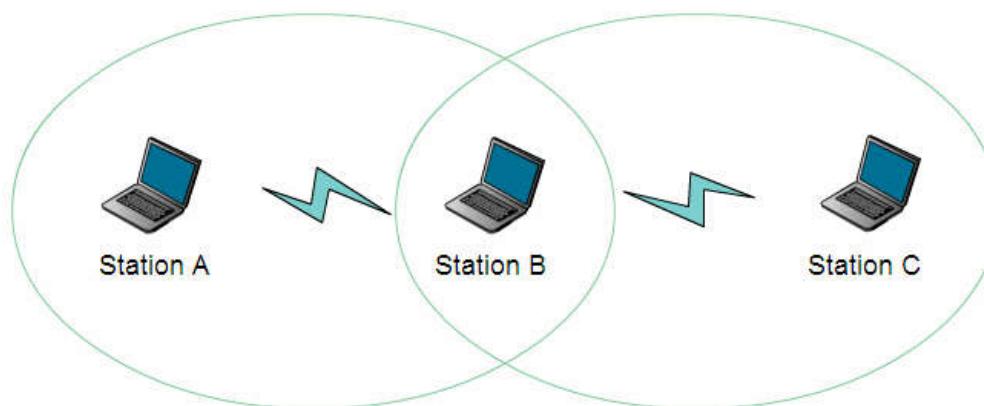
Subnivelul MAC

- În cazul WLAN subnivelul MAC trebuie să îndeplinească următoarele operații:
 - fragmentarea pachetelor
 - transmisia pachetelor
 - retransmisia pachetelor
 - confirmarea pachetelor
- Metoda de acces la mediu este CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), ceea ce înseamnă că se încearcă pe cât posibil evitarea coliziunilor.

Rețele WLAN (802.11)

Subnivelul MAC

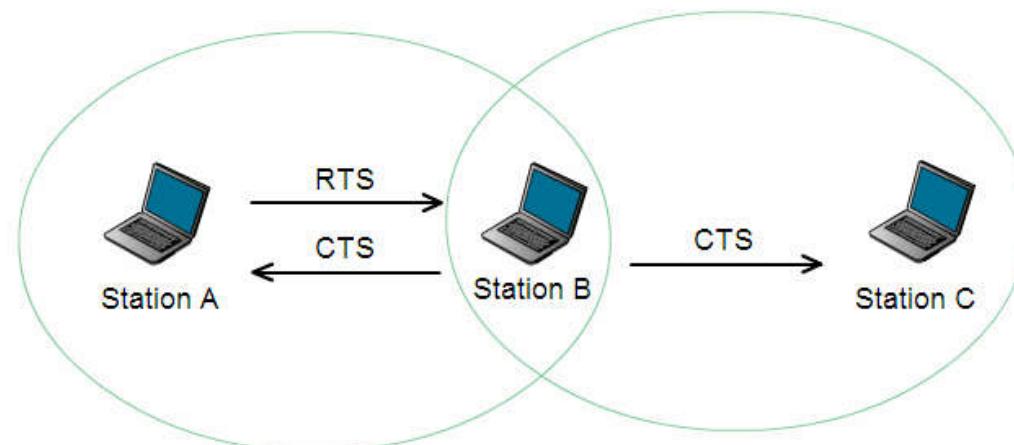
- Metoda *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) folosită în rețelele Ethernet nu ar fi practică în acest caz din două motive:
 - necesită implementarea unui mecanism full duplex de comunicație între stații, ceea ce ar conduce la costuri ridicate de producție
 - nu există certitudinea că stațiile se „aud” toate între ele, adică este posibil ca cel care transmite, să credă că mediul este liber, dar de fapt în zona receptorului mediul să fie ocupat (*the hidden node problem*). În figura de mai jos se observă că stația A nu „aude” conversația dintre B și C și nici stația C nu „aude” conversația dintre A și B.



Rețele WLAN (802.11)

Subnivelul MAC

- Există două modalități de a detecta dacă mediul fizic este liber:
 - detectarea prezenței altor transmisii prin ascultarea propriu-zisă a mediului (*Physical Carrier Sense*).
 - ascultare virtuală a mediului (*Virtual Carrier Sense*). Această metodă presupune folosirea unor pachete de control numite RTS (Request to Send) și CTS (Clear to Send).



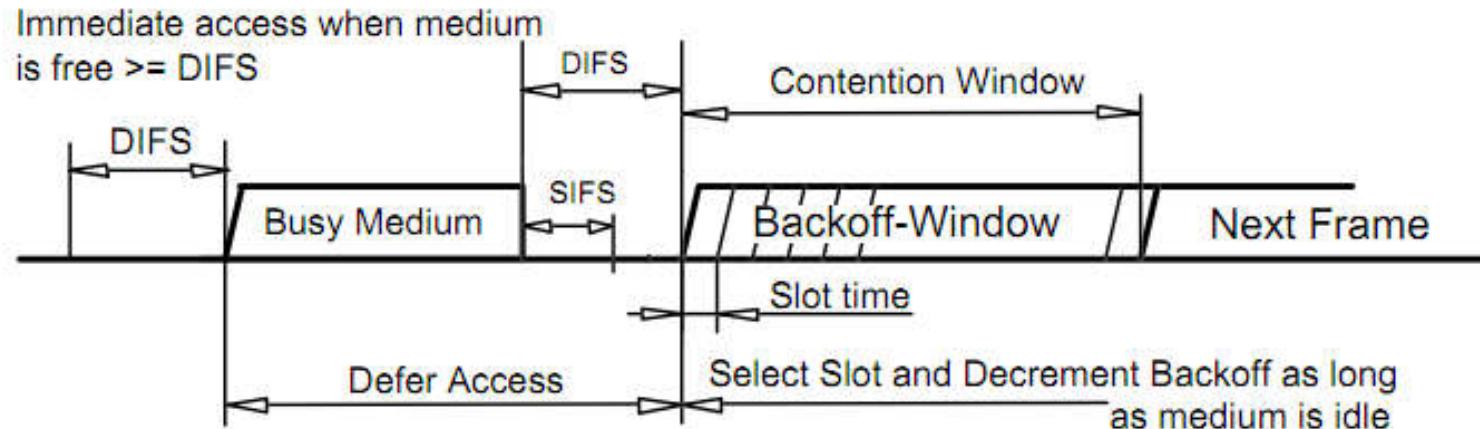
Rețele WLAN (802.11)

Subnivelul MAC - *Physical Carrier Sense*:

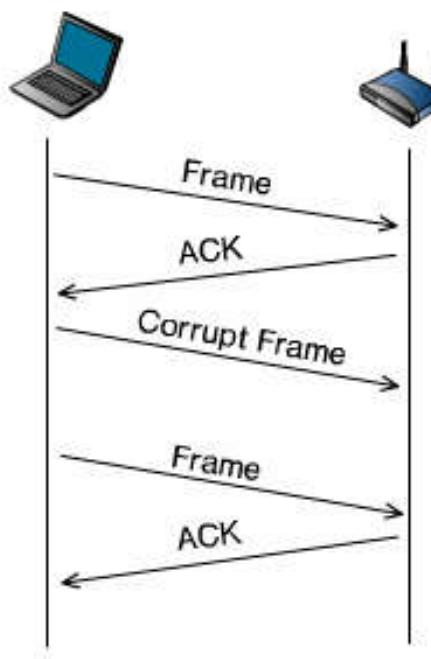
- Stația care transmite ascultă mediul. Dacă acesta este ocupat amână transmisia, iar dacă este liber pentru o perioadă de timp egală cu DIFS (Distributed Inter Frame Space) poate trece la transmiterea pachetelor. Deoarece există o probabilitate destul de mare ca două stații care sesizează că mediul este liber să încerce să transmită simultan, există un mecanism de evitare a unor astfel de situații, prin care stațiile mai așteaptă un interval de timp aleator, și doar după scurgerea acestui interval de timp, dacă mediul este în continuare liber, stația poate trece la transmiterea datelor.
- Stația care recepționează pachetele verifică suma de control care le însoțește, iar apoi le confirmă printr-un pachet de tip ACK. Dacă sursa primește pachetele de confirmare înseamnă că nu a avut loc nici o coliziune. Dacă nu se primește confirmarea înseamnă că a avut loc o coliziune și pachetul care nu a fost confirmat este retransmis.

Rețele WLAN (802.11)

Subnivelul MAC - *Physical Carrier Sense*



- În 802.11 sunt practicate confirmările pozitive (*positive acknowledge*).



Rețele WLAN (802.11)

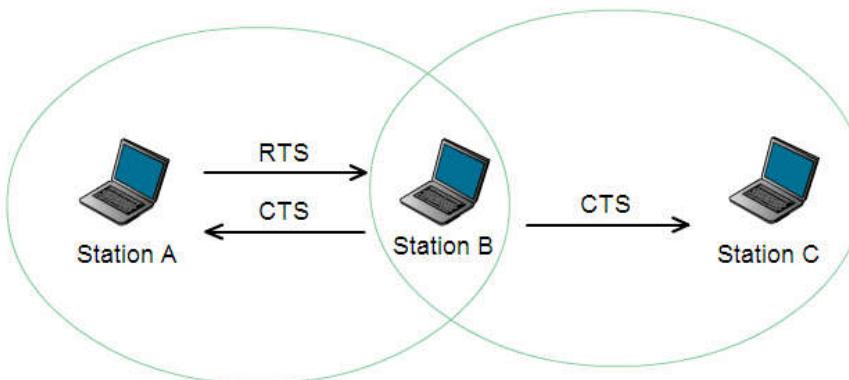
Subnivelul MAC - Virtual Carrier Sense

- Pentru a reduce probabilitatea unor coliziuni, situație care apar frecvent pentru cazul descris de către „the hidden node problem”, standardul a prevăzut și metoda Virtual Carrier Sense.
- O stație care vrea să transmită date, mai întâi trimite un scurt pachet de control numit RTS (Request to Send), care include adresa sursei, adresa destinației și durata transmisiei care urmează să aibă loc, această durată incluzând și receptia pachetului de confirmare, în scopul de a rezerva mediul pentru toate etapele unei transmisii. Dacă mediul este liber, atunci stația destinație răspunde cu un pachet numit CTS (Clear to Send), care conține aceleași informații legate de durata transmisiei.
- Când stațiile învecinate recepționează fie un pachet RTS fie un pachet CTS își setează un indicator numit NAV (*Network Allocation Vector*) în conformitate cu informația de timp conținută în aceste pachete. Acesta este de fapt un timer care este decrementat și doar când ajunge la zero stația poate încerca să transmită din nou, dacă mediul este liber. Dacă una dintre stații nu recepționează pachetul RTS, nefiind în aria de acoperire a acelei stații, atunci ea va recepționa pachetul CTS, care vine ca răspuns la RTS. Prin acest mecanism este rezolvată și „problema nodului ascuns”.

Rețele WLAN (802.11)

Subnivelul MAC - Virtual Carrier Sense

- Chiar dacă stația C nu „aude” pachetul de tip RTS, ea va recepționa pachetul CTS trimis de stația B. Pe baza informației din acest pachet își va seta indicatorul NAV.

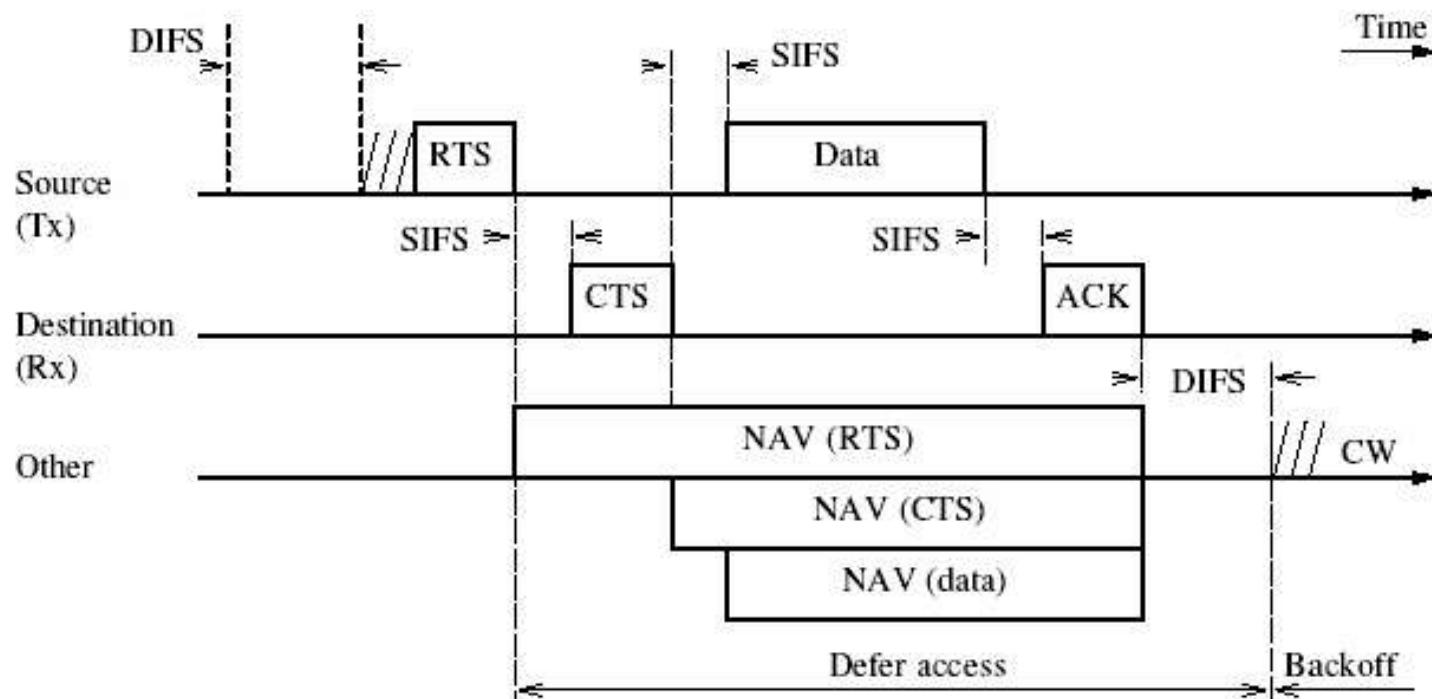


- O stație care vrea să transmită va aștepta un interval de timp egal cu valoarea dată de NAV, iar apoi apelează la algoritmul de tip *backoff* pentru a calcula momentul transmisiei. Mecanismul oferit de timer-ul NAV nu implică neapărat folosirea pachetelor RTS/CTS. Există situații când pachetele de date conțin informații de timp care duc la actualizarea timer-ului NAV.
- Dacă este activat mecanismul RTS/CTS, capacitatea de transfer a rețelei este diminuată. De aceea, acest mecanism este eficient, doar în cazul în care există o densitate relativ mare de stații și există riscul apariției fenomenului *the hidden node problem*.

Rețele WLAN (802.11)

Subnivelul MAC - *Virtual Carrier Sense*

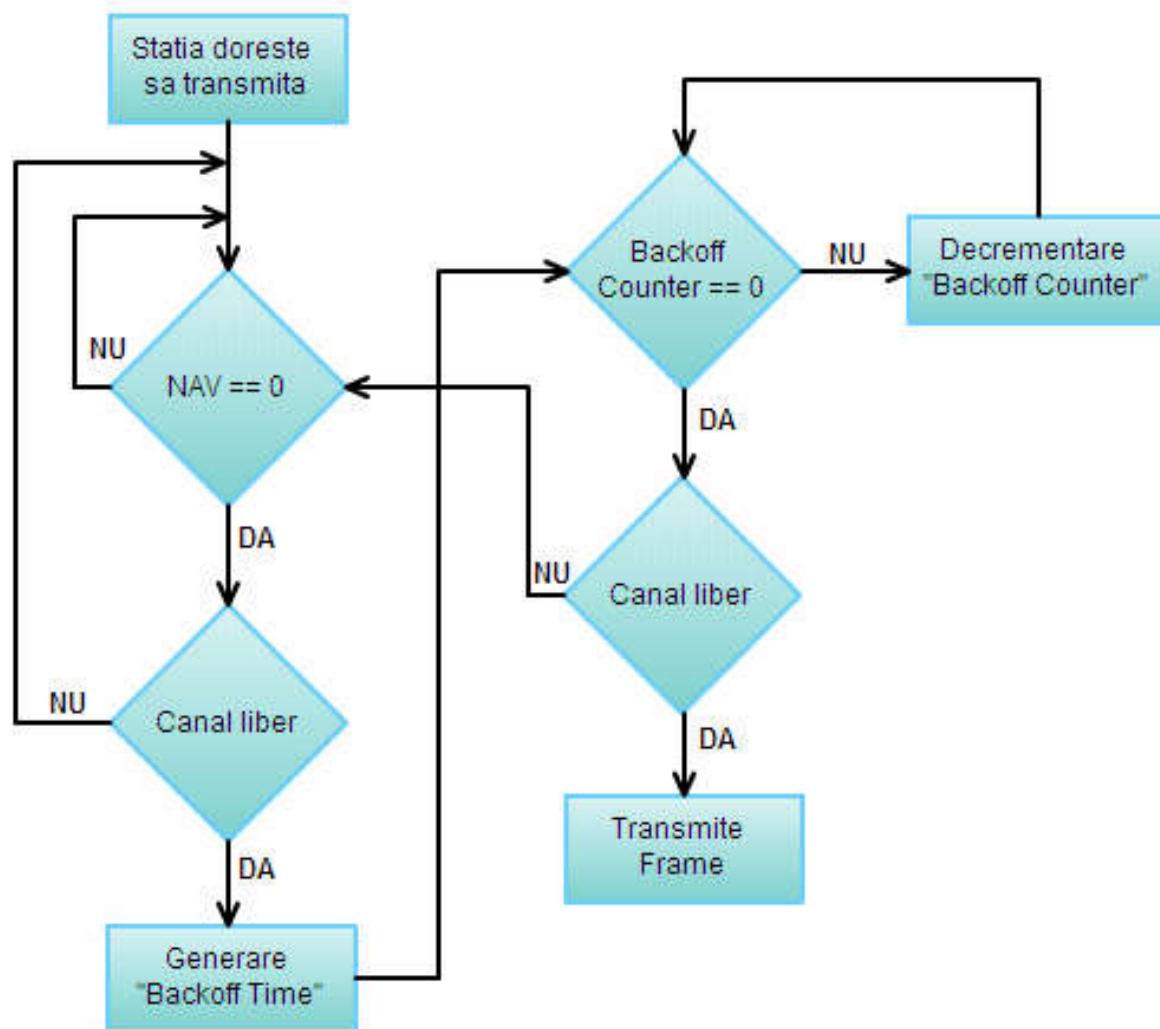
- Pentru cazul ilustrat în figura de mai jos, presupunem că înainte de transmisia datelor, au fost parcursi toți pașii prezențați în diagrama de pe slide-ul următor, pentru a determina dacă sunt îndeplinite toate condițiile care să permită transmisia unui frame.
- *DIFS* (*Distributed InterFrame Space*) este intervalul minim pe durata căruia mediul de transmisie trebuie să fie liber.
- *SIFS* (*Short Inter Frame Space*) Este ales în aşa fel încât să îi permită stației transmițătoare să treacă din modul de transmisie în modul de receptie.
- *NAV* (*Network Allocation Vector*)



Rețele WLAN (802.11)

Subnivelul MAC

- În diagrama de mai jos sunt sintetizate condițiile care trebuie îndeplinite pentru ca o stație sau un access point să poată trece la transmisia unui frame.



Computer Networks

Part 4_II

Data Link Layer

Summary

- Overview
- Protocols: PPP, ATM, Ethernet
- Ethernet in more details

Data Link Layer

Overview

- **The first point to address is why the data link layer is needed at all?**
 - We have addressing in Layer 3, so why is this not sufficient to move a packet from one system to another
- **The answer** to this lies in the **separation of duties** performed by each layer, which provides for a wide variety of **disparate technologies**
 - It could be the case that a Layer 3 protocol such as IP could talk directly to the hardware layer
 - Imagine how many different IP protocol stacks would have to be written
 - Or the IP stack code would be enormously large to account for all the separate devices it would need to interact with
 - Any changes to the IP protocol would necessitate massive code changes for every conceivable type of hardware

Data Link Layer

Overview

- Benefits for separation of duties:
 - Layer 3 protocols usually have no knowledge of the underlying physical infrastructure
 - The data link layer hides the details of the interaction with the physical medium entirely from upper protocols such as IP
 - In theory, this would allow IP to run over any possible physical medium
 - This has led to the mantra, “**IP Over Everything**”
- In the real world, most data link layer protocols support only a very limited number of physical media.
 - Ethernet can be run on only a few carefully specified physical media

Data Link Layer

Overview

- Layer 2/Data Link networks can be classified broadly into three types:
 - Point-to-Point Networks
 - Point-to-point network protocols do not usually require source and destination addresses since they are established between two networking devices only
 - Circuit-Based Networks
 - Circuit-based networks create virtual circuits between different devices over a shared infrastructure
 - Shared Networks
 - Shared networks provide each device with a share of the underlying network medium such as a physical cable or a switch
- These Layer 2 frames (packets) usually consist of:
 - A **circuit identifier** in the case of circuit-based networks
 - An **address** that directs the packet to the required destination in the case of shared media
 - A **maximum transmission unit (MTU)** established between the source and receiving component. Data from higher layers is broken into fixed-length frames
 - An **error check** that is inserted by the source component and verified by the receiving component to verify data integrity on each data link segment

Data Link Layer

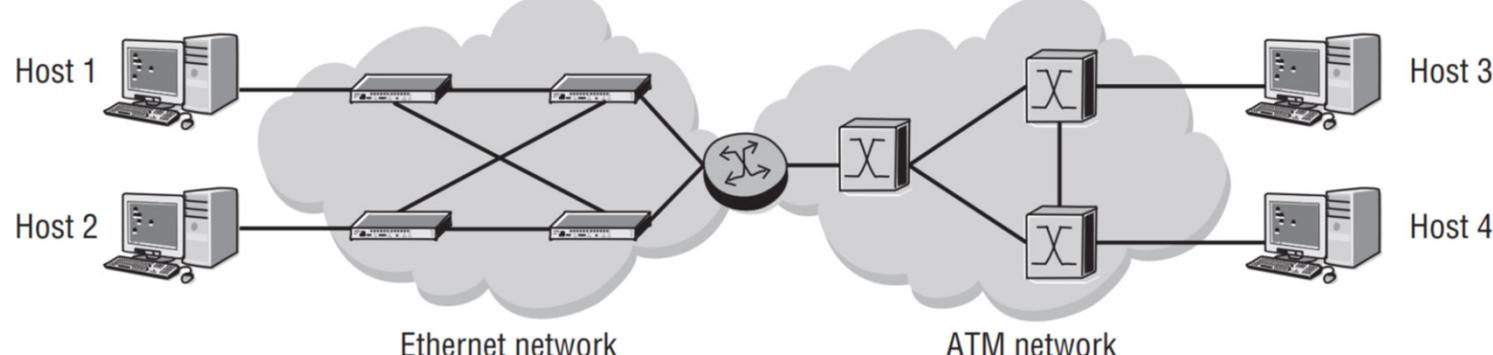
Overview

- Scope of the Data Link Layer
 - The Data Link frame remains intact while it traverses the Layer 2 devices in a particular IP subnet
 - If the IP packet needs to be routed to another subnet via an IP router, the original Data Link frame will be removed after it ingresses the IP router.
 - When forwarding the IP packet out from the appropriate port, the IP router constructs a new Data Link frame with a new header
 - This new Data Link header is used as the frame traverses the next subnet
 - This process continues until the destination host is reached
- The application data sent between two host stations can traverse several physically different networks
 - Each network has a different Data Link header and may even use different Data Link protocols that depend on the physical wire itself, for example, Ethernet, PPP, ATM, or Frame-Relay

Data Link Layer

Overview

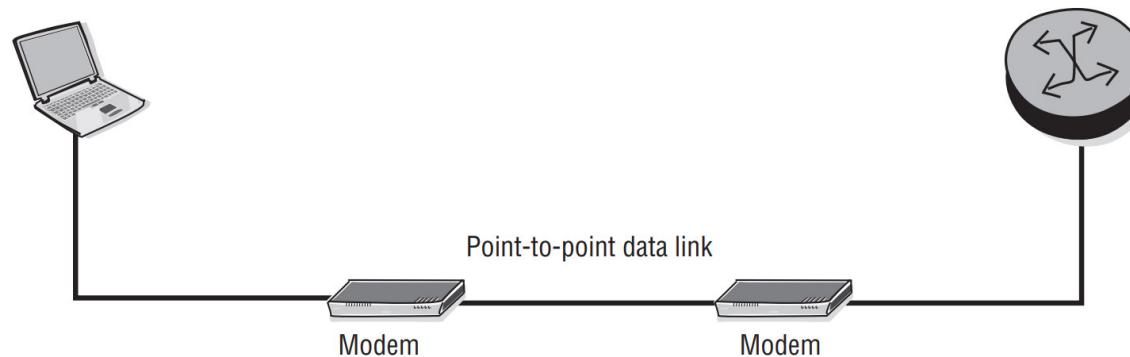
- The PCs on the left side of the Ethernet network do not require anything other than Ethernet Layer 2 framing to talk to each other
- The PCs on the right side of the network similarly require only ATM Layer 2 framing to talk to each other
- The Layer 2 networks are separated by routers, which are Layer 3 OSI devices. The PCs on the Ethernet network can only communicate with the PCs on the ATM network through Layer 3 addresses
- Note that the devices in the ATM cloud represent ATM switches and the devices in the Ethernet cloud represent Ethernet switches
- The device connecting the two clouds is a router



Data Link Layer

PPP: The Point-to-Point Protocol

- In the early days of the Internet, **point-to-point data links** allowed hosts to communicate with each other through the telephone network
- Older protocols such as **Serial Line IP (SLIP)** provided a simple mechanism for framing higher-layer applications for **transmission along serial lines**
- **Serial lines** allow for data to be sent in a **single-byte stream one after another** in “serial”
- **SLIP** was simple enough but could not control the characteristics of the connection
- **Point-to-Point Protocol (PPP)** provides advantages such as link control to negotiate the link characteristics



Typical configuration: a PC using a modem to connect to the Internet or any other dial-up network would use the PPP protocol

Data Link Layer

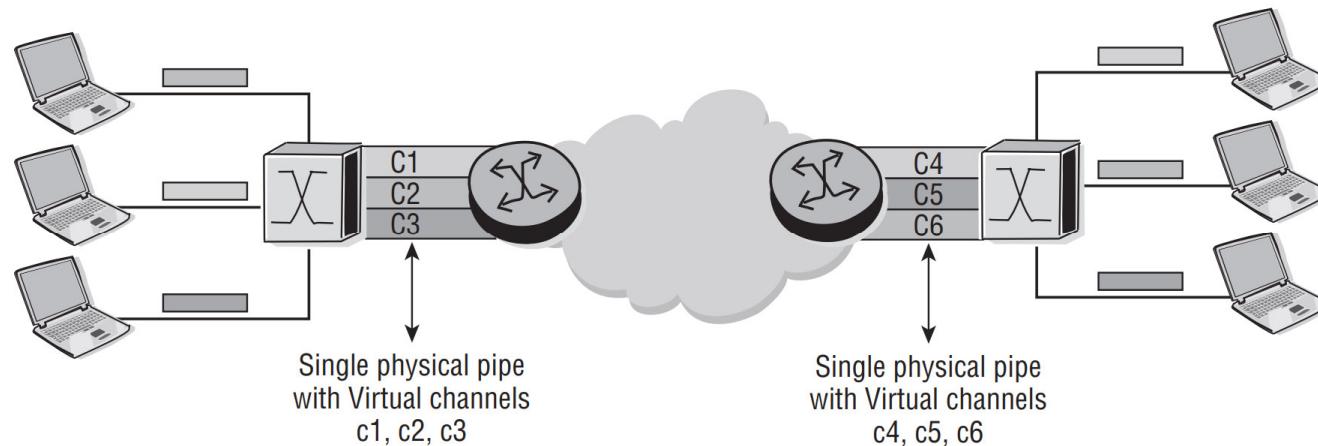
Circuit-Switching

- **Circuit-switched protocols** allow the transfer of user information as a unique set of packets identified by **Virtual Circuits (VC)**
 - A virtual connection may exist only for the duration of a particular network conversation
 - In some cases, the connection can exist for much longer than this
- The reason the **circuit is virtual** is that a VC can be configured over an infrastructure that can support multiple connections
 - This is beneficial because if a given path fails, it is very easy to reconfigure the VC to take another path through the network
 - It also allows for multiple VCs to share the same physical infrastructure

Data Link Layer

Circuit-Switching

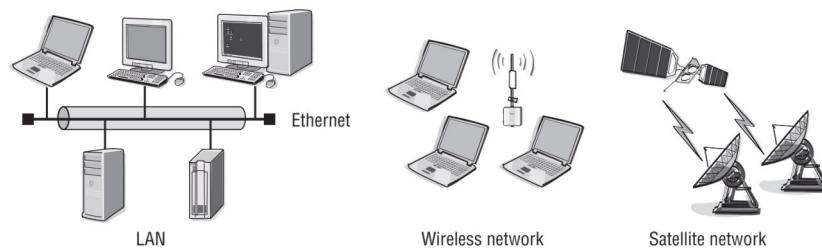
- In the figure, the switch on the left accepts traffic from each host PC into a virtual circuit and switches to another virtual circuit when going to the router.
- The **virtual circuit number** is the same between the host PC and the switch, and between the switch and the router.
- Traffic from each PC is uniquely identified by a **virtual circuit number** at every hop. This allows for **many logical connections to be configured over a single physical connection** and is the predominant way that WAN connections are handled in modern networks.
- The two most predominant circuit-switching technologies are Frame-Relay and Asynchronous Transfer Mode (ATM).



Data Link Layer

Broadcast and Shared Access Data Links

- Unlike point-to-point and circuit-switching networks, **broadcast networks** typically **use a shared media** to communicate to all the devices that are attached to that shared media.
- For data to be reliably delivered from the source to the destination, each of the devices on the shared media is identified by a **unique address**.
- To transmit data reliably, the device on the shared media
 - Must compose the frame
 - Obtain control of the media
 - Then transmit the information
- Since the media is shared, it is possible for multiple stations to transmit their information simultaneously, resulting in a collision.
- This collision causes data corruption
 - An algorithm needs to be followed to ensure a minimum number of collisions and also to ensure proper recovery from collisions.



Examples of shared media technologies where every station receives the same information simultaneously.

Data Link Layer

Ethernet Overview

- Ethernet is a broadcast technology that relies on a shared media for communication
 - It uses a “passive,” wait-and-listen protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
 - It uses data link layer addressing known as Media Access Control (MAC) addresses
 - it provides the ability to send a data frame to all devices on the network simultaneously (broadcasting)
- Two very similar but different standards were developed
 - Both of the standards are still in use, but the Ethernet II standard is by far the more widely accepted

Data Link Layer

Ethernet Overview

- The frame of each type is shown in figure below

802.3

- Frame type defined by IEEE
- Used mainly for IPX



Ethernet II

- Length replaced by type to identify upper-layer protocols
- Most commonly used frame today

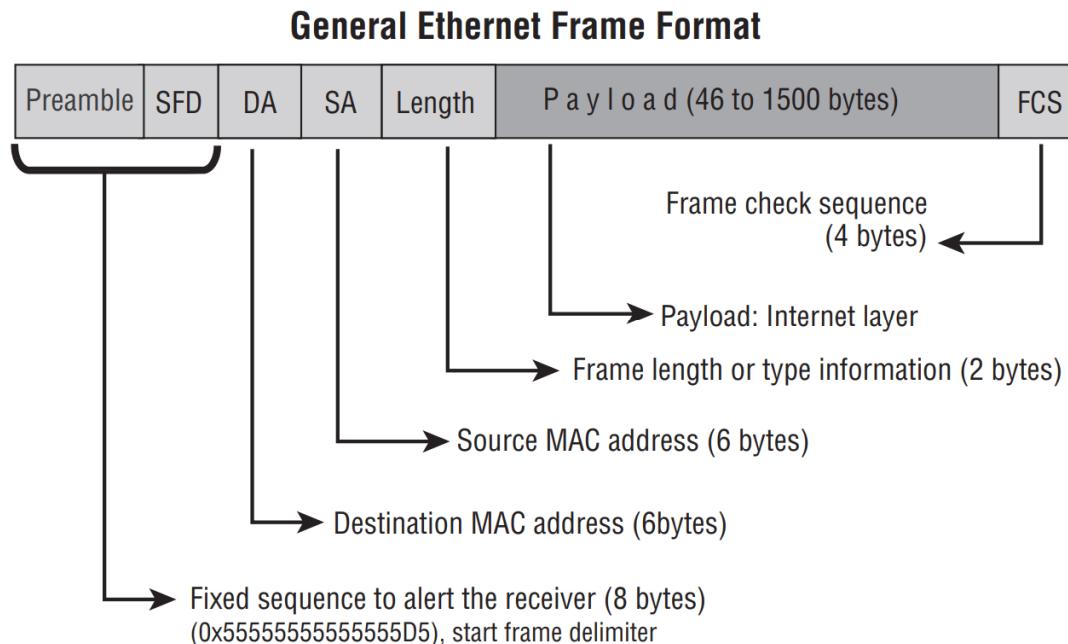


- The 16-bit field that follows the source address (SA) indicates whether the frame is Ethernet II or 802.3
 - If the value is 1,500 or less, the frame is treated as 802.3. If the value is greater than 1,500, the frame is treated as Ethernet II.

Data Link Layer

Ethernet Overview

- The original Ethernet standards defined the minimum frame size as 64 bytes and the maximum as 1,518 bytes
 - These numbers include all bytes from the destination MAC address field to the Frame Check Sequence (FCS) field
 - The preamble and the SFD fields are not included when quoting the size of a frame
 - The IEEE 802.3ac standard released in 1998 extended the maximum allowable frame size to 1,522 bytes to allow for a virtual LAN (VLAN) tag to be inserted into the Ethernet frame format
 - Gigabit Ethernet and 10 gigabit Ethernet ports may support jumbo frames that can be 9,000 bytes



Data Link Layer

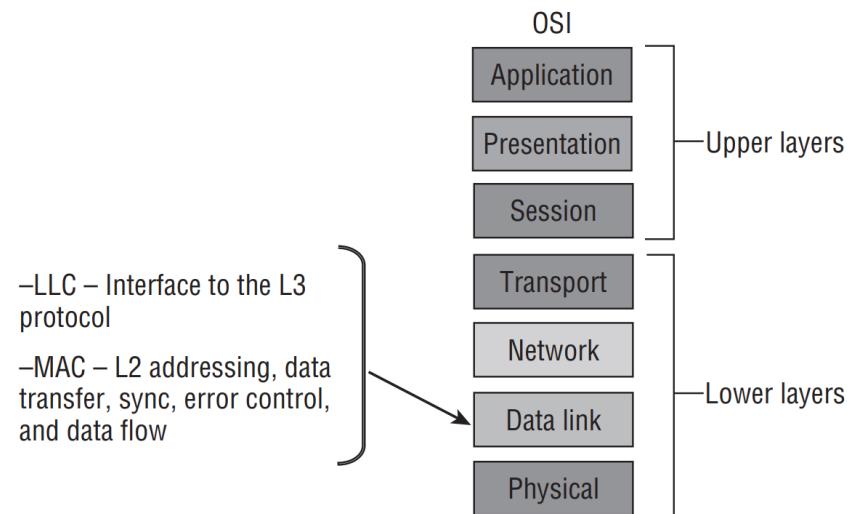
Ethernet Overview

- **Preamble:** a stream of bits used to allow the transmitter and receiver to synchronize their communication
 - The preamble is an alternating pattern of binary 56 ones and zeros
- **Start of Frame Delimiter:** this is always 10101011 and is used to indicate the beginning of the frame information
- **Length/Type:** The payload length or type field, also known as Ethertype
 - If the Ethernet frame is in the 802.3 format, this field is interpreted as length
 - If the Ethernet frame is in the Ethernet II or original DIX format, this field is interpreted as type, or Ethertype
 - The numeric value in this field determines whether the frame is an 802.3 frame or Ethernet II frame. If the value is less than 1,536 (hex value 0x600), it is an 802.3 frame. If the value is equal to or greater than 1,536, it is an Ethernet II frame
- **Data (a.k.a. Payload):** The data is inserted here
 - This is where the IP header and data are placed if you are running IP over Ethernet
- **Frame Check Sequence (FCS):** This is a part of the frame put in place to verify that the information each frame contains is not damaged during transmission
 - If a frame is corrupted during transmission, the FCS carried in the frame will not match with the recipient's calculated FCS. Any frames that do not match the calculated FCS will be discarded

Data Link Layer

Ethernet Overview

- Ethernet resides at the data link layer, and this layer can be subdivided further into two sublayers:
 - the Logical Link Control (LLC) and
 - the Media Access Control (MAC)
- The LLC interfaces between the network interface layer and the higher L3 protocol
 - may provide additional functions such as flow control or retransmission
- The MAC layer is responsible:
 - for determining the physical source and destination addresses for a particular frame and
 - for the synchronization of data transmission and
 - for error checking.
- The Ethernet II type of frame used for IP does not contain an LLC header and therefore does not provide any LLC functions
- IP uses Ethernet II simply for the transmission of frames on a shared media.



Data Link Layer

Ethernet Transmission—CSMA/CD

➤ Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- Make sure that only one host can access the shared media at a time, and provide for a corrective mechanism in the event that two or more hosts try to talk simultaneously

➤ Carrier Sense (CS)

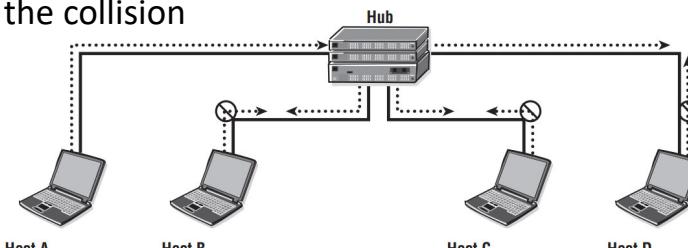
- Each Ethernet LAN-attached host continuously listens for traffic on the medium to determine when gaps between frame transmissions occur. In other words, “sense” (listen to) the “carrier” (the physical media)

➤ Multiple Access (MA)

- When the host senses that the carrier has no other host accessing the physical media, then it can begin transmitting

➤ Collision Detect (CD)

- If two or more hosts in the same CSMA/CD network or collision domain begin transmitting at approximately the same time, the bit streams from the transmitting hosts will interfere (collide) with each other
- Each host must stop transmitting as soon as it has detected the collision and then must wait a random length of time as determined by a back-off algorithm before attempting to retransmit the frame
- In this event, each transmitting host will transmit a 32-bit jam signal alerting all LAN-attached hosts of a collision before running the back-off algorithm. The purpose of the jam signal is to ensure that all hosts have received notice of the collision

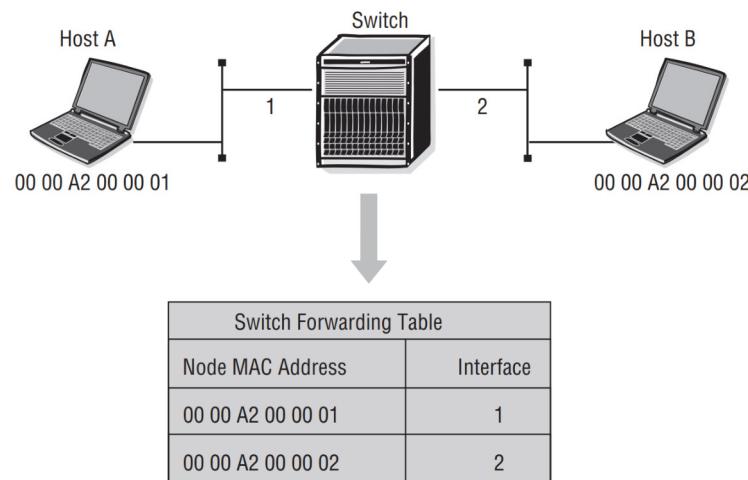


- All hosts constantly listen to the line.
- Host A transmits.
- Host B, C, and D listen to Host A and do not transmit.
- All hosts receive Host A's message.

Data Link Layer

Ethernet Switching Operations

- The Ethernet switch will forward a frame only to the port that needs to receive it
- It performs this function by building a dynamic MAC address table (FDB - forwarding database) that matches MAC addresses to ports so that it “knows” which ports correspond to which MAC addresses
- When the switch receives an Ethernet frame:
 - It records the source MAC address and the interface on which it arrived
 - It looks at the destination MAC address of the frame, compares it to the entries in its MAC FDB, and then transmits the frame out of the appropriate interface
 - If no entry is found, the switch floods the frame out of all its interfaces except the interface on which the frame arrived

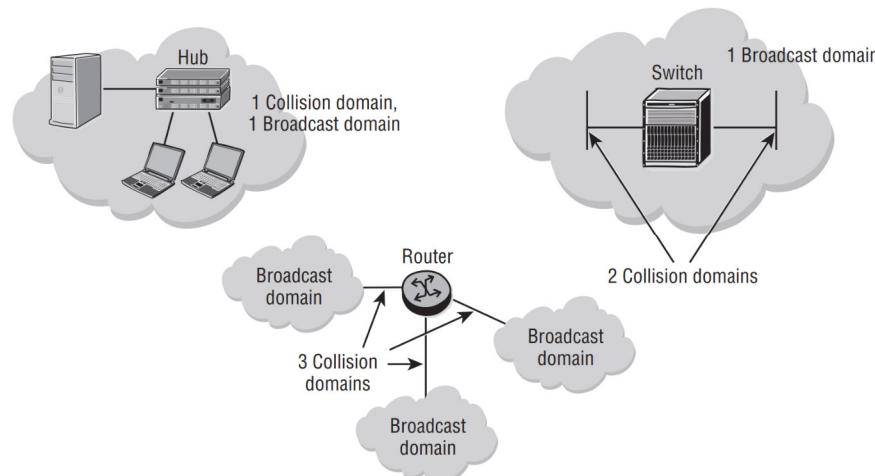


Switches build up their FDB table by recording the source address of frames as they enter each port on the switch

Data Link Layer

Ethernet Switching Operations

- Routers operate at Layer 3 of the OSI model, and so an Ethernet frame would not be forwarded across a router
- This boundary or domain that includes all the Ethernet switches contained by a router boundary is known as a **broadcast domain**
 - Within a broadcast domain, every Ethernet device will receive and process all broadcast packets
- In contrast to a broadcast domain, a **collision domain** exists between devices only within a single wire or hub
 - A collision domain is a group of Ethernet or Fast Ethernet devices in a CSMA/CD LAN that are connected by repeaters or hubs and that compete for access in the network
 - Only one device in the collision domain may transmit at any one time
- Devices on a hub are in a single collision domain, whereas each device on a switch has its own collision domain between the device and its individual port



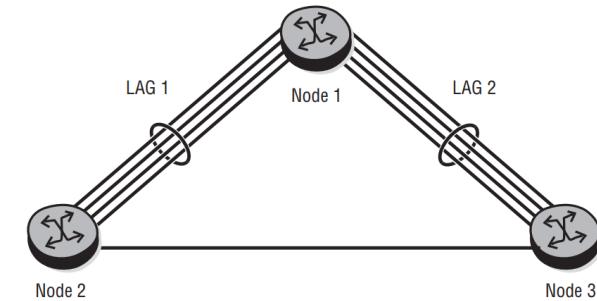
- Hubs provide no separation for collision or broadcast domains
- Switches provide collision domain separation
- Routers provide both collision and broadcast domain separation

Data Link Layer

Ethernet Link Redundancy: LAG

- There are two basic types of redundancy available with Ethernet networks: **link redundancy** and **path redundancy**

- Link redundancy is provided via the **Link Aggregation Group (LAG)** protocol
- Path redundancy is provided by the **Spanning Tree Protocol (STP)**
- The primary difference between link redundancy and path redundancy is that the former does not provide redundancy in the event of a switch failure
- A failure of a single or multiple links between LAG-connected switches would be survivable



- A LAG is based on the IEEE 802.3ad standard
- LAG allows you to aggregate multiple physical links between Ethernet devices so that they are functionally equivalent to a single logical link
 - All frames transmitted between the same source/destination MAC address pair (referred to as a **conversation**) will be transmitted across the same physical link in the bundle
- The primary benefits of LAG are that it increases the bandwidth available between two Ethernet devices by grouping several ports into one logical link
- The aggregation of multiple physical links allows for statistical load sharing and offers seamless redundancy

Data Link Layer

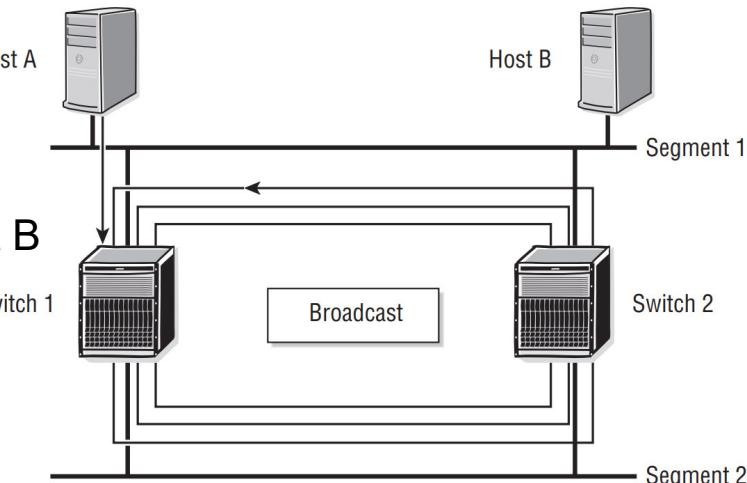
Ethernet Path Redundancy: STP

- LAG is a good solution for providing link redundancy between neighboring Ethernet devices
- If you require end-to-end path redundancy, LAG cannot provide this functionality
- Path redundancy is provided by the Spanning Tree Protocol (STP)
- There are some potential problems associated with providing path redundancy because of the nature of Ethernet switches
 - Providing redundant Ethernet switch paths can result in broadcast storms due to constant “looping” of Ethernet frames
 - A loop exists in a network when a frame or packet exits one interface on a device and then re-enters the device on a second interface.
 - It may also lead to FDB table instability as switches might see source addresses coming in on different interfaces
- In looping scenarios, it is often the case that the FDB table is unstable, resulting in excessive flooding.
 - Without STP, broadcast traffic may increase exponentially because, as the switch receives multiple copies of a frame, it further replicates each frame and transmits them out one or more ports on the switch

Data Link Layer

Ethernet Path Redundancy: STP

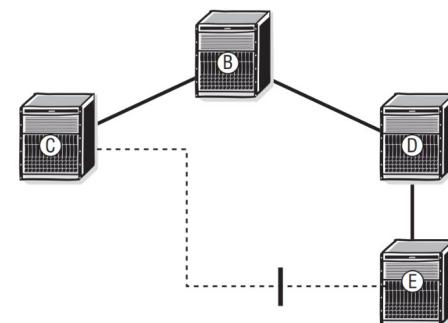
- Host A sends a frame with the destination MAC address of Host B
 - One copy of the frame is received by Host B and processed
- The original frame from Host A is also received by Switch 1
 - Switch 1 records the source MAC of Host A to be on Segment 1
 - Since Switch 1 does not know where Host B is, it replicates the frame and sends it out the port connected to Segment 2
- The original frame is also received by Switch 2 on Segment 1
 - Switch 2 also records the source MAC of Host A to be on Segment 1
 - Since Switch 2 does not know where Host B is, like Switch 1 it replicates the frame and sends it out the port connected to Segment 2
- Switch 2 receives the replicated frame from Switch 1 in Step 2 above via Segment 2
 - Switch 2 removes the existing entry for Host A in the MAC FDB and records that Host A belongs to the port attached to Segment 2
 - Switch 2 then replicates the frame and transmits it out the port attached to Segment 1, where it will be received by Switch 1 on Segment 1
- This process continues indefinitely as both Switch 1 and Switch 2 replicate the original frame from Host A onto Segments 1 and 2, causing excessive flooding and MAC FDB instability



Data Link Layer

Ethernet Path Redundancy: STP

- The Spanning Tree Protocol (STP) was developed to solve these instability and broadcast-storm issues
- STP is intended to prevent loops in an Ethernet switched network
 - It does this by selectively blocking ports to achieve a loop-free topology
 - It determines what ports it can put into a nonfunctioning state to prevent loops from occurring, while still allowing frames to reach every destination in the Ethernet network
- STP uses a root/branch/leaf model, which determines a single path to each leaf spanning the entire switched network
- The sole purpose of STP is to build an active loop-free topology (active in the sense that the ports that are blocked can change in response to changed network conditions)
- Spanning Tree topology can be thought of as a tree that includes the following components:
 - A root (a root bridge/switch)
 - Branches (LANs and designated bridges/switches)
 - Leaves (end nodes)

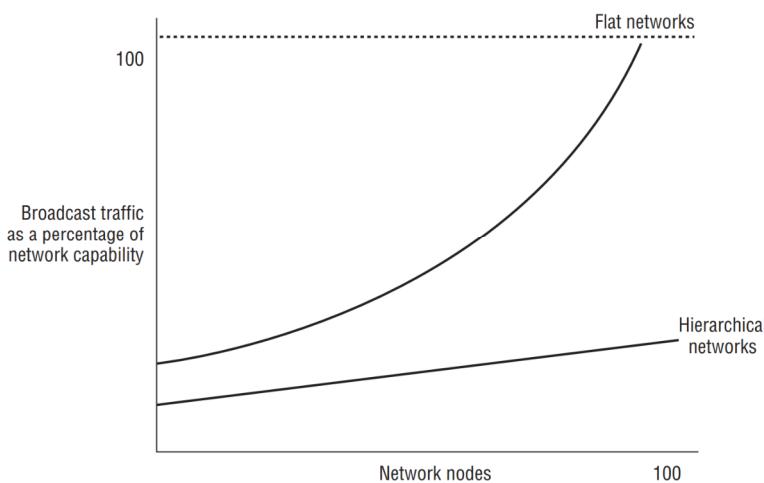


- STP will block the ports between Switches C and E, ensuring a loop-free topology in the switched network

Data Link Layer

Virtual LANs

- A virtual LAN (VLAN) is a mechanism that allows you to segregate devices, and their associated traffic from other devices and traffic
- There are two main reasons to use VLANs:
 - To decrease the amount of broadcast traffic
 - To increase the security of your network



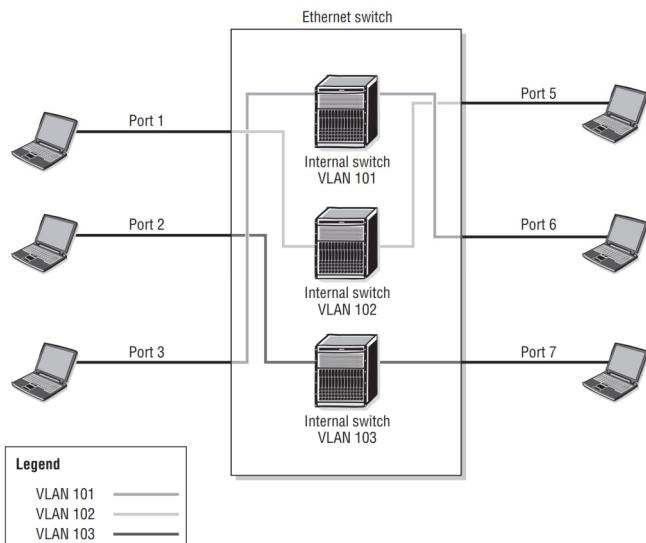
- As broadcasts increase on a flat network, they quickly consume all available network resources
- A flat network means a network without routers or without VLANs or both
- By segregating a group of devices to a particular VLAN, a switch will block broadcasts from devices in that VLAN to devices that are not in that VLAN

- VLANs also have the benefit of added security by separating the network into distinct logical networks.
 - Traffic in one VLAN is separated from another VLAN as if they were physically separate networks

Data Link Layer

Virtual LANs

- While in theory there are many ways to create VLANs, such as by MAC address, IP address, workstation names, and so on, in practice, these methods are very cumbersome
- The primary way that VLANs are created in modern networks is by physical port
- Each VLAN is identified by a VLAN ID (VID),
 - This is usually a number such as 100, 101, and the like
 - They can reside on only a single switch, or they can be distributed throughout the entire network on each switch
- You can think of a VLAN as a broadcast domain, and, in fact, that is exactly what it is



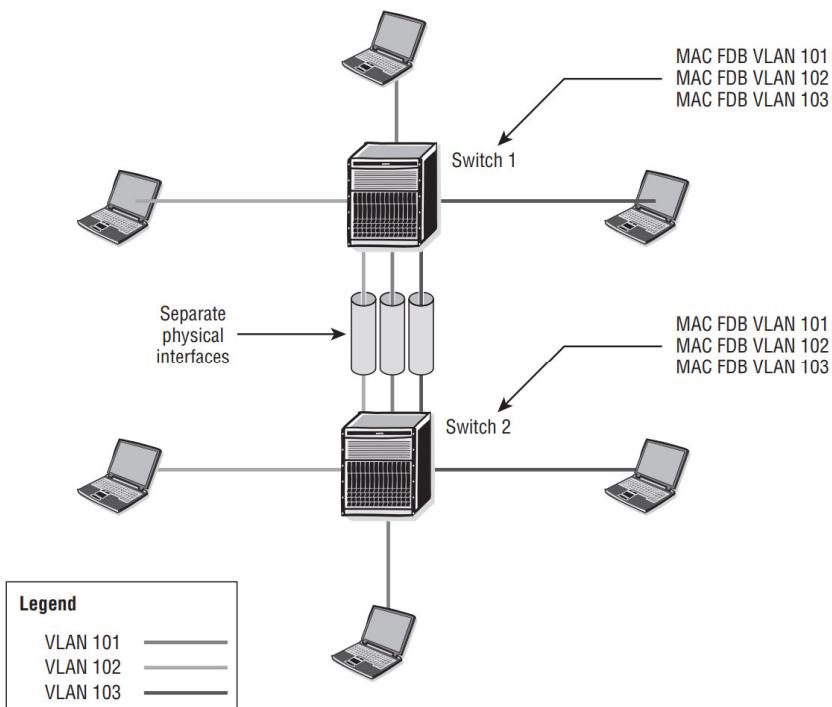
VLANs provide for logical separation of devices on the same physical switch

Data Link Layer

Virtual LANs

VLAN Trunking

- Considering VLANs that are shared across multiple switches
 - Frames ingressing a port in a particular VLAN will only be allowed to egress a port on the same VLAN, regardless of the switch it exits

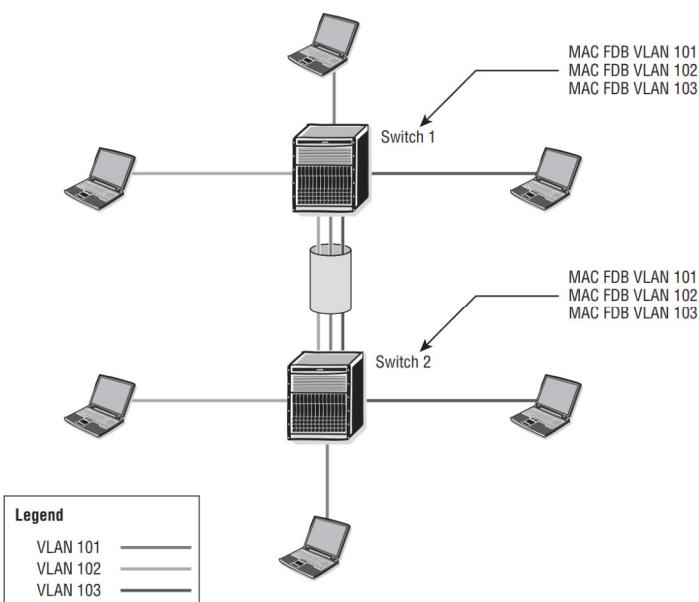


- In this case, there is a separate physical interswitch link for each VLAN
- That might be acceptable for two or three VLANs, but it is not a very scalable or practical solution
- This is where VLAN “trunking” comes into play

Data Link Layer

Virtual LANs - VLAN trunking

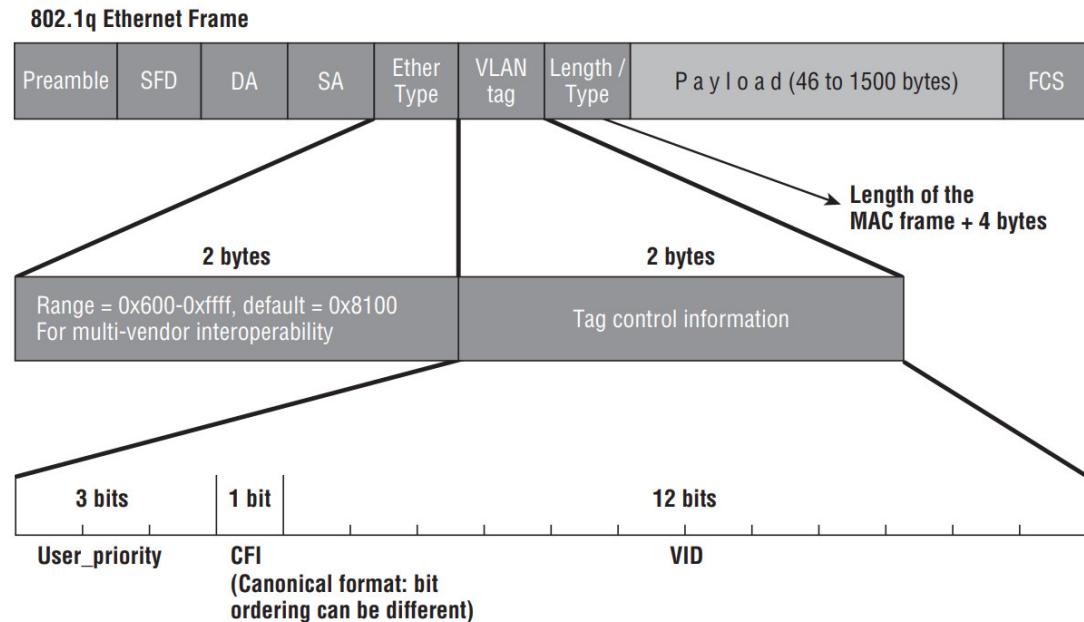
- When a frame is leaving a switch with another switch as its destination, the egress switch will tag the frames with a VID so that the ingress switch knows which VLAN the frame belongs to
 - The IEEE 802.1q standard governs the format of the assigned tag
 - The procedure works by inserting a 32-bit VLAN header into the Ethernet frame of all network traffic for a VLAN as it exits the egress switch
 - The VID uses 12 bits of the 32-bit VLAN header
 - The ingress switch then uses the VID to determine which FDB it will use to find the destination
 - After a frame reaches the destination switch port and before the frame is forwarded to the end destination, the VLAN header is removed



- There is a single VLAN trunk port between the switches that carries traffic for all VLANs by tagging the frames with the correct VID on egress to the other switch

Data Link Layer

Virtual LANs - VLAN trunking

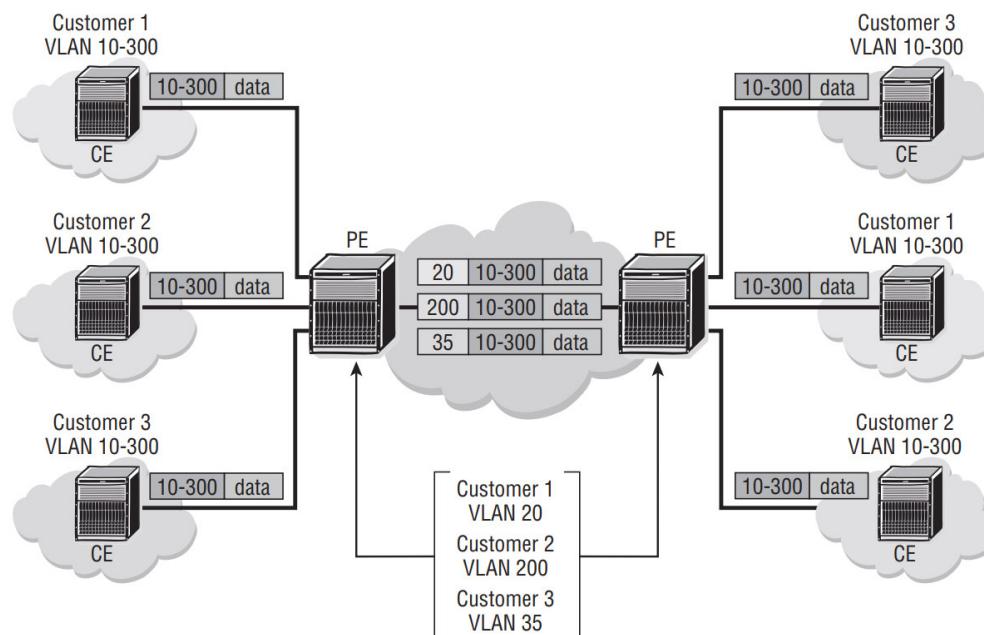


- **Priority Value (User Priority):** A 3-bit value that specifies a frame's priority
 - **CFI:** A single bit. A setting of 0 means that the MAC address information is in its simplest form.
 - **VID:** A 12-bit value that identifies the VLAN that the frame belongs to. If the VID is 0, the tag header contains only priority information
-
- https://en.wikipedia.org/wiki/IEEE_802.1Q

Data Link Layer

Virtual LANs - VLAN trunking

- A restriction of Ethernet VLANs is the limited number of VIDs
 - Because VLANs 0 and 4095 are reserved, a Provider Edge (PE) router (connection to the customer) is really only capable of supporting 4094 VLANs - not a significant number if it is compared with the expanding rates of networks
 - While 4094 might seem sufficient, a single PE router might support hundreds or even thousands of customers, and the number of available VIDs can quickly evaporate
- One of the solutions to this restriction is VLAN stacking, also known as Q-in-Q



Data Link Layer

Virtual LANs

VLAN Trunking

- VLAN trunking provides efficient interswitch forwarding of VLAN frames
 - It allows a single Ethernet port to carry frames from multiple VLANs instead of the “one link per VLAN” approach
- The sharing of VLANs between switches is achieved by the insertion of a header or “tag” with a 12-bit VID
- A VID must be assigned for each VLAN
 - Assigning the same VID to VLANs on different connected switches can extend the VLAN (broadcast domain) across a network
- When a frame is leaving a switch with another switch as its destination, the egress switch will tag the frames with a VID so that the ingress switch knows which VLAN the frame belongs to
- The IEEE 802.1q standard governs the format of the assigned tag
 - The procedure works by inserting a 32-bit VLAN header into the Ethernet frame of all network traffic for a VLAN as it exits the egress switch
 - The VID uses 12 bits of the 32-bit VLAN header
 - The ingress switch then uses the VID to determine which FDB it will use to find the destination

Data Link Layer

Reliable Transmission

- Link: <https://book.systemsapproach.org/direct/reliable.html>

- CRC is used to detect errors
- Some error codes are strong enough to correct errors
- The overhead is typically too high
- Corrupted frames must be discarded
- A link-level protocol that wants to deliver frames reliably must recover from these discarded frames
- This is accomplished using a combination of two fundamental mechanisms:
 - Acknowledgements and
 - Timeouts

Data Link Layer

Reliable Transmission

- An **acknowledgement** (ACK for short) is a small control frame that a protocol sends back to its peer saying that it has received the earlier frame
 - A control frame is a frame with header only (no data)
- The receipt of an acknowledgement indicates to the sender of the original frame that its frame was successfully delivered
- If the sender does not receive an acknowledgment after a reasonable amount of time, then it retransmits the original frame
 - The action of waiting a reasonable amount of time is called a **timeout**
- The general strategy of using acknowledgements and timeouts to implement reliable delivery is sometimes called **Automatic Repeat reQuest (ARQ)**

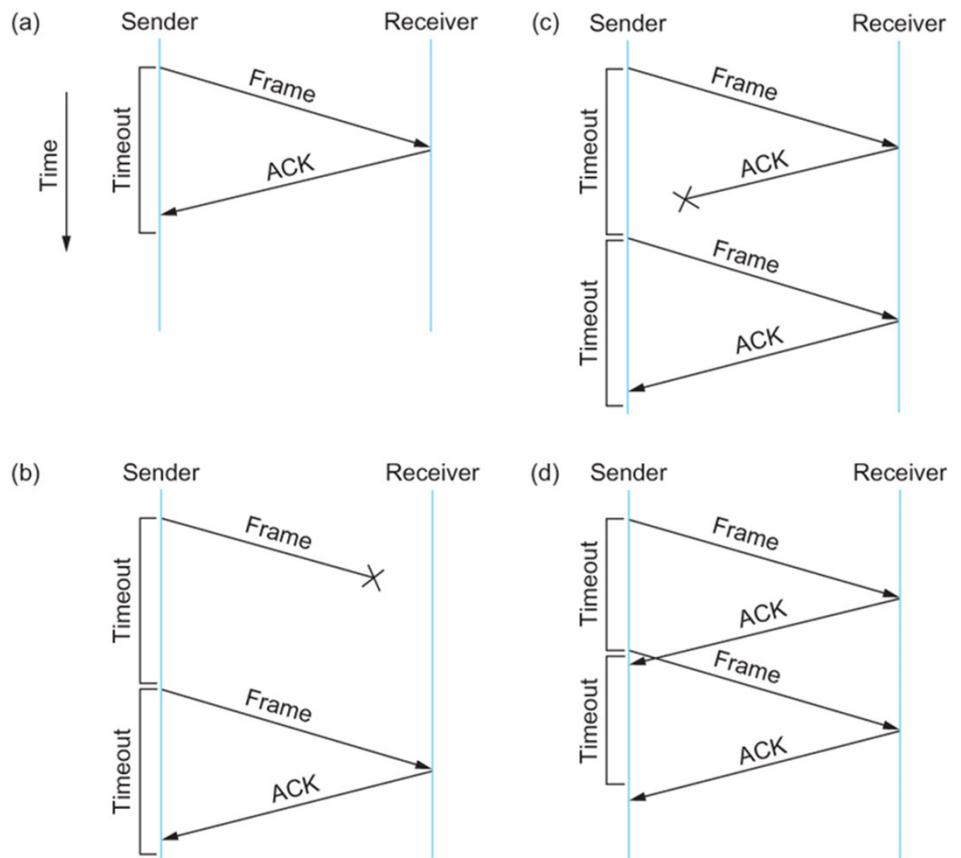
Data Link Layer

Stop-and-Wait

- Idea of stop-and-wait algorithm is straightforward
 - After transmitting one frame, the sender waits for an acknowledgement before transmitting the next frame
 - If the acknowledgement does not arrive after a certain period of time, the sender times out and retransmits the original frame

- Timeline showing four different scenarios for the stop-and-wait algorithm

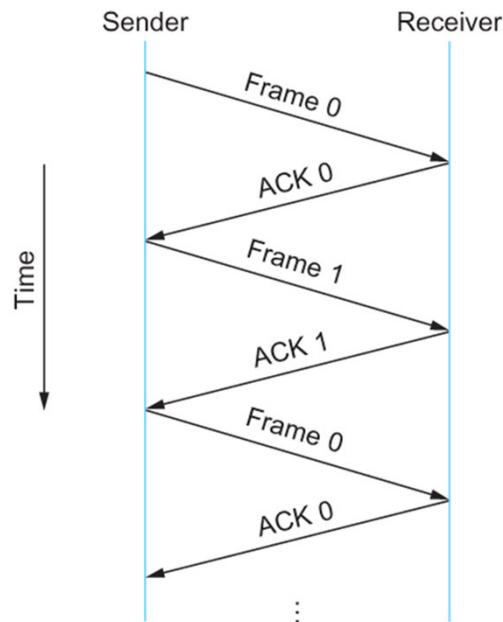
- (a) The ACK is received before the timer expires
- (b) The original frame is lost
- (c) The ACK is lost
- (d) The timeout fires too soon



Data Link Layer

Stop-and-Wait

- If the acknowledgment is lost or delayed in arriving:
 - The sender times out and retransmits the original frame, but the receiver will think that it is the next frame since it has correctly received and acknowledged the first frame
 - As a result, duplicate copies of frames will be delivered
- How to solve this:
 - Use 1 bit sequence number (0 or 1)
 - When the sender retransmits frame 0, the receiver can determine that it is seeing a second copy of frame 0 rather than the first copy of frame 1 and therefore can ignore it (the receiver still acknowledges it, in case the first acknowledgement was lost)



Data Link Layer

Stop-and-Wait

- The sender has only one outstanding frame on the link at a time
 - This may be far below the link's capacity
- Consider a 1.5 Mbps link with a 45 ms RTT
 - The link has a **delay × bandwidth** product of 67.5 Kb or approximately 8 KB
 - Since the sender can send only one frame per RTT and assuming a frame size of 1 KB
 - Maximum Sending rate is

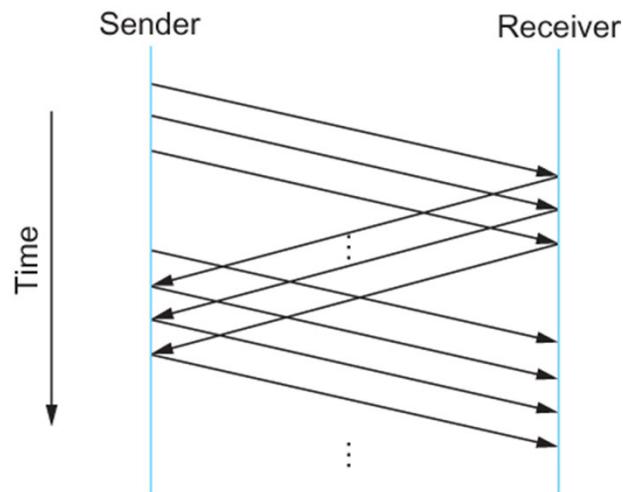
Bits per frame / Time per frame = $1024 \times 8 / 0.045 = 182 \text{ Kbps}$

or about one-eighth of the link's capacity
 - To use the link fully, then sender should transmit up to eight frames before having to wait for an acknowledgement

Data Link Layer

Sliding Window

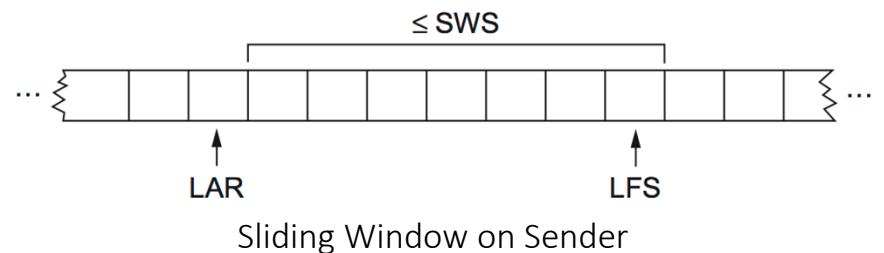
- The significance of the **delay × bandwidth** product is that:
 - It represents the amount of data that could be in transit
 - We would like to be able to send this much data without waiting for the first acknowledgment
 - The principle at work here is often referred to as keeping the pipe full
 - The algorithms presented in the following section do exactly this



Data Link Layer

Sliding Window

- The sliding window algorithm works as follows
- Sender assigns a sequence number denoted as **SeqNum** to each frame
 - Assume it can grow infinitely large
- Sender maintains three variables:
 - **Sending Window Size (SWS)**
 - Upper bound on the number of outstanding (unacknowledged) frames that the sender can transmit
 - **Last Acknowledgement Received (LAR)**
 - Sequence number of the last acknowledgement received
 - **Last Frame Sent (LFS)**
 - Sequence number of the last frame sent
- Sender also maintains the following invariant
$$\text{LFS} - \text{LAR} \leq \text{SWS}$$
- When an acknowledgement arrives
 - the sender moves LAR to right, thereby allowing the sender to transmit another frame
- Also, the sender associates a timer with each frame it transmits
 - It retransmits the frame if the timer expires before the ACK is received
- Note that the sender has to be willing to buffer up to SWS frames (WHY?)



Data Link Layer

Sliding Window

- Receiver maintains three variables

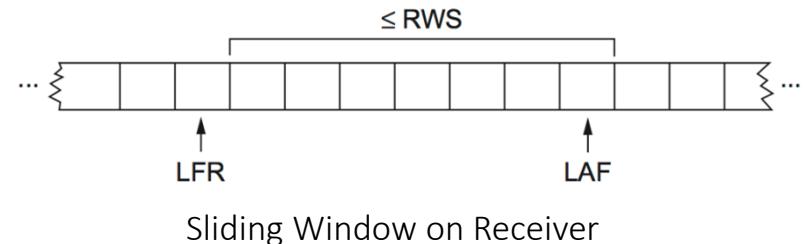
- **Receiving Window Size (RWS)**
 - Upper bound on the number of out-of-order frames that the receiver is willing to accept
- **Largest Acceptable Frame (LAF)**
 - Sequence number of the largest acceptable frame
- **Last Frame Received (LFR)**
 - Sequence number of the last frame received

- Receiver also maintains the following invariant

$$\text{LAF} - \text{LFR} \leq \text{RWS}$$

- When a frame with sequence number SeqNum arrives:

- If $\text{SeqNum} \leq \text{LFR}$ or $\text{SeqNum} > \text{LAF}$
 - Discard it (the frame is outside the receiver window)
- If $\text{LFR} < \text{SeqNum} \leq \text{LAF}$
 - Accept it
 - Now the receiver needs to decide whether or not to send an ACK
 - Let SeqNumToAck
 - The largest sequence number not yet acknowledged, such that all frames with sequence number less than or equal to SeqNumToAck have been received
 - The receiver acknowledges the receipt of SeqNumToAck even if high-numbered packets have been received
 - This acknowledgement is said to be cumulative.
 - The receiver then sets
 - $\text{LFR} = \text{SeqNumToAck}$ and adjusts
 - $\text{LAF} = \text{LFR} + \text{RWS}$



Data Link Layer

Sliding Window

- For example, suppose LFR = 5 and RWS = 4
 - (i.e. the last ACK that the receiver sent was for seq. no. 5)
 - LAF = 9
- If frames 7 and 8 arrive, they will be buffered because they are within the receiver window
 - But no ACK will be sent since frame 6 is yet to arrive
 - Frames 7 and 8 are out of order
- Frame 6 arrives (it is late because it was lost first time and had to be retransmitted)
 - Now Receiver Acknowledges Frame 8
 - and bumps LFR to 8
 - and LAF to 12

Data Link Layer

Sliding Window

- When timeout occurs, the amount of data in transit decreases
 - Since the sender is unable to advance its window
- When the packet loss occurs, this scheme is no longer keeping the pipe full
 - The longer it takes to notice that a packet loss has occurred, the more severe the problem becomes
- How to improve this
 - *Negative Acknowledgement (NAK)*
 - *Additional Acknowledgement*
 - *Selective Acknowledgement*

Data Link Layer

Sliding Window

- *Negative Acknowledgement (NAK)*
 - Receiver sends NAK for frame 6 when frame 7 arrives (in the previous example)
 - However, this is unnecessary since sender's timeout mechanism will be sufficient to catch the situation
- *Additional Acknowledgement*
 - Receiver sends additional ACK for frame 5 when frame 7 arrives
 - Sender uses duplicate ACK as a clue for frame loss
- *Selective Acknowledgement*
 - Receiver will acknowledge exactly those frames it has received, rather than the highest number frames
 - Receiver will acknowledge frames 7 and 8
 - Sender knows frame 6 is lost
 - Sender can keep the pipe full (additional complexity)

Data Link Layer

Sliding Window

- Sequence Number serves three different roles
 - Reliable
 - Preserve the order
 - Each frame has a sequence number
 - The receiver makes sure that it does not pass a frame up to the next higher-level protocol until it has already passed up all frames with a smaller sequence number
 - Frame control
 - Receiver is able to throttle the sender
 - Keeps the sender from overrunning the receiver
 - From transmitting more data than the receiver is able to process

Rețele de calculatoare

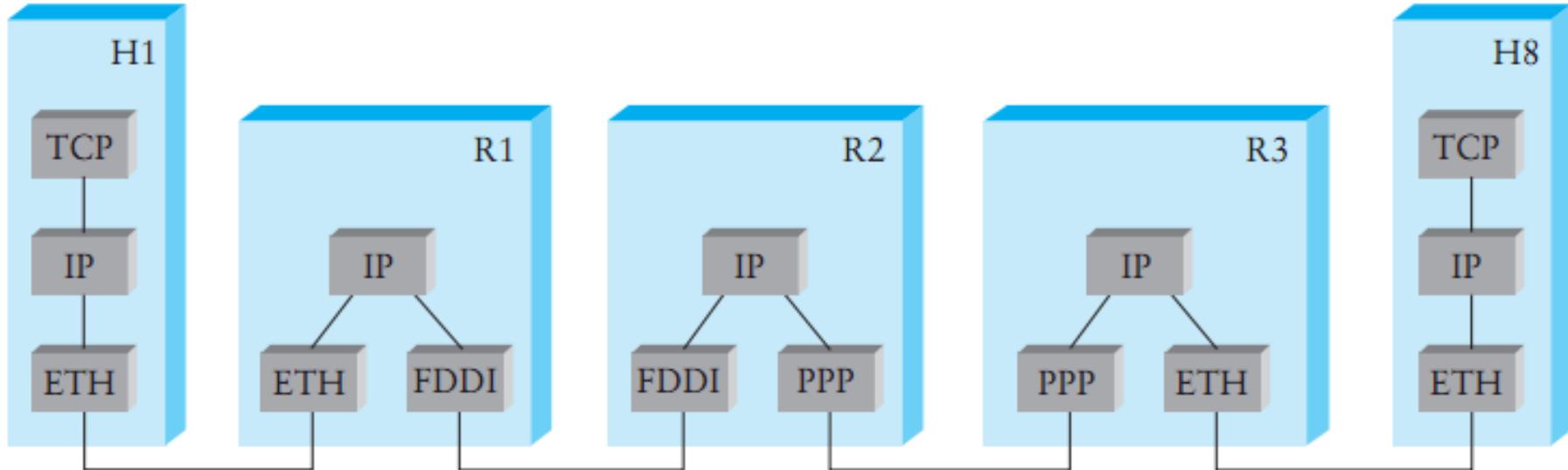
Partea a 5-a

Sebastian Fuicu

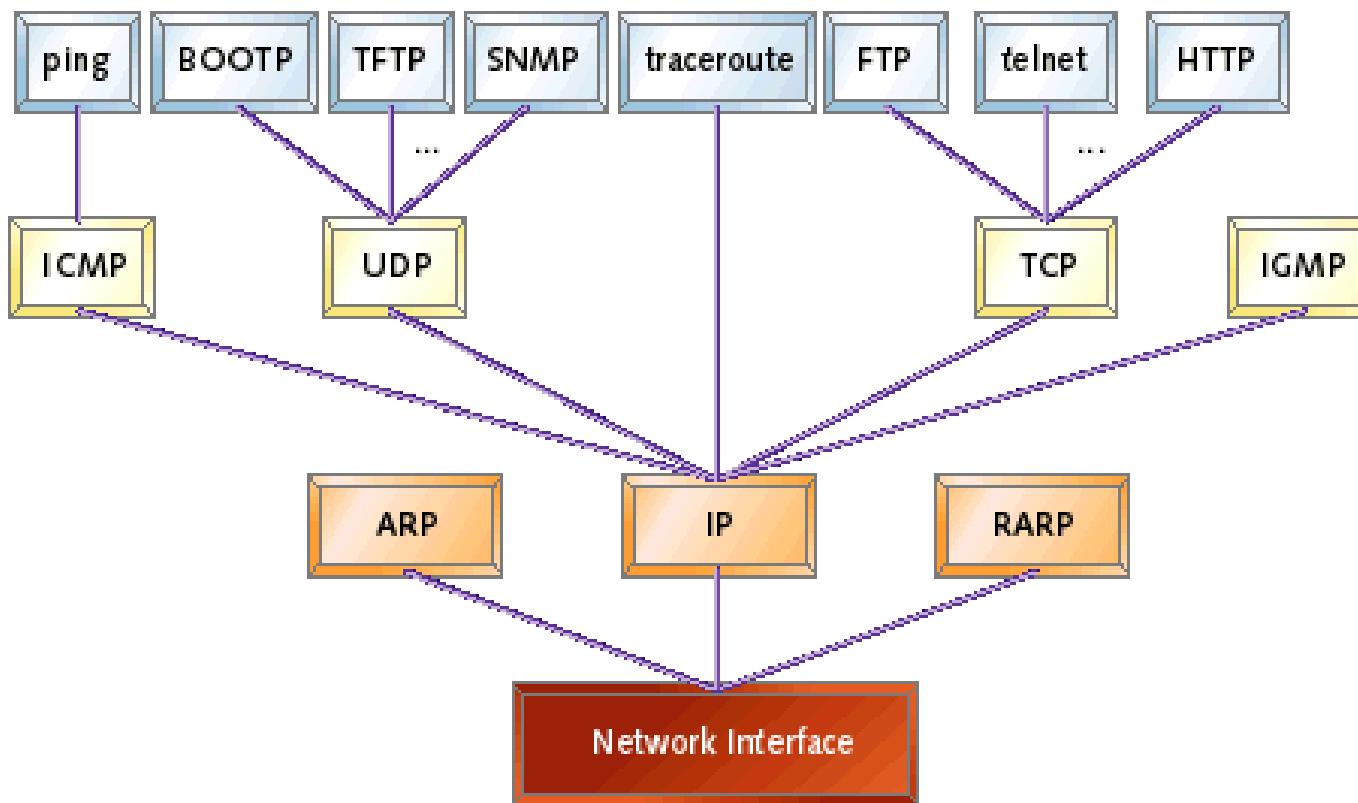
- **Interconectarea rețelelor**
- **Protocolul IP**
- **Protocolul TCP**
- **Controlul congestiei**

Interconectarea rețelelor

- Interconectarea rețelor reprezintă mecanismul prin care o colecție arbitrară de rețele oferă împreună un serviciu de transport de tip host-to-host.
- Protocolul IP este elementul cheie folosit pentru a construi rețele interconectate.
- O instanță a acestui protocol trebuie să ruleze pe fiecare din nodurile rețelelor interconectate.



Protocolul IP



Protocolul IP

Serviciul oferit de protocolul IP

- Modelul serviciului prezintă două componente:
 - o schemă de adresare care permite identificarea tuturor hosturilor interconectate.
 - un model de transfer al datelor, de tip datagramă, fără conexiune, model numit și „best effort”.
- Noțiunea de „best effort” se traduce prin aceea că dacă un pachet se pierde sau este afectat de eroare, rețeaua nu face nimic pentru a-l recupera, pentru că ea a depus deja tot efortul pentru a transporta acel pachet.
- Este posibil ca pachetele care au fost transmise să ajungă în altă ordine decât cea în care au fost transmise sau unele să fie duplicate.

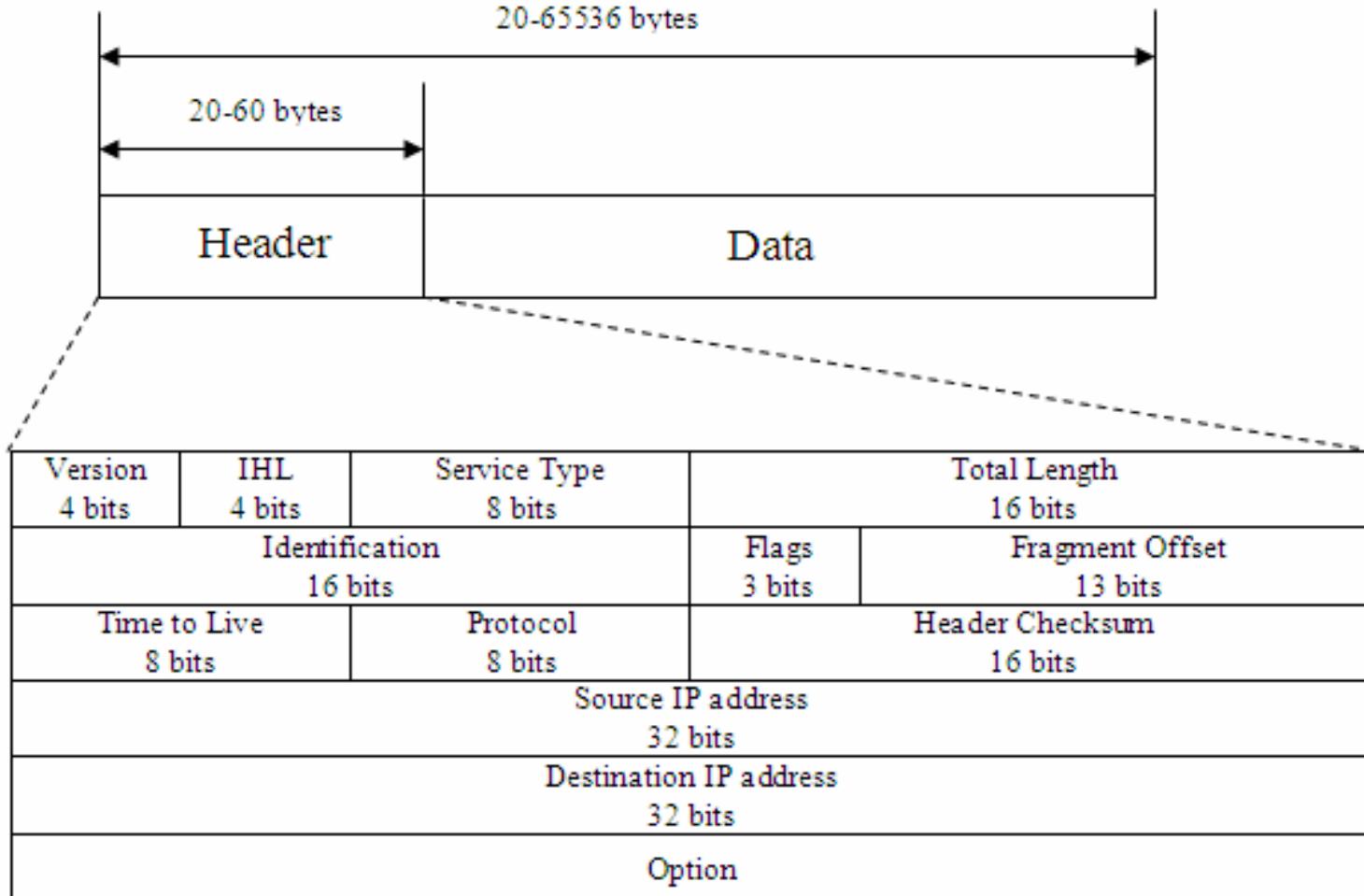
Protocolul IP

Serviciul oferit de protocolul IP

- Serviciul de tipul „best effort” fără conexiune este cel mai simplu serviciu care poate fi oferit în cazul interconectării rețelelor.
- Păstrarea simplității software-ului care rulează pe routere a fost unul din principalele scopuri ale protocolului IP.
- Abilitatea protocolului IP de a rula peste orice tip de rețea reprezintă o altă caracteristică fundamentală a acestuia.
- Multe din tehnologiile peste care protocolul IP rulează nici măcar nu erau inventate în momentul apariției acestuia.

Protocolul IP

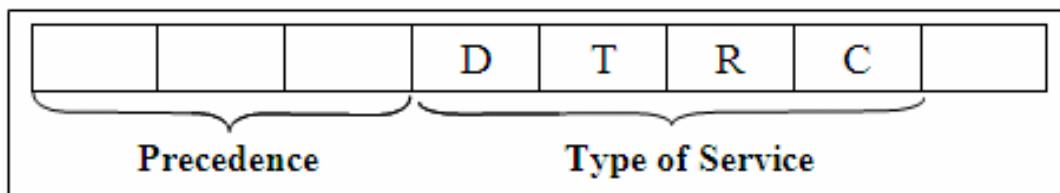
Formatul unui pachet IP



Protocolul IP

Formatul unui pachet IP

- **Version:** Versiunea protocolului. Există două versiuni funcționale, 4 și 6. În acest material este prezentată doar versiunea 4.
- **IHL:** Câmpul ne spune lungimea header-ului, exprimată în număr de cuvinte de 4 octeți. Dacă lungimea este 20, atunci valoarea lui IHL este 5.
- **Service Type:** Câmpul acesta este împărțit la rândul lui în mai multe subcâmpuri.



- **Precedence:** a fost gândit pentru a defini prioritatea unui pachet. În versiunea 4 a protocolului IP acest subcâmp nu este folosit.

Protocolul IP

Formatul unui pachet IP

- **Type of Service:** acest subcâmp este format din 4 biți. Fiecare dintre ei are o anumite semnificație și doar unul poate fi setat la un moment dat.

ToS	Semnificație
0000	Normal
0001	Minimizează costul
0010	Maximizează siguranța
0100	Maximizează capacitatea de transfer
1000	Minimizează întâzierea

- **Total Length:** Acest câmp conține lungimea totală a pachetului. Dacă se dorește să se afle lungimea datelor, se scade din lungimea totală, valoarea câmpului Header Length înmulțită cu 4.
- **Identification, Flags, Fragmentation Offset:** Folosite în procesul de fragmentare a pachetelor.

Protocolul IP

Formatul unui pachet IP

- **Time to Live:** Acest câmp este folosit pentru a stabili numărul maxim de hop-uri (routere) prin care un pachet poate trece.
 - Fiecare router care procesează pachetul decrementează câmpul cu o unitate. Când valoarea ajunge la zero, pachetul este eliminat din rețea și un mesaj de eroare este generat către nodul care avea adresa trecută în câmpul *Source IP Address*.
 - Valoarea de initializare a acestui câmp este de obicei dublul numărului maxim de router-e care se pot interpune între sursă și destinație.
 - Este necesar acest mecanism deoarece în absența lui și în anumite circumstanțe (tabele de rutare corupte) anumite pachete ar putea călători la infinit în rețea, consumând inutil resursele rețelei.

Protocolul IP

Formatul unui pachet IP

- **Protocol:** Prin acest câmp se identifică protocolul de nivel superior care face uz de protocolul IP pentru a-și transporta datele

Protocol	Valoare
TCP	6
UDP	17
ICMP	1
OSPF	89
EGP	8
Ipv6	41

- **Checksum:** Sumă de control aplicată pachetului.
- **Source Address:** Câmpul conține adresa nodului care a trimis pachetul.
 - **Destination Address:** Câmpul conține adresa nodului căruia îi este destinat pachetul.

Protocolul IP

Fragmentarea și reasamblarea pachetelor

- Fiecare protocol de pe Nivelul Legătură de Date are propriul format pentru frame-urile utilizate la transportul informației. Când un pachet IP traversează rețelele el trebuie să fie încapsulat în aceste frame-uri ale Nivelului Legătură de Date.
- Fiecare frame acceptă o anumită dimensiune maximă pentru câmpul de date. Astfel, dacă dimensiunea pachetului IP depășește această valoare, pachetul va trebui să fie fragmentat.
- Câmpul *Flags* conține 3 biți. Primul este rezervat, iar următorii doi sunt notați cu D (*do not fragment*), respectiv M(*more fragment*). Dacă bitul D are valoarea 1, atunci pachetul nu poate fi fragmentat. Dacă bitul M are valoarea 1 aceasta semnifică că pachetul nu este ultimul fragment ci mai sunt și altele. Dacă valoarea este 0, atunci fragmentul este ultimul.
 - Câmpul *Fragmentation Offset* indică poziția relativă a unui fragment în cadrul unui pachet. Poziția este indicată sub formă de deplasament exprimat ca multiplu de 8 octeți.

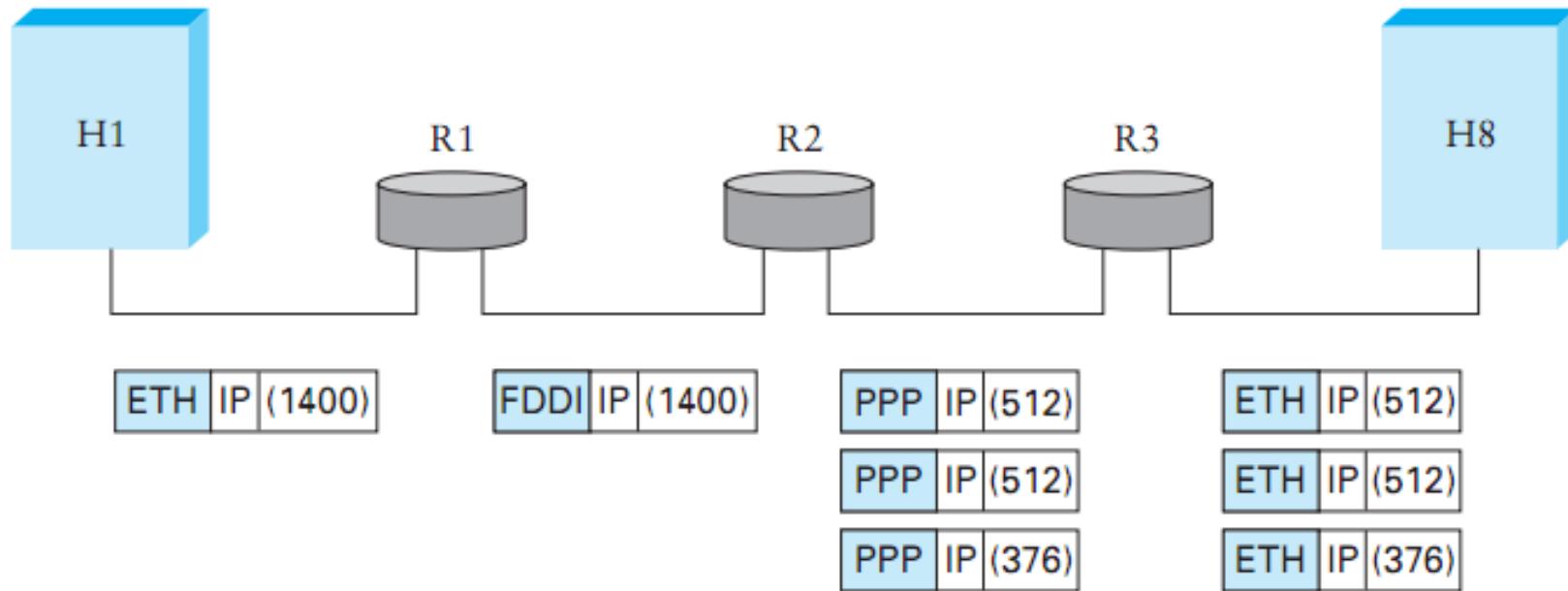
Protocolul IP

Fragmentarea și reasamblarea pachetelor

- Câmpul *Identification* are și el un rol important în procesul de fragmentare. Fiecare pachet primește un număr de identificare care va fi stocat în acest câmp. Acest număr este generat prin incrementare cu unu pentru fiecare nou pachet trimis. Valoarea de la care se pornește este una pozitivă aleasă în mod aleator. Astfel, pachetele vor fi identificate în mod unic folosind această etichetă și adresa sursei. Atunci când este necesar ca un pachet să fie fragmentat, fragmentele care au rezultat vor avea același număr de identificare cu cel al pachetului din care provin. În acest fel va putea fi refăcut pachetul original.

Protocolul IP

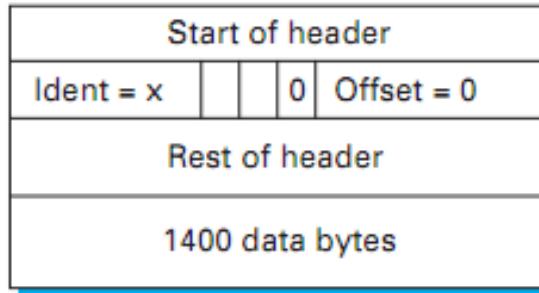
Fragmentarea și reasamblarea pachetelor



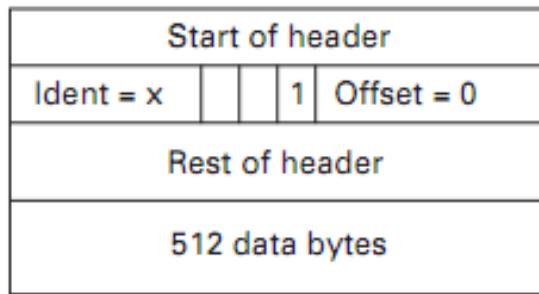
Protocolul IP

Fagmentarea și reasamblarea pachetelor

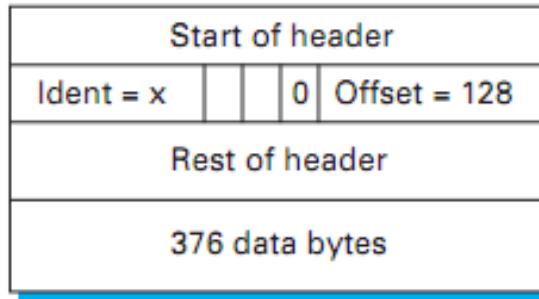
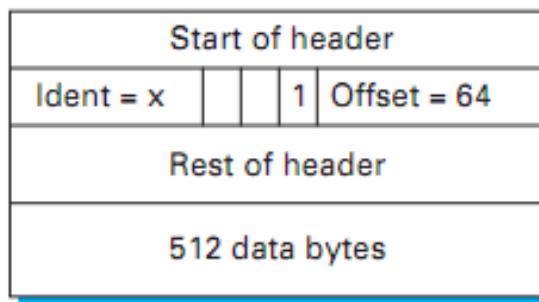
(a)



(b)



(b)

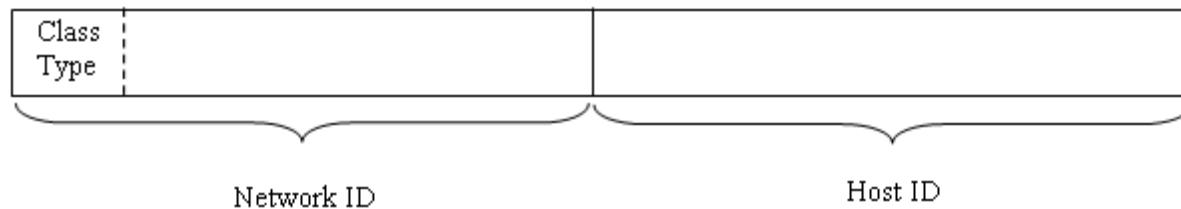


Header fields used in IP fragmentation. (a) Unfragmented packet; (b) fragmented packets.

Protocolul IP

Adresarea

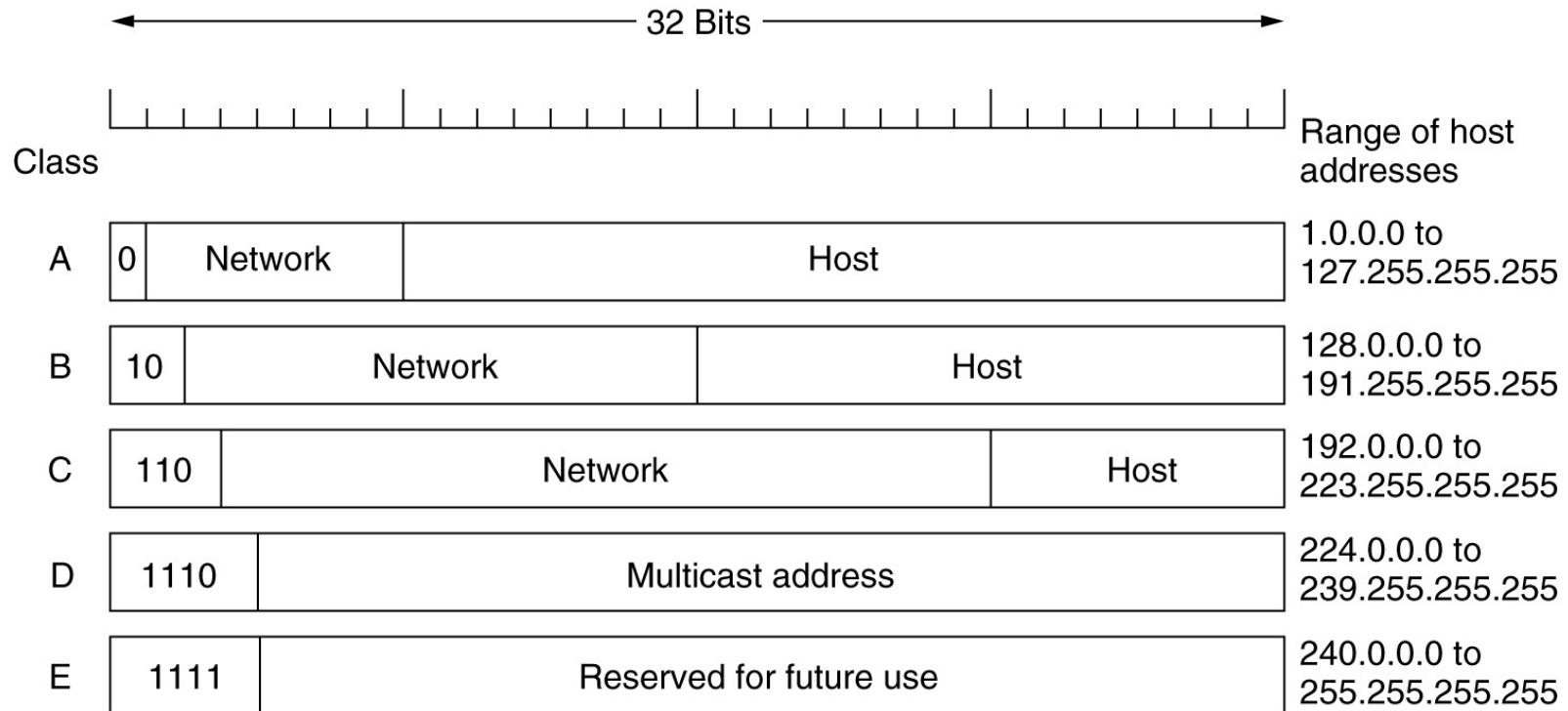
- Protocolul IP implementează o schemă de adresare globală.
- Întregul spațiu de adrese IP a fost împărțit în 5 clase notate de la A la E.
- O adresă IP conține două părți: una care ne dă adresa rețelei și cealaltă ne spune adresa host-ului în cadrul rețelei.
- Valuarea primilor biți dintr-o adresă IP ne spune clasa din care face parte acea adresă. Fiecare clasă alocă un număr diferit de biți pentru partea de Network ID și pentru Host ID.



Class	High Order Bits	Netid	Hostid
A	0	7 bits	24 bits
B	10	14 bits	16 bits
C	110	21 bits	8 bits
D	1110	28 bits multicast group number	
E	1111	reserved	

Protocolul IP

Adresarea



Protocolul TCP

- Protocolul TCP are sarcina de a transforma Nivelul Rețea, reprezentat în cazul nostru prin protocolul IP, dintr-un nivel nesigur într-unul sigur.
- Tot protocolul TCP este responsabil și cu implementarea anumitor mecanisme de **control al fluxului și al congestiei**.
- Protocolul **TCP este de tipul capăt la capăt**, adică este necesar să existe o instanță a acestui protocol doar pe mașina sursă și pe mașina destinație, nu și în nodurile intermediare care vor fi tranzitate de către pachete.

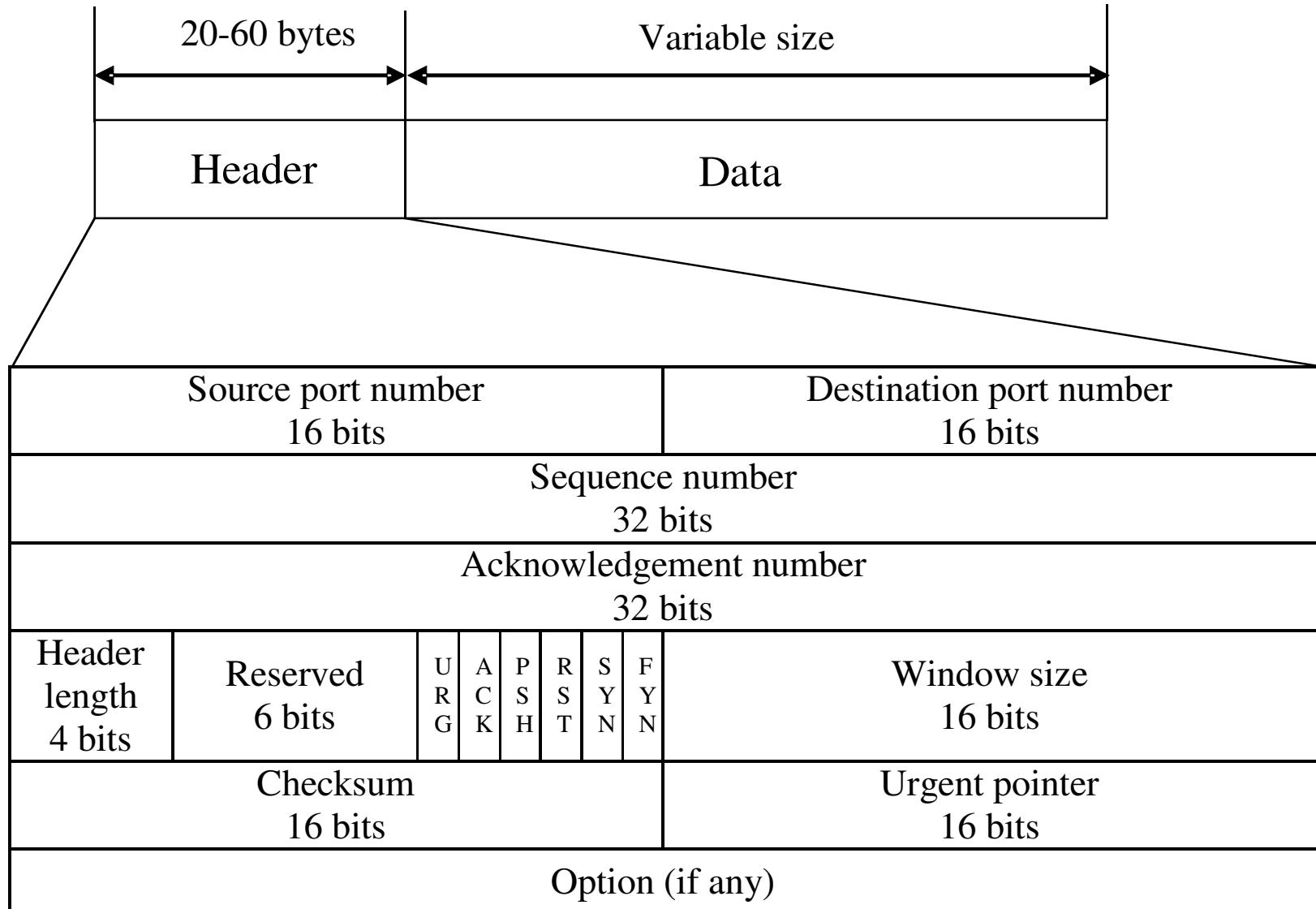
Protocolul TCP

Serviciile oferite de către TCP sunt:

- Stream Data Service
 - Aceasta presupune că pachetele care ajung la destinație să fie recepționate exact în ordinea în care au fost trimise.
 - Dacă trebuie transmis un mesaj, care din cauza lungimii va fi fragmentat în mai multe pachete, la receptor se verifică sosirea tuturor pachetelor și se aşeză în ordinea în care au fost trimise.
 - Pentru a putea furniza acest serviciu, TCP face uz de buffer-e atât pe partea de transmisie, cât și pe partea de recepție.
- Full-duplex service
 - Presupune că transferul de informații între două noduri poate fi făcut simultan în ambele direcții.
 - Când un pachet pleacă de la unul din noduri către celălalt, transportă și un mesaj de confirmare pentru un pachet recepționat anterior (piggybacking).
- Reliable service
 - TCP folosește un mecanism de confirmări pentru a se asigura că pachetele nu au fost afectate de erori și ca au sosit în ordinea corectă.

Protocolul TCP

Formatul unui pachet TCP



Protocolul TCP

Formatul unui pachet TCP

- **Source port number** : Numărul portului folosit de către aplicația care rulează pe mașina care trimite pachetele.
- **Destination port number** : Numărul portului folosit de aplicația care rulează pe mașina care primește pachetele.
- **Sequence number**: TCP numerotează fiecare octet trimis și folosește acest câmp pentru a indica numărul de secvență al primului octet de date din pachet. Când se inițiază o comunicație între două mașini, numărul de secvență pentru primul octet trimis este ales aleator.
- **Acknowledgement number**: Acest câmp este folosit pentru a confirma octetii recepționați. El se calculează însumând la valoarea din câmpul *Sequence number* al pachetului primit, dimensiunea câmpului de date recepționat, plus 1. Astfel valoarea acestui câmp reprezintă de fapt următoarea valoare pentru câmpul *Sequence number* care va fi folosită de către mașina care recepționează pachetul de confirmare.

Protocolul TCP

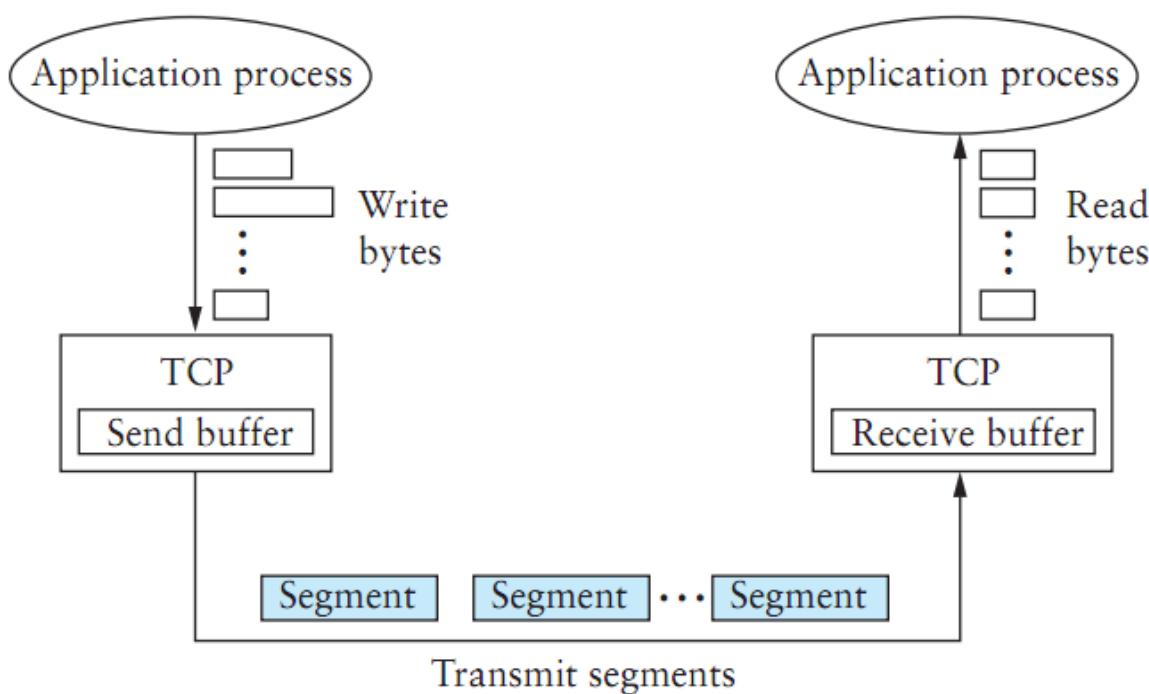
Formatul unui pachet TCP

- **Header length:** Conține dimensiunea header-ului exprimată în cuvinte de 32 de biți.
- **Control field:** Conține 6 biți a căror semnificație va fi explicitată atunci când se va discuta modul cum se inițiază și se desfășoară o sesiune de comunicație.
- **Window size:** reprezintă spațiul disponibil din buffer-ul de recepție.

Protocolul TCP

Conexiunea de tip end-to-end

- TCP-ul își bazează funcționarea pe un algoritm cu fereastră glisantă.
- Protocolele cu fereastră glisantă au fost gândite pentru legături punct la punct.
- În cazul TCP-ului se vor stabili conexiuni logice între procese care pot rula pe oricare două calculatoare din Internet.



Protocolul TCP

Conexiunea de tip end-to-end

- Mecanismul ferestrei glisante garantează un transfer sigur al datelor:
 - mecanism de retransmisie al datelor
 - recepția pachetelor în ordinea în care au fost transmise
 - multiplexarea traficului
 - permite controlul fluxului între transmițător și receptor
- Un alt mecanism esențial implementat de către TCP este controlul congestiei.

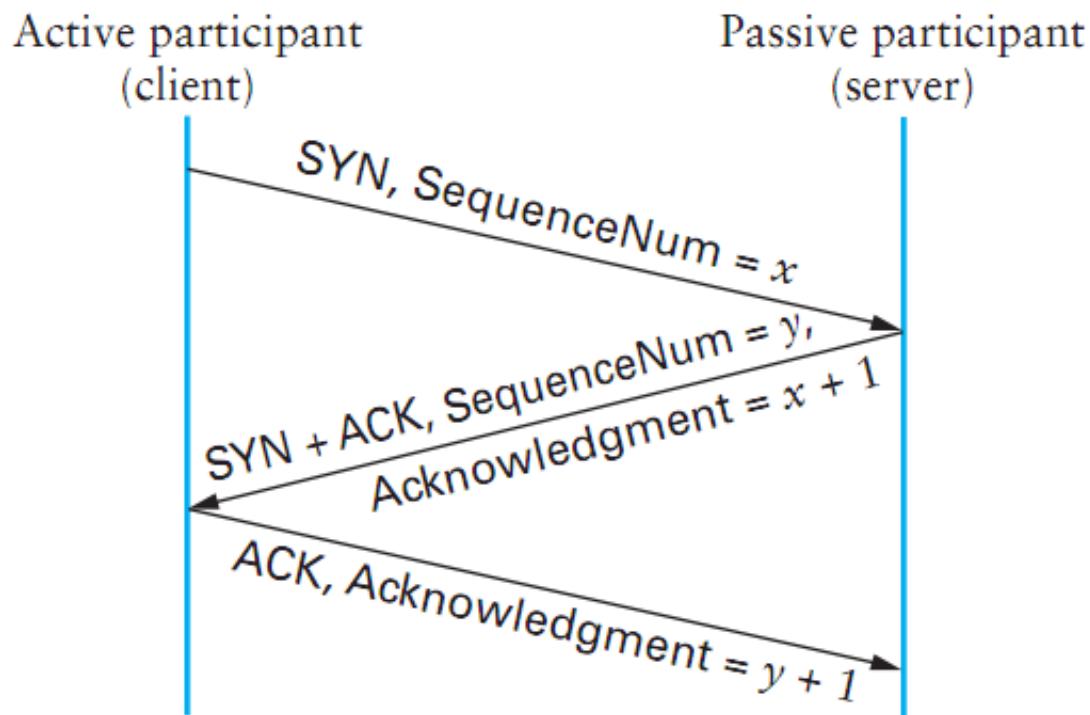
Protocolul TCP

Inițierea conexiunii:

- Nodul care inițiază conexiunea, de obicei este de tip *client*, iar cel care răspunde cererii venite de la *client*, este *server-ul*.
- Pentru a iniția o conexiune este nevoie de trei pachete ([three-way handshake](#)):
 - Primul pachet vine din partea clientului și are setat flag-ul SYN. Acest pachet conține numărul portului pe care clientul îl va folosi pe durata conexiunii, precum și numărul de secvență inițial.
 - Al doilea pachet implicat în stabilirea conexiunii vine ca răspuns din partea serverului, având setat flag-ul SYN, precum și flag-ul ACK, fiind un pachet de confirmare pentru primul pachet. El mai conține, de asemenea și numărul de secvență inițial folosit de server.
 - Al treilea pachet vine din partea clientului și conține confirmarea pentru pachetul SYN trimis de server.

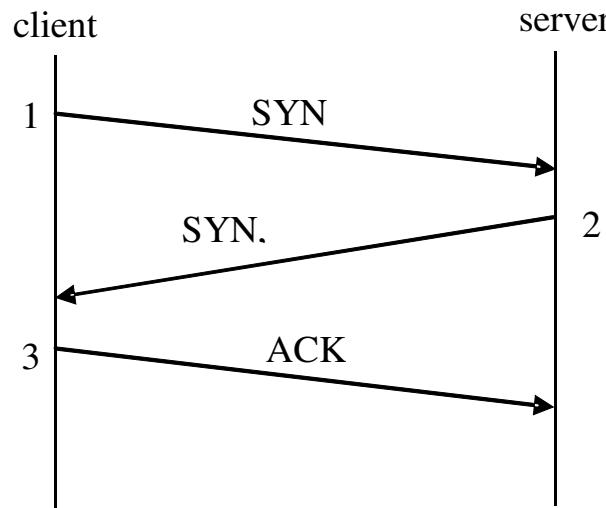
Protocolul TCP

Inițierea conexiunii:



Protocolul TCP

Inițierea conexiunii:

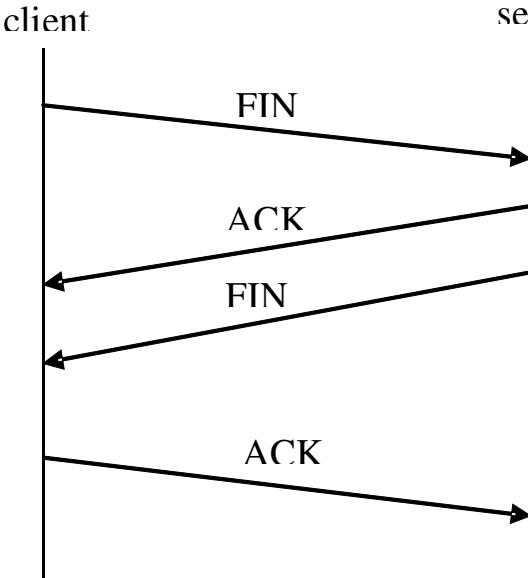


No. .	Time	Source	Destination	Protocol	Info
3	0.000309	192.168.0.13	212.112.238.74	TCP	1193 > ftp [SYN] Seq=0 Ack=0 win=64240 Len=0 MSS=1460
4	0.131235	212.112.238.74	192.168.0.13	TCP	ftp > 1193 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460
5	0.131369	192.168.0.13	212.112.238.74	TCP	1193 > ftp [ACK] Seq=1 Ack=1 win=64240 Len=0

Protocolul TCP

Încheierea unei conexiuni

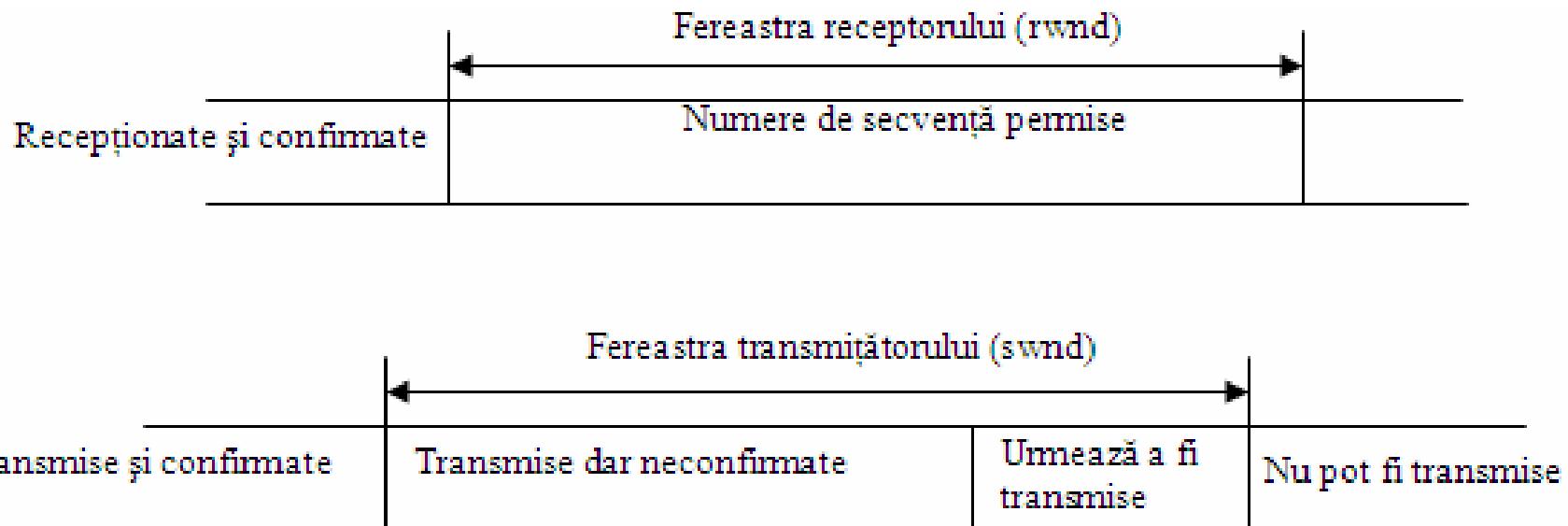
- Încheierea unei conexiuni poate fi făcută în mod bilateral, adică ambele noduri vor trimite câte un pachet de tip FIN, confirmate prin câte un pachet ACK, sau unilateral, când doar unul dintre noduri trimite un pachet FIN care va fi confirmat print-un pachet ACK.



No.	Time	Source	Destination	Protocol	Info
158	24.365722	10.23.3.21	10.23.3.11	TCP	telnet > 1100 [FIN, ACK] Seq=6442 Ack=129 Win=5840 Len=0
159	24.366030	10.23.3.11	10.23.3.21	TCP	1100 > telnet [ACK] Seq=129 Ack=6443 Win=64171
160	24.366613	10.23.3.11	10.23.3.21	TCP	1100 > telnet [FIN, ACK] Seq=129 Ack=6443 Win=64171
161	24.366778	10.23.3.21	10.23.3.11	TCP	telnet > 1100 [ACK] Seq=6443 Ack=130 Win=5840 Len=0

Protocolul TCP

Funcționarea ferestrei glisante:



Protocolul TCP

Funcționarea ferestrei glisante:

- Atunci când un pachet este recepționat se verifică dacă numărul lui de secvență coincide cu numărul de secvență de la începutul ferestrei receptorului, adică este următorul număr de secvență așteptat.
- Dacă numărul de secvență nu coincide dar se află în interiorul ferestrei, atunci este introdus în buff-er dar nu este confirmat, iar receptorul trimite un pachet de confirmare care conține în câmpul ACK aceiași valoare cu cea din pachetul de confirmare corespunzător ultimului pachet de date valid.
- Dacă numărul de secvență nu este cel așteptat și nici nu se află în interiorul ferestrei de recepție, atunci pachetul este ignorat.
- În momentul când sosește pachetul așteptat, atunci acesta este confirmat, iar limita din stânga a ferestrei se deplasează spre dreapta.
- Limita din dreapta a ferestrei se va deplasa spre dreapta doar în momentul în care pachetele care au fost confirmate sunt scoase din buffer pentru a fi procesate.

Protocolul TCP

Retransmisia datelor:

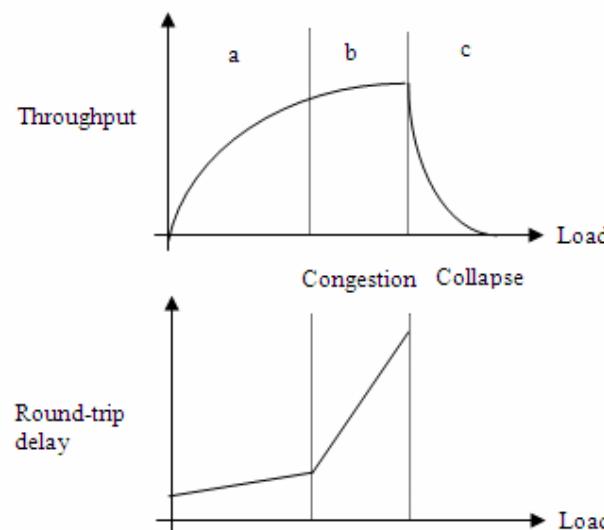
- Retransmisia datelor se realizează fie când pachetele ajung la destinație dar sunt afectate de erori, fie când s-au pierdut pe drum.
- Pentru ca procedeul de retransmisie să funcționeze este nevoie de folosirea unor timere și a unui mecanism de confirmări pozitive (*positive acknowledgements*).
- Confirmarea pozitivă însemnă că sunt confirmate doar pachetele care au ajuns neafectate de erori.
- Pentru a optimiza mecanismul de retransmisie se folosește o metodă cumulativă de confirmări, care permite confirmarea printr-un singur mesaj a unui grup de pachete consecutive.
- Pentru transmisia confirmărilor sunt folosite pachete de date, procedeul purtând denumirea de *piggybacking*. Există și posibilitatea de a transmite distinct doar pachete de confirmare.

Controlul congestiei

- Atunci când resursele rețelei nu mai reușesc să facă față traficului și rețeaua devine suprasolicitată, parametrii în care se desfășoară traficul se degradează tot mai mult, în felul acesta instalându-se congestia. Congestia se manifestă prin întârzieri tot mai mari, printr-un număr mare de pachete pierdute și în final se poate ajunge la colaps total, adică blocarea rețelei.
- În funcție de locul unde apare congestia există două categorii de congestie:
 - Congestie care apare din cauza suprasolicitării server-elor de aplicații (Server Side Congestion): apar prea multe cereri din partea clienților la un moment dat astfel încât noile cereri pentru conexiuni vor fi respinse.
 - Cealaltă categorie de congestie apare pe partea de client, atunci când mai mulți clienți împart în comun aceleași conexiune fizice. În acest caz congestia apare în nodurile intermediare care nu mai pot să gestioneze numărul mare de conexiuni care apar la un moment dat.
- Într-o rețea bazată pe comutarea de pachete resursele sunt distribuite la nivelul fiecărui nod din rețea. Aceste resurse pot fi definite prin trei elemente: capacitatea de procesare a informațiilor, dimensiunea buffer-elor și capacitatea de transport a liniilor.

Controlul congestiei

- Luând în considerare capacitatea de transport și întârzierile care apar, instalarea congestiei arată grafic ca în figura de mai jos.
- Se observă că atunci când rețeaua lucrează în parametrii optimi, ea răspunde corect atunci când apare o încărcare mai mare (zona a). La început apare o creștere exponențială, apoi o zonă în care rețeaua nu mai reacționează la o creștere suplimentară a încărcării (zona b), pentru ca apoi dintr-un anumit punct capacitatea de transfer să scadă brusc, iar dacă nu sunt luate măsuri, se ajunge la colaps (zona c).
- În ceea ce privește întârzierea introdusă de rețea se observă că se păstrează o valoare aproximativ constantă în zona a, apoi pe măsură ce rețeaua începe să fie congestionată (zona b) întârzierea introdusă de rețea devine tot mai mare.



Controlul congestiei

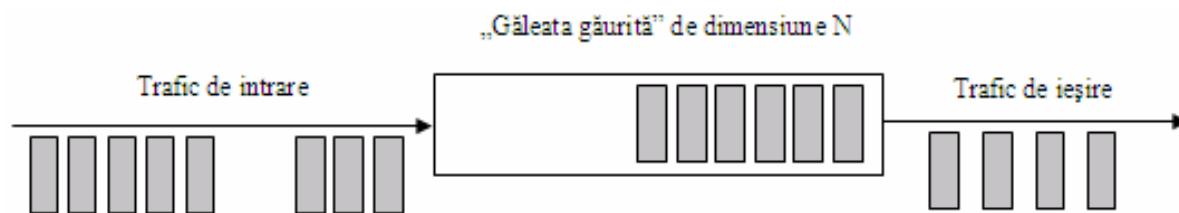
- Pentru controlul congestiei există două abordări și anume:
 - un control în buclă deschisă
 - un control în buclă închisă (cu *feedback*)
- În primul caz se stabilesc de la început parametrii în care va funcționa rețeaua, luându-se măsuri în faza de proiectare pentru prevenirea apariției problemelor, deoarece controlul se va face fără informații despre situația concretă de la un moment dat din rețea. Controlul în buclă deschisă se face de obicei la periferia rețelei, prin supravegherea traficului (*traffic policing*) și prin formarea traficului (*traffic shaping*) pentru nodul care a primit acces la resursele rețelei. Formarea traficului presupune uniformizarea ratei medii de transmisie a datelor.
- Al doilea mod de control al congestiei este cel în buclă închisă. Aici măsurile care sunt întreprinse se bazează pe informații culese în permanentă din interiorul rețelei. Informațiile primite pot fi implicite sau explice. Un tip de informație implicită ar putea fi numărul de pachete pierdute, sau întârzierile din rețea. Informațiile explice sunt cele generate în mod special pentru a avertiza despre apariția congestiei. Aceste informații pot fi pachete suplimentare care să conțină date despre congestie sau ar putea fi folosite anumite câmpuri în cadrul pachetelor și care vor fi setate cu anumite valori atunci când apare congestia.

Controlul congestiei

- Pentru controlul congestiei în buclă deschisă, vom da două exemple de algoritmi pentru formarea traficului:

Algoritmul găleții găurite

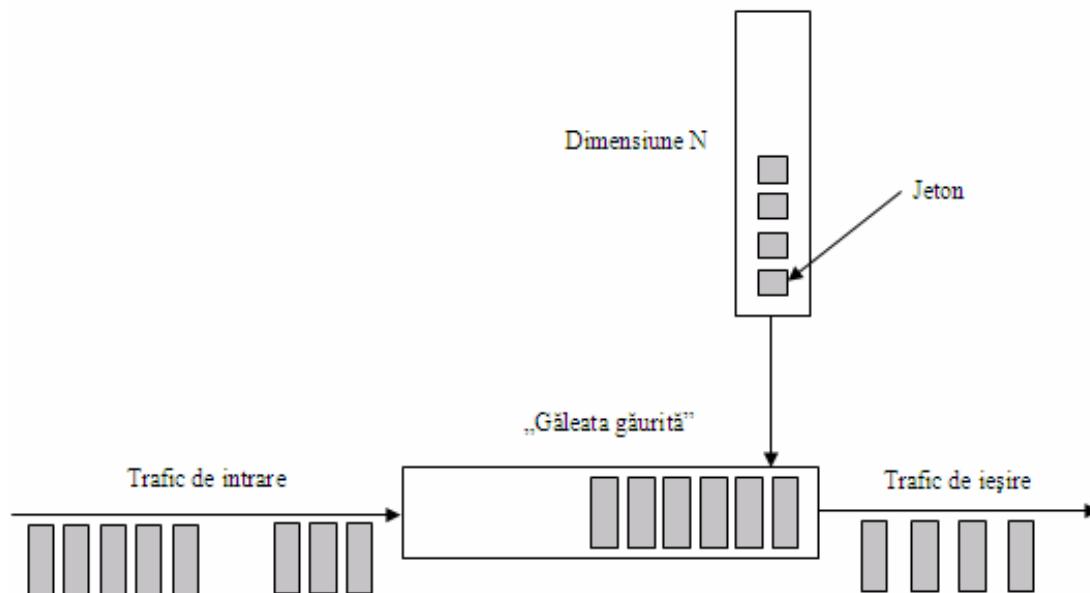
- Algoritmul poartă această denumire deoarece se face analogie cu o găleată care are un orificiu pe fund. Indiferent de debitul cu care apa intră în găleată, ea se va scurge prin orificiu cu un debit constant. La un moment dat, dacă găleata continuă să fie alimentată ea se poate umple și apa se pierde. Același principiu poate fi aplicat și în cazul unei transmisii de date. Pachetele care sunt trimise dintr-un anumit nod în rețea sunt trecute printr-o „găleată găurită”, adică un buffer de tip FIFO care acceptă pachete la orice rată de transfer, dar le transmite mai departe cu o rată fixă. Dacă buffer-ul se umple atunci pachetele care sosesc ulterior se pierd. În felul acesta se realizează în mod implicit și controlul traficului, adică urmărirea dacă un utilizator depășește parametrii de trafic care i-au fost atribuți. Marele avantaj al acestei metode este acela că nu permite traficului în rafală care ar putea veni din partea nodului transmițător să pătrundă sub această formă în rețea, aceasta fiind principala cauză a apariției congestiei, chiar și atunci când resursele rețelei ar părea ca sunt suficiente.



Controlul congestiei

Algoritmul găleții cu jeton

- Acest algoritm se aseamănă în mare măsură cu cel anterior, dar spre deosebire de acesta permite flexibilitate în ceea ce privește rata traficului de ieșire. În cazul găleții găurile rata acestui trafic era fixă. Algoritmul funcționează în felul următor: găleata acumulează jetoane generate cu o rată de un jeton la ΔT secunde. Pentru ca un pachet să poată fi trimis el trebuie să găsească un jeton în găleată, pe care să-l distrugă. Acest mod de abordare a problemei permite ca în momentul în care la intrare avem date în rafală, iar în găleată avem jetoane disponibile, atunci datele vor fi transmise tot în rafală, dar lungimea rafalei are maxim valoarea egală cu numărul de jetoane din găleată.



Controlul congestiei

Controlul în buclă închisă

- Pentru a se evita funcționarea rețelei în zona de congestie este nevoie să se folosească procedee de monitorizare și control a congestiei. Când se pune problema controlului congestiei două mecanisme trebuie luate în discuție: evitarea congestiei (*congestion avoidance*) și ieșirea din congestie (*congestion recovery*). Aceste mecanisme pot fi implementate pe de o parte la nivelul routerelor, care reprezintă nodurile intermediare, atunci când are loc un transfer de date între sursă și destinație, cât și la nivelul sursei și a destinației, adică al celor două noduri între care se desfășoară transferul. Se spune că în acest caz realizăm un control capăt la capăt.
- Într-o rețea ideală, pentru ca cele două tipuri de control să fie eficiente este nevoie ca pe de o parte rețeaua să ofere feedback, pentru ca resursele să fie folosite în mod eficient, iar pe de altă parte este nevoie ca fluxurile de date să fie protejate unele față de altele, în cazul în care unii utilizatori ar avea tendința să acapareze mai multe resurse decât cele care le-au fost alocate. Această protejare a fluxurilor de date se poate face prin mecanisme de tip QoS.
- Principalul avantaj al unei rețele bazate pe comutarea de pachete este faptul că toate resursele rețelei vor fi folosite împreună de către toți utilizatorii, iar rețeaua va încerca să aloce maximul de resurse disponibile fiecărui utilizator în parte. Ceea ce creează probleme este natura unpredictibilă și în rafală a traficului, aceasta putând să conducă la apariția congestiei. Pentru scurte momente de timp rețeaua poate deveni supraîncărcată și pentru a se evita intrarea în congestie este necesar ca într-un anume fel, utilizatorii să fie înștiințați de acest lucru și să treacă la o diminuare a încărcării rețelei, evitându-se astfel apariția congestiei. Acest mecanism va funcționa, făcând presupunerea că toți utilizatorii rețelei vor coopera și vor lua în considerare semnalele care avertizează asupra apariției congestiei. Pentru a se asigura acest lucru au fost implementate mecanisme de evitare a congestiei chiar în protocoalele de comunicație în Internet.

Rețele de calculatoare

Partea a 6-a

Sebastian Fuicu

.Dirijarea pachetelor IP

.Protocole de rutare

Dirijarea pachetelor IP

Principalele elemente ale procesului de dirijare sunt:

- .Fiecare pachet IP conține **adresa IP a sursei și a destinației**.
- .**Adresa de rețea** din cadrul adresei IP identifică o singură rețea fizică.
- .Toate host-urile și toate router-ele care au **aceiași adresă de rețea** sunt conectate la **aceiași rețea fizică** și pot comunica direct, trimițându-și unul altuia pachete.
- .**Fiecare rețea care face parte din Internet** are cel puțin un **router** care este conectat și la o altă rețea fizică. El poate schimba pachete cu host-uri și router-e din alte rețele.

Dirijarea pachetelor IP

Modalități de realizare a dirijării:

.Un pachet este trimis de la sursă la destinație traversând mai multe rutere intermediare.

.Orice nod (host sau router) încearcă să determine dacă nodul destinație se află în aceeași rețea fizică în care se află și el.

- Pentru aceasta compară adresa de rețea a destinației cu adresa de rețea asociată interfeței sale de rețea.
- Dacă cele două adrese coincid, atunci pachetul poate fi livrat în mod direct, deoarece destinația se află în aceeași rețea.
- Dacă nu coincid, pachetul va trebui să fie trimis către un router.

Dirijarea pachetelor IP

Modalități de realizare a dirijării:

. Dacă nodul care se află în posesia unui pachet este un router și dispune de mai multe interfețe de rețea, va trebui să selecteze una dintre acele interfețe.

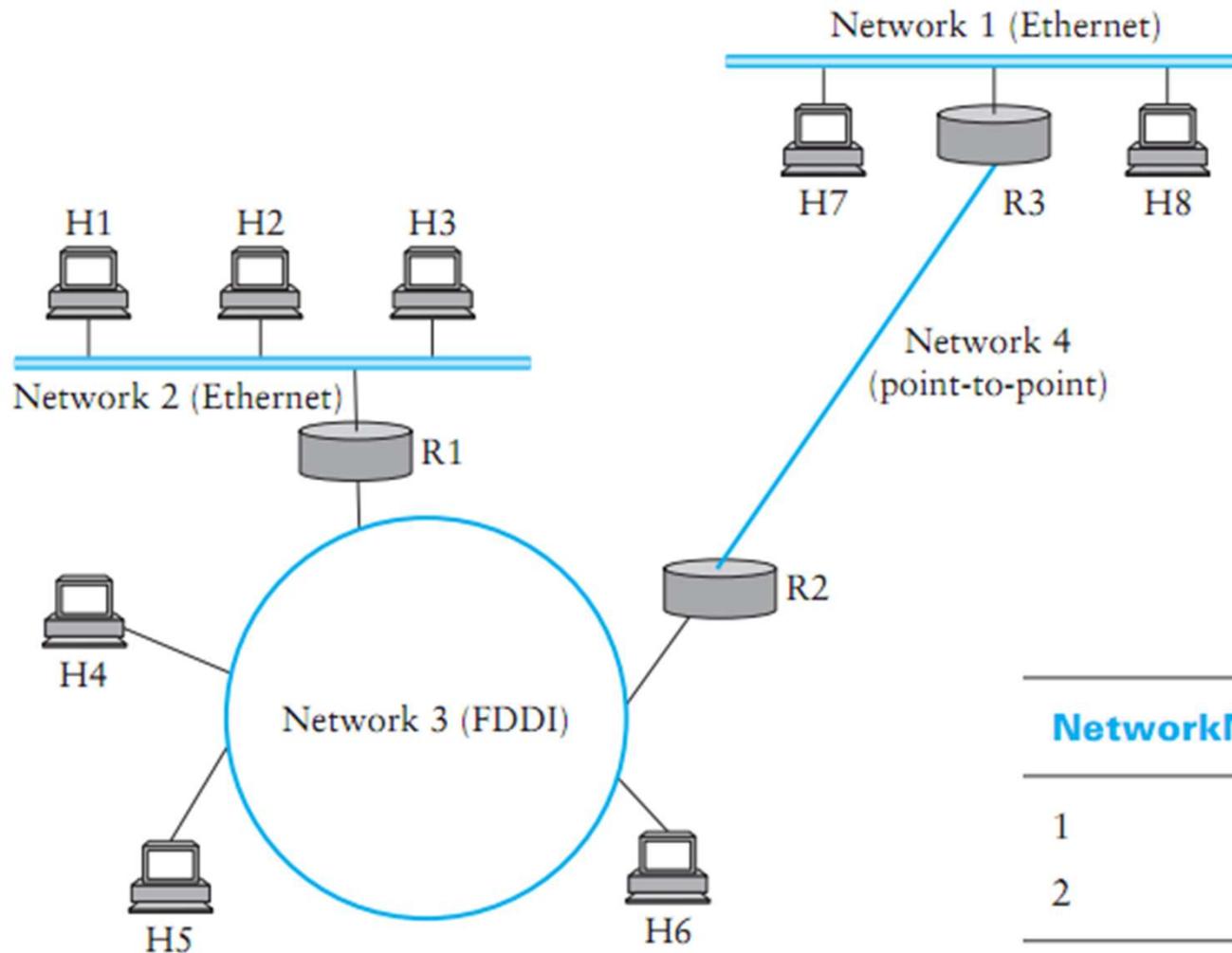
. Selectarea se face pe baza unei tabele de rutare care conține perechi de genul:

<NetworkNumber, NextHop>

. În cazul în care nici una dintre destinații nu se regăsește în listă, există un router predefinit căruia îi este trimis pachetul.

Dirijarea pachetelor IP

Modalități de realizare a dirijării:



NetworkNum	NextHop
1	R3
2	R1

Tabela de rutare
pentru R2

Dirijarea pachetelor IP

- .Pentru a obține o rețea scalabilă trebuie redusă cât mai mult cantitatea de informații ce trebuie stocată în fiecare nod.
- .Cea mai simplă modalitate de a obține acest lucru este **agregarea ierarhică**.
- .Protocolul IP introduce o ierarhie pe două nivele, cu rețelele pe primul nivel și noduri pe al doilea.
- .Agregarea presupune ca routerele să caute doar rețeaua destinație, indiferent de numărul host-urilor din acea rețea.

Dirijarea pachetelor IP

Translatarea adreselor (ARP – Address Resolution Protocol)

.Când un pachet IP trebuie trimis într-o rețea fizică este nevoie de un mecanism de translatare a adresei IP a nodului destinație în adresa de pe nivelul legătură de date a interfeței de rețea a acelui nod.

.În cazul protocolului Ethernet, aceasta este adresa MAC.

.Scopul protocolului ARP este de a permite fiecărui host din rețea să-și construiască o tabelă de mapări între adresele IP și adresele MAC.

.Condiția ca acest protocol să funcționeze este posibilitatea de a transmite mesaje de tip broadcast în acea rețea.

Dirijarea pachetelor IP

Raportarea erorilor (ICMP – Internet Control Message Protocol)

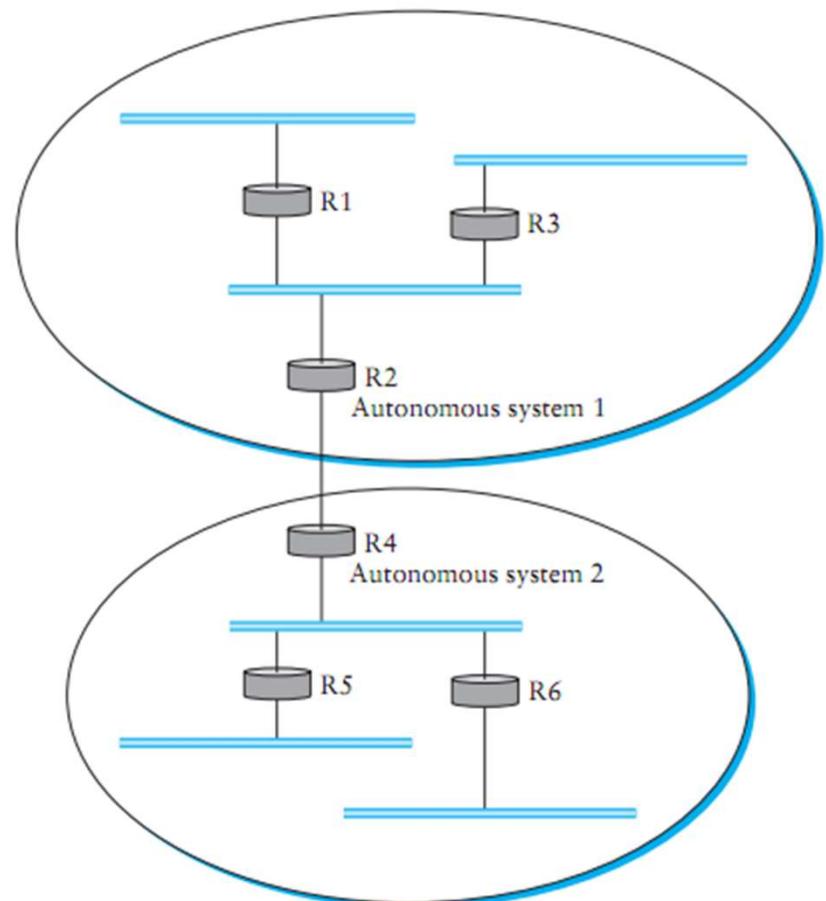
.IP este configurat să funcționeze totdeauna cu un protocol adițional, numit ICMP, care definește o colecție de mesaje de eroare ce sunt transmise înapoi sursei pachetului atunci când un router sau un host nu poate să proceseze cu succes un pachet IP.

.Mesaje de eroare pot fi transmise atunci când:

- destinația nu a fost găsită
- procesul de reasamblare nu s-a putut efectua
- TTL a atins valoarea zero
- checksum-ul header-ului a dat o valoare greșită, etc.

Protocole de rutare

- .Protocolele de rutare sunt necesare pentru actualizarea tabelelor de rutare.
- .Un **sistem autonom** este o regiune din Internet care este sub controlul administrativ al unei singure entități.
- .Ideeă de bază a unui sistem autonom este de a oferi o modalitate suplimentară de a agraga ierarhic informațiile de rutare.
- .Rezultă o **rutare intradomeniu** și o **rutare interdomenii**.



Protocole de rutare

- .Există două tipuri de rutare: *rutare statică* și *rutare dinamică*.
- .Doar al doilea tip de rutare, cea dinamică, implică existența protocoalelor de rutare.
- .Există trei mari categorii de algoritmi pentru rutarea dinamică:
 - protocole de rutare cu *vectori distanță*
 - protocole de rutare care folosesc *starea legăturilor*
 - protocole mixte care le îmbină pe primele două metode

Protocoloale de rutare

Rutarea statică

- . Este cea mai simplă metodă de rutare și se realizează folosind rute predefinite, stabilirea acestor rute căzând în sarcina administratorului de rețea.
- . Atunci când topologia rețelei se schimbă și tabelele de rutare trebuie actualizate, aceasta făcându-se tot de către administratorul de rețea.
- . Rutarea statică funcționează foarte bine în cazul unor rețele de dimensiuni reduse unde traficul este predictibil.
- . De asemenea acest tip de rutare conduce și la o alocare mai eficientă a resurselor: nu se ocupă din capacitatea de transport a rețelei cu informații de rutare, nu se încarcă procesoarele routerelor cu calcule pentru aflarea rutei optime și nici nu necesită multă memorie.

Protocoloale de rutare

Rutarea dinamică

.Algoritmii de rutare dinamici își modifică deciziile de dirijare a traficului pentru a reflecta modificările din topologia rețelei și în unele cazuri și pe cele de trafic. Acești algoritmi diferă prin:

- locul de unde își iau informația
- momentul la care își schimbă rutele
- metrica folosită.

.Metrici folosite de către algoritmii de rutare

- lungimea căii de transmisie
- siguranța căii de transmisie
- întârzierea
- lățimea de bandă
- încărcarea

Protocole de rutare

Protocole de rutare cu vectori distanță

.Algoritmul de dirijare cu vectori distanță impune ca fiecare router să mențină o tabelă care păstrează cea mai bună distanță cunoscută spre fiecare destinație și interfața pe care trebuie trimise datele pentru a ajunge la acea destinație.

.Denumirea de *distanță* nu trebuie înțeleasă ca o distanță fizică între două noduri, ci o denumire generică, putând fi folosite oricare din metricile enumerate anterior.

.Pentru actualizarea tabelelor de rutare, routerele fac schimb de informații care constau chiar în aceste tabele de rutare pe care fiecare router îl transmite către celelalte routere.

.Fiecare intrare din tabelă conține o parte care indică interfața de ieșire și o estimare a distanței până la acea destinație. În loc de distanță poate fi folosită și altă metrică, ca de exemplu numărul de salturi sau numărul de pachete care așteptă în cozi de-a lungul căii.

.Principalul dezavantaj al acestui tip de rutare este o convergență mai lentă.

Protocole de rutare

Protocole de rutare bazate pe starea legăturilor

. Protocolele din această familie sunt mult mai complexe decât protocolele care folosesc vectori distanță.

. Aceste protocole funcționează pe principiul că routerele fac schimb de informații cuprinse sub denumirea generică de starea legăturilor, aceasta reprezentând, de fapt, informații despre legături și nodurile din subretea.

. Routerele care rulează un protocol bazat pe starea legăturilor nu vor trebui să facă schimb de tabele de rutare ca în cazul protocolelor bazate pe vectori distanță, fiind suficient să transmită informații despre nodurile adiacente și metrica asociată fiecărei conexiuni.

. Cea mai potrivită comparație a acestei familii de protocole este comparația cu un joc de puzzle. Fiecare router din rețea generează o piesă din puzzle pe care o distribuie în întreaga rețea prin inundare. În final, fiecare router din rețea deține câte o copie a fiecărei piese de puzzle, reprezentând routerele din rețea.

Protocole de rutare

Protocole de rutare interioare și exterioare

.Deoarece dimensiunea Internet-ului a crescut foarte mult, s-a ajuns la concluzia că un singur protocol de rutare nu ar fi putut face față tuturor cerințelor și de aceea s-a trecut la separarea Internet-ului în entități numite sisteme autonome (*autonomous systems - AS*).

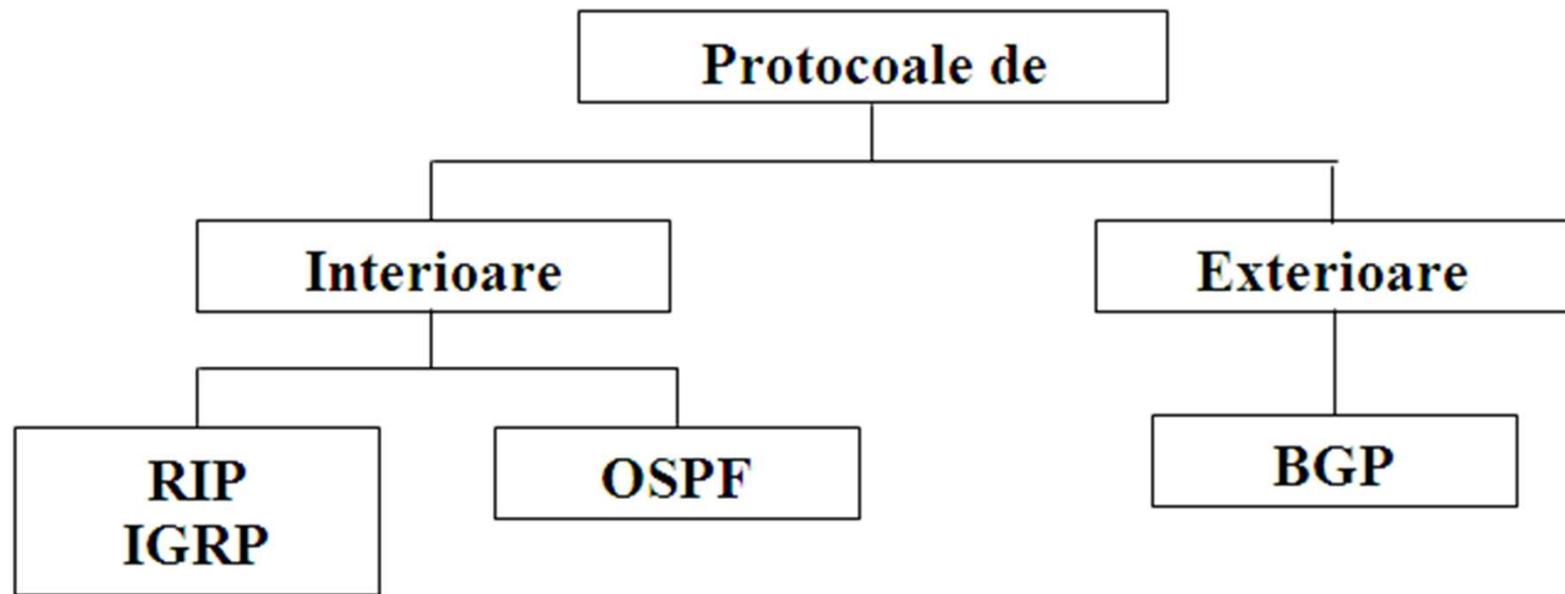
.Rutarea în interiorul unui AS se numește rutare interioară (*interior routing*), iar rutarea între sisteme autonome se numește rutare exterioară (*exterior routing*).

.Fiecare AS are libertatea de a-și alege propriul protocol de rutare interioară, dar pentru rutarea exterioară este ales un protocol unic care să fie folosit de sistemele care fac legătura dintre AS-uri. .

Protocole de rutare

Protocole de rutare interioare și exterioare

Cele mai populare protocole de rutare interioară și exterioară sunt redate în figura de mai jos:



.RIP (Routing Information Protocol)

.OSPF (Open Shortest Path First)

.IGRP (Interior Gateway Routing Protocol)

.BGP (Border Gateway Protocol)

Protocole de rutare

Protocole de rutare interioare și exterioare

.RIP, IGRP și OSPF sunt folosite pentru a actualiza tabelele routerelor din interiorul unui AS

.BGP este folosit pentru actualizarea tabelelor folosite de către routerele care fac legătura între AS-uri.

