

Retele de calculatoare – Informatica anul 3 (2019-2020)

Note de Laborator
Retele de calculatoare

Specializare: Informatica anul 3

Contact:
retelecdsd@gmail.com
<http://www.cdssd.ro>

Comunicatii de
Date si
Sisteme
Distribuite



<http://www.cdssd.ro>

Laborator 4

1. Obiective:

- Intelegerea rolului protocoalelor Ethernet si ARP in retelele de calculatoare
- Wireshark: Frame-uri Ethernet si ARP
- Studii de caz: **Riverbed Modeler Academic Edition** – mediu de simulare a retelelor de calculatoare (Varianta programare C++: [OMNeT++ Network Simulation Framework](http://www.omnetpp.org/) <http://www.omnetpp.org/>)
- Aplicatii de retea in Python

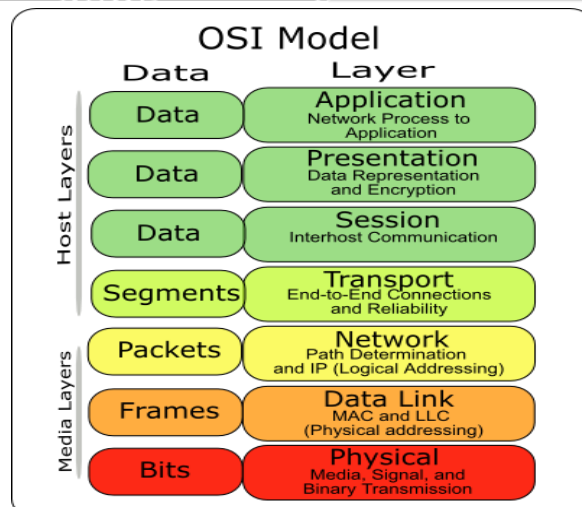
2. Consideratii teoretice (Partea practica – pag. 11; Tema – pag.30)

2.1. Protocoale de comunicatie. Suita de protocoale TCP/IP

Un protocol de comunicatie este un set de reguli ce determina formatul si transmitia datelor, controland aspecte legate de : Constructia fizica a retelei, Modul de conectare a calculatoarelor in retea, Formatul si transmiterea datelor, Rezolvarea erorilor.

In Internet standardul tehnic il reprezinta suita de protocoale TCP/IP (Transmission Control Protocol/Internet Protocol). Aceasta este dezvoltata ca un standard deschis putand fi folosita in mod liber si contine, printre altele, protocoalele precizate mai jos, corespunzator modelelor OSI si TCP/IP:

Modelul OSI	Modelul TCP/IP	Protocoale
Application	Application	Telnet, SSH SMTP, POP, IMAP FTP, TFTP, NFS HTTP DNS
Presentation		
Session		
Transport	Transport	TCP, UDP
Network	Internet	IP, ICMP, ARP, RARP
Data Link	Network Access	Internet, Ethernet, FDDI, ATM SLIP, PPP ARP, RARP
Physical		



Retele de calculatoare – Informatica anul 3 (2019-2020)

2.1.1. ETHERNET (IEEE 802.3)

- Metoda de acces : **CSMA / CD – Carrier Sense Multiple Access / Collision Detection** (Sesizarea purtatoarei, Acces multiplu, Detecatarea coliziunilor);
- Intr-un mediu cu sesizarea coliziunilor o placa adaptoare (**NIC – Network Interface Card - placa de retea**) „asculta” rețeaua atunci când are de transmis un frame. Când sesizează că o altă placă adaptoare trimite un frame intervine un timp de așteptare pentru toate stațiile, după care se reîncearcă transmiterea. Se definește **coliziunea** ca reprezentând încercarea simultană a două stații de lucru (SL-uri) de a transmite un frame **pe un mediu comun de comunicație**. Prima stație care detectează o coliziune își va suspenda emisia și va trimite un semnal special care anunță existența unei **interferențe (jamming)**. Acest semnal va avea o frecvență specială și o durată echivalentă unei transmisii de 32 de biți. Toate stațiile vor detecta acest semnal și își vor suspenda activitatea. Pentru a micșora “sansele de apariție” a unei noi coliziuni când stațiile vor începe din nou să transmită, timpul care va trece până la reluarea transmisiilor va fi diferit pentru fiecare stație. **Intervalele de timp sunt stabilite pe baza unui algoritm de revenire (back-off algorithm)**. Ethernet: protocol nedeterminist; nu asigură prioritate în vederea retransmiterii pentru nici-un host.

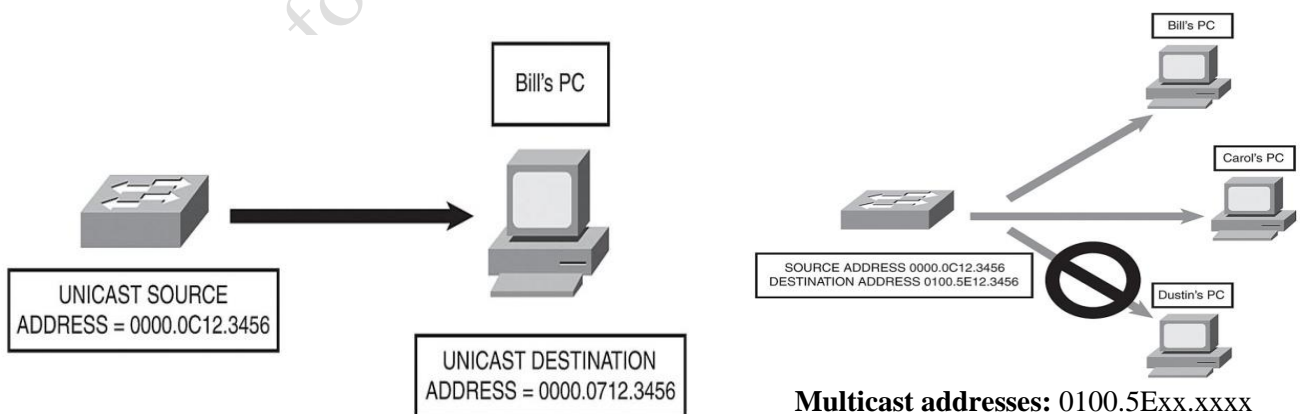
2.1.2. Adresarea MAC

Adresele MAC au o lungime de șase octeți, sunt înscrise în memoria ROM a plăcii de rețea și sunt stabilite la nivel mondial. Ulterior, ele pot fi modificate prin software, folosindu-se un buffer care conține noua adresă. Primii trei octeți identifică producătorul plăcii de rețea sau organizația care a proiectat acel tip de rețea. Acest câmp poartă denumirea OUI (Organization Unique Identifier). Ceilalți trei octeți sunt completați de producător și reprezintă un număr de ordine. **Ex: 02:76:4C:08:89:67**

Primii doi biți transmiși (I/G și U/L) au o importanță specială:

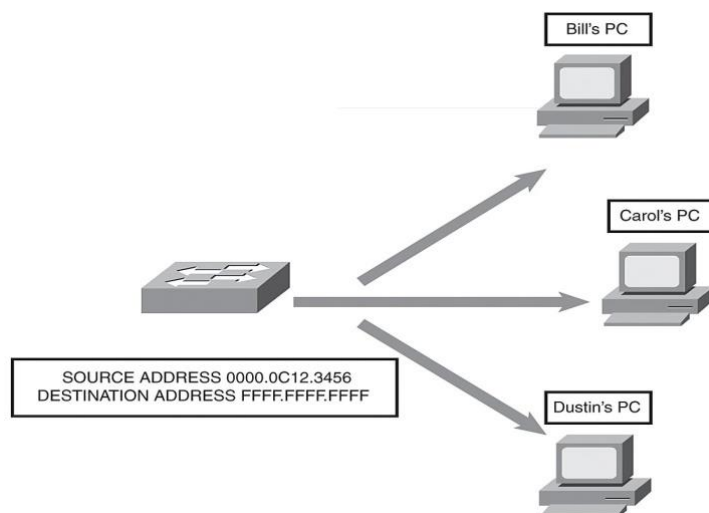
- **Bitul I/G (Individual/Grup)** arată tipul destinației: o singură stație (valoarea 0 – Individual UNICAST) sau un grup de stații (valoarea 1 – Grup MULTICAST SAU BROADCAST).
- **Bitul U/L (Universal/Local)** arată dacă adresa a fost atribuită oficial de IEEE (valoarea 0 - Universal) sau a fost atribuită local (valoarea 1 - Local).

Informații suplimentare – Anexe Lab.

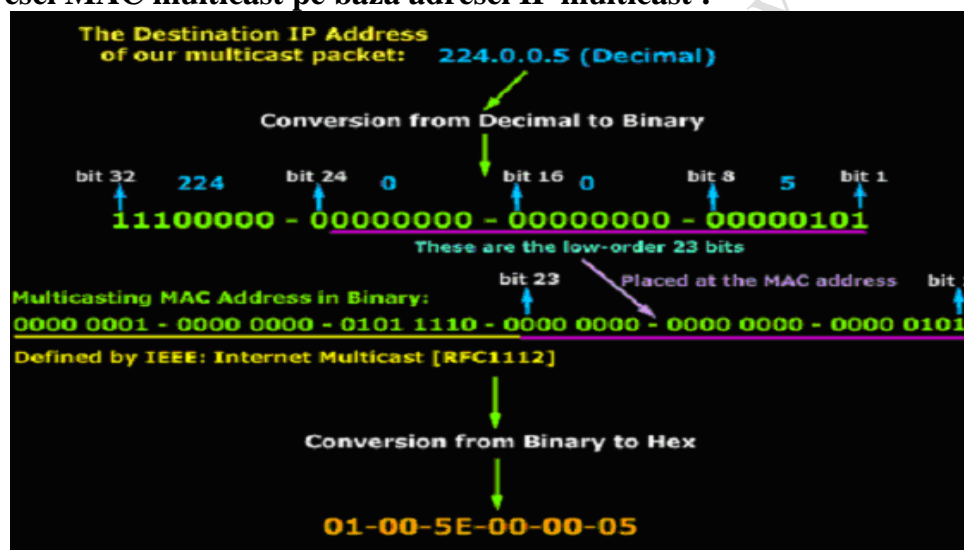


Broadcast address: FFFF.FFFF.FFFF.

Retele de calculatoare – Informatica anul 3 (2019-2020)



Calculul adresei MAC multicast pe baza adresei IP multicast :

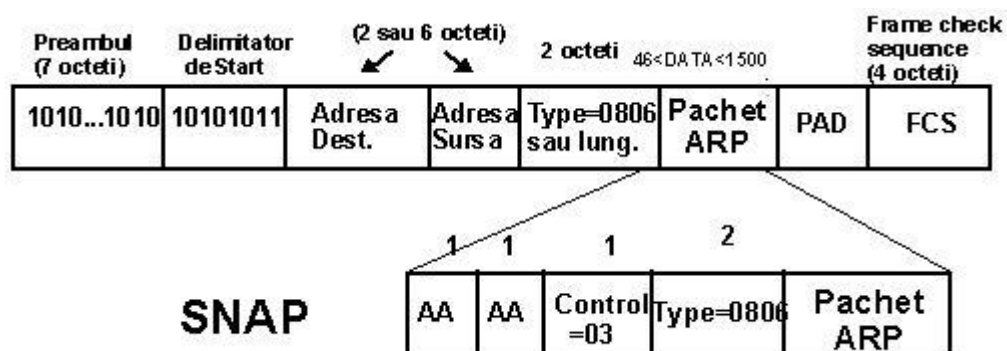


2.1.2. ARP (Address Resolution Protocol)

- Pentru a livra un pachet Ethernet, emitatorul are nevoie de adresa fizică (MAC) a receptorului. Dacă emitatorul are doar adresa IP a receptorului, el trebuie să afle adresa MAC corespunzătoare.
- Protocolul ARP servește găsirii adresei MAC pornind de la adresa IP, presupunând că mașina căutată este în aceeași rețea Ethernet.
- Protocolul presupune că emitatorul trimite un pachet Ethernet de broadcast în rețeaua locală, întrebând "cine are adresa IP cutare". Mașina care are adresa IP căutată va răspunde cu un mesaj de genul "eu am adresa IP cutare" - mesaj ce conține adresa MAC. Celelalte mașini ignoră mesajul.
- O corespondență IP - MAC este păstrată în memorie un anumit timp.

Încapsularea pachetului ARP într-un cadru Ethernet sau 802.2 SNAP (Sub-Network Access Protocol) este exemplificată în figură:

Rețele de calculatoare – Informatica anul 3 (2019-2020)



Structura unei cereri ARP este:

0	8	16	24	31
HARDWARE TYPE		PROTOCOL TYPE		
Lung. Adr. MAC	Lung. Adr. IP	COD OPERATIE		
ADRESA MAC Sursa (32biti)				
Adresa MAC Sursa (16biti)		Adresa IP Sursa (16biti)		
Adresa IP Sursa (16bits)		Adresa MAC Dest. (16biti)		
ADRESA MAC DEST. (32biti)				
ADRESA IP DEST. (32biti)				

Descrierea câmpurilor:

- Tipul hardware-ului: tipul de hardware al interfeței;
- Tipul protocolului: tipul de protocol pe care emițătorul îl folosește;
- Lungimea adresei MAC: lungimea fiecărei adrese hardware din cadru, dată în octeți;
- Lungimea adresei de protocol: lungimea adresei de protocol din datagramă dată în octeți;
- Cod operație (Op Code) indică: tipul de datagramă, cerere ARP sau răspuns la aceasta. Dacă datagrama este cerere valoarea acestui câmp este 1, iar dacă este răspuns valoarea este 0;
- Adresa MAC sursă: adresa hardware a dispozitivului transmițător;
- Adresa IP sursă: adresa IP a dispozitivului transmițător;
- Adresa IP a destinației: adresa IP a dispozitivului destinație;
- Adresa MAC a destinației: adresa hardware a dispozitivului destinație.

Câmpul "Tipul hardware-ului"

Acest câmp identifică tipul interfeței hardware folosit:

Tip Descriere

1 = Ethernet

2 = Ethernet experimental

3 = X.25

4 = Proteon ProNET (Token Ring)

5 = Chaos

6 = IEEE 802.x

7 = ARCnet

Retele de calculatoare – Informatica anul 3 (2019-2020)

16 = ATM

Câmpul "Tipul de protocol"

Tipul de protocol identifică, așa cum îi spune și numele, tipul de protocol de nivel rețea pe care dispozitivul transmițător îl folosește. Aceasta astfel indică, în mod implicit, și tipul de adresă de nivel rețea folosit. Cu TCP/IP, aceste protocoale sunt de obicei EtherType. Câteva exemple de valori ale acestui câmp:

În zecimal Descriere

2048 = Internet Protocol (IP)

2049 = X.75

2053 = X.25 Level 3

2054 = ARP

2055 = XNS

32821 = Reverse ARP

32824 = DEC LANBridge

32823 = Apple Talk

Dacă protocolul nu este EtherType, vor fi folosite alte valori.

Filter: arp

No.	Time	Source	Destination	Protocol	Info
32	0.300572	Cisco_73:d2:00	Broadcast	ARP	who has 82.77.81.157? Tell 82.77.80.1
33	0.439970	EdimaxTe_e7:eb:0d	Broadcast	ARP	who has 172.16.0.92? Tell 172.16.0.3
51	0.711456	Mototech_18:ae:8e	Broadcast	ARP	who has 172.16.3.14? Tell 172.16.0.91
53	0.823986	AsustekC_77:c4:c8	Broadcast	ARP	who has 172.16.0.14? Tell 172.16.0.185
67	0.900534	Cisco_73:d2:00	Broadcast	ARP	who has 81.196.163.170? Tell 81.196.74.1
100	1.500512	Cisco_73:d2:00	Broadcast	ARP	who has 81.196.74.43? Tell 81.196.74.1
116	1.800499	Cisco_73:d2:00	Broadcast	ARP	who has 82.77.81.158? Tell 82.77.80.1
133	2.319656	Cisco-Li_60:a9:67	Broadcast	ARP	who has 172.16.0.84? Tell 172.16.0.246
167	2.700425	Cisco_73:d2:00	Broadcast	ARP	who has 82.77.81.189? Tell 82.77.80.1
197	3.300411	Cisco_73:d2:00	Broadcast	ARP	who has 82.77.81.159? Tell 82.77.80.1
200	3.500792	EdimaxTe_e7:eb:0d	Broadcast	ARP	who has 172.16.0.92? Tell 172.16.0.3
216	3.600397	Cisco_73:d2:00	Broadcast	ARP	who has 82.77.81.116? Tell 82.77.80.1
234	4.003991	Cisco-Li_60:a9:67	Broadcast	ARP	who has 172.16.2.194? Tell 172.16.0.246
235	4.066202	HewlettP_0b:37:22	Broadcast	ARP	who has 172.16.1.126? Tell 172.16.0.1
282	4.800308	Cisco_73:d2:00	Broadcast	ARP	who has 81.196.74.76? Tell 81.196.74.1
284	4.820702	Mototech_18:ae:8e	Broadcast	ARP	who has 172.16.3.15? Tell 172.16.0.91

Frame 200 (60 bytes on wire, 60 bytes captured)

Ethernet II, Src: EdimaxTe_e7:eb:0d (00:50:fc:e7:eb:0d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Address Resolution Protocol (request)

Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: EdimaxTe_e7:eb:0d (00:50:fc:e7:eb:0d)
Sender IP address: 172.16.0.3 (172.16.0.3)
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 172.16.0.92 (172.16.0.92)

0000 ff ff ff ff ff ff 00 50 fc e7 eb 0d 08 06 00 01P.....
0010 08 00 06 04 00 01 00 50 fc e7 eb 0d ac 10 00 03P.....
0020 00 00 00 00 00 00 ac 10 5c 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00
.....

File: "F:\977345\info_3_2009-2010\C2_L2\lab\wr... Packets: 118897 Displayed: 6414 Marked: 0 Profile: Default

- Informatii suplimentare – Anexe Lab.

Retele de calculatoare – Informatica anul 3 (2019-2020)

2.1.3. RARP (Reverse ARP) – mapeaza adresa IP corespunzatoare adresei MAC

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Marquett_12:dd:88	Broadcast	RARP	who is 00:00:a1:12:dd:88? Tell 00:00:a1:12:dd:88

Frame 1 (60 bytes on wire, 60 bytes captured)

- Ethernet II, Src: Marquett_12:dd:88 (00:00:a1:12:dd:88), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Source: Marquett_12:dd:88 (00:00:a1:12:dd:88)
 - Type: ARP (0x0806)
 - Trailer: 00000000000000000000000000000000
- Address Resolution Protocol (reverse request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: reverse request (0x0003)
 - [Is gratuitous: False]
 - Sender MAC address: Marquett_12:dd:88 (00:00:a1:12:dd:88)
 - Sender IP address: 0.0.0.0 (0.0.0.0)
 - Target MAC address: Marquett_12:dd:88 (00:00:a1:12:dd:88)
 - Target IP address: 0.0.0.0 (0.0.0.0)

0000 ff ff ff ff ff ff 00 00 a1 12 dd 88 08 06 00 01
0010 08 00 06 04 00 03 00 00 a1 12 dd 88 00 00 00 00
0020 00 00 a1 12 dd 88 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Protocol Structure - RARP (Reverse Address Resolution Protocol)

<-----16 bits----->		<-----16 bits----->	
Hardware Type		Protocol Type	
HLen	Plen	Operation	
Sender Hardware Address			
Sender Protocol Address			
Target Hardware Address			
Target Protocol Address			

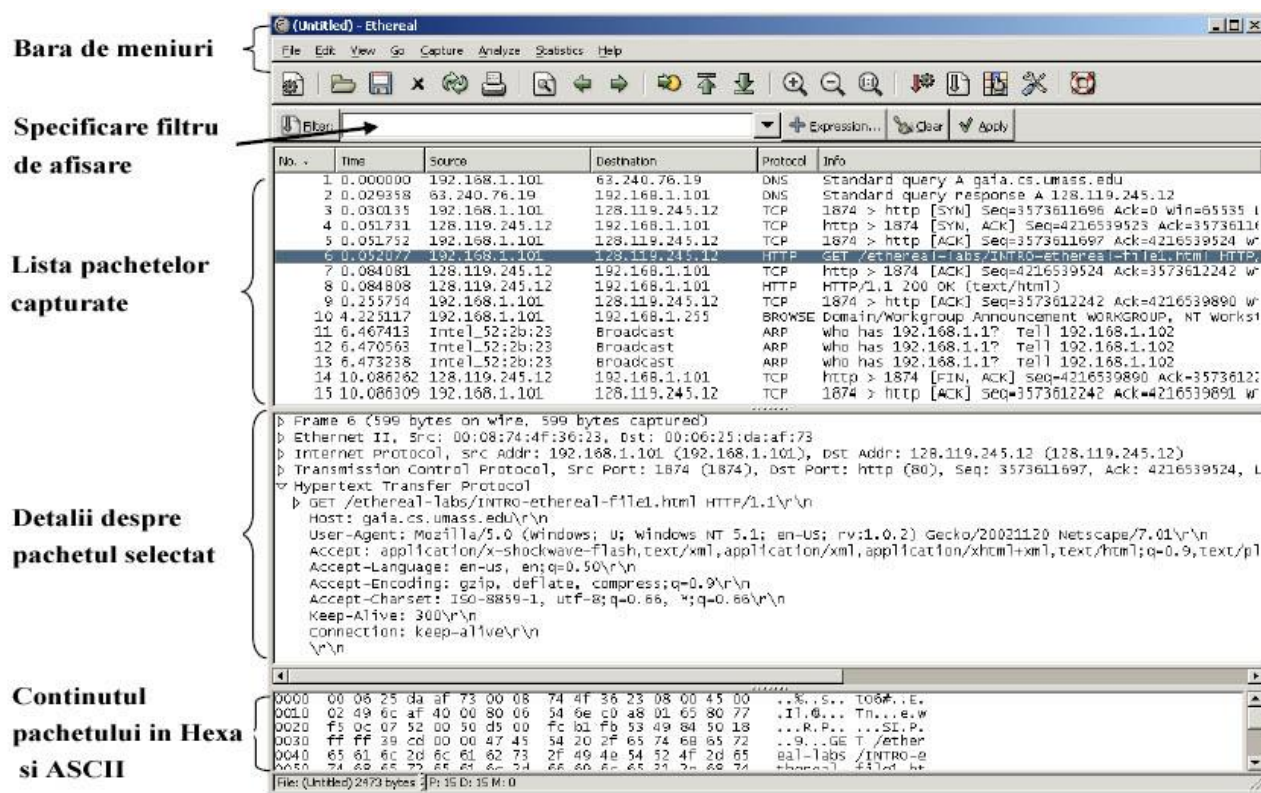
In the above figure it is clear that RARP and ARP has the same structure:

- Hardware type - which specifies a hardware interface type for which the sender requires a response.
- Protocol type - which specifies the type of the high-level protocol address the sender has supplied.
- Hlen - Hardware address length.
- Plen - Protocol address length.
- Operation - The values are as follows:
 - ARP request.
 - ARP response.
 - RARP request.
 - RARP response.
 - Dynamic RARP request.
 - Dynamic RARP reply.
 - Dynamic RARP error.
 - InARP request.
 - InARP reply.
- Sender hardware address - HLen bytes in length.
- Sender protocol address - Plen bytes in length.
- Target hardware address - HLen bytes in length.
- Target protocol address - Plen bytes in length.

Retele de calculatoare – Informatica anul 3 (2019-2020)

2.2. Wireshark (<http://www.wireshark.org/download.html>)

– este un *packet sniffer* (network protocol analyzer). Programul permite examinarea on-line a traficului dintr-o rețea, sau capturarea și salvarea traficului într-un fișier, cu analiză ulterioară a datelor. Pentru fiecare pachet capturat, programul afișează informații detaliate. **Wireshark** include un limbaj propriu pentru definirea expresiilor de filtrare și permite reconstruirea unei sesiuni TCP pe baza pachetelor capturate. (+ **Recapitulare Lab 03!**)



2.3. Riverbed Modeler Academic Edition

Oyervatie: (Lab_1 + Lab_2 si toate celelalte materiale prezente la www.cdsd.ro ...F.F.F.Importante)

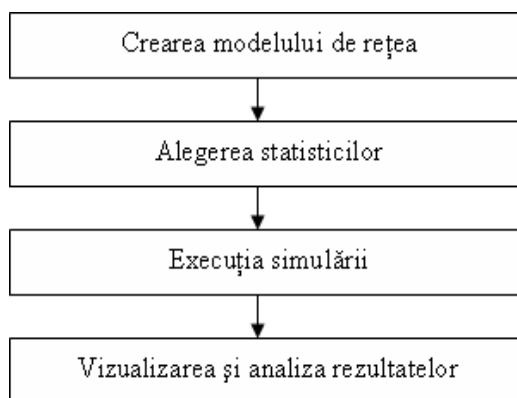
Riverbed Modeler Academic Edition (versiune actuala a Opnet-ului - **Optimized Network Application and Network Performance**) – mediu de simulare a rețelelor de calculatoare - furnizează software de management pentru aplicații și rețele, care oferă soluții pentru:

- Planificarea capacității rețelelor,
- Modelare și simulare pentru rețele și aplicații
- Managementul configurării rețelelor
- Managementul performanțelor aplicațiilor

Variantă “programare” C++: OMNeT++ Network Simulation Framework <http://www.omnetpp.org/>

Riverbed oferă o versiune academică (**Modeler Academic Edition**) - include modele standard pentru protocoale și echipamentele disponibile în tehnologia IT (disponibile, după instalare, în subdirectoare). Etapele de lucru avute în vedere:

Retele de calculatoare – Informatica anul 3 (2019-2020)



Etapele de lucru pentru Modeler Academic Edition pentru simularea și analiza unei rețele

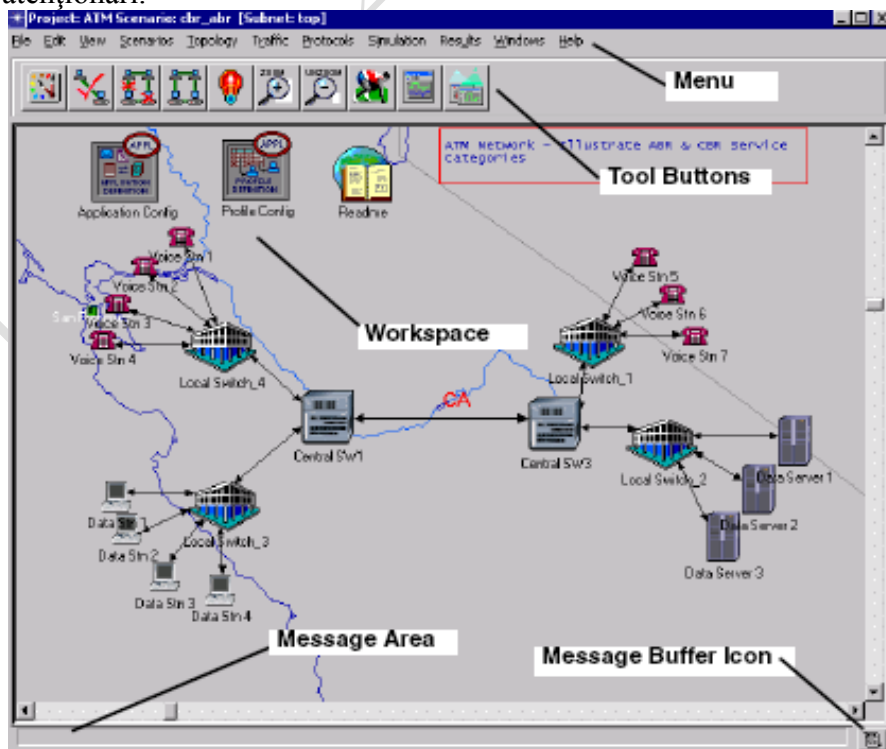
Obs: O statistica este o caracteristica numerica a unui esantion (Anexa 3, Lab_03)

- **Statistica** este stiinta colectarii, clasificarii, prezentarii, interpretarii datelor numerice si a folosirii acestora pentru a formula concluzii si a lua decizii.
- **Statistica descriptiva** (Descriptive Statistics) se ocupa cu colectarea, clasificarea si prezentarea datelor numerice.
- **Statistica inferentiala** (Inferential Statistics) se ocupa cu interpretarea datelor oferite de statistica descriptiva si cu folosirea acestora pentru a formula concluzii si lua decizii.

Workspace este spațiul de lucru din partea centrală a ferestrei editorului, care este folosit pentru crearea modelului rețelei, selectarea și deplasarea obiectelor rețelei, alegerea operațiilor specifice conextului.

Message Area, plasată în partea de jos a ferestrei, furnizează informații despre starea *tool-ului*.

Message Buffer Window, plasata în partea de jos în stânga, permite accesul la o listă de mesaje, notificări, atenționări.



Project Editor Window

Retele de calculatoare – Informatica anul 3 (2019-2020)



Butoane folosite în Project Editor

Semnificația butoanelor din Project Editor

1. <i>Open object palette</i>	6. <i>Zoom</i>
2. <i>Check link consistency</i>	7. <i>Restore</i>
3. <i>Fail Selected objects</i>	8. <i>Configure discrete event simulation</i>
4. <i>Recover selected objects</i>	9. <i>View simulation results</i>
5. <i>Return to parent subnet</i>	10. <i>Hide or show all graphs</i>

2.3.2. Studiu de caz (Modeler Academic Edition)

Obiectiv: Utilizarea Modeler pentru studiul îmbunătățirii performanțelor unei rețele LAN prin configurarea de VLAN-uri (Virtual LAN-uri)

Scenariu: O firmă de consultanță are birouri localizate în mai multe clădiri și recent a angajat personal suplimentar, volumul de trafic din rețeaua LAN înregistrând o creștere semnificativă. Deoarece firma nu vrea să investească într-o nouă infrastructură de rețea, inginerul de sistem trebuie să găsească cea mai bună soluție de utilizare a infrastructurii existente.

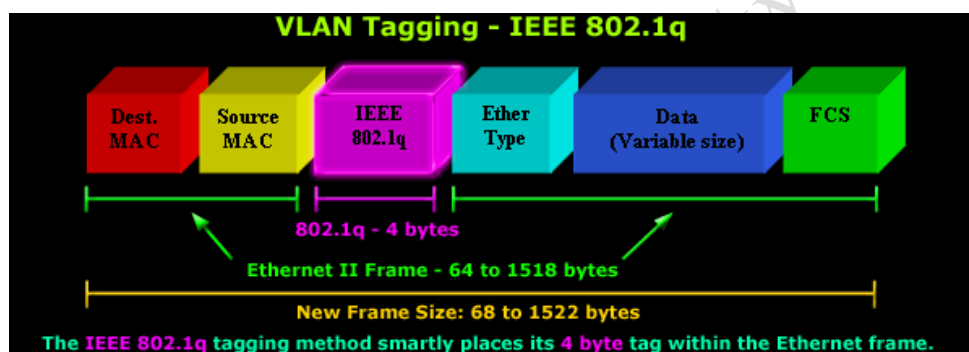
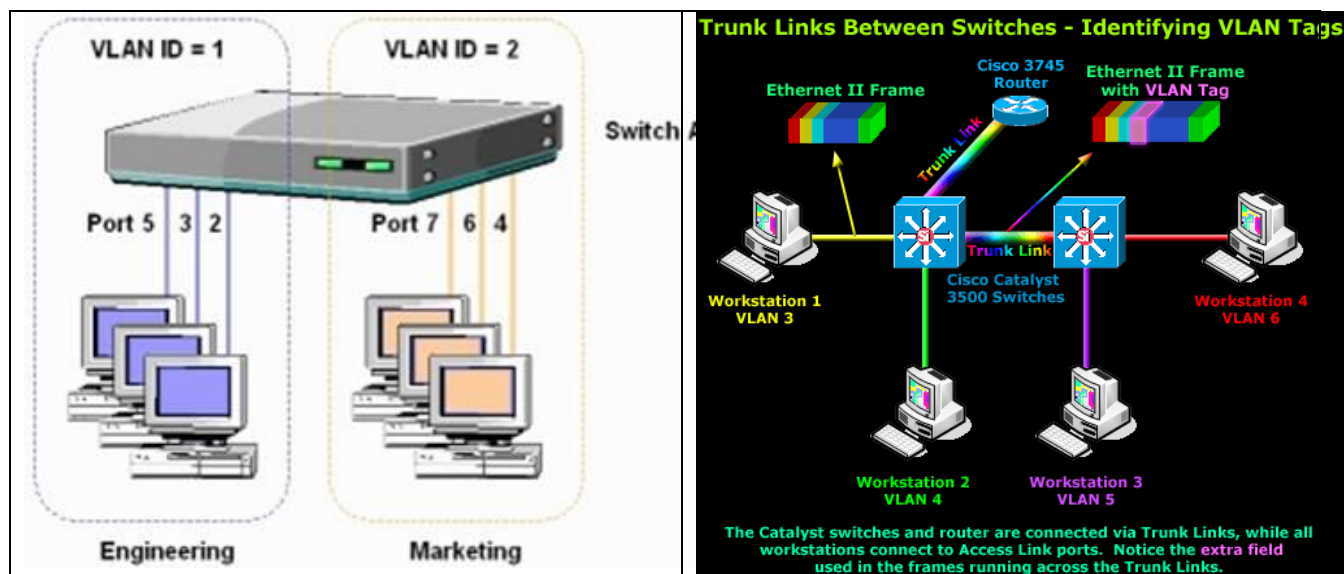
Laboratorul evaluează două scenarii legate de performanțele rețelei într-o rețea Ethernet, respectiv o rețea Ethernet în care au fost configurate VLAN-uri.

- **Informații suplimentare – Anexe Lab.**

2.3.3. Segmentarea rețelilor. Rețele virtuale (Virtual LAN-VLAN)

- În rețele LAN tradiționale, stațiile conectate la același switch împart același domeniu de broadcast. În acest domeniu, fiecare stație primește pachetele de broadcast trimise de fiecare dintre celelalte stații. **Cu cât numărul de stații crește, crește și broadcastul, ceea ce duce la creșterea încărcării rețelei cu efecte negative asupra performanței.** Pentru a împiedica acest lucru se poate realiza o **separare fizică în sensul că putem lega câteva stații la switch-ul A, altele la switch-ul B, apoi fiecare switch să fie conectat într-un router. Această soluție necesită investiția în echipamente hardware și nu este scalabilă.**
- VLAN-urile oferă izolarea logică în loc de segregarea fizică. Un VLAN este un set de stații care sunt tratate ca un singur domeniu de broadcast. Stațiile din același VLAN pot comunica unele cu altele, dar nu pot comunica cu stațiile din alt VLAN. Această izolare se realizează folosind **VLAN Tagging.**
- **Un tag VLAN** este o extensie de patru octeți a frame-ului Ethernet care transportă o prioritate (0-7) și un identificator (1-4096). Stațiile care suportă VLAN-urile pot aplica aceste tag-uri. Tag-urile pot fi adăugate și de switch-urile care suportă acest lucru, pe baza portului pe care sosește frame-ul.
- **Exemplu:** un switch poate fi programat în așa fel încât să știe că porturile 5, 3 și 2 aparțin VLAN-ului 1 și porturile 7, 6, 4 aparțin VLAN-ului 2. Switch-ul va înainta pachetele de broadcast spre toate porturile de pe același VLAN, dar niciodată spre porturile celuilalt VLAN. În figura de mai jos este prezentată această situație:

Retele de calculatoare – Informatica anul 3 (2019-2020)



Crearea si folosirea VLAN-urilor se face dupa cateva reguli, cum ar fi:

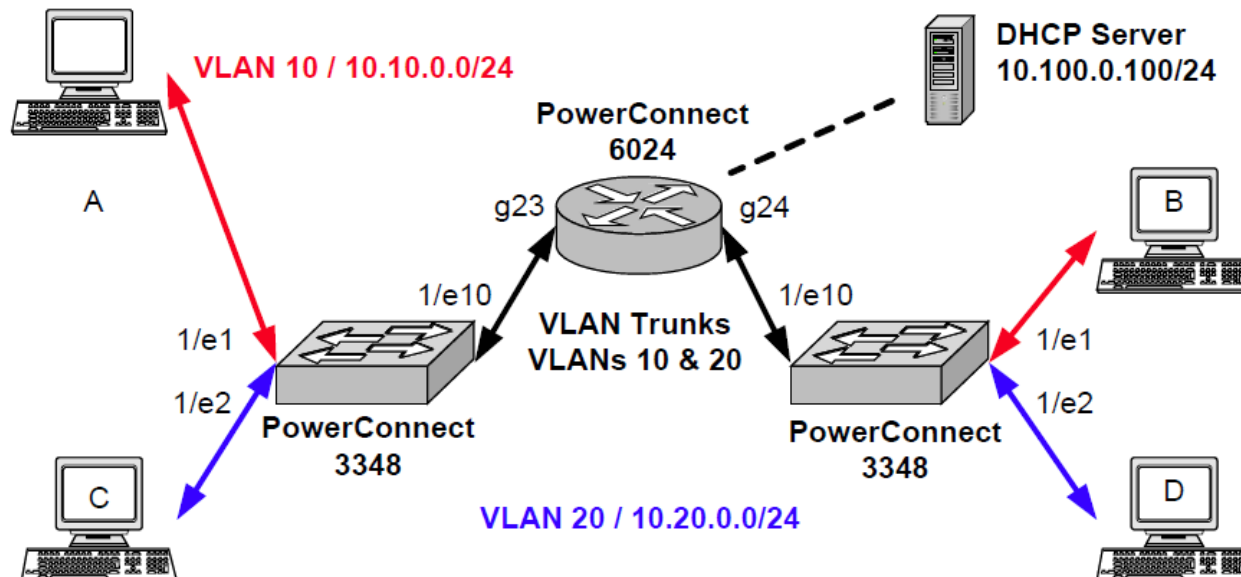
- traficul Layer 2 nu poate ajunge dintr-un VLAN in altul
- fiecare port trebuie sa apartina cel putin unui VLAN static. Implicit un port este membru untagged al VLAN-ului default
- un port poate exista in unul sau mai multe VLAN-uri, in acest caz, "tagging" este folosit pentru a identifica carui VLAN apartine un anumit frame
- un port poate fi definit ca untagged pentru nici un VLAN sau doar pentru un VLAN. Un port care este untagged pentru un VLAN transmite frame-urile destinate acelui VLAN fara tag-ul VLAN in frame-ul Ethernet
- un port poate fi definit ca tagged pentru nici un VLAN sau pentru mai multe VLAN-uri. Un port care este tagged pentru un VLAN transmite frame-urile destinate acelui VLAN cu tag-ul VLAN, incluzand si identificatorul numeric al VLAN-ului
- un port nu poate fi untagged si tagged in acelasi VLAN

Este posibila situatia in care se doreste ca sistemele din VLAN-uri diferite sa aiba acces la un server comun. In acest caz se defineste un port comun si se configureaza in mod corespunzator serverul. Pe server va aparea in cazul in care avem 2 VLAN-uri ca si cum ar fi doua placi de retea. Folosirea VLAN-urilor aduce pe langa alte avantaje si imbunatatirea performantelor retelei.

<http://www.commsdesign.com/showArticle.jhtml?articleID=26806942>

Retele de calculatoare – Informatica anul 3 (2019-2020)

Exemplu:



3. Partea practica (Tema – pag.30)

Informatii suplimentare Anexe Lab.

3.1. Aplicatii Wireshark

Se vor parcurge toate etapele de mai jos, folosind ca studiu de caz adrese convenabil alese. Capturile (snipping tools) realizate de studenti vor fi salvate intr-un document word, analizate si comentate. (+ Recapitulare Lab_03)

3.1.1. Frame-uri Ethernet – captura si analiza

Captura Ethernet frames (recomandare: stergere cache browser)

- Internet Explorer: *Tools->Internet Options->Delete Files.*
- Firefox: *Tools->Clear Private Data.*

• Porniti Wireshark (capture!);

• Introduceti URL-ul de mai jos in browser

<http://www.oracle.com/us/sun/index.htm>

• Opriti Wireshark

1. Identificati pachetul ce corespunde mesajului HTTP GET trimis de pe statia de lucru la

<http://www.oracle.com/us/sun/index.htm>

Rețele de calculatoare – Informatica anul 3 (2019-2020)

The screenshot shows the Wireshark network protocol analyzer interface. The top pane displays a list of 21 captured packets. The middle pane shows the details of the selected packet (Frame 4), which is an HTTP GET request. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.145	128.119.245.12	TCP	2038 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.050606	128.119.245.12	192.168.2.145	TCP	http > 2038 [SYN, ACK] Seq=0 Ack=1 Win=5
3	0.050729	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=1 Ack=1 Win=65535
4	0.055906	192.168.2.145	128.119.245.12	HTTP	GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1
5	0.128700	128.119.245.12	192.168.2.145	TCP	http > 2038 [ACK] Seq=1 Ack=453 Win=6432
6	0.134167	128.119.245.12	192.168.2.145	TCP	[TCP segment of a reassembled PDU]
7	0.150302	128.119.245.12	192.168.2.145	TCP	[TCP segment of a reassembled PDU]
8	0.150487	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=453 Ack=1762 Win=6
9	0.213639	128.119.245.12	192.168.2.145	TCP	[TCP segment of a reassembled PDU]
10	0.215724	128.119.245.12	192.168.2.145	TCP	[TCP Previous segment lost] [TCP segment
11	0.215947	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=453 Ack=3214 Win=6
12	0.231749	128.119.245.12	192.168.2.145	HTTP	[TCP Retransmission] HTTP/1.1 200 OK (text/html)
13	0.232145	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=453 Ack=4810 Win=6
14	0.320470	192.168.2.145	128.119.245.12	HTTP	GET /favicon.ico HTTP/1.1
15	0.403428	128.119.245.12	192.168.2.145	HTTP	HTTP/1.1 404 Not Found (text/html)
16	0.423932	192.168.2.145	168.66.12.224	TCP	2039 > http [SYN] Seq=0 Len=0 MSS=1460
17	0.579522	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=793 Ack=6235 Win=6
18	3.383584	192.168.2.145	168.66.12.224	TCP	2039 > http [SYN] Seq=0 Len=0 MSS=1460
19	9.392197	192.168.2.145	168.66.12.224	TCP	2039 > http [SYN] Seq=0 Len=0 MSS=1460
20	10.389131	128.119.245.12	192.168.2.145	TCP	http > 2038 [FIN, ACK] Seq=6235 Ack=793
21	10.389258	192.168.2.145	128.119.245.12	TCP	2038 > http [ACK] Seq=793 Ack=6236 Win=6

Frame 4 (506 bytes on wire, 506 bytes captured)

- Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: LinksysG_45:90:a8 (00:0c:41:45:90:a8)
- Internet Protocol, Src: 192.168.2.145 (192.168.2.145), Dst: 128.119.245.12 (128.119.245.12)
- Transmission Control Protocol, Src Port: 2038 (2038), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452
- Hypertext Transfer Protocol
 - GET /wireshark-labs/HTTP-ethereal-lab-file3.html HTTP/1.1\r\n
 - Request Method: GET
 - Request URI: /wireshark-labs/HTTP-ethereal-lab-file3.html
 - Request Version: HTTP/1.1
 - Host: gaia.cs.umass.edu\r\n
 - User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4\r\n
 - Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
 - Accept-Language: en-us,en;q=0.5\r\n
 - Accept-Encoding: gzip,deflate\r\n
 - Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
 - Keep-Alive: 300\r\n
 - Connection: keep-alive\r\n
 - \r\n

Raw packet data (hex/ascii):

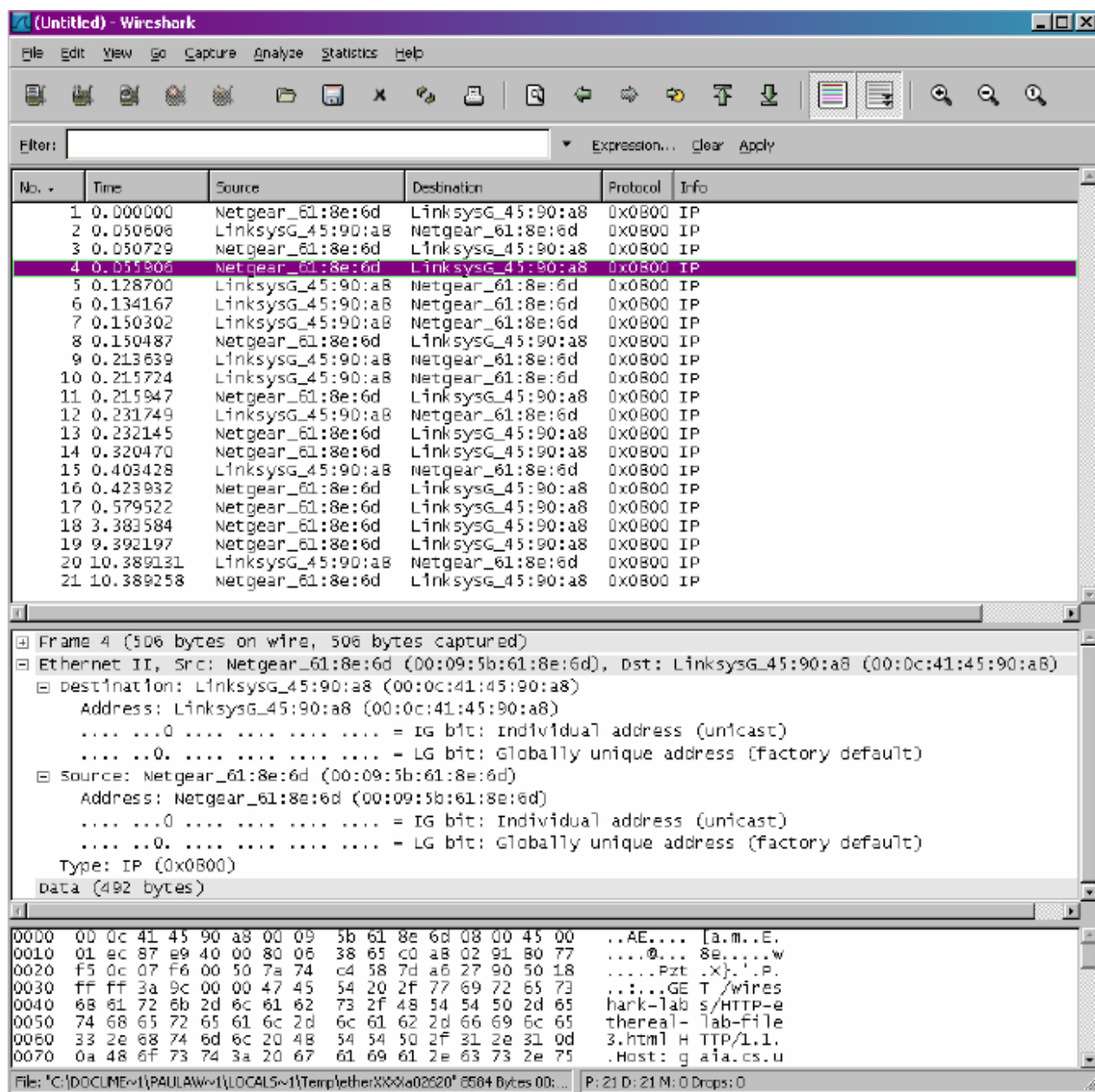
```

0040  58 61 72 65 2d 6c 61 62 73 2f 48 54 54 50 2d 65  hark-lab s/HTTP-
0050  74 68 65 72 65 61 6c 2d 6c 61 62 2d 66 69 6c 65  thereal- lab-file
0060  33 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d  3.html H TTP/1.1
0070  0a 48 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75  .Host: gaia.cs.u
0080  6d 61 73 73 2e 65 64 75 0d 0a 55 73 65 72 2d 41  mass.edu .user-A
0090  67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e  gent: Mo zilla/5.

```

- Pentru Ethernet si ARP nu suntem interesati de IP sau protocoalele de pe nivelele superioare. Fereastra “listing of captured packets” informatii numai despre protocoalele mai jos de IP: Selectati *Analyze->Enabled Protocols*; Debifati IP ; OK (Obs: La final revenim cu bifarea IP (enable))
- Se obtine o fereasta aseanatoare celei de mai jos:

Rețele de calculatoare – Informatica anul 3 (2019-2020)



- Pentru a raspunde urmatoarelor intrebari este necesar sa se vizualizeze ferestrele Detalii pachet si Continut:
- Selectati frame-ul Ethernet ce contine mesajul HTTP GET ; Expandati informatiile Ethernet II in fereastra Detalii pachet; Continutul frame-ului Ethernet este disponibil in fereastra Continut pachet.

Tiparire pachet: *File->Print; Selected packet only; Packet summary line* si selectati dimensiunea minima a pachetului de care este nevoie pentru a raspunde la intrebari:

1. Adresa MAC sursa (48-bit Ethernet) a calculatorului pe care lucrati;

Retele de calculatoare – Informatica anul 3 (2019-2020)

2. Adresa Ethernet destinatie. Este aceasta corespunzatoare masinii

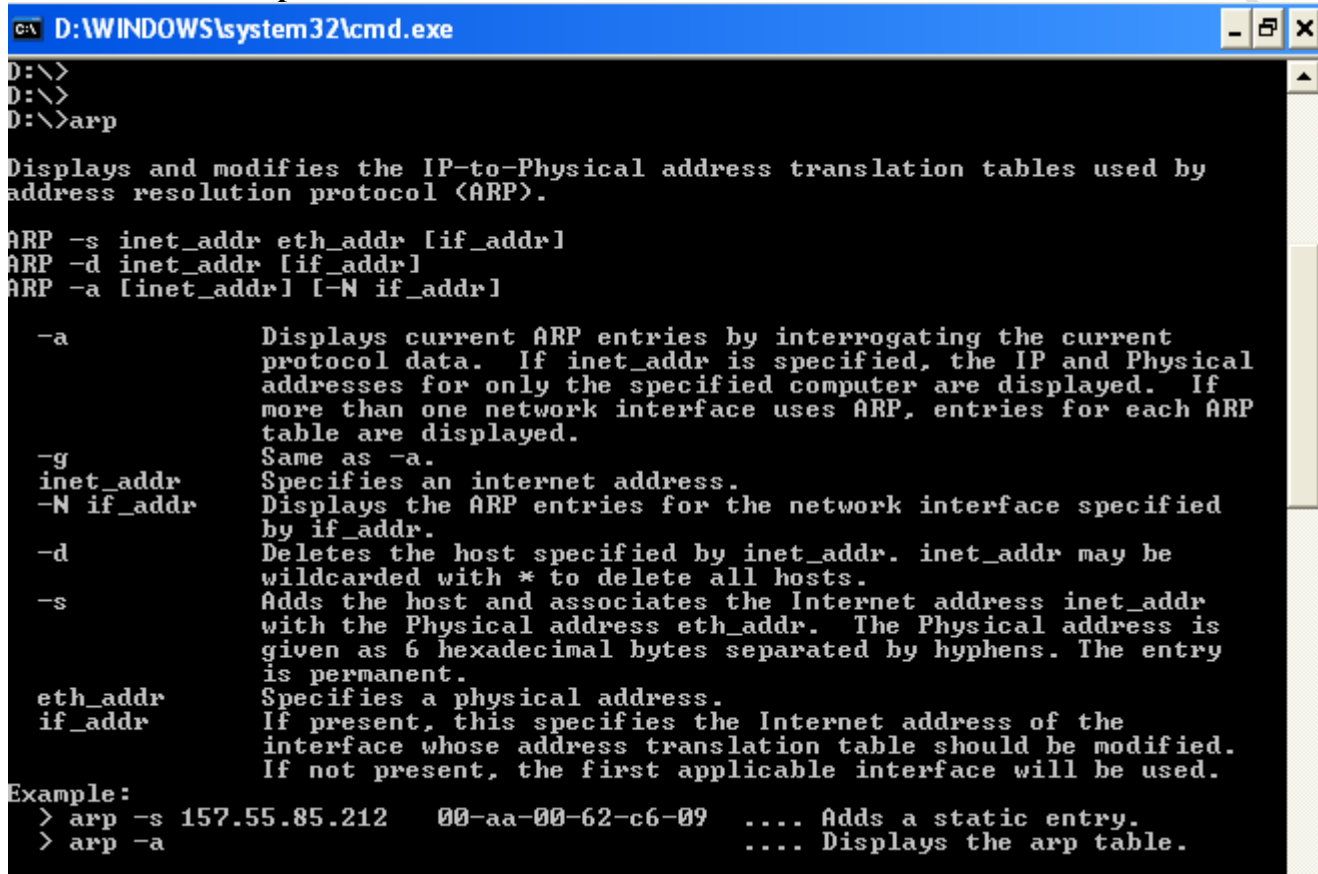
<http://www.oracle.com/us/sun/index.htm>

(Raspuns propus: Nu. **Precizati de ce?**). **Ce dispozitiv are aceasta adresa hardware?**

3. Explicati inregistrările campurilor a 2 frame-uri Ethernet in contextul informatiilor din anexa 1.

3.1.2. Address Resolution Protocol

3.1.2.1. Comanda arp :



```
C:\D:\WINDOWS\system32\cmd.exe
D:\>
D:\>
D:\>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]

-a          Displays current ARP entries by interrogating the current
            protocol data.  If inet_addr is specified, the IP and Physical
            addresses for only the specified computer are displayed.  If
            more than one network interface uses ARP, entries for each ARP
            table are displayed.
-g          Same as -a.
inet_addr   Specifies an internet address.
-N if_addr  Displays the ARP entries for the network interface specified
            by if_addr.
-d          Deletes the host specified by inet_addr.  inet_addr may be
            wildcarded with * to delete all hosts.
-s          Adds the host and associates the Internet address inet_addr
            with the Physical address eth_addr.  The Physical address is
            given as 6 hexadecimal bytes separated by hyphens.  The entry
            is permanent.
eth_addr    Specifies a physical address.
if_addr     If present, this specifies the Internet address of the
            interface whose address translation table should be modified.
            If not present, the first applicable interface will be used.

Example:
> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry.
> arp -a .... Displays the arp table.
```

Verificati comanda arp cu toate optiunile de mai sus; efectuati ping-uri catre 3 IP-uri din reteaua interna (LAN) si 2 IP-uri din exterior (exemplu: 104.127.29.79; 192.0.32.8) ; vizualizati si faceti capture ale tabelului arp inainte si dupa pinguri. Ce observati? Comentati raspunsul/ raspunsurile

3.1.2.2. Captura Ethernet frames (cu stergere cache browser)

Stergeti cache-ul arp (folositi **arp -d**)

- Internet Explorer: *Tools->Internet Options->Delete Files.*
- Firefox: *Tools->Clear Private Data.*
- Porniti Wireshark
- Introduceti URL in browser <http://www.oracle.com/us/sun/index.htm>
- Opriti Wireshark

Retele de calculatoare – Informatica anul 3 (2019-2020)

Pentru ARP nu suntem interesati de IP sau protocoalele de pe nivelele superioare. Fereastra “listing of captured packets” informatii numai despre protocoalele mai jos de IP: Selectati *Analyze->Enabled Protocols*; Debifati IP; OK.

Se obtine o fereasta aseanatoare celei de mai jos:

The image shows a Wireshark packet capture window titled "ethernet-ethereal-trace-1 - Wireshark". The packet list on the left shows 17 packets. Packet 1 is an ARP request from AmbitMic_a9:3d:68 to Broadcast (ff:ff:ff:ff:ff:ff) asking for the IP address 192.168.1.1. The packet details pane on the right shows the structure of the ARP request, including the Ethernet II header, ARP request opcode, and the sender and target MAC and IP addresses. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	AmbitMic_a9:3d:68	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.105
2	0.001018	LinksysG_da:af:73	AmbitMic_a9:3d:68	ARP	192.168.1.1 is at 00:06:25:da:af:73
3	0.001028	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
4	2.962850	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
5	8.971488	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
6	13.542974	Telebit_73:8d:ce	Broadcast	ARP	Who has 192.168.1.117? Tell 192.168.1.104
7	17.444423	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
8	17.465902	LinksysG_da:af:73	AmbitMic_a9:3d:68	0x0800	IP
9	17.465927	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
10	17.466468	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
11	17.494766	LinksysG_da:af:73	AmbitMic_a9:3d:68	0x0800	IP
12	17.498935	LinksysG_da:af:73	AmbitMic_a9:3d:68	0x0800	IP
13	17.500025	LinksysG_da:af:73	AmbitMic_a9:3d:68	0x0800	IP
14	17.500069	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP
15	17.527057	LinksysG_da:af:73	AmbitMic_a9:3d:68	0x0800	IP
16	17.527422	LinksysG_da:af:73	AmbitMic_a9:3d:68	0x0800	IP
17	17.527457	AmbitMic_a9:3d:68	LinksysG_da:af:73	0x0800	IP

Frame 1 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: AmbitMic_a9:3d:68 (00:d0:59:a9:3d:68)
Sender IP address: 192.168.1.105 (192.168.1.105)
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.1.1 (192.168.1.1)

0000 ff ff ff ff ff ff 00 d0 59 a9 3d 68 08 06 00 01 Y.=h....
0010 08 00 06 04 00 01 00 d0 59 a9 3d 68 c0 a8 01 69 Y.=h...1
0020 00 00 00 00 00 00 c0 a8 01 01

Raspundeti la urmatoarele intrebari argumentatnd prin informatiile corespunzatoare capturilor efectuate si atasate:

1. Care sunt adresele sursa si destinatie pentru un mesaj ARP request?
2. Indicati valoarea in hexadecimal pentru Ethernet Frame type field (2 octeti). Comentati valoarea obtinuta in binar..
3. Testati filtrele indicate in anexa 3.

Retele de calculatoare – Informatica anul 3 (2019-2020)

3.2. Studiu de caz (Riverbed Modeler Academic Edition)

3.2.1. Obiectiv: Utilizarea Modeler pentru studiul îmbunătățirii performanțelor unei rețele LAN prin configurarea de VLAN-uri.

- **Cursanții sunt încurajați** să folosească materialul de mai jos într-un mod constructiv, astfel încât să evalueze caracteristicile legăturilor fizice și a dispozitivelor de rețea folosite (*click dreapta, view link/node description*), a modelelor de rețea precum și elementele de bază privind simularea **sistemelor cu evenimente discrete** analizate; rezultatele obținute grafic vor fi analizate și interpretate în contextul cerințelor prezentate mai jos. **Indicație: Anexa 8 și referințele bibliografice (+ Lab_01, Lab_02, Lab_03).**

Sistem cu eveniment discret: “fie un sistem real, fie un model matematic (ce descrie funcționarea unui sistem real), a cărui evoluție este raportată la apariția unor evenimente. Astfel, producerea evenimentelor joacă rolul de *cauză* pentru dinamica sistemului și are drept *efect* modificare stărilor sistemului, evidențiind o certă similitudine cu așa-numita „tratare pe stare” a sistemelor continue sau discrete în timp. Mai mult chiar, și în cazul unui sistem cu evenimente discrete se poate vorbi despre o *funcție de tranziție a stărilor*, care formalizează riguros faptul că sistemul trece dintr-o stare în alta numai ca urmare a producerii unui eveniment și că sistemul păstrează starea în care se află până la producerea unui nou eveniment. Analogia cu sistemele continue sau discrete în timp, pe care le vom referi sub numele de „sisteme clasice” trebuie însă utilizată concomitent cu înțelegerea corectă și completă a deosebirilor privind interpretarea cauzală a comportării. Dacă în cazul sistemelor clasice, cauzele și efectele sunt valorile unor semnale, care, cel puțin sub raport teoretic, prin variații acoperă intervale (adică mulțimi cu aceeași cardinalitate ca \mathbf{R}), în cazul sistemelor cu evenimente discrete, *mulțimea evenimentelor* ce pot apărea, precum și *mulțimea stărilor* în care poate tranzita sistemul sunt *discrete* (adică au cel mult cardinalitatea lui \mathbf{N}).”

3.2.2. Scenariu: O firmă de consultanță are birouri localizate în mai multe clădiri și recent a angajat personal suplimentar, volumul de trafic din rețeaua LAN înregistrând o creștere semnificativă. Deoarece firma nu vrea să investească într-o nouă infrastructură de rețea, inginerul de sistem trebuie să găsească cea mai bună soluție de utilizare a infrastructurii existente.

Laboratorul evaluează 2 scenarii legate de performanțele rețelei într-o rețea Ethernet, respectiv o rețea Ethernet în care au fost configurate VLAN-uri.

Instrucțiuni

1. Porniți Modeler

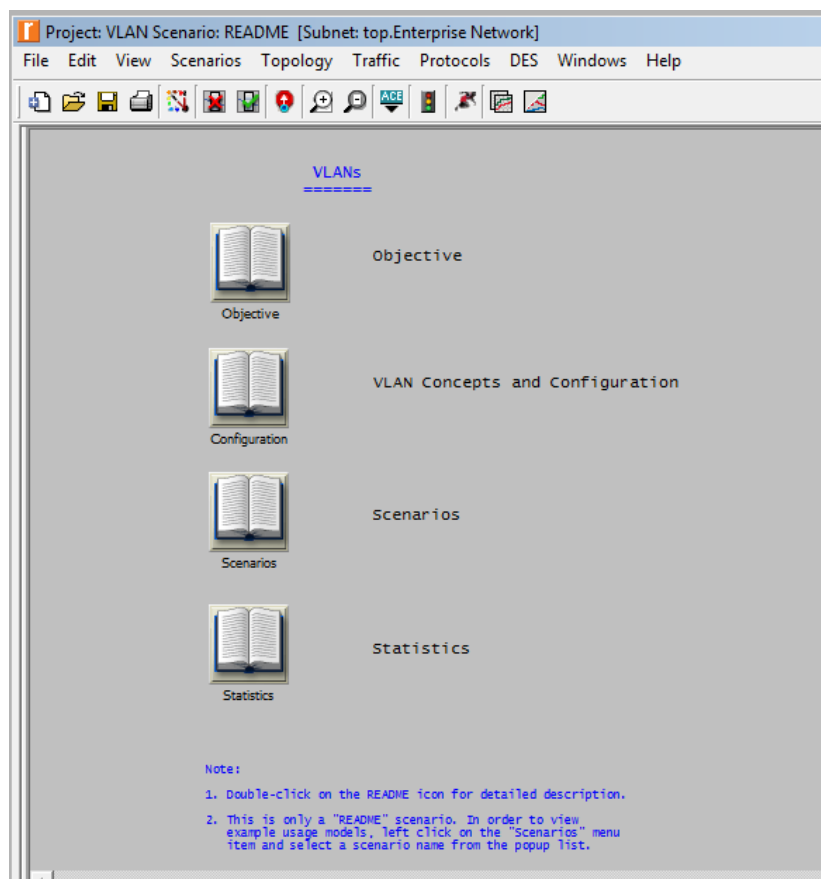
2. [Se deschide Proiectul VLAN din Example Networks și se salvează \(Save As\) în op. models →](#)

[Se rulează scenariile I. no vlan și II. three vlans](#)

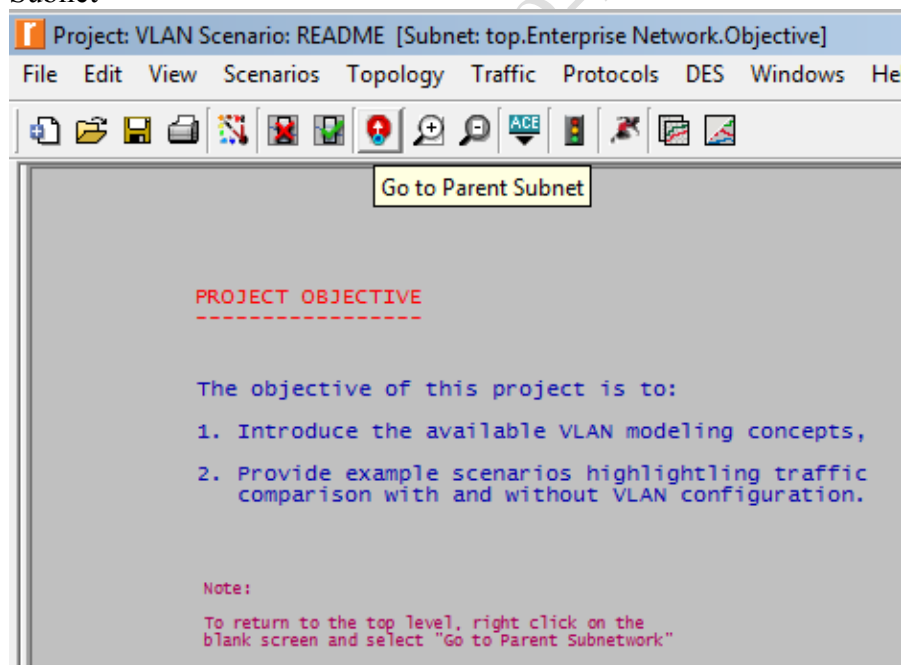
Atenție: Dacă se lucrează pe proiectul original VLAN.Project NU SE SALVEAZA nici-o modificare! Aceasta pentru a se putea reveni în bune condiții la proiectul functional.

File -> Open -> example_networks (în stanga!) -> VLAN. Project (dublu click) -> VLAN (dublu click)

Retele de calculatoare – Informatica anul 3 (2019-2020)



2.1 Documentare: Objective, Configuration, Scenarios, Statistics (pentru revenire, “Go to Parent Subnet”)



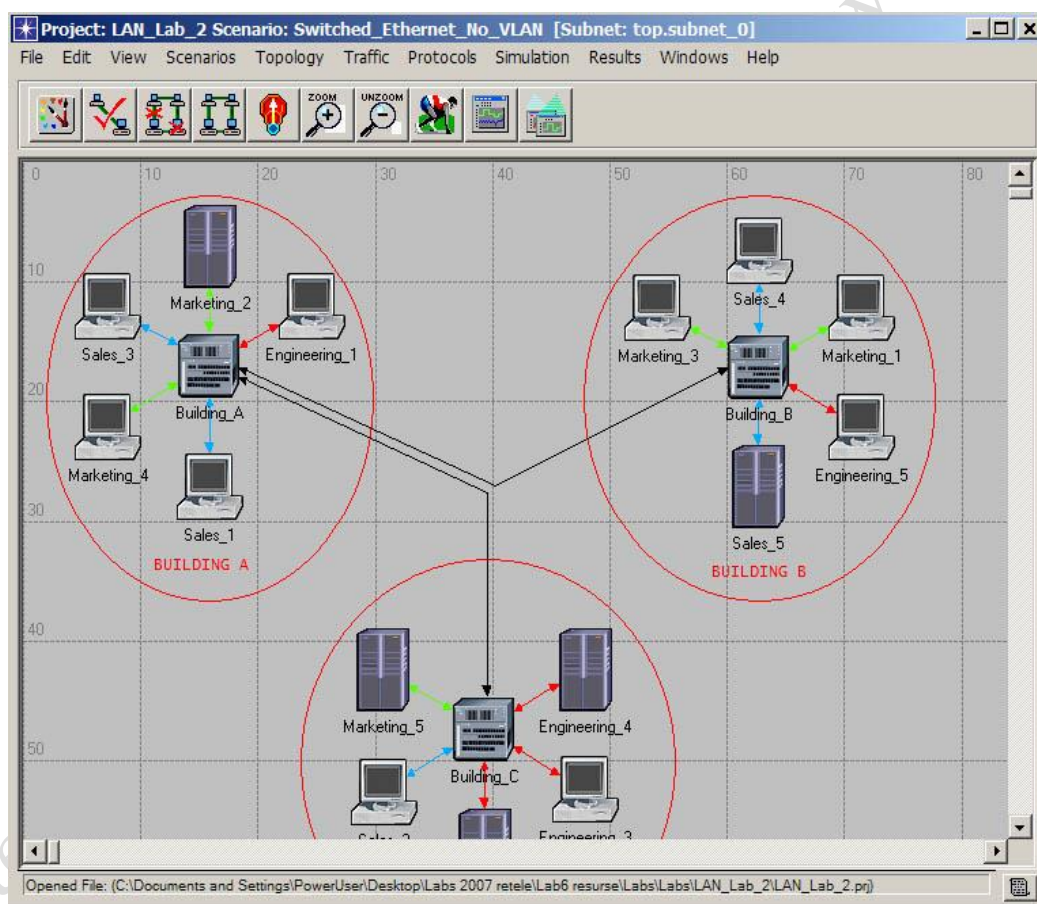
3. Primul scenariu se numeste Switched_Ehernet_No_VLAN (lucru vizibil in bara de titlu a aplicatiei). Pentru schimbarea scenariului se poate apela meniul **Scenarios – Switch to Scenario**.

Retele de calculatoare – Informatica anul 3 (2019-2020)

În acest scenariu, utilizatorii care fac parte din departamente diferite: vânzări, marketing, proiectare – sunt conectați la switch-uri Ethernet denumite Building_A, Building_B și Building_C, câte unul pentru fiecare clădire. Grupurile de utilizatori accesează diverse servere de fișiere, baze de date, HTTP, FTP sau email. Laboratorul va simula traficul generat de utilizatori între servere și stațiile lor de lucru. **Pentru a vizualiza fluxurile de date între clienți și servere se apelează comanda View - Demand Objects – Show All.**

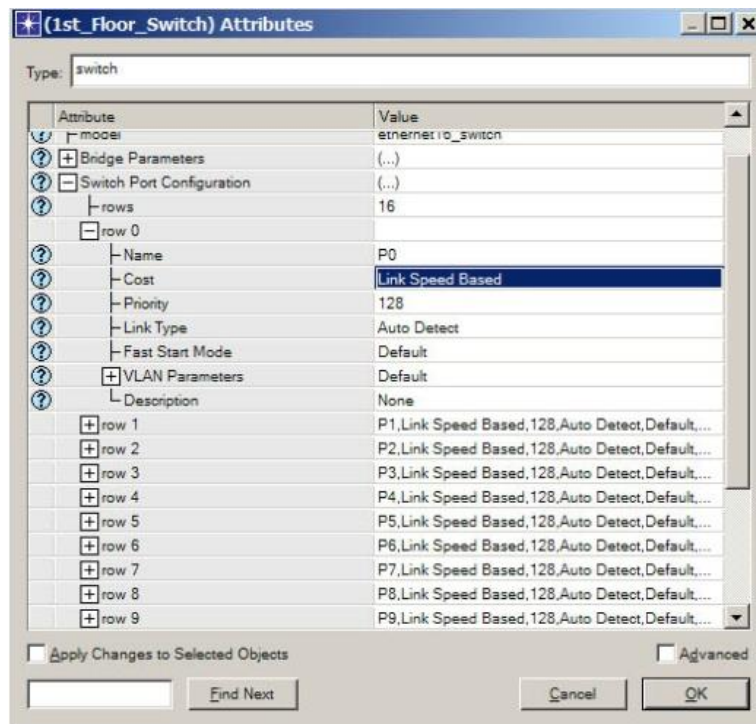
[Varianta tutorial în limba engleză](#)

Modelele Riverbed Modeler sunt formate din obiecte, care sunt descrise prin atribute. Atributele unui obiect pot fi analizate prin selectarea Edit Attributes din meniul contextual. Explorați atributele următoarelor obiecte: Application Config, Profile Config, Tasks și Workstation. Aceste atribute definesc traficul care este generat în cadrul rețelei.



4. Selectați unul dintre switch-uri (Building_A, Building_B sau Building_C) și deschideți fereastra Attributes.

Retele de calculatoare – Informatica anul 3 (2019-2020)



5. Selectati **Switch Port Configuration** si apoi **row 0** (sau alt port) pentru a avea acces la detalii. Observati valoarea costului Link Speed Based. Daca se face click pe ? atunci sunt afisate informatii despre semnificatia costului. Deschideti **VLAN Parameters** si verificati daca campul *Schemes* are valoarea **No VLANs**. Aceasta indica faptul ca nici un VLAN nu este configurat in retea.

6. In modelul Opnet, rata de transfer care este suportata de un switch este specificata de legatura catre respectivul echipament. Selecati Edit Attributes pentru o legatura si observati rata de transfer.

Configurarea si lansarea simularii

Scopul simularii este evaluarea performantelor retelei prin simularea utilizarii timp de 8 ore.

- click pe butonul configure/run simulation sau **Simulation** → **Configure Discrete Event Simulation**
- se selecteaza durata simularii
- se lanseaza in executie simulatorul

Pe durata simularii sunt furnizate informatii privind rezultatele.

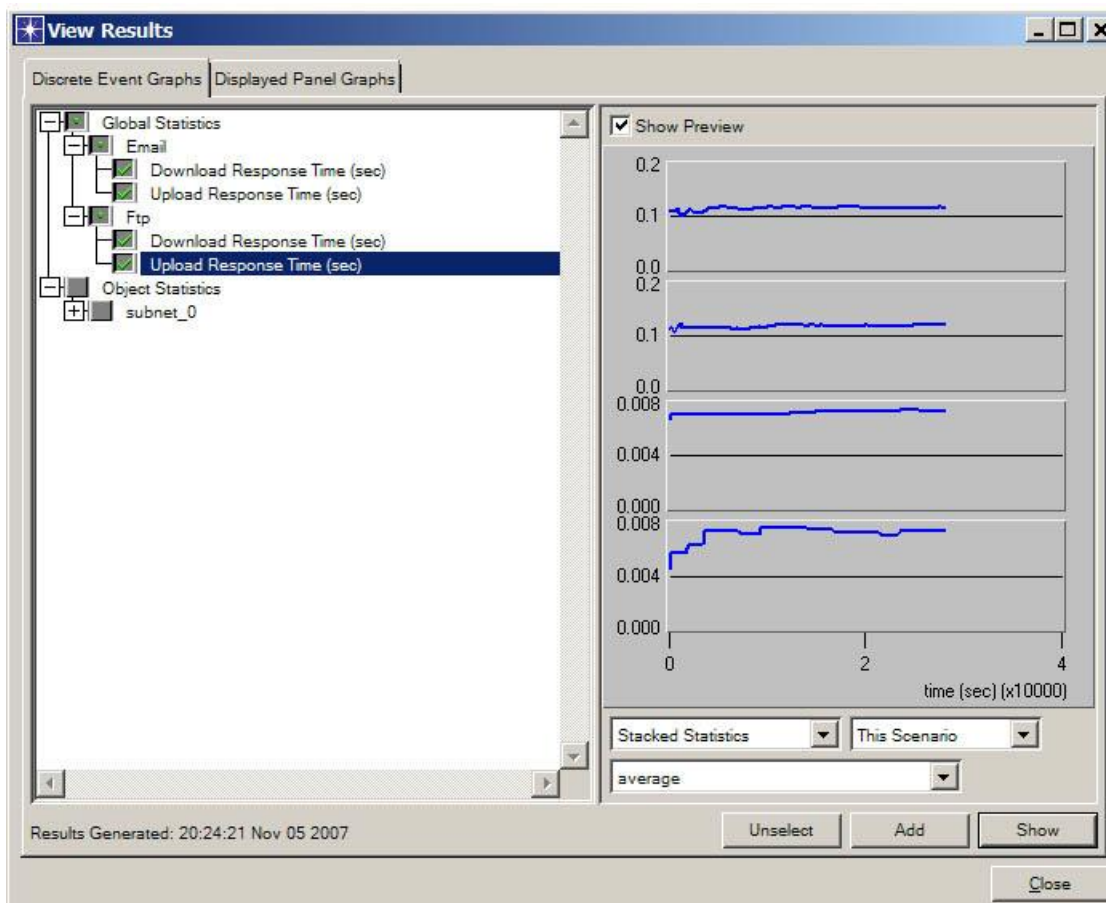
Se inchide fereastra prin selectarea butonului **Close**

Vizualizarea rezultatelor

Se va studia timpul de raspuns al aplicatiilor pentru Email si FTP asa cum este observat de utilizatori si statistici privind throughput-ul pentru switch-uri.

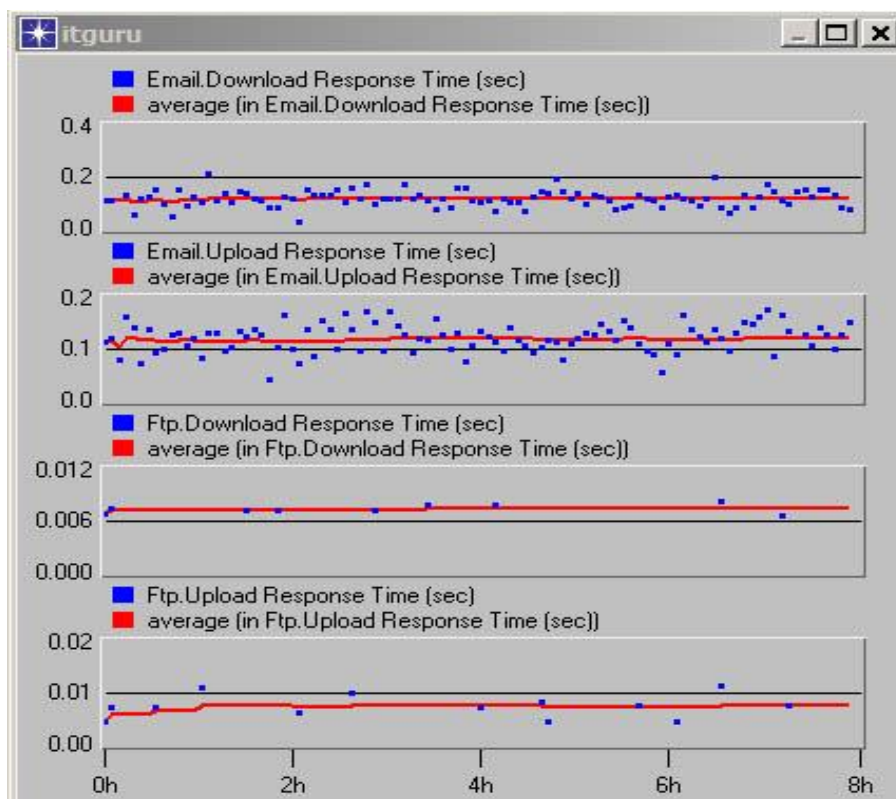
Retele de calculatoare – Informatica anul 3 (2019-2020)

1. Click Results – View Results
2. Expand Global Statistics, Email si FTP
3. Selecatați Download Response Time (sec) si Upload Response Time (sec) pentru Email si FTP.
4. Selecatați meniul **As Is** si alegeti **average**. Selectati Add si faceti click pe primul grafic realizat la pasul precedent.

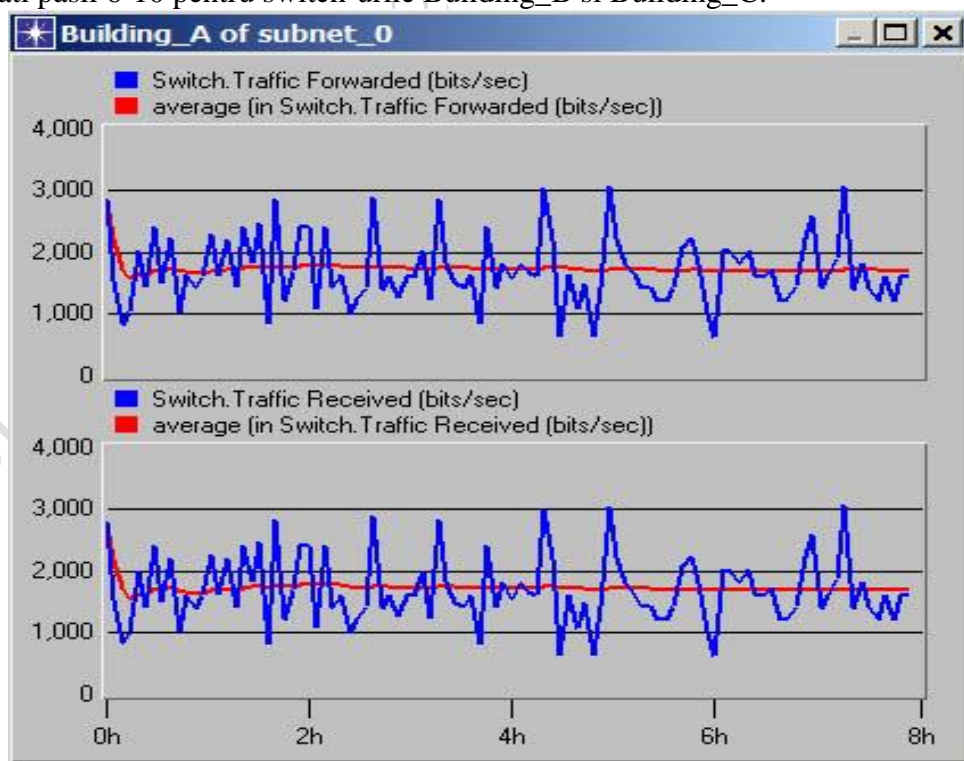


5. Deselectati graficele pentru Email si FTP
6. Selecati **Object Statistics** si activati **subnet_0**
7. Selectati **Building_A** si apoi **Switch**
8. Selectati **Traffic Received (bits/sec)** si **Traffic Forwarded (bits/sec)**
9. Click **Show** (Stacked Statistics, This Scenario , As Is)

Retele de calculatoare – Informatica anul 3 (2019-2020)



10. Selecatați meniul **As Is** și alegeți **average**. Selectați Add și faceți click pe primul grafic realizat la pasul precedent.
11. Repetați pașii 6-10 pentru switch-urile Building_B și Building_C.

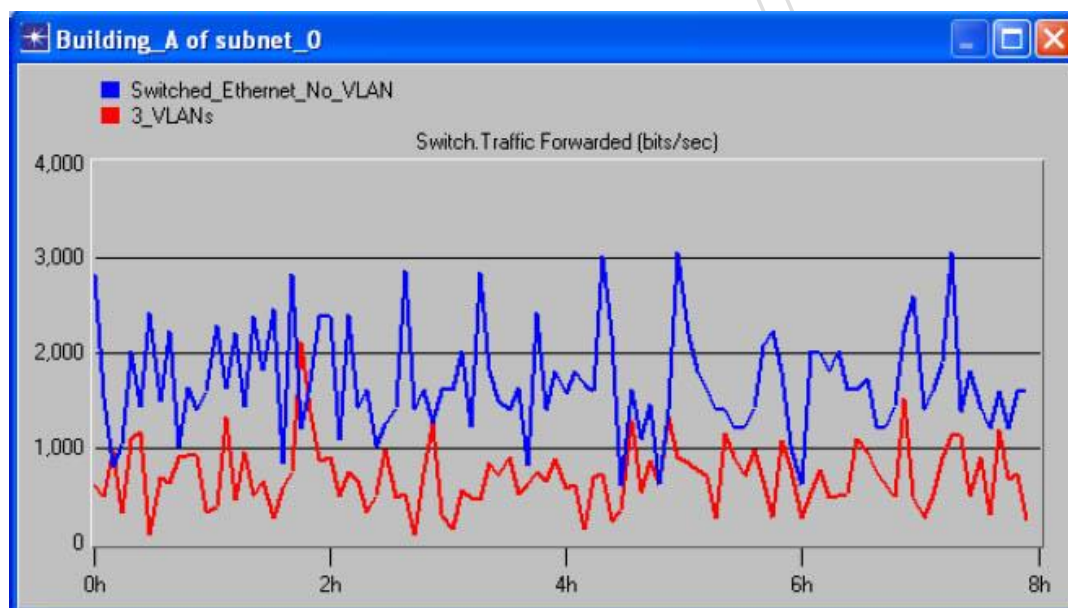


Retele de calculatoare – Informatica anul 3 (2019-2020)

Nota: Notati valorile minime, maxime si medii pentru timpul de raspuns al aplicatiilor si pentru switch throughput.

Schimbati scenariul pentru a simula utilizarea VLAN-urilor

1. Click Scenarios – Switch Scenarios
2. Selectati **3_VLANs**
3. Pentru a examina modificarile necesare configurarii VLAN-urilor, selectati oricare dintre cele 3 switch-uri (Building_A, Building_B sau Building_C) si selectati Edit Attributes.
4. Selectati **VLAN Parameters**. Campul *Scheme* are acum valoarea **Port-Based VLAN**, fata de **No VLAN** ca in primul scenariu.
5. Click (...) din campul valoare al atributului **Supported VLAN** si selectati Edit. Observati cei trei identificatori de VLAN-uri (10, 20 si 30).
6. Repetati toti pasii anteriori pentru lansarea in executie a simularii si vizualizarea rezultatelor.
7. Comparati rezultatele **Results – Compare Results**. Expand Global Statistics si selectati una dintre aplicatii, de exemplu (Object Statistics, subnet_0, Building_A/Switch/Traffic Forwarded(bits/sec))
8. Click Show.



Ce se observa?

(Sugeram o analiza a variantei de raspuns din **Anexa 5**, in raport cu un punct de vedere argumentat prin propriile rezultate)

3.3. Studiu de caz (Modeler/IT Guru Academic Edition):

Retea Ethernet

Se va parcurge integral [Ethernet Network.pdf](#) (document atasat!)

Cursantii sunt incurajati sa foloseasca materialul de mai jos intr-un mod constructiv/ mobilizator !!!!!, astfel incat sa evalueze caracteristicile legaturilor fizice si a dispozitivelor de retea folosite (*click*

Retele de calculatoare – Informatica anul 3 (2019-2020)

dreapta, view link/node description), a modelelor de retea precum si elementele de baza privind simularea *sistemelor cu evenimente discrete* analizate; **Indicatie: Anexe Lab.**

- rezultatele obtinute grafic vor fi analizate si interpretate in contextul cerintelor prezentate in [Ethernet Network.pdf](#).

3.4. Aplicatii de retea in Python

3.4.1. Recapitulare (Lab_02, Lab_03)

- Python_intro
- Programare_Python
- Byte-of-python

3.4.2. Aplicatie: Conversia IP Multicast – MAC multicast

Indicatii:

ipTOMac.py

```
1 import sys
2
3 '''
4 def verify_ip(mcast_ip):
5     This function takes a multicast IP (string) as an argument \
6     and returns True if IP address is correct
7
8     Multicast IP address length
9     Number of octets
10    Format of every octet (0-255)
11    First octet is from multicast range (224-239)
12 '''
13
14
15 def ipTOMac(mcast_ip):
16     '''
17     Function ipTOMac takes multicast IP address as an argument and returns \
18     multicast MAC address
19     '''
20     # if not(verify_ip(mcast_ip)):
21     #     sys.exit(0)
22     mcast_mac = '01:00:5e:'
23     octets = mcast_ip.split('.')
24     second_oct = int(octets[1]) & 127
25     third_oct = int(octets[2])
26     fourth_oct = int(octets[3])
27     mcast_mac = mcast_mac + format(second_oct, '02x') + ':' + format\
28     (third_oct, '02x') + ':' + format(fourth_oct, '02x')
29     return mcast_mac
30 if __name__ == '__main__':
31     if len(sys.argv) != 2:
32         print ("Usage: ./ipTOMac.py <IP Multicast>")
33     else:
34         print (ipTOMac(sys.argv[1]))
35
```

Retele de calculatoare – Informatica anul 3 (2019-2020)

Output:

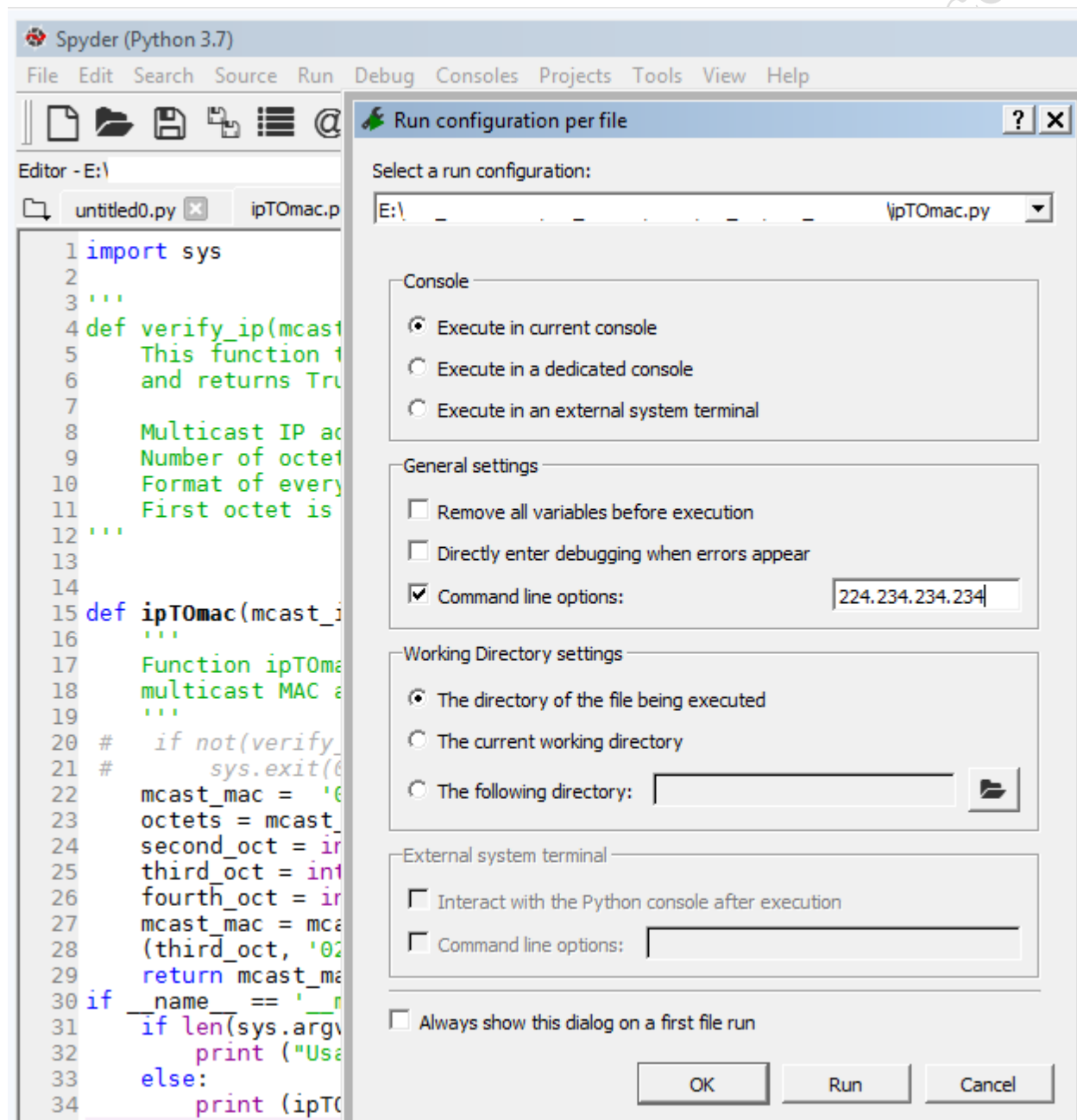
Anaconda Prompt:

```
Anaconda Prompt

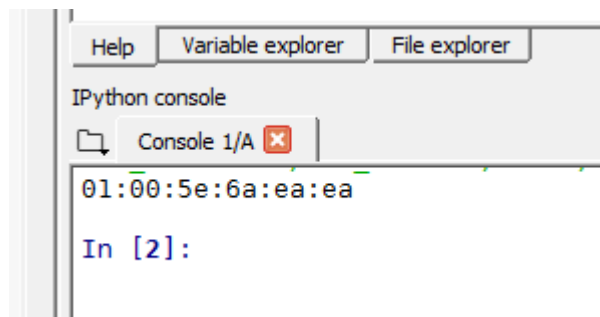
(base) E:\>python ipT0mac.py 224.234.234.234
01:00:5e:6a:ea:ea
```

Spyder:

Run -> Configure -> Command line options



Retele de calculatoare – Informatica anul 3 (2019-2020)



Tema Python 3.4.2:

Pentru aplicatia **ipTOMac_Nume_Prenume.py** :

- Documentarea solutiei partiale prezentate (algoritm, instructiuni etc) - <https://docs.python.org/3.7/genindex.html>
- Implementarea functiei pentru verificarea adresei IP ca fiind IP Multicast
- Completarea programului cu afisarea tuturor adreselor IP multicast care au aceeasi adresa MAC Multicast (conform figurii de mai jos – **All Multicast IPs**)

d. **Challenge:** Interfata grafica pentru aplicatia care raspunde cerintelor de la punctele 1 si 2

Recomandare: Qt Designer , cu Designer din Anaconda prompt).

<http://pythonforengineers.com/your-first-gui-app-with-python-and-pyqt/>,

<https://www.codementor.io/deepakasingh04/design-simple-dialog-using-pyqt5-designer-tool-ajskrd09n>, <https://wiki.python.org/moin/PyQt/Tutorials>

Multicast IP		Multicast MAC		All Multicast IPs
224.234.234.234	>>>	01-00-5e-6a-ea-ea	>>>	224.106.234.234
				224.106.234.234
				224.234.234.234
				225.106.234.234
				225.234.234.234
				226.106.234.234
				226.234.234.234
				227.106.234.234
				227.234.234.234
				228.106.234.234
				228.234.234.234
				229.106.234.234
				229.234.234.234
				230.106.234.234
				230.234.234.234
				231.106.234.234
				231.234.234.234
				232.106.234.234
				232.234.234.234
				233.106.234.234
				233.234.234.234
				234.106.234.234
				234.234.234.234
				235.106.234.234
				235.234.234.234
				236.106.234.234
				236.234.234.234
				237.106.234.234
				237.234.234.234
				238.106.234.234
				238.234.234.234
				239.106.234.234
				239.234.234.234

Cerinta aplicatie: All Multicast IPs

3.4.3 Aplicații *pyshark*

pyshark – pachet Python care permite analiza PDU-urilor folosind decodarea Wireshark

3.4.3.1 Documentare ([4_pyshark.pdf](#))

3.4.3.2 Aplicație - Citirea pachetelor stocate într-un fișier pcap

Aplicația permite acces la atribute precum numărul de pachet și informații complete pentru fiecare strat, cum ar fi protocolul său, adresa IP, adresa mac și flag-uri etc.

Indicații:

```
pyshark_1.py
1 import pyshark
2 cap = pyshark.FileCapture('D:/.../.../http.pcap') # path for http.pcap with "/"
3 cap
4 print(cap[0])
```

Output:

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\EP>d:
(base) D:\>cd D:\ .....
(base) D:\ ..... >python pyshark_1.py
Packet (Length: 62)
Layer ETH:
  Destination: fe:ff:20:00:01:00
  Address: fe:ff:20:00:01:00
  .... 1. .... = LG bit: Locally administered address (this is NOT the factory default)
  .... 0 .... = IG bit: Individual address (unicast)
  Source: 00:00:01:00:00:00
  Type: IPv4 (0x0800)
  Address: 00:00:01:00:00:00
  .... 0 .... = LG bit: Globally unique address (factory default)
  .... 0 .... = IG bit: Individual address (unicast)
Layer IP:
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  .... 000 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 48
  Identification: 0xf41 (3905)
  Flags: 0x4000, Don't fragment
  0... .... = Reserved bit: Not set
  1... .... = Don't fragment: Set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
  Header checksum: 0x91eb [validation disabled]
  Header checksum status: Unverified
  Source: 145.254.160.237
  Destination: 65.208.228.223
Layer TCP:
  Source Port: 3372
  Destination Port: 80
  Stream index: 0
  TCP Segment Len: 0
  Sequence number: 0 (relative sequence number)
  Next sequence number: 0 (relative sequence number)
  Acknowledgment number: 0
```


3.4.3.3 Aplicatie - Filtrarea frame-urilor dns stocate într-un fișier pcap

Indicatii

```
pyshark_2.py
1 import pyshark
2 cap = pyshark.FileCapture('D:/http.cap', display_filter="dns")
3 for pkt in cap:
4     print(pkt.ip)
```

Output

```
Anaconda Prompt (Anaconda3)

(base) D:\>python pyshark_2.py
Layer IP:
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 75
Identification: 0x0f49 (3913)
Flags: 0x0000
0... .. = Reserved bit: Not set
.0.. .. = Don't fragment: Not set
..0. .. = More fragments: Not set
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 128
Protocol: UDP (17)
Header checksum: 0x63a5 [validation disabled]
Header checksum status: Unverified
Source: 145.254.160.237
Destination: 145.253.2.203

Layer IP:
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 174
Identification: 0x1595 (5525)
Flags: 0x4000, Don't fragment
0... .. = Reserved bit: Not set
.1.. .. = Don't fragment: Set
..0. .. = More fragments: Not set
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 249
Protocol: UDP (17)
Header checksum: 0xa3f5 [validation disabled]
Header checksum status: Unverified
Source: 145.253.2.203
Destination: 145.254.160.237
```

Tema: Modificarea programului pentru afisarea numarului frame-ului si afisarea informatiilor de pe fiecare strat in parte

3.4.3.4 Aplicatie – Afisarea numarului de frame-uri stocate într-un fișier pcap si a continutului acestora (a tuturor sau a unuia anume, de exemplu cap[0])

Retele de calculatoare – Informatica anul 3 (2019-2020)

Indicatii

```
pyshark_3.py
1 # python3
2 # https://realpython.com/python-kwargs-and-args/
3
4 import pyshark
5
6 packets_array = []
7
8 def counter(*args):
9     packets_array.append(args[0])
10
11 def count_packets():
12     cap = pyshark.FileCapture('D:/http.cap', keep_packets=False)
13     cap.apply_on_packets(counter, timeout=10000)
14     return len(packets_array)
15
16 print("Packets number:"+str(count_packets()))
17
18 for packet in packets_array[0]:
19     print(packet)
20
```

Output

```
Anaconda Prompt (Anaconda3)
(base) D:\>python pyshark_3.py
Packets number:43
Layer ETH:
  Destination: fe:ff:20:00:01:00
  Address: fe:ff:20:00:01:00
  .... 1. .... = LG bit: Locally administered address (this is NOT the factory default)
  .... 0. .... = IG bit: Individual address (unicast)
  Source: 00:00:01:00:00:00
  Type: IPv4 (0x0800)
  Address: 00:00:01:00:00:00
  .... 0. .... = LG bit: Globally unique address (factory default)
  .... 0. .... = IG bit: Individual address (unicast)
Layer IP:
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  .... 00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 48
  Identification: 0x0f41 (3905)
  Flags: 0x4000, Don't fragment
  0... .. = Reserved bit: Not set
  .1... .. = Don't fragment: Set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
  Header checksum: 0x91eb [validation disabled]
  Header checksum status: Unverified
  Source: 145.254.160.237
  Destination: 65.208.228.223
Layer TCP:
  Source Port: 3372
  Destination Port: 80
  Stream index: 0
  TCP Segment Len: 0
```

Challenge: Salvarea numarului de frame-uri si a continutului tuturor frame-urilor intr-un fisier

Retele de calculatoare – Informatica anul 3 (2019-2020)

Observatii

1. Atentie (Modeler) – Proiectul creat se salveaza implicit in:

C:\Users\student(NUMÉ user)\op_model\NUMÉ_PROIÉCT

NUMÉ_PROIÉCT *contine proiectul modeler propriu-zis*

VARIANTA

se arhiveaza intreg folderul *Folder creat mai jos...el contine proiectul opnet propriu-zis*

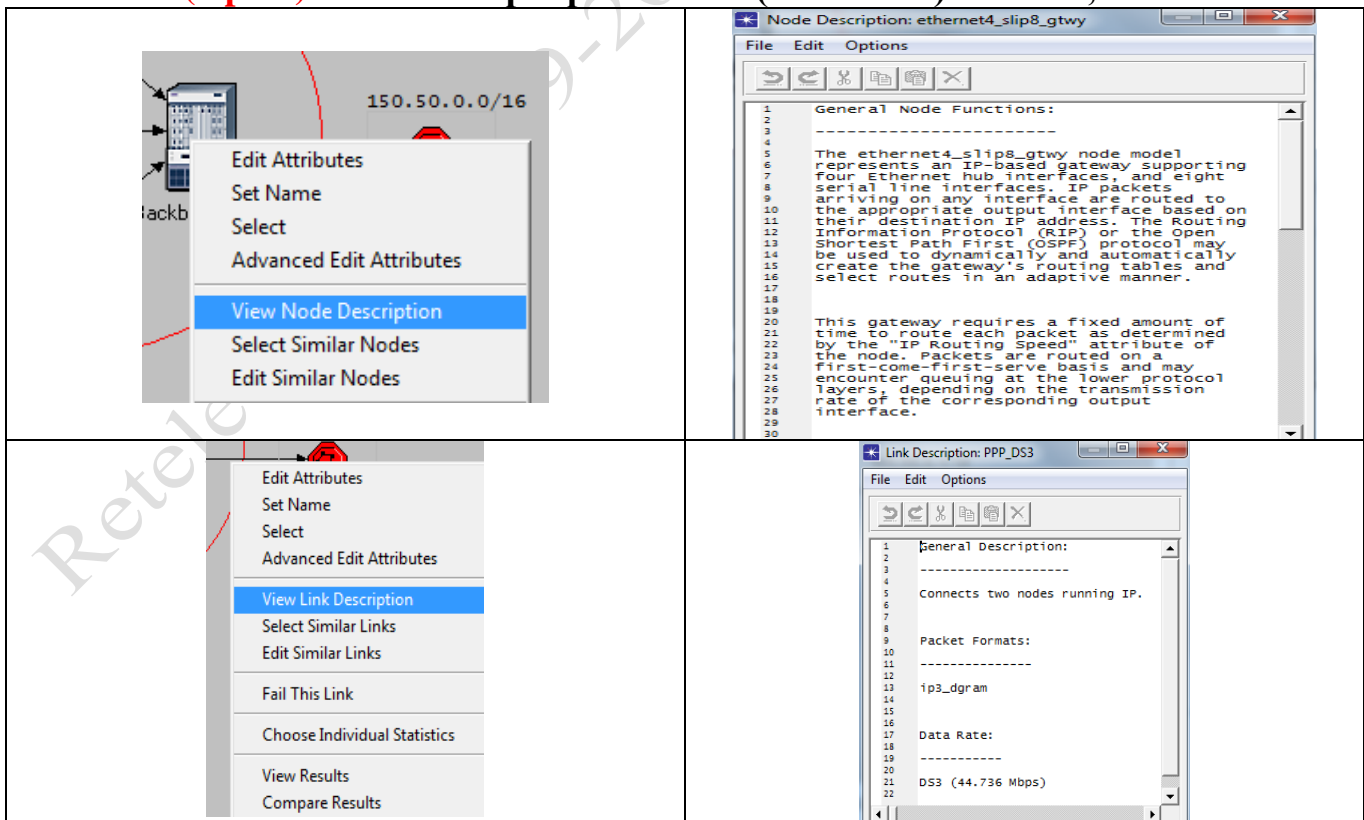
- In directorul\Studenti\Info3\Nume_Prenume se creează directorul \L2_Modeler_Nume_Prenume folosind:
 - **File** → **New** → **Folder**
- Se lansează în execuție IT Guru.
- Se selectează directorul în care vor fi plasate fişierele proiectului.
 - **File** → **Model Files** → **Add Model Directory**
 - Se selectează directorul în care se va lucra (în acest director vor fi salvate fişierele proiectului curent)
 - Se arhiveaza (eventual!) L2_Modeler_Nume_Prenume (**Atentie, gmail nu “prea vrea” .rar in .rar** <http://www.makeuseof.com/tag/4-ways-email-attachments-file-extension-blocked/>)

Atentie (Modeler Academic Edition) – se foloseste

readme_mod_work_dir.pdf (este prezent in arhiva Lab_02)

pentru a identifica folderul op_models in care se salveaza default *proiectul Modeler*.

2. Atentie (Opnet) : Click dreapta pe “obiect” (ex. Router)...”Judec, deci exist!”



Retele de calculatoare – Informatica anul 3 (2019-2020)

4. Tema (Partea practica: pag.5):

- Toate punctele din secțiunea 3 “partea practica” se vor relua de către cursanți, folosind etapele de lucru indicate.
- Arhiva cu numele **L4_num+prenume_info3.rar** va conține
 - a. **L4_num+prenume_Modeler (folder)** - conține proiectele Modeler/Omnet++ (**3.2, 3.3**) și **L4_num+prenume_Modeler.doc** (document .doc): rezultatele experimentale: comentarii însoțite de capturi corespunzătoare proiectelor Modeler/ Omnet++ (**3.2; 3.3**) pași intermediari importanți/topologia fizică, rezultate/capturi pentru View node description și View link description (obs.2 anterioară), exercitiile rezolvate, răspunsuri la întrebări, rezultate finale, observații finale)
 - b. **L4_num+prenume_Wireshark (folder – conform Tema Wireshark (3.1))**: Fișiere wireshark demonstrative, pe care s-a lucrat și care au fost analizate + .doc cu capturi (snipping tool) și comentarii pentru toate scenariile
 - c. **L4_num+prenume_Python (folder)** – cu subfolderele 3.4.3 și 3.4.3 (fiecare din acestea conține scripturile .py și document .doc (snipping tool) pentru aplicațiile Python, conform Tema Python, pag.24 (Tema Python 3.4.2) și a temelor indicate la aplicațiile de la punctul 3.4.3)

Arhiva cu numele **L4_num+prenume_info3.rar** se va trimite prin e-mail la adresa retelecdsd@gmail.com precizându-se la subject: **L4_num+prenume_info3**, până pe data de **30 octombrie 2019, ora 8.00 a.m.** (**Atentie, gmail nu “prea vrea” .rar în .rar** <http://www.makeuseof.com/tag/4-ways-email-attachments-file-extension-blocked/>)

Cursanții sunt încurajați să analizeze și să comenteze rezultatele obținute, studiind și materialele indicate în bibliografie și anexe.

Obs:

Punctaj maxim (Data trimerii temei)			
<= 30.10. 2019	04.11. 2019	08.11.2019	12.11.2019
100 pct	80 pct	60 pct	50 pct

Atentie la TEMA Wireshark:

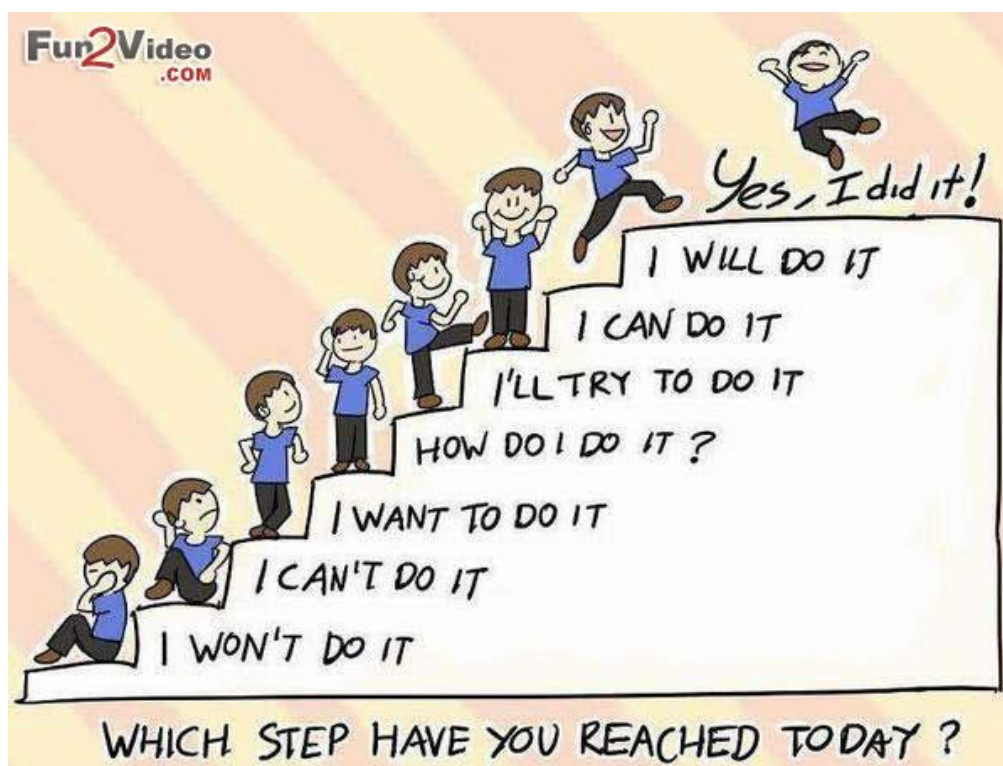
- Proiectarea scenariului (pe rand, fiecare !!!)
- Pornire captura (placă de rețea în promiscuous mode !!!! – vezi laboratorul 3 discuții.... și soluție oferită de wireshark!!!!)
- Scenariu propriu-zis
- Oprește captura
- Salvarea fișierului
- Analiza rezultatelor obținute: identificarea protocoalelor; filtrarea adecvată etc!!!!

Obs. **Studentii “pasionați” de programare C++**, dornici de afirmare pot opta să folosească pe lângă Modeler (sau ca variantă), [OMNeT++ Network Simulation Framework](http://www.omnetpp.org/) www.omnetpp.org/ , cu păstrarea scenariilor pentru aplicație, descrise în laborator. **Se acordă “bonusuri! ...SUBSTANTIALE!”**

Retele de calculatoare – Informatica anul 3 (2019-2020)

OBSERVATIE:

- **Cursantii sunt incurajati** sa foloseasca materialele prezentate intr-un mod constructiv, astfel incat sa evalueze caracteristicile legaturilor fizice si ale dispozitivelor de retea folosite (**click dreapta, view link/node description**), a modelelor de retea precum si elementele de baza privind simularea sistemelor discrete de evenimente analizate; rezultatele obtinute grafic vor fi analizate si interpretate in contextul cerintelor formulate .



Sursa: <http://www.funfun.in/wp-content/uploads/2013/06/steps-of-success-encouraging-quote.jpg>

How to send an e-mail

<http://lifesacker.com/5803366/how-to-send-an-email-with-an-attachment-for-beginners>

<https://support.google.com/mail/answer/6584?hl=en> "As a security measure to prevent potential viruses, Gmail doesn't allow you to send or receive **executable files** (such as files ending in .exe)."

<https://support.google.com/mail/answer/2480713?hl=en>

<http://fastupload.ro/free.php>

<http://www.computerica.ro/siteuri-transfer-fisiere-mari-upload/>

Bibliografie:

Lab_01, Lab_02, Lab_03, TL_01

<http://www.cdsd.ro/cursuri.html>

<http://www.wireshark.org/download.html>

http://www.wireshark.org/docs/wsug_html_chunked/

Retele de calculatoare – Informatica anul 3 (2019-2020)

<http://www.iana.org/assignments/port-numbers>

https://rpmapps.riverbed.com/ae/4dcgi/SIGNUP_NewUser

<https://supportkb.riverbed.com/support/index?page=content&id=S24443>

https://rpmapps.riverbed.com/ae/4dcgi/DOWNLOAD_HOME

https://rpmapps.riverbed.com/ae/4dcgi/REG_TransactionCode

- Install Riverbed Modeler 17.5 Windows 10, 8.1, 8 and 7 (<https://www.youtube.com/watch?v=TpenN2jYbHQ>)
- Install Riverbed Modeler (<https://www.youtube.com/watch?v=DQ3XhHYuFGA>)
- How to activate riverbed modeler 17.5 (<https://www.youtube.com/watch?v=h-lmeJMqiSA>)
- How to solve invalid activation of Opnet Modeler 17.5 (<https://www.youtube.com/watch?v=13ZBcXkW46s>)
- Riverbed Modeler 17.5 Tutorial - Switched Lan (<https://www.youtube.com/watch?v=XdebwQLrr0w>)
- 6-Virtual LAN (VLAN) configuration in OPNET Riverbed (<https://www.youtube.com/watch?v=Ajz7bVO5WJM>)
- Riverbed Modeler Configuracion VLAN (<https://www.youtube.com/watch?v=rP3jPMcyEFk>)
- Ethernet (lab 04)
- Riverbed Opnet 17.5 Tutorial - The Ethernet network (https://www.youtube.com/watch?v=fS_J6ApFJtc)
- 6-Virtual LAN (VLAN) configuration in OPNET Riverbed (<https://www.youtube.com/watch?v=Ajz7bVO5WJM>)
- Riverbed Modeler Tutorial 3 Configuracion VLAN (<https://www.youtube.com/watch?v=rP3jPMcyEFk>)

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

Anexa 1: Frame-uri Ethernet

1.1. Ethernet (Ethernet II)

```
+-----+-----+-----+-----+
|  Dst   |  Src   |  Type  |  Data...|
+-----+-----+-----+-----+
<-- 6 --> <-- 6 --> <-- 2 --> <-46-1500->
Type 0x80 0x00 = TCP/IP
Type 0x06 0x00 = XNS
Type 0x81 0x37 = Novell NetWare
```

1.2. 802.3

```
+-----+-----+-----+-----+
|  Dst   |  Src   | Length |  Data...|
+-----+-----+-----+-----+
<-- 6 --> <-- 6 --> <-- 2 --> <-46-1500->
```

1.3. 802.2 (802.3 cu header 802.2)

```
+-----+-----+-----+-----+-----+-----+
|  Dst   |  Src   | Length | DSAP   | SSAP   | Control |  Data...|
+-----+-----+-----+-----+-----+-----+
<- 1 -> <- 1 -> <- 1 -> <-43-1497->
```


Rețele de calculatoare – Informatica anul 3 (2019-2020)

1.4. SNAP (802.3 cu headere 802.2 si SNAP)

-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
Dst Src Length 0xAA 0xAA 0x03 Org Code Type Data...
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
<-- 3 --> <-- 2 --> <-38-1492->

Observatie: SubNetwork Access Protocol (SNAP) este un standard pentru transmisia datagramelor IP peste rețele IEEE 802. Datagramelor IP pot fi trimise pe rețele IEEE 802 încapsulate cu 802.2 LLC si SNAP (data link layers) si 802.3, 802.4 or 802.5 (physical network layers).

1.5. Ethernet II (DIX)

Denumire camp	P	AD	AS	Tip	D	CRC
Nr. Octeti	8	6	6	2	46÷1500	4

- a) **P** = preambul – camp folosit pentru sincronizare si incadrare
 - i) Primii 7 octeti au forma 10101010
 - ii) Al 8-lea are forma 10101011
- b) **AD** = Adresa de Destinatie (MAC)
- c) **AS** = Adresa Sursa (MAC)
- d) **Tip** – contine 2 octeti ce identifica tipul protocolului de nivel superior care a emis sau vrea sa receptioneze frame-ul. Este asignat de Xerox si neinterpretat de 802.3. Campul permite protocoalelor de nivel inalt (client) sa imparta reteaua fara „a intra” unul in mesajele celuilalt, asigurand asa-numita multiplexare
- e) **D** = Date – contine mesajul de date ce se intentioneaza a fi transmis la destinatie prin intermediul frame-ului (reamintim ca frame-ul este un produs al încapsularii de pe nivelul 2 OSI)
- f) **CRC** = Cyclic Redundance Code – contine restul sumei de verificare ciclica a redundantei calculat polinomial prin CRC-32. Secvența CRC (FCS – Frame Check Sequence) este dată de restul împărțirii acestui polinom la un polinom standard.

Standardul IEEE 802 foloseste următorul polinom:

$$\text{CRC32} = X^{32} + X^{26} + X^{25} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1.$$

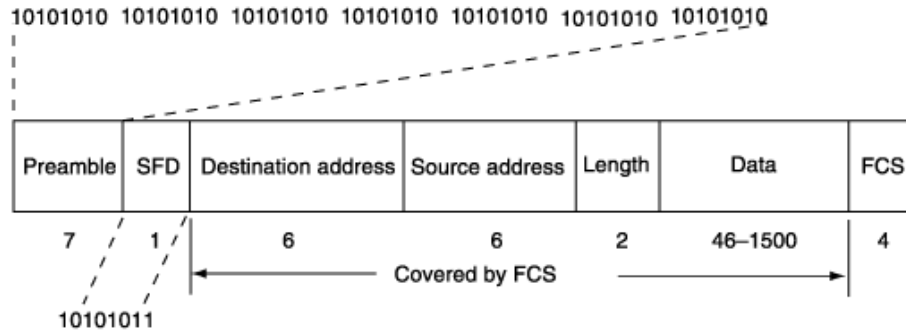
Notam: SLE = Statia de Lucru Emitatoare; SLD = Statia de Lucru Destinatie

SLD primește frame-ul, face propriul calcul CRC-32 si compara valoarea calculata cu cea aflata in campul CRC din pachet, concluzionand daca frame-ul a sosit intact. In caz de alterare informatia este distrusa bit cu bit cerandu-se retransmiterea.

1.6. IEEE 802.3

Denumire camp	P	SFD	AD	AS	L	D	PAD	CRC
Nr. Octeti	7	1	6	6	2	0÷1500	!	4

Retele de calculatoare – Informatica anul 3 (2019-2020)



SFD: Start frame delimiter

FCS: Frame check sequence (32-bit CRC value)

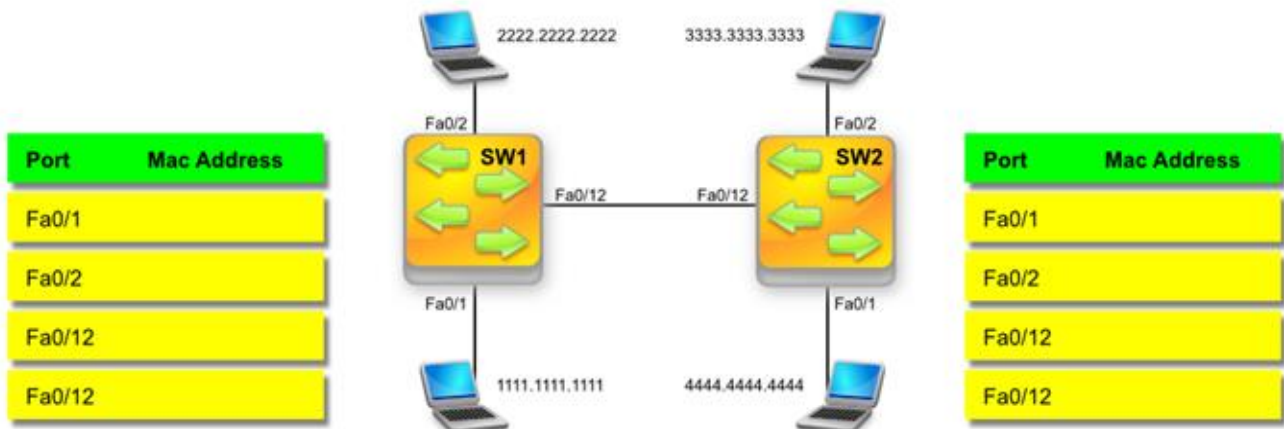
- P** = preambul – 7 octeti de forma 10101010
- SFD** = Start Frame Delimiter – 1 octet de forma 10101011
- AD** = Adresa de Destinatie (MAC)
- AS** = Adresa Sursa (MAC)
- L** = lungimea (pe 2 octeti) – exprima lungimea campului **D**
- D** = date – Variaza intre 0 si 1500 octeti cu observatia ca daca are lungimea mai mica de 46 de octeti atunci **PAD**-ul este utilizat pentru a completa lungimea frame-ului pana la o dimensiune minima acceptabila.
- PAD** = camp tampon de lungime variabila mai mare sau egala cu 0
- CRC** – 4 octeti

Obs Frame-ul de la Ethernet are dimensiunea intre 64 si 1518 octeti (fara **P**) cu o dimensiune minima a campului **D** de 46 de octeti. La IEEE 802.3 dimensiunea frame-ului (fara **P** si **SPD**) este aceeaasi. In plus la 802.3 este permis ca aplicatia sau un nivel superior de protocol sa trimita o zona de date **D** mai mica de 46 octeti, deoarece frame-ul este completat automat in **PAD** pe subnivelul MAC. La Ethernet frame-urile care sunt prea mici sunt considerate erori

Terminologie: Literatura indica folosirea termenului de Ethernet pentru standardul IEEE 802.3.

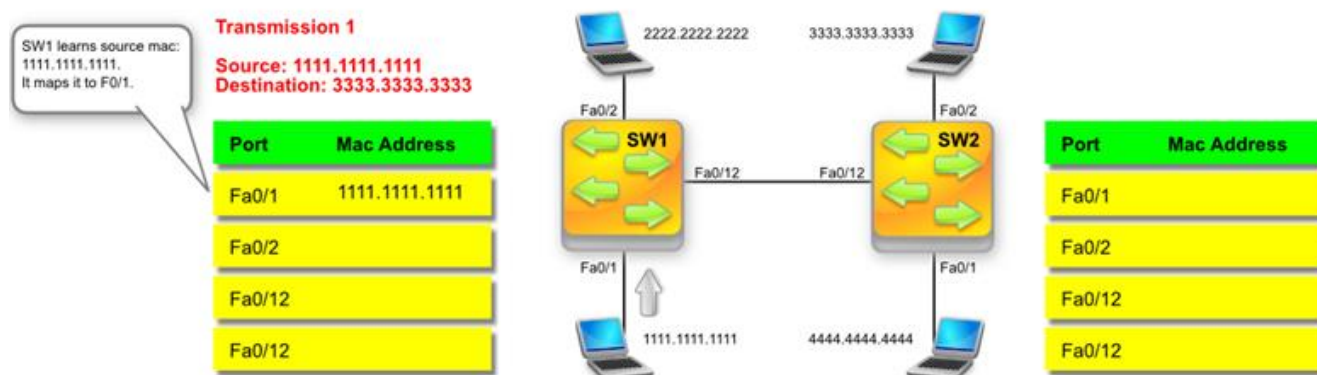
Anexa 2: **Exercitiu - Completare tabele CAM (Bridge/ Switch Tables)**

Step1:



Retele de calculatoare – Informatica anul 3 (2019-2020)

Step 2:

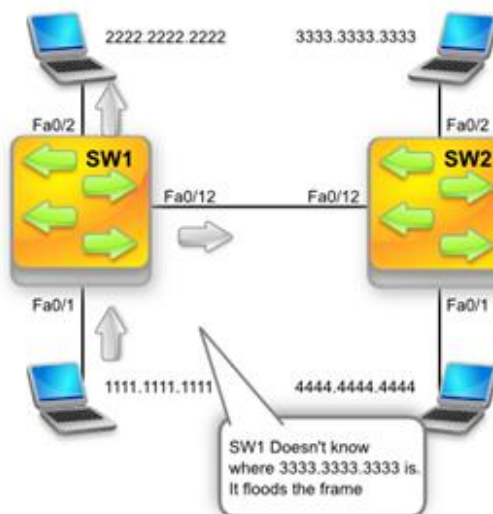


Step 3:

Transmission 1

Source: 1111.1111.1111
Destination: 3333.3333.3333

Port	Mac Address
Fa0/1	1111.1111.1111
Fa0/2	
Fa0/12	
Fa0/12	



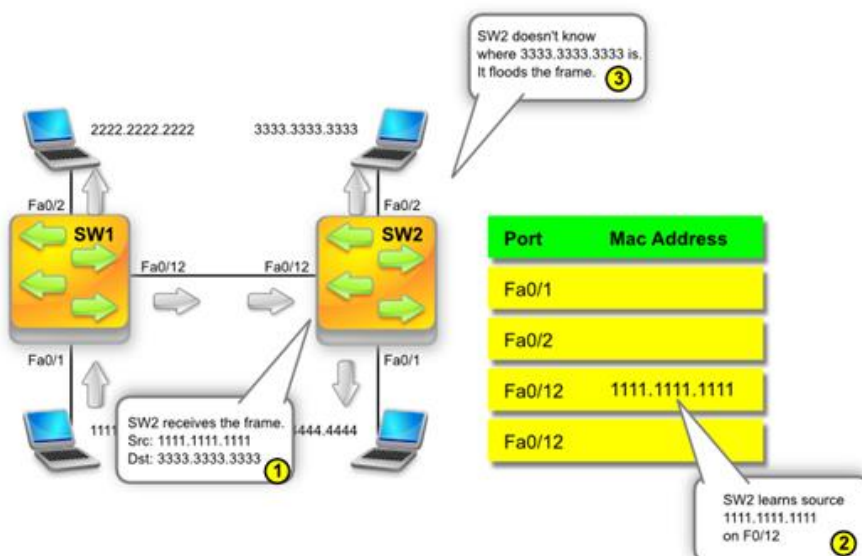
Port	Mac Address
Fa0/1	
Fa0/2	
Fa0/12	
Fa0/12	

Step 4:

Transmission 1

Source: 1111.1111.1111
Destination: 3333.3333.3333

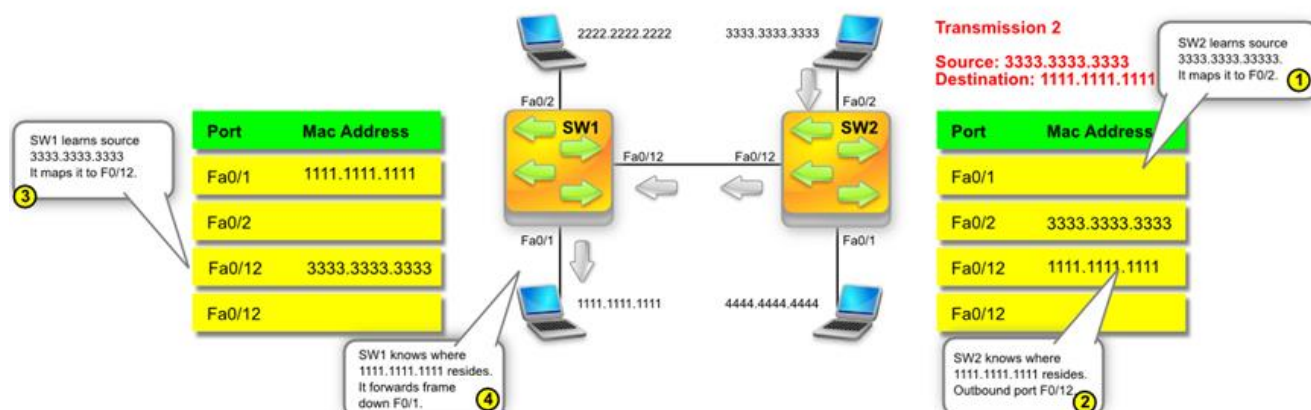
Port	Mac Address
Fa0/1	1111.1111.1111
Fa0/2	
Fa0/12	
Fa0/12	



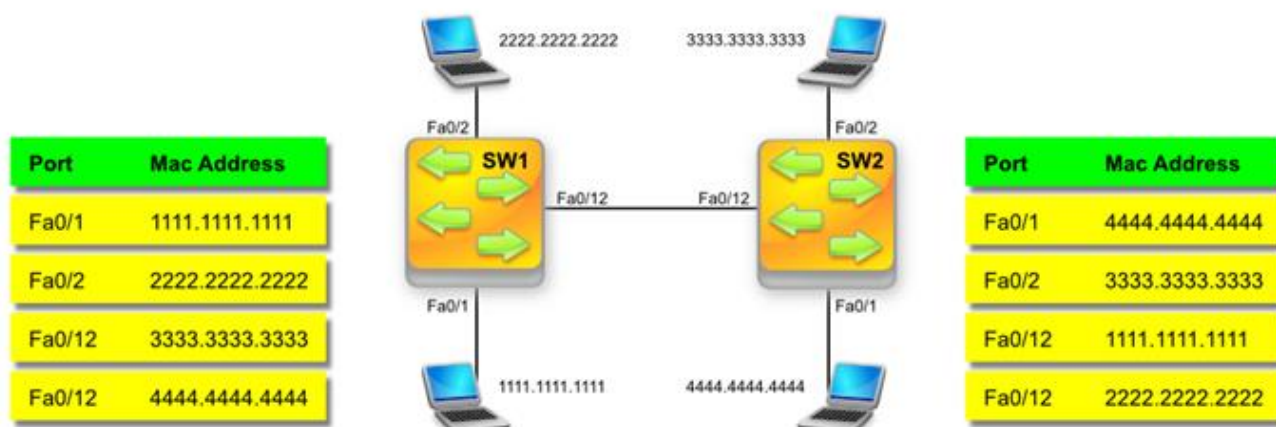
Port	Mac Address
Fa0/1	
Fa0/2	
Fa0/12	1111.1111.1111
Fa0/12	

Retele de calculatoare – Informatica anul 3 (2019-2020)

Step 5:

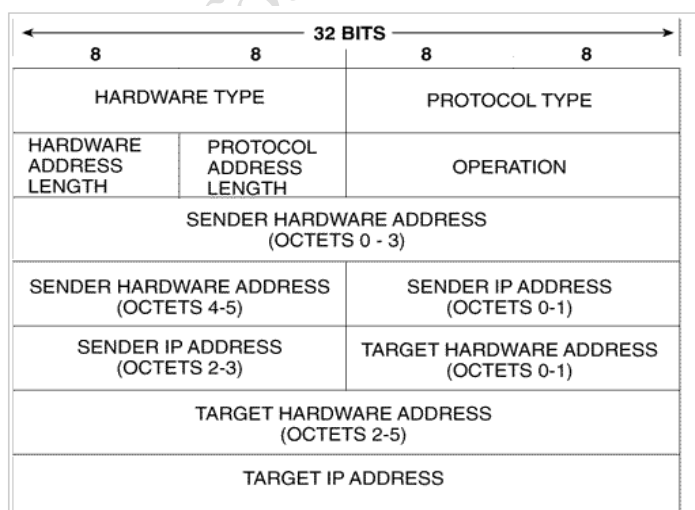


..... End:



Anexa 4

ARP Frame Format and types



Retele de calculatoare – Informatica anul 3 (2019-2020)

The above fig shows the ARP format used , below is the explanation of each field:

Hardware type

Each data link layer protocol is assigned a number used in this field. For Ethernet it is 1.

Protocol type

Each protocol is assigned a number used in this field. For example, IPv4 is 0x0800.

Hardware length

Length in bytes of a hardware address. Ethernet addresses are 6 bytes long.

Protocol length

Length in bytes of a logical address. IPv4 addresses are 4 bytes long.

Operation

Specifies the operation the sender is performing: 1 for request, and 2 for reply.

There are actually four types of ARP messages that may be sent by the ARP protocol. These are identified by four values in the "operation" field of an ARP message. The types of message are:

- 1.ARP request
- 2.ARP reply
- 3.RARP request
- 4.RARP reply

Sender hardware address

Hardware address of the sender.

Sender protocol address

Protocol address of the sender.

Target hardware address

Hardware address of the intended receiver. This field is zero on request.

Target protocol address

Protocol address of the intended receiver.

ARP Function explained

ARP is used in four cases when two hosts are communicating:

- 1.When two hosts are on the same network and one desires to send a packet to the other
- 2.When two hosts are on the different networks and must use a gateway or router to reach the other host
- 3.When a router needs to forward a packet for one host through another router
- 4.When a router needs to forward a packet from one host to the destination host on the same network

When an ARP response arrives, the receiver inserts a binding into an ARP cache so that it can be used for further packets. The oldest entry is removed if the table is either full or after an entry has not been updated recently. When an ARP request arrives, the receiver checks if it has the senders protocol address in the cache; if so, then the receiver updates the cache entry with the sender's binding. After a host replies to an ARP request, it adds the sender's binding to the cache - if a message travels from one host to another,

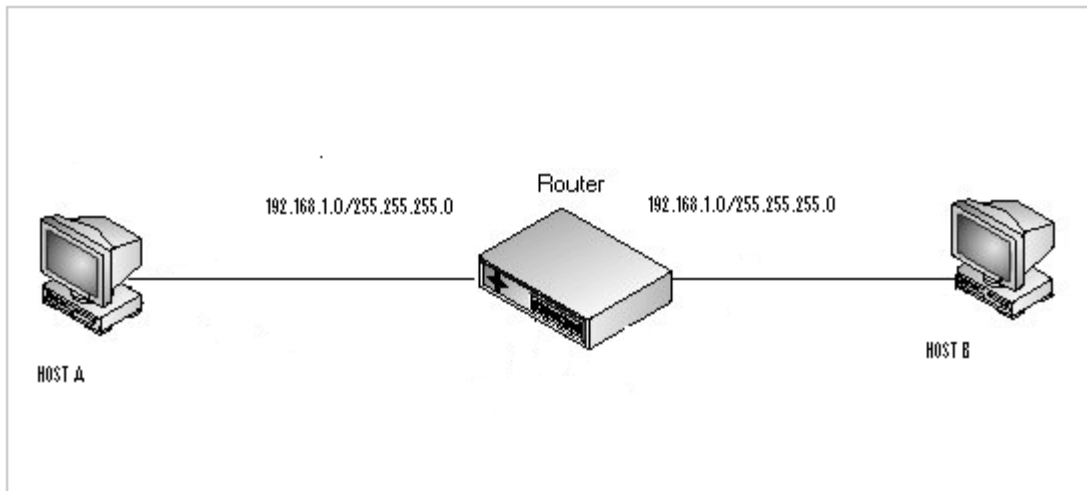
Retele de calculatoare – Informatica anul 3 (2019-2020)

then a reply will often travel back.

To understand this further let's see how an ARP actually works:

ARP works by broadcasting the packet to all hosts attached to an Ethernet network. The packet contains the IP address the sender is interested in communicating with. The target machine, recognizing that the IP address in the packet matches its own, returns an answer. Hosts actually keep a cache of ARP responses

Let's take an example here to study this concept by ARP across subnet:

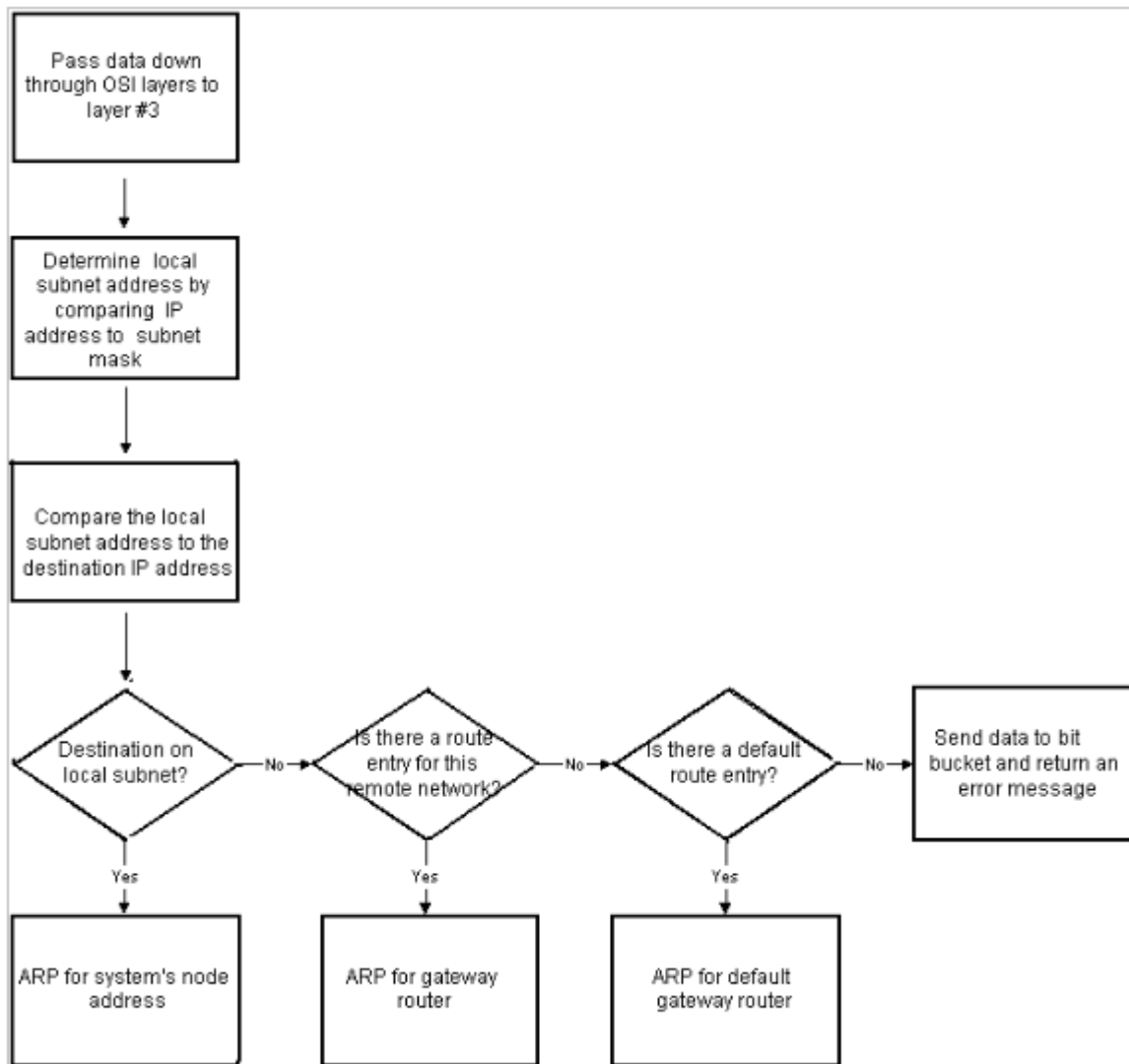


From the fig above let's say:

- computer A needs to send some data to computer B
- Since host B is not on the same subnet, before sending computer A transmits an ARP request in order to discover the MAC address of port A on the local router. This is done after the A checks its ARP cache and it does not find an entry for the MAC address of port A.
- Once host A knows the MAC address, it transmits an Ethernet frame to the router.
- This router C will send an ARP request out of port B in order to discover the MAC address of computer B.
- Once Computer B replies to this ARP request, the router will strip off the Ethernet frame from the data and create a new one.
- The router replaces the source MAC address (originally host A address) with the MAC address of port B. It will also replace the destination MAC address (originally port A) with the MAC address of host B. The fig 1 shows the Message format used.

The following fig shows the basic strategy and principle used by ARP:

Retele de calculatoare – Informatica anul 3 (2019-2020)



ARP Cache concept

The ARP cache contains a table containing matched sets of MAC and IP addresses. Each device on the network manages its own ARP cache table. There are two ways in which ARP cache is populated:

- **Static ARP Cache Entries:** In this type address resolutions are manually added to the cache table for a device and are kept in the cache on a permanent basis.
- **Dynamic ARP Cache Entries:** These are hardware and IP address pairs that are added to the cache by the software itself as a result of successfully completed past ARP resolutions. They are kept in the cache only for a period of time and are then flushed. After a particular entry times out, it is removed from the cache. The next time that address mapping is needed a fresh resolution is performed to update the cache.

Note: A device's ARP cache can contain both static and dynamic entries.

Retele de calculatoare – Informatica anul 3 (2019-2020)

Reverse ARP and Proxy ARP defined

Reverse Address Resolution Protocol (RARP) is a complement of the Address Resolution Protocol. It is a network layer protocol used to obtain an IP address for a given MAC address. The primary limitation of RARP is that each MAC address must be configured manually on a centralised server, and that the protocol only conveys an IP address. Its useful for diskless systems.

Proxy ARP a protocol that is used to hide a machine with a public IP on a private network behind a router, and still have the machine appear to be on the public network "in front of" the router. For this example, let's assume that host A is on a network segment connected to Router A's interface A, and host B is on a network segment connected to Router A's interface B. Host A wants to send data directly to host B, but doesn't have host B's MAC address. An ARP Request sent to host B from host A will stop at the router as it is a broadcast - but with Proxy ARP, the router A will actually answer the ARP Request with the MAC address of the router interface that received the ARP Request.

In this case, Router A will respond to the ARP Request with the MAC address of it's own interface A. This is transparent to the host A - when host A sends data to host B, the destination IP address will be that of host B, but the destination MAC address will be that of RouterA's A interface.

Though ARP is a simple resolution protocol its features and use with regards to network is immense.

Anexa 5:

Capture Filter

Capture only the Ethernet-based traffic to and from Ethernet MAC address 08:00:08:15:ca:fe:

```
• ether host 08:00:08:15:ca:fe
```

Ethernet [Multicast](#) traffic only:

```
• ether multicast
```

Ethernet [Broadcast](#) traffic only:

```
• ether broadcast
```

Ethernet traffic to/from a range of addresses:

```
• (ether[0:4]>=0x00804400 and ether[0:4]<=0x008044ff) or  
(ether[6:4]>=0x00804400 and ether[6:4]<=0x008044ff)
```

Display Filter Reference: Ethernet

Field name	Type	Description	Versions
eth.addr	6-byte Hardware (MAC) Address	Address	0.99.0 to 1.0.3
eth.dst	6-byte Hardware (MAC) Address	Destination	0.99.0 to 1.0.3
eth.ig	Boolean	IG bit	0.99.4 to 1.0.3
eth.len	Unsigned 16-bit integer	Length	0.99.0 to 1.0.3

Rețele de calculatoare – Informatica anul 3 (2019-2020)

eth.lg	Boolean	LG bit	0.99.4 to 1.0.3
eth.local_admin	Boolean	Locally Administrated Address	0.99.0 to 0.99.4
eth.multicast	Boolean	Multicast	0.99.0 to 0.99.4
eth.src	6-byte Hardware (MAC) Address	Source	0.99.0 to 1.0.3
eth.trailer	Byte array	Trailer	0.99.0 to 1.0.3
eth.type	Unsigned 16-bit integer	Type	0.99.0 to 1.0.3

Anexa 5: Display Filter Reference: Address Resolution Protocol

Field name	Type	Description	Versions
arp.dst.atm_num_e164	String	Target ATM number (E.164)	0.99.0 to 1.0.3
arp.dst.atm_num_nsap	Byte array	Target ATM number (NSAP)	0.99.0 to 1.0.3
arp.dst.atm_subaddr	Byte array	Target ATM subaddress	0.99.0 to 1.0.3
arp.dst.hlen	Unsigned 8-bit integer	Target ATM number length	0.99.0 to 1.0.3
arp.dst.htype	Boolean	Target ATM number type	0.99.0 to 1.0.3
arp.dst.hw	Byte array	Target hardware address	0.99.0 to 1.0.3
arp.dst.hw_mac	6-byte Hardware (MAC) Address	Target MAC address	0.99.0 to 1.0.3
arp.dst.pln	Unsigned 8-bit integer	Target protocol size	0.99.0 to 1.0.3
arp.dst.proto	Byte array	Target protocol address	0.99.0 to 1.0.3
arp.dst.proto_ipv4	IPv4 address	Target IP address	0.99.0 to 1.0.3
arp.dst.slen	Unsigned 8-bit integer	Target ATM subaddress length	0.99.0 to 1.0.3
arp.dst.stype	Boolean	Target ATM subaddress type	0.99.0 to 1.0.3
arp.duplicate-address-detected	None	Duplicate IP address detected	0.99.8 to 1.0.3
arp.duplicate-address-frame	Frame number	Frame showing earlier use of IP address	0.99.8 to 1.0.3
arp.hw.size	Unsigned 8-bit integer	Hardware size	0.99.0 to

Rețele de calculatoare – Informatica anul 3 (2019-2020)

			1.0.3
arp.hw.type	Unsigned 16-bit integer	Hardware type	0.99.0 to 1.0.3
arp.opcode	Unsigned 16-bit integer	Opcode	0.99.0 to 1.0.3
arp.packet-storm-detected	None	Packet storm detected	0.99.5 to 1.0.3
arp.proto.size	Unsigned 8-bit integer	Protocol size	0.99.0 to 1.0.3
arp.proto.type	Unsigned 16-bit integer	Protocol type	0.99.0 to 1.0.3
arp.seconds-since-duplicate-address-frame	Unsigned 32-bit integer	Seconds since earlier frame seen	0.99.8 to 1.0.3
arp.src.atm_num_e164	String	Sender ATM number (E.164)	0.99.0 to 1.0.3
arp.src.atm_num_nsap	Byte array	Sender ATM number (NSAP)	0.99.0 to 1.0.3
arp.src.atm_subaddr	Byte array	Sender ATM subaddress	0.99.0 to 1.0.3
arp.src.hlen	Unsigned 8-bit integer	Sender ATM number length	0.99.0 to 1.0.3
arp.src.htype	Boolean	Sender ATM number type	0.99.0 to 1.0.3
arp.src.hw	Byte array	Sender hardware address	0.99.0 to 1.0.3
arp.src.hw_mac	6-byte Hardware (MAC) Address	Sender MAC address	0.99.0 to 1.0.3
arp.src.pln	Unsigned 8-bit integer	Sender protocol size	0.99.0 to 1.0.3
arp.src.proto	Byte array	Sender protocol address	0.99.0 to 1.0.3
arp.src.proto_ipv4	IPv4 address	Sender IP address	0.99.0 to 1.0.3
arp.src.slen	Unsigned 8-bit integer	Sender ATM subaddress length	0.99.0 to 1.0.3
arp.src.stype	Boolean	Sender ATM subaddress type	0.99.0 to 1.0.3

Anexa 6:

Broadcast Switch *Troughput* (viteza de transfer reală, traficul real de broadcast) a fost redus(a) considerabil prin configurarea VLAN-urilor.....mai multe domenii de broadcast, corespunzătoare numărului de VLAN-uri implementate.

Retele de calculatoare – Informatica anul 3 (2019-2020)

<http://www.commsdesign.com/showArticle.jhtml?articleID=26806942>

http://www.bandwidth.com/wiki/article/Benefits_of_VLANs

Anexa 7: Capturi Wireshark

The image shows a Wireshark packet capture window titled 'vlan.cap - Wireshark'. The main pane displays a list of network packets. Packet 53 is selected, showing details for an 802.1Q Virtual LAN frame. The frame is encapsulated in an Ethernet II frame with source MAC 00:40:05:40:ef:24 and destination MAC 00:60:08:9f:b1:f3. The 802.1Q header shows Priority 0, CFI 0, and ID 32. The payload is an IP packet from 131.151.32.129 to 131.151.32.21, which is a TCP segment for port 6000 (x11) with sequence number 1 and acknowledgment number 129. The packet list pane shows the following details for packet 53:

No.	Time	Source	Destination	Protocol	Info
42	0.050121	131.151.32.21	131.151.32.129	X11	Event: ConfigureNotify, ConfigureNotify, Expose, Co
43	0.055805	131.151.5.55	131.151.5.255	BROWSER	Host Announcement ANANNI, workstation, Server, NT W
44	0.056091	CisTechn_62:73:a1	NETBIOS-	BROWSER	Host Announcement ANANNI, workstation, Server, NT W
45	0.056766	131.151.32.129	131.151.32.21	X11	Requests: ConfigureWindow, ConfigureWindow, Configu
46	0.058641	131.151.32.21	131.151.32.129	X11	Event: ConfigureNotify, Expose, Expose, Expose, Exp
47	0.059984	131.151.32.129	131.151.32.21	X11	Requests: ConfigureWindow, ConfigureWindow, Configu
48	0.061021	131.151.32.129	131.151.32.21	X11	Requests: PolyFillRectangle, PolySegment, PolySegme
49	0.061696	131.151.32.129	131.151.32.21	X11	Requests: PolyFillRectangle, PolySegment, PolySegme
50	0.062000	00050500.0020186273a1	00000000.ffffffffffff	NBIPX	Find name ERL<Id>
51	0.068679	00056800.0800078412de	00000000.ffffffffffff	IPX RIP	Request
52	0.106124	131.151.32.21	131.151.32.129	X11	Event: MapNotify, MapNotify, MapNotify, MapNotify
53	0.122476	131.151.32.129	131.151.32.21	TCP	d-cinema-rp > x11 [ACK] Seq=1 Ack=129 Win=31856 Le
54	0.132724	00056800.0800078412de	00000000.ffffffffffff	IPX RIP	Request
55	0.164528	00056800.0800078412de	00000000.ffffffffffff	IPX RIP	Request

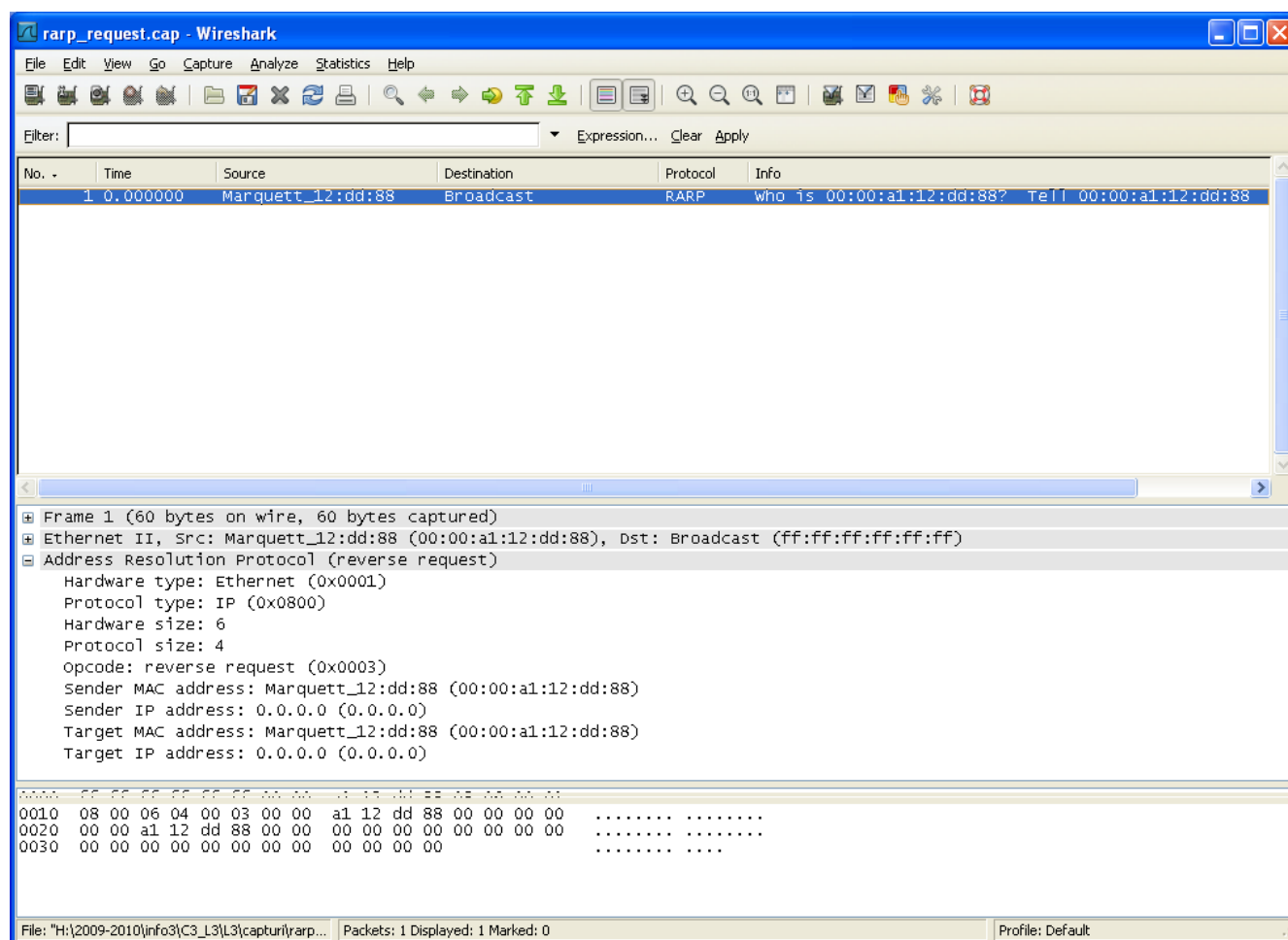
The packet details pane for packet 53 shows the following structure:

- Frame 53 (70 bytes on wire, 70 bytes captured)
- Ethernet II, Src: AnCommu_40:ef:24 (00:40:05:40:ef:24), Dst: 3com_9f:b1:f3 (00:60:08:9f:b1:f3)
- 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 32
 - 000. = Priority: 0
 - ...0 = CFI: 0
 - 0000 0010 0000 = ID: 32
 - Type: IP (0x0800)
- Internet Protocol, Src: 131.151.32.129 (131.151.32.129), Dst: 131.151.32.21 (131.151.32.21)
- Transmission Control Protocol, Src Port: d-cinema-rp (1173), Dst Port: x11 (6000), Seq: 1, Ack: 129, Len: 0

The packet bytes pane shows the raw data of the selected packet, with the 802.1Q header highlighted in red:

```
0000 00 60 08 9f b1 f3 00 40 05 40 ef 24 81 00 00 20 .....@.@.$...
0010 08 00 45 00 00 34 3b 64 40 00 40 06 b7 9b 83 97 ..E..4;d @.@....
0020 20 81 83 97 20 15 04 95 17 70 51 d4 ee 9c 51 a5 .... .pQ...Q.
0030 5b 36 80 10 7c 70 12 c7 00 00 01 01 08 0a 00 04 [6..|p.. .....
0040 f0 d4 01 99 a3 fd .....
```

Retele de calculatoare – Informatica anul 3 (2019-2020)



Retele de calculatoare – Informatica anul 3 (2019-2020)

arp-storm.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.159? Tell 24.166.172.1
2	0.098594	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.172.141? Tell 24.166.172.1
3	0.110617	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.161? Tell 24.166.172.1
4	0.211791	Cisco_af:f4:54	Broadcast	ARP	who has 65.28.78.76? Tell 65.28.78.1
5	0.216744	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.163? Tell 24.166.172.1
6	0.307909	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.175.123? Tell 24.166.172.1
7	0.330433	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.165? Tell 24.166.172.1
8	0.408556	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.175.82? Tell 24.166.172.1
9	0.455104	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.220.131? Tell 69.76.216.1
10	0.486666	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.168? Tell 24.166.172.1
11	0.504694	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.221.27? Tell 69.76.216.1
12	0.510684	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.174.184? Tell 24.166.172.1
13	0.540733	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.169? Tell 24.166.172.1
14	0.587308	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.174.181? Tell 24.166.172.1
15	0.662937	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.223.216? Tell 69.76.216.1
16	0.690450	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.173.172? Tell 24.166.172.1
17	0.692934	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.223.217? Tell 69.76.216.1
18	0.771600	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.217.186? Tell 69.76.216.1
19	0.792105	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.174.221? Tell 24.166.172.1
20	0.801633	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.218.94? Tell 69.76.216.1
21	0.806611	Cisco_af:f4:54	Broadcast	ARP	who has 24.166.174.207? Tell 24.166.172.1
22	0.856709	Cisco_af:f4:54	Broadcast	ARP	who has 69.76.223.222? Tell 69.76.216.1

Frame 20 (60 bytes on wire, 60 bytes captured)

- Ethernet II, Src: Cisco_af:f4:54 (00:07:0d:af:f4:54), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Address: Broadcast (ff:ff:ff:ff:ff:ff)
 -1.... = IG bit: Group address (multicast/broadcast)
 - ...1.... = LG bit: Locally administered address (this is NOT the factory default)
 - Source: Cisco_af:f4:54 (00:07:0d:af:f4:54)
 - Address: Cisco_af:f4:54 (00:07:0d:af:f4:54)
 -0.... = IG bit: Individual address (unicast)
 - ...0.... = LG bit: Globally unique address (factory default)
 - Type: ARP (0x0806)
 - Trailer: 040104000000000201000302000005010301
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (0x0001)
 - Sender MAC address: Cisco_af:f4:54 (00:07:0d:af:f4:54)
 - Sender IP address: 69.76.216.1 (69.76.216.1)
 - Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)

0000 08 00 06 04 00 01 00 07 0d af f4 54 45 4c d8 01TEL..

0020 00 00 00 00 00 00 45 4c da 5e 04 01 04 00 00 00EL.A.....

0030 00 02 01 00 03 02 00 00 05 01 03 01

Frame (frame), 60 bytes Packets: 622 Displayed: 622 Marked: 0 Profile: Default

Anexa 8

8 Ethernet Model Description

Background / Operational Description

Ethernet is a bus-based local area network (LAN) technology commonly used in the technical and business communities.

Detailed information about the Ethernet protocol is in the *IEEE 802.3 Standard*.

Model Scope and Limitations

The Ethernet MAC model provided with OPNET implements the carrier sensing, collision detection, and retransmission mechanisms specified in the *IEEE 802.3*, *IEEE 8-2.3u*, and *IEEE 802.3z Standard*. Explicit modeling is performed for all features other than serialization of bit transfers to and from the physical layer. The following list itemizes the features provided in this model:

- FIFO processing of Transmission Requests
- Propagation Delay based on Distance Between Individual Stations
- Carrier Sensing from Physical Layer
- Collision Detection from Physical Layer
- Truncated Binary Exponential Backoff
- Transmission Attempt Limit of 16
- Interframe Gap Timing for Deference
- Jam Sequence Transmission after Collisions
- 802.3 Minimum and Maximum Frame Sizes
- Frame Bursting (1000BaseX Ethernet operating in half-duplex only)
- Full- and half-duplex transmissions

You can configure port-based VLANs on all generic bridge and switch models, and on any vendor-specific models that support this technology. Ethernet link models allow you to simulate point-to-point trunk links; a single trunk link can carry traffic for multiple VLANs as specified by IEEE 802.1q.

To configure a VLAN, set the VLAN Scheme attribute to "Port-based VLAN" on the bridge or switch supporting the VLAN. You can assign VLAN identifiers to specific port numbers in the VLAN Port Configuration Table. (To find a link's port numbers, use Link Interfaces on the Edit Attributes (Advanced) dialog box.) Note that you can assign only one VLAN identifier to a specific port. However, multiple ports can belong to the same VLAN.

The Ethernet models also support Fast EtherChannel technology. This allows multiple Ethernet point-to-point links to be bundled into one logical full-duplex channel of up to 800 Mbps (for Fast Ethernet) or 8000 Mbps (for Gigabit Ethernet). You can use a Fast EtherChannel or Gigabit EtherChannel link in place of any regular Ethernet link (10BaseT, 100BaseT, or 1000BaseX). EtherChannel links support flow-based balancing of traffic, and are useful for upgrading bottleneck links in Ethernet LAN networks.

Note—You can only use EtherChannel links when Ethernet is running in full-duplex mode.

The Ethernet models can be deployed either in a bus (10Base2) or a hub (10BaseT, 100BaseT or fast ethernet, and 1000BaseX or gigabit ethernet) configuration. The following list itemizes the main differences between these two configurations:

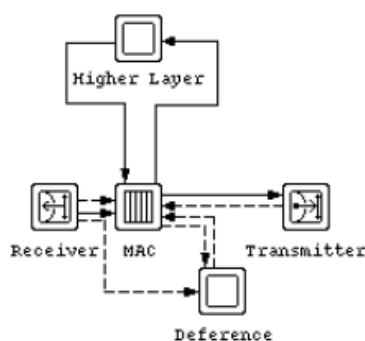
- Connections from the MAC processes to the hub are via duplex point-to-point links, as opposed to a bus medium.
- Collision Detection in the hub configuration is handled by the hub, rather than individual MAC processes.
- Deference mechanism is handled by the hub, rather than a separate deference process.
- Ethernet hubs cannot be directly connected to one another. Instead, a bridge must be used to link two or more hubs together.

Retele Info3 201

Model Architecture

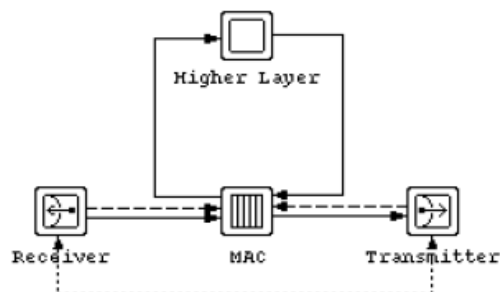
The following diagram illustrates the relevant portions of a typical 10Base2 node model using the process models defined for the Ethernet model suite.

Figure 8-1 Node Model Structure Surrounding Ethernet



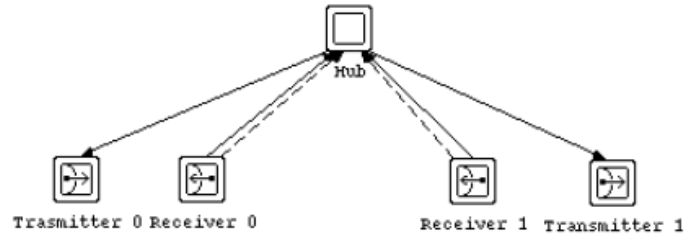
The following diagram illustrates the relevant portions of a typical node model.

Figure 8-2 Node Model Structure Surrounding Ethernet Hub



The following diagram illustrates the relevant portions of a typical hub node model.

Figure 8-3 Node Model Structure Surrounding Ethernet Hub



Process Models

The following table enumerates the process models used by the Ethernet model suite. Additional details on these process models are provided in subsequent sections.

Table 8-1 Ethernet Process Models (Part 1 of 2)

name	location	summary
eth_defer_v2	Deference Module	Performs carrier sensing and computes the deference variable.
eth_gen_v2	Higher Layer Module	Provides a simple example of higher-layer interfacing with the Ethernet MAC protocol. The process generates packets to the other stations on the LAN.
ethernet_hub_v2	Hub Module	Implements the Ethernet hub functions such as collision detection.

Retele

Table 8-1 Ethernet Process Models (Part 2 of 2)

name	location	summary
ethernet_mac_v2	MAC Module	Performs media access control for the Ethernet interface in a 10BaseT, 100BaseT, or 1000BaseX configuration. The module can operate in either half or full duplex mode, and if operating at gigabit ethernet speed, can have frame bursting either enabled or disabled.
eth_mac_v2	MAC Module	Performs media access control for the Ethernet interface in a 10Base2 configuration.
eth_sink_v2	Higher Layer Module	Provides a simple example of higher-layer interfacing with the Ethernet MAC protocol. The process simply discards arriving packets.
End of Table 8-1		

Model Interfaces

Packet Formats

The following table enumerates the packet formats used in the Ethernet model suite.

Table 8-2 Ethernet Packet Formats

name	description
ethernet	Represents the Ethernet packet format. This packet format allows for the encapsulation of higher level protocol data and carries control fields such as destination and source address, and protocol type. In addition, the frame check sequence (FCS) and preamble components of transmissions are modeled as fields of the packet.
End of Table 8-2	

ICI Formats

The following table enumerates the interface control information (ICI) formats used in the Ethernet model suite.

Table 8-3 Ethernet ICI Formats

name	description
eth_mac_ind	Used in conjunction with packet transfers from eth_mac_v2 (or ethernet_mac_v2) to the higher layers. The attributes carried in this ICI specify the destination address (normally that of this station), the address of the sending station, and the type of higher level protocol used, respectively.
mac_req	Used in conjunction with packet transfers to eth_mac_v2 (or ethernet_mac_v2). It carries the destination address of the node to which the packet is being transmitted. The higher level protocol type may also be passed in this ICI, but it is not currently used by MAC processes.
End of Table 8-3	

Ethernet Addressing

Each MAC entity has a physical address, specified by the "station_address" attribute of the eth_mac_v2 (and ethernet_mac_v2) process models. When using these models in conjunction with the Spanning Tree Bridge models, the Ethernet addresses must be unique for the entire OPNET network to support correct bridge address learning. The addresses can be automatically assigned to meet this requirement.

Retele Info3