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

Note de Laborator Retele de calculatoare	**Specializare: Informatica anul 3** **Contact:** **retelecdsd@gmail.com** **http://www.cdsd.ro**

Comunicatii de Date si Sisteme Distribuite
http://www.cdsd.ro

Laborator 3

1. Obiective:

- Intelegerea rolului protocoalelor in retelele de calculatoare
- Identificarea nivelurilor (layer-elor) modelelor OSI si TCP/IP.
- Analiza conectivitatii si traficului dintr-o retea folosind Wireshark
- Aplicatii de retea in Pyton

2. Consideratii teoretice

(Partea practica: pag.3; Tema: pag.26)

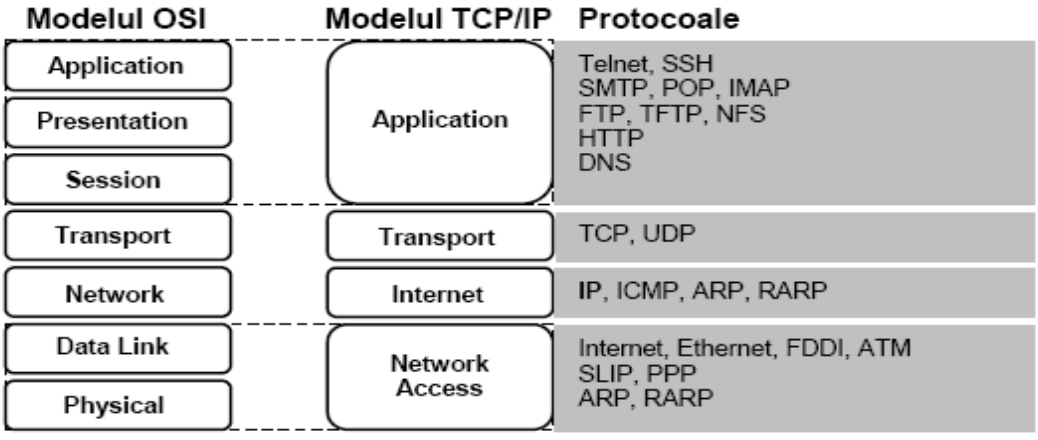
2.1. Protocoale de comunicatie. Suita de protocoale TCP/IP

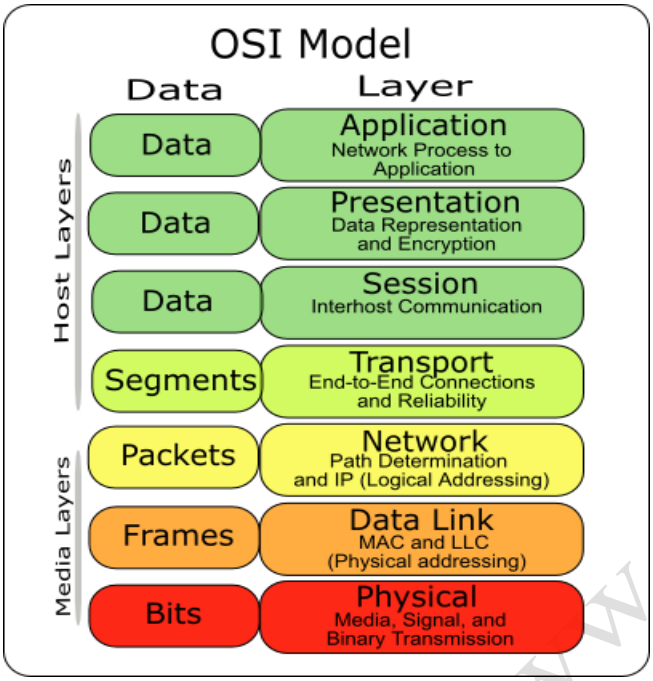
Un protocol de comunicatie corespunde unui set de reguli ce determina formatul si transmisia datelor, controland aspecte legate de :

- Constructia fizica a retelei.
- Modul de conectare a calculatoarelor in retea.
- Formatul si transmiterea datelor.
- Rezolvarea erorilor.

Aceste protocoale sunt create si mentinute de diferite comitete sau organizatii cum ar fi: Institute of Electrical and Electronic Engineers (IEEE), American National Standards Institute (ANSI), International Organization for Standardization (ISO) etc.

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:





- **Nivelul Acces Retea (Network Access)**

Gestioneaza conexiunea la mediul fizic al retelei indiferent de tehnologia folosita. Include nivelul Legatura de Date (Data Link) si Fizic (Physical) al modelului OSI. Adresarea se face prin intermediul adreselor MAC (adrese fizice)

- **Nivelul Internet (Internet)**

Asigura rutarea pachetelor de date la destinatie determinand cel mai bun drum si comunatarea de pachete. Protocolul determinant de pe acest nivel este Internet Protocol (IP). Adresarea se face prin intermediul adreselor IP (adrese logice), corespunzator versiunilor IPv4 / IPv6.

- **Nivelul Transport (Transport)**

Ofera servicii de transport intre punctul sursa si punctul destinatie. De asemenea, asigura controlul fluxului de date, corectia erorilor si calitatea serviciilor. Cele doua protocoale de pe acest nivel sunt : Transport Control Protocol (TCP) si User Datagram Protocol (UDP). TCP este un protocol orientat pe conexiune, garantandu-se receptia informatiei la destinatie asa cum a fost transmisa. Se asigura astfel o comunicatie stabila si fara erori. Spre deosebire de TCP, protocolul UDP nu necesita stabilirea unei conexiuni cu destinatarul, fiind prin urmare un protocol neorientat pe conexiune. Adresarea se face prin intermediul porturilor, exprimate ca numere intregi cuprinse in intervalul intre 0-65535 (http://www.iana.org/assignments/port-numbers)

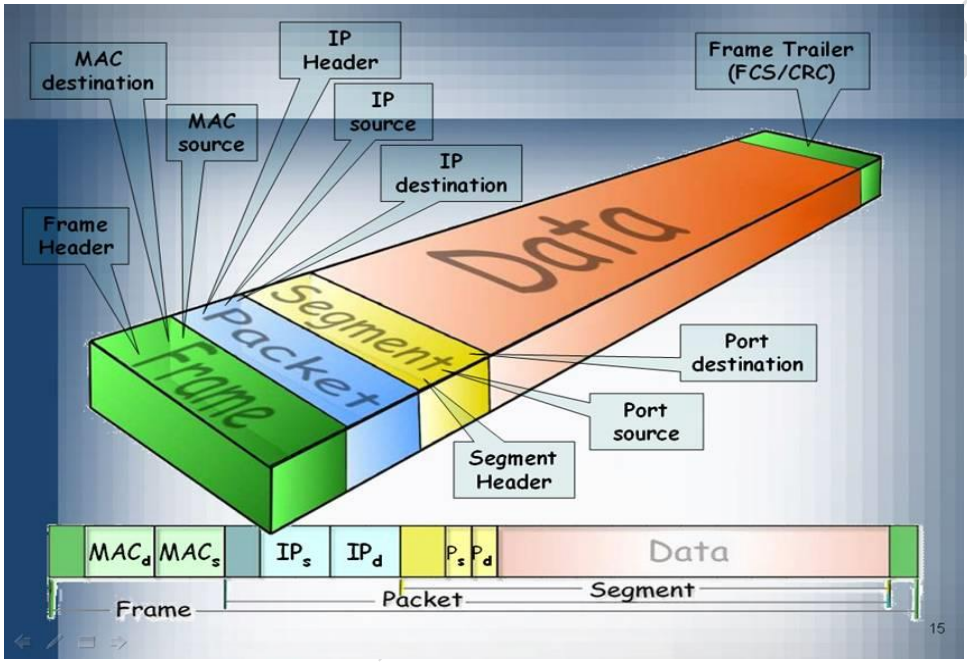
- **Nivelul Aplicatie (Application)**

Asigura reprezentarea corecta a datelor. Acest nivel include nivelul Aplicatie (Application) si Prezentare (Presentation) a modelului OSI (Open System Interconnection). Exista o serie de protocoale la acest nivel: Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) utilizate pentru trasmiterea si receptionarea mesajelor, Hyper Text Transfer Protocol (HTTP) pentru www, File Transfer Protocol (FTP) pentru transferul de fisiere, Domain Name System (DNS) pentru rezolvarea numelor de domeniu etc. Fiecare din aceste protocoale are asociat un numar de port de comunicatie.

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 retea, sau capturarea si salvarea traficului intr-un fisier, cu analiza ulterioara a datelor. Pentru fiecare pachet capturat, programul afiseaza informatii detaliate. **Wireshark** include un limbaj propriu pentru definirea expresiilor de filtrare si permite reconstruirea unei sesiuni TCP pe baza pachetelor capturate.

Referinte bibliografice: 3_Wireshark_Tutorial_ro.pdf, 4_Wireshark_HTTP_SOLUTION_v6.1.pdf, 5_Wireshark_user-guide.pdf

2.3.



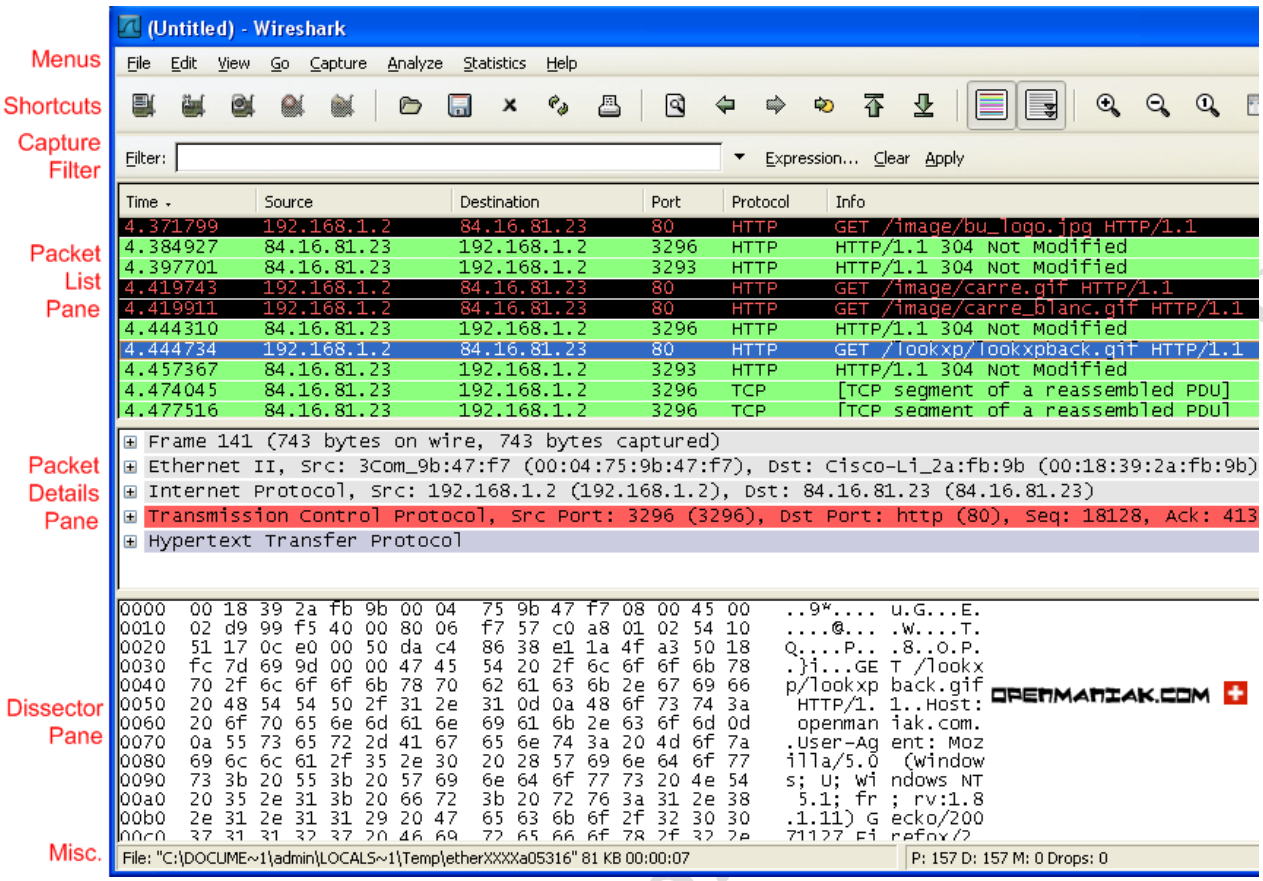
3. Partea practica (Tema: pag.26)

3.1. Utilizare Wireshark

Referinte bibliografice: 3_Wireshark_Tutorial_ro.pdf, 4_Wireshark_HTTP_SOLUTION_v6.1.pdf, 5_Wireshark_user-guide.pdf

Se vor parcurge toate etapele de mai jos, folosind ca studiu de caz adrese convenabil alese. Capturile similare realizate de studenti vor fi salvate intr-un document word, analizate si comentate. Se preda impreuna cu fisierul/fisierele wireshark (de dimensiuni limitate!!!!)

Captura de ecran facuta pentru hostul 192.168.1.2 cand naviga pe siteul *openmaniak.com*



TEMA Wireshark:

Se va parcurge materialul de mai jos, se vor documenta raspunsurile propuse si se vor formula raspunsuri corespunzatoare propriilor date.

Pentru fiecare task in parte se vor salva scrreenshot-uri folosind snipping tool.

Referinte bibliografice: 3_Wireshark_Tutorial_ro.pdf, 4_Wireshark_HTTP_SOLUTION_v6.1.pdf, 5_Wireshark_user-guide.pdf

Using Wireshark™ to View Protocol Data Units

Learning Objectives

Be able to explain the purpose of a protocol analyzer (Wireshark).
Be able to perform basic PDU capture using Wireshark.
Be able to perform basic PDU analysis on straightforward network data traffic.
Experiment with Wireshark features and options such as PDU capture and display filtering.

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

Background

Wireshark is a software protocol analyzer, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. Before June 2006, Wireshark was known as Ethereal.

A packet sniffer (also known as a network analyzer or protocol analyzer) is computer software that can intercept and log data tra passing over a data network. As data streams travel back and forth over the network, the sniffer "captures" each protocol data u (PDU) and can decode and analyze its content according to the appropriate RFC or other specifications.

Wireshark is programmed to recognize the structure of different network protocols. This enables it to display the encapsulation and individual fields of a PDU and interpret their meaning.

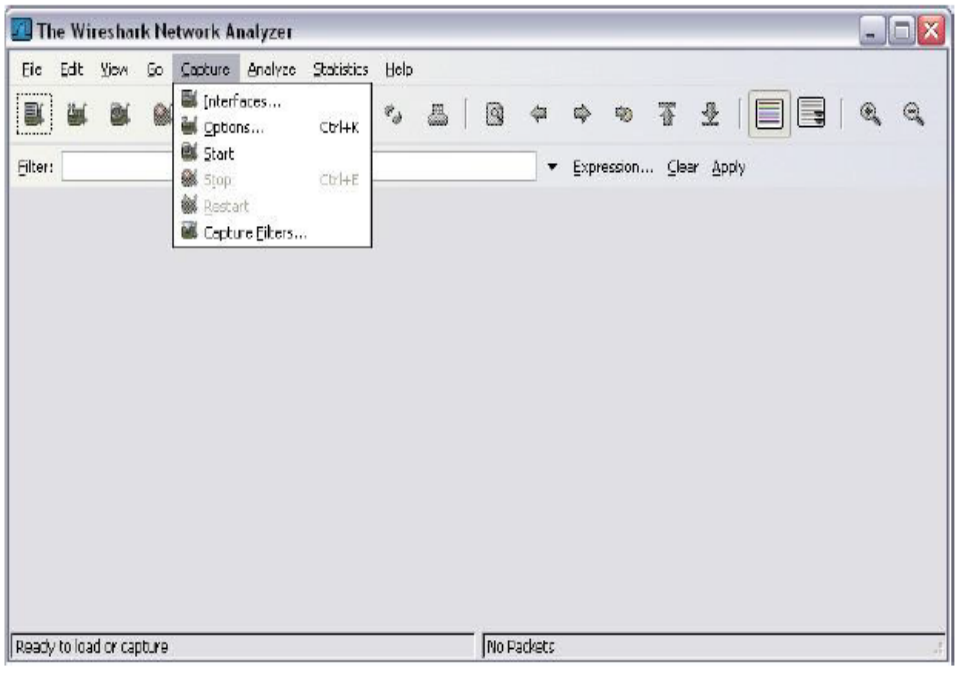
It is a useful tool for anyone working with networks and can be used with most labs in the CCNA courses for data analysis and troubleshooting.

For information and to download the program go to -http://www.Wireshark.org

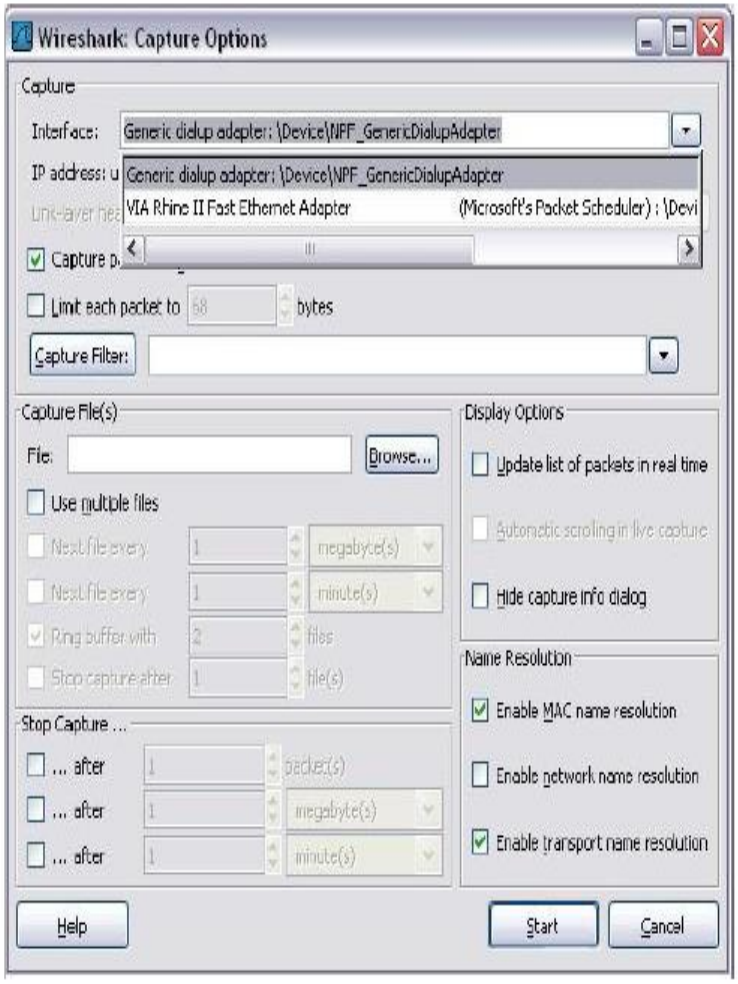
Scenario

To capture PDUs the computer on which Wireshark is installed must have a working connection to the network and Wireshark must be running before any data can be captured.

When Wireshark is launched, the screen below is displayed.

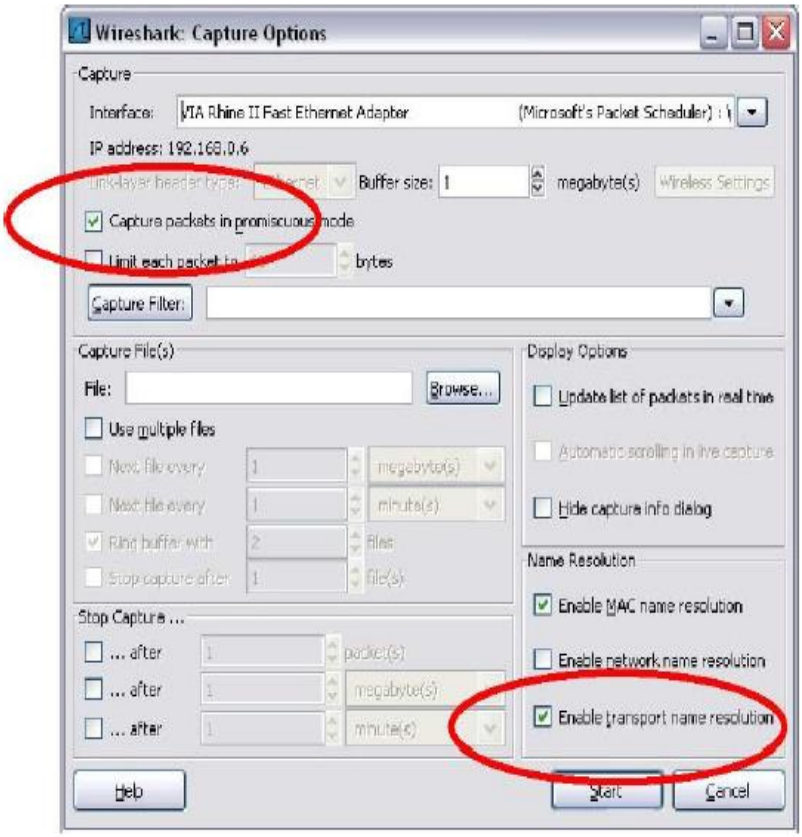


To start data capture it is first necessary to go to the **Capture** menu and select the **Options** choice. The **Options** dialog provides a range of settings and filters which determines which and how much data traffic is captured.



First, it is necessary to ensure that Wireshark is set to monitor the correct interface. From the **Interface** drop down list, select the network adapter in use. Typically, for a computer this will be the connected Ethernet Adapter.

Then other Options can be set. Among those available in **Capture Options**, the two highlighted below are worth examination.



Setting Wireshark to capture packets in promiscuous mode

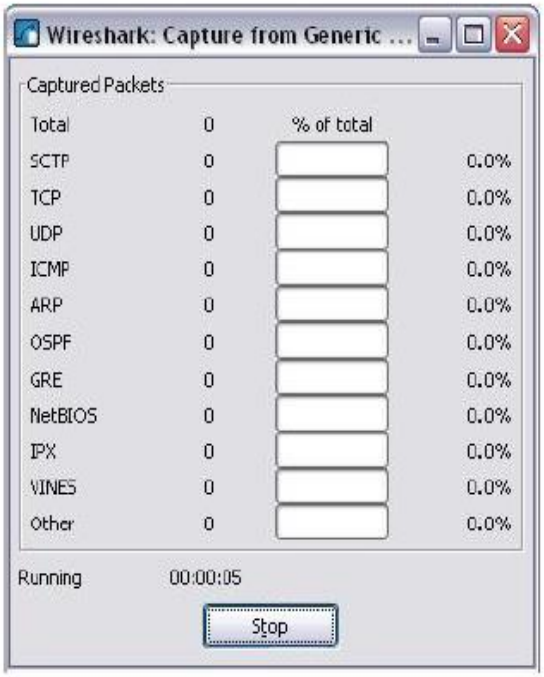
If this feature is NOT checked, only PDUs destined for this computer will be captured. If this feature is checked, all PDUs destined for this computer AND all those detected by the computer NIC on the same network segment (i.e., those that "pass by" the NIC but are not destined for the computer) are captured. Note: The capturing of these other PDUs depends on the intermediary device connecting the end device computers on this network. As you use different intermediary devices (hubs, switches, routers) throughout these courses, you will experience the different Wireshark results.

Setting Wireshark for network name resolution

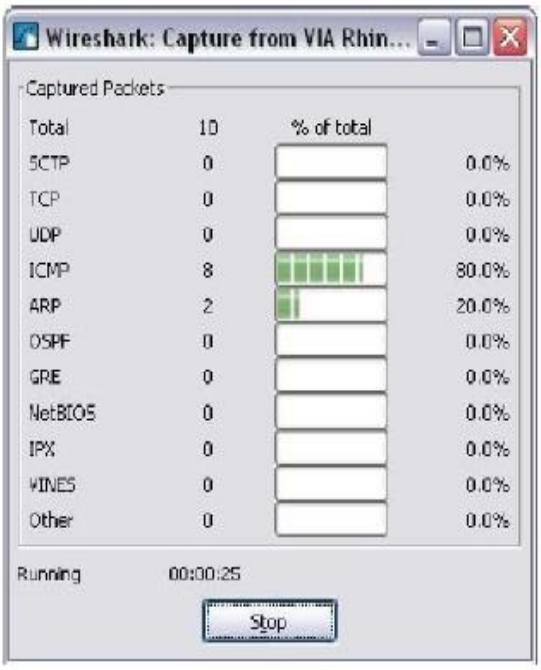
This option allows you to control whether or not Wireshark translates network addresses found in PDUs into names. Although this is a useful feature, the name resolution process may add extra PDUs to your captured data perhaps distorting the analysis.

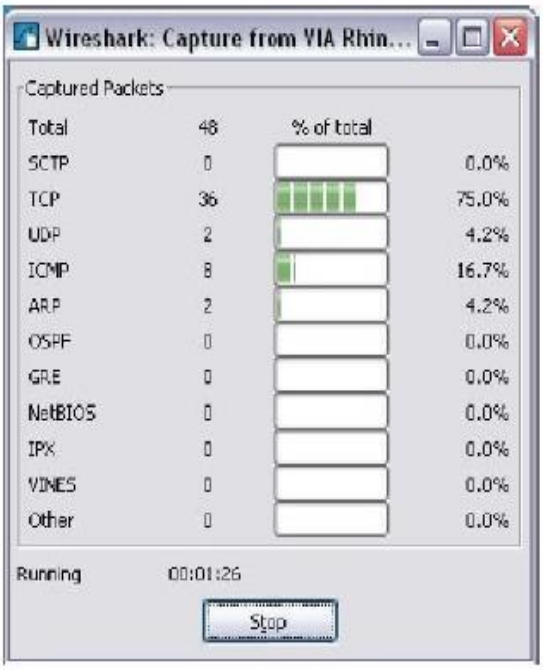
There are also a number of other capture filtering and process settings available.

Clicking on the **Start** button starts the data capture process and a message box displays the progress of this process.



As data PDUs are captured, the types and number are indicated in the message box

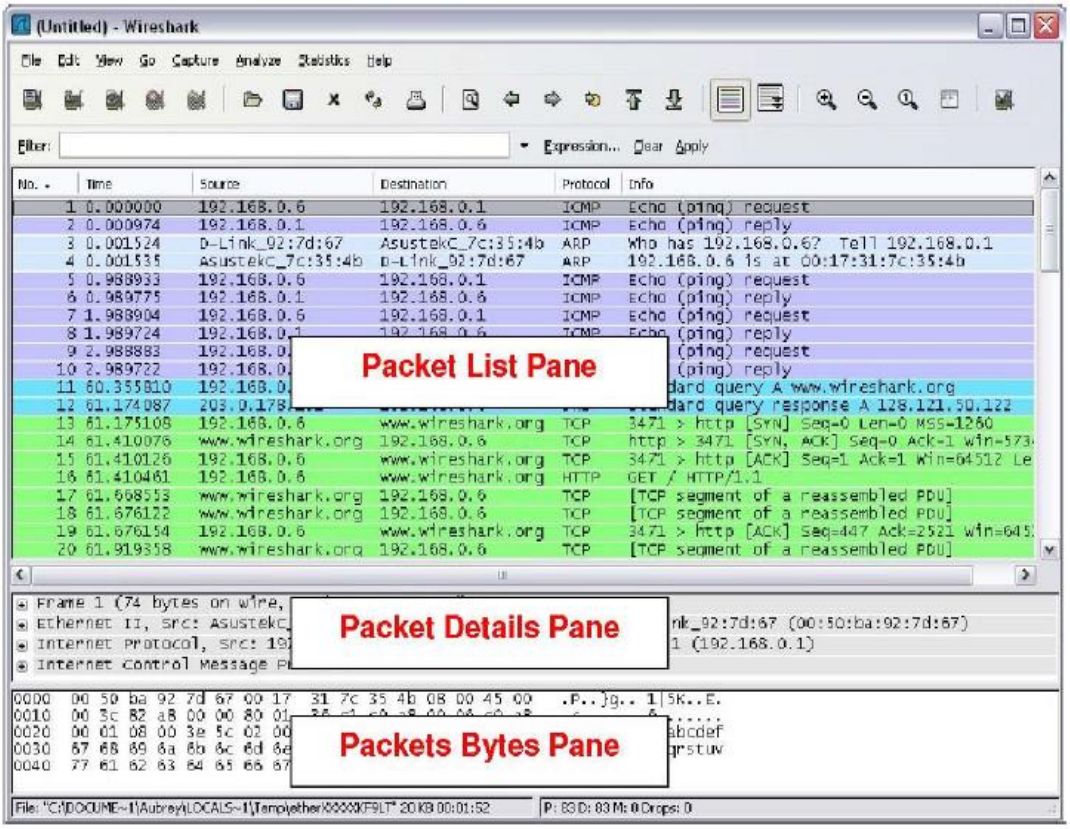




The examples above show the capture of a ping process and then accessing a web page.

When the **Stop** button is clicked, the capture process is terminated and the main screen is displayed.

This main display window of Wireshark has three panes.



The PDU (or Packet) List Pane at the top of the diagram displays a summary of each packet captured. By clicking on packets in this pane, you control what is displayed in the other two panes.

The PDU (or Packet) Details Pane in the middle of the diagram displays the packet selected in the Packet List Pane in more detail.

The PDU (or Packet) Bytes Pane at the bottom of the diagram displays the actual data (in hexadecimal form representing the actual binary) from the packet selected in the Packet List Pane, and highlights the field selected in the Packet Details Pane.

Each line in the Packet List corresponds to one PDU or packet of the captured data. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes. The example above shows the PDUs captured when the ping utility was used and http://www.Wireshark.org was accessed. Packet number 1 is selected in this pane.

The Packet Details pane shows the current packet (selected in the "Packet List" pane) in a more detailed form. This pane shows the protocols and protocol fields of the selected packet. The protocols and fields of the packet are displayed using a tree, which can be expanded and collapsed.

The Packet Bytes pane shows the data of the current packet (selected in the "Packet List" pane) in what is known as "hexdump" style. In this lab, this pane will not be examined in detail. However, when a more in-depth analysis is required this displayed information is useful for examining the binary values and content of PDUs.

The information captured for the data PDUs can be saved in a file. This file can then be opened in Wireshark for analysis some time in the future without the need to re-capture the same data traffic again. The information displayed when a capture file is opened is the same as the original capture.

When closing a data capture screen or exiting Wireshark you are prompted to save the captured PDUs.

Clicking on **Continue without Saving** closes the file or exits Wireshark without saving the displayed captured data.

Task 1: Ping PDU Capture

Step 1: After ensuring that the standard lab topology and configuration is correct, launch Wireshark on a computer in a lab pod.

Set the Capture Options as described above in the overview and start the capture process.

From the command line of the computer, ping the IP address of another network connected and powered on end device on in the lab topology. In this case, ping the Eagle Server at using the command ping

192.168.254.254.

After receiving the successful replies to the ping in the command line window, stop the packet capture.

Step 2: Examine the Packet List pane.

The Packet List pane on Wireshark should now look something like this:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
2	2.000032	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
3	4.000059	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
4	4.072858	QuantaCo_bd:0c:7c	Broadcast	ARP	Who has 10.1.1.254? Tell 10.1.1.1
5	4.073609	Cisco_cf:66:40	QuantaCo_bd:0c:7c	ARP	10.1.1.254 is at 00:0c:85:cf:66:40
6	4.073626	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
7	4.074122	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
8	5.067535	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
9	5.068007	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
10	6.000113	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
11	6.067548	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
12	6.068019	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
13	6.084103	Cisco_9f:6c:c9	Cisco_9f:6c:c9	LOOP	Reply
14	7.067603	10.1.1.1	192.168.254.254	ICMP	Echo (ping) request
15	7.068131	192.168.254.254	10.1.1.1	ICMP	Echo (ping) reply
16	8.000126	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =
17	9.975700	Cisco_9f:6c:c9	CDP/VTP/DTP/PAgP/U	DTP	Dynamic Trunking Protocol
18	10.000134	Cisco_9f:6c:c9	Spanning-tree-(for	STP	Conf. Root = 32769/00:0f:f7:9f:6c:c0 Cost =

Look at the packets listed above; we are interested in packet numbers 6, 7, 8, 9, 11, 12, 14 and 15. Locate the equivalent packets on the packet list on your computer.

Observatie: Pentru **Task 1** de mai sus, se va folosi o adresa IP a unui calculator/host din aceeasi retea (aflata cu ipconfig pe acel calculator)

If you performed Step 1A above match the messages displayed in the command line window when the ping was issued with the six packets captured by Wireshark.

From the Wireshark Packet List answer the following:

What protocol is used by ping? _____ICMP_____

What is the full protocol name? ___Internet Control Message Protocol____

What are the names of the two ping messages? _____Echo Request____

_____Echo Reply_____

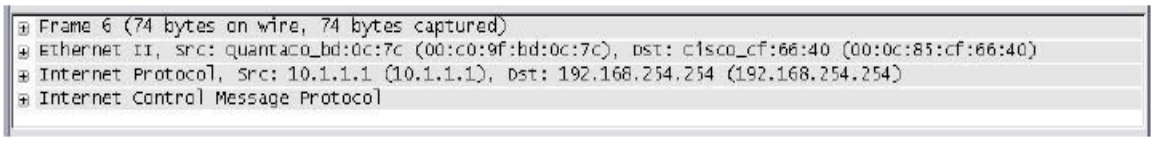
Are the listed source and destination IP addresses what you expected? Yes / No

Why? _____

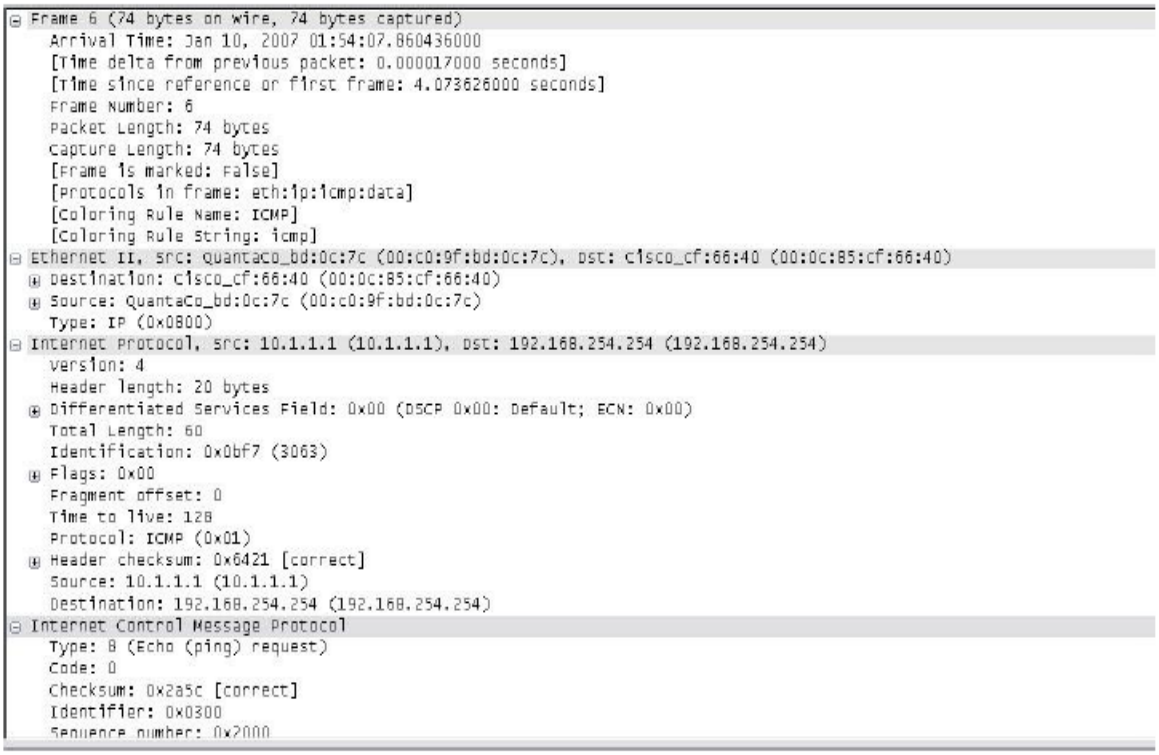
Answers may vary-Yes, the source address is my computer and the destination is the Eagle server

Step 3: Select (highlight) the first echo request packet on the list with the mouse.

The Packet Detail pane will now display something similar to:



Click on each of the four "**+**" to expand the information. The packet Detail Pane will now be similar to:



As you can see, the details for each section and protocol can be expanded further. Spend some time scrolling through this information. At this stage of the course, you may not fully understand the information displayed but make a note of the information you do recognize.

Locate the two different types of 'Source" and "Destination". Why are there two types?

The Ethernet II shows the MAC addresses and the Internet Protocol shows the IP addresses

What protocols are in the Ethernet frame?

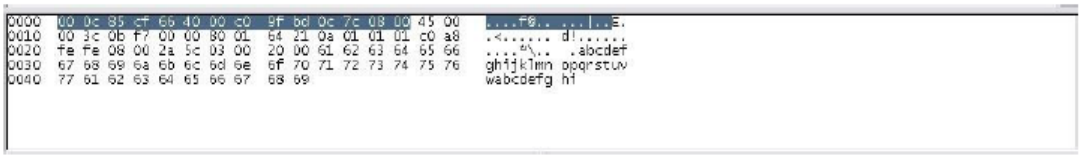
___ eth:ip:icmp:data _____

As you select a line in the Packets Detail pane all or part of the information in the Packet Bytes pane also

becomes highlighted.

For example, if the second line (+ Ethernet II) is highlighted in the Details pane the Bytes pane now

highlights the corresponding values.



This shows the particular binary values that represent that information in the PDU. At this stage of the course, it is not necessary to understand this information in detail.

Step 4: Go to the File menu and select Close.

Click on **Continue without Saving** when this message box appears.

Task 2: FTP PDU Capture

Step 1: Start packet capture.

Assuming Wireshark is still running from the previous steps, start packet capture by clicking on the **Start** option on the **Capture** menu of Wireshark.

At the command line on your computer running Wireshark, enter **ftp 192.168.254.254**

When the connection is established, enter **anonymous** as the user without a password.

Userid: **anonymous**

Password: <ENTER>

You may alternatively use login with userid **cisco** and with password **cisco**.

When successfully logged in enter **get** `/pub/eagle_labs/eagle1/chapter1/gaim-1.5.0.exe` and press the enter key <ENTER>. This will start downloading the file from the ftp server. The output will look similar to:

http://kb2.adobe.com/cps/164/tn_16418.html

Observatie: Pentru **TASK 2** de mai sus, se va utiliza:

```
C:\Windows\system32>ftp ftp.redhat.com
Connected to ftp.redhat.com.
220 Red Hat FTP server ready. All transfers are logged. (FTP) [no EPSV]
User (ftp.redhat.com:(none)): anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> ?
Commands may be abbreviated.  Commands are:

!               delete          literal         prompt          send
?               debug           ls              put             status
append          dir             mdelete         pwd             trace
ascii           disconnect      mdir            quit            type
bell            get             mget            quote           user
binary          glob            mkdir           recv            verbose
bye             hash            mls             remotehelp
cd              help            mput            rename
close           lcd             open            rmdir
```

```
ftp> status
Connected to ftp.redhat.com.
Type: ascii; Verbose: On ; Bell: Off ; Prompting: On ; Globbing: On
Debugging: Off ; Hash mark printing: Off .
ftp> trace
Packet tracing On .
ftp> user
Username anonymous
331 Can't change from guest user.
Password:
230 Already logged in.
ftp> bye
221 Goodbye.
```

```
C:\Documents and Settings\ccna1>ftp eagle-server.example.com
Connected to eagle-server.example.com.
220 Welcome to the eagle-server FTP service.
User (eagle-server.example.com:(none)): anonymous
331 Please specify the password.
Password:<ENTER>
230 Login successful.
ftp> get /pub/eagle_labs/eagle1/chapter1/gaim-1.5.0.exe
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for
pub/eagle_labs/eagle1/chapter1/gaim-1.5.0.exe (6967072 bytes).
226 File send OK.
ftp: 6967072 bytes received in 0.59Seconds 11729.08Kbytes/sec.
```

When the file download is complete enter **quit**

```
ftp> quit
221 Goodbye.
C:\Documents and Settings\ccna1>
```

Step 2: Increase the size of the Wireshark Packet List pane and scroll through the PDUs listed.

Locate and note those PDUs associated with the file download.
These will be the PDUs from the Layer 4 protocol TCP and the Layer 7 protocol FTP.

Identify the three groups of PDUs associated with the file transfer.

If you performed the step above, match the packets with the messages and prompts in the FTP command line window.

The first group is associated with the "connection" phase and logging into the server.
List examples of messages exchanged in this phase.

Answers will vary- 1292 > ftp [SYN], FTP > 1292 [SYN, ACK], Response: 220 Welcome to the eagle-server FTP service, 1292 > ftp [ACK], Request: User anonymous, Response: 331 Please specify the password, Request: Pass

Locate and list examples of messages exchanged in the second phase that is the actual download request and the data transfer.

Answers will vary- FTP Data: 1448 bytes, 1294 > ftp-data [ACK],

The third group of PDUs relate to logging out and "breaking the connection". List examples of messages exchanged during this process.

Answers will vary- Request:QUIT, Response: 221 Goodbye, 1292 > ftp [FIN, ACK], ftp >1292 [FIN, ACK]

Locate recurring TCP exchanges throughout the FTP process. What feature of TCP does this indicate? ___Send and receipt of data_____

Step 3: Examine Packet Details.

Select (highlight) a packet on the list associated with the first phase of the FTP process.

View the packet details in the Details pane.

What are the protocols encapsulated in the frame?

____ Eth:ip:tcp:ftp-data _____

Highlight the packets containing the user name and password.

Examine the highlighted portion in the Packet Byte pane.

What does this say about the security of this FTP login process?

_____ Security isn't very high because the name and password are visible._____

Highlight a packet associated with the second phase.

From any pane, locate the packet containing the file name.

The filename is: ___gaim-1.5.0.exe_____

Highlight a packet containing the actual file content -note the plain text visible in the Byte pane.

Highlight and examine, in the Details and Byte panes, some packets exchanged in the third phase of the

file download.
What features distinguish the content of these packets?
____ A [FIN, ACK] is issued to close the connection._____

When finished, close the Wireshark file and continue without saving

Task 3: HTTP PDU Capture Step 1: Start packet capture.

Assuming Wireshark is still running from the previous steps, start packet capture by clicking on the **Start** option on the **Capture** menu of Wireshark.

Note: Capture Options do not have to be set if continuing from previous steps of this lab.

Launch a web browser on the computer that is running Wireshark.

Observatie: Pentru **TASK 3** de mai sus, se va folosi **https://www.python.org**

Enter the URL of the Eagle Server of **example.com** or enter the IP address-192.168.254.254. When the webpage has fully downloaded, stop the Wireshark packet capture.

Step 2: Increase the size of the Wireshark Packet List pane and scroll through the PDUs listed.

Locate and identify the TCP and HTTP packets associated with the webpage download. Note the similarity between this message exchange and the FTP exchange.

Step 3: In the Packet List pane, highlight an HTTP packet that has the notation "(text/html)" in the Info column.

In the Packet Detail pane click on the "+" next to "**Line-based text data: html**"
When this information expands what is displayed?
_____HTML code for the web page_____

Examine the highlighted portion of the Byte Panel.
This shows the HTML data carried by the packet.

When finished close the Wireshark file and continue without saving

Task 4: Reflection

Consider the encapsulation information pertaining to captured network data Wireshark can provide. Relate this to the OSI and TCP/IP layer models. It is important that you can recognize and link both the protocols represented and the protocol layer and encapsulation types of the models with the information provided by Wireshark.

Task 5: Challenge

Discuss how you could use a protocol analyzer such as Wireshark to:
(1) Troubleshoot the failure of a webpage to download successfully to a browser on a computer. and
(2) Identify data traffic on a network that is requested by users.

Answers could vary-Wireshark could show when request for a web page failed due to incorrect URL. User traffic could be monitored to identify errors in source or destination.

More Information:

Transfer files through FTP | Command line

Sursa: http://kb2.adobe.com/cps/164/tn_16418.html

A command line FTP utility is available on Windows and Mac OS X and can be used without installing additional software. Command line FTP is a reliable tool for transferring files. You can also use it for testing when troubleshooting problems with the Dreamweaver or UltraDev FTP client. The instructions provided below show you how to use the command line FTP utility for Windows and Mac OS X.

Note: This TechNote provides basic instructions for using the command line FTP utility. For more detailed information, consult the documentation for your operating system.

Windows:

1 To launch command line FTP utility, go to Start > Run and enter"cmd" into the Open field of the run dialog box. Click OK or press enter.

Note: If using Windows 98/95/Me, enter "command" instead of"cmd".

2 To change to a local directory that contains your site files, type cd followed by the path to the desired directory after the prompt.

Example: If your site files are stored in c:\sites\my_site, enter cd c:\sites\my_site.

3 At the prompt, enter ftp + FTP server name and press enter.

```
C:\Temp>ftp ftp.redhat.com
Connected to ftp.redhat.com.
220 Red Hat FTP server ready. All transfers are logged.
User (ftp.redhat.com:(none)): ■
```

Note: The server name would be the same as the ftp host name used in the define sites dialog box in Dreamweaver and UltraDev.

Example: ftp ftp.redhat.com.

4 You are prompted to enter a user name. Following the prompt, enter the user name for the FTP account and press enter.

Note: If you are connecting to an FTP server that allows anonymous FTP, you don't need a valid account. However, use the preconfigured account called ftp (ftp is the same as anonymous) to access the server.

5 Enter your password when the password prompt appears and press enter.

```
C:\Temp>ftp ftp.redhat.com
Connected to ftp.redhat.com.
220 Red Hat FTP server ready. All transfers are logged.
User (ftp.redhat.com:(none)): ftp
331 Please specify the password.
Password:
230 Login successful. Have fun.
ftp> ■
```

Note: For security reasons, the password entered is not displayed on the screen.

6 When the' Login successful' message appears, you have successfully connected. You now know that the ftp server is up and running and that you can successfully connect to it from your machine.
At the ftp> prompt, you can run any FTP commands. To get a listing of acceptable FTP commands, simply enter a question mark (?) and press enter.

Note: You still see a list of FTP commands even if you are not successfully connected to the FTP server.

Commonly used FTP commands:

ls: directory listing equivalent to dir.
ls -l: long directory listing, more detail.
pwd: display the name of current directory.

cd: change directory.
lcd: change the local current directory.
get: to download the file from the FTP server.
put: to transfer file and place it on the FTP server.
mget: to download multiple files from the FTP server.
mput: to transfer multiple files to the FTP server.
prompt: to turn on/off interactive mode.
binary: to turn on binary mode.
ascii: to turn on ascii mode.
delete: to turn a file on FTP server.
status: to display how the current FTP session is configured.
mkdir: to make directory on FTP server.
quit/close/bye/disconnect: to disconnect from the FTP server.

Note: All FTP commands are case sensitive and in lower case.

Mac OS X

1 To launch terminal, choose Hard Drive > Applications > Utilities > Terminal.

2 At the UNIX shell prompt, you can change the current directory by typing cd + the path to the desired directory.

3 At the UNIX shell prompt, enter ftp + the FTP host name and press return.

4 Follow Windows steps 4 - 6.

3.2. Aplicatii de retea in Pyton

3.2.1. Recapitulare

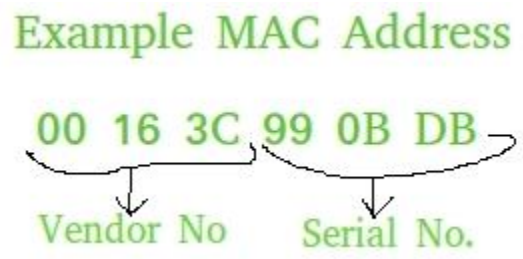
- **Python_intro**
- **Programare_Python**
- **Byte-of-python**

3.2.2.Aplicatii MAC

MAC address also known as physical address is the unique identifier that is assigned to the NIC (Network Interface Card) of the computer. NIC helps in connection of a computer with other computers in the network. MAC address is unique for all the NIC's.

Uses of MAC address :

- Useful in places where IP address change frequently. Helps network admin. to get information regarding network traffic.
- Helps us to configure which computers can be connected to our computers. By this way we can filter potential spam/virus attacks.
- Helps in uniquely identifying computers from other computers around the world.



3.2.2.1 Using uuid.getnode() (pip install uuid)

getnode() can be used to extract the MAC address of the computer. This function is defined in **uuid** module (pip install uuid).
The illustrated code given below shows how to generate a UUID for a given host, identified by its MAC address, using the uuid1() function.

3_2_2_1.py

```
import uuid

# printing the value of unique MAC
# address using uuid and getnode() function
print (hex(uuid.getnode()))
```

Output :

```
0x163e990bdb
```

3.2.2.2 Using getnode() + format() [for better formatting]

3_2_2_2.py

```
import uuid

# joins elements of getnode() after each 2 digits.

print ("The MAC address in formatted way is : ", end="")
print (':'.join(['{:02x}'.format((uuid.getnode() >> ele) & 0xff)
for ele in range(0,8*6,8)][::-1]))
```

Output :

```
The MAC address in formatted way is : 00:16:3e:99:0b:db
```

3.2.2.3 Using getnode() + findall() + re() [for reducing complexity]

3_2_2_3.py

```
import re, uuid

# joins elements of getnode() after each 2 digits.
# using regex expression
print ("The MAC address in formatted and less complex way is : ", end="")
print (':'.join(re.findall('..', '%012x' % uuid.getnode())))
```

Output :

```
The MAC address in formatted and less complex way is : 00:16:3e:99:0b:db
```

https://stackoverflow.com/questions/4258822/mac-ethernet-id-using-python

3.2.2.4

3_2_2_4.py

Windows O.S.

```
import sys
import os

def getMacAddress():
    if sys.platform == 'win32':
        for line in os.popen("ipconfig /all"):
            if line.lstrip().startswith('Physical Address'):
                mac = line.split(':')[1].strip().replace('-',':')
                break
    else:
        for line in os.popen("/sbin/ifconfig"):
            if line.find('Ether') > -1:
                mac = line.split()[4]
                break
    return mac

print(getMacAddress())
```

Output:

```
78:e7:g1:84:b5:ed
```

References:

https://docs.python.org/3.7/library/os.html

```
sys — System-specific parameters and functions
This module provides access to some variables used or maintained by the interpreter
and to functions that interact strongly with the interpreter. It is always
available.
```

https://docs.python.org/3/tutorial/stdlib.html
The os module provides dozens of functions for interacting with the operating system:

`os.popen`(*cmd, mode='r', buffering=-1*)

> Open a pipe to or from command *cmd*. The return value is an open file object connected to the pipe, which can be read or written depending on whether *mode* is `'r'` (default) or `'w'`. The *buffering* argument has the same meaning as the corresponding argument to the built-in `open()` function. The returned file object reads or writes text strings rather than bytes.
>
> The `close` method returns `None` if the subprocess exited successfully, or the subprocess's return code if there was an error. On POSIX systems, if the return code is positive it represents the return value of the process left-shifted by one byte. If the return code is negative, the process was terminated by the signal given by the negated value of the return code. (For example, the return value might be `- signal.SIGKILL` if the subprocess was killed.) On Windows systems, the return value contains the signed integer return code from the child process.
>
> This is implemented using `subprocess.Popen`; see that class's documentation for more powerful ways to manage and communicate with subprocesses.

lstrip

Returns a copy of the string with leading characters removed.

startswith()

The startswith() method returns True if a string starts with the specified prefix(string). If not, it returns False.

split()

The split() method breaks up a string at the specified separator and returns a list of strings.

strip()

The strip() method returns a copy of the string with both leading and trailing characters removed (based on the string argument passed).

Linux O.S.

```
>>> ifname = 'eth0'
>>> print open('/sys/class/net/%s/address' % ifname).read()
```

```
Output:
78:e7:g1:84:b5:ed
```

3.2.3.Aplicatii ARP

Displays and modifies entries in the Address Resolution Protocol (ARP) cache, which contains one or more tables that are used to store IP addresses and their resolved Ethernet or Token Ring physical addresses. There is a separate table for each Ethernet or Token Ring network adapter installed on your computer. Used without parameters, **arp** displays help.

Syntax

arp[**-a** [*InetAddr*] [**-N***IfaceAddr*]] [**-g** [*InetAddr*] [**-N***IfaceAddr*]] [**-d***InetAddr* [*IfaceAddr*]] [**-s***InetAddr EtherAddr* [*IfaceAddr*]]

3.2.3.1

3_2_3_1.py

arp –a in cmd ➔ text

```
text = """  Internet Address      Physical Address      Type
  156.178.1.1           5h-c9-6f-78-g9-91     dynamic
  156.178.1.255         ff-ff-ff-ff-ff-ff     static
  167.0.0.11            05-00-9b-00-00-10     static
  167.0.0.123           05-00-9b-00-00-ad     static
  ....."""

for item in text.split():
    if item.count('-') == 5:
        print (item)
```

Output:

```
['e4-d5-3d-3c-36-a8', 'b0-48-7a-f6-a7-e0', 'ff-ff-ff-ff-ff-ff',
'01-00-5e-00-00-16', '01-00-5e-00-00-fc', '01-00-5e-7f-ff-fa']
```

3.2.3.2

3_2_3_2.py

```
import re

addresses = """  Internet Address      Physical Address      Type
156.178.1.1           5f-c9-6f-78-f9-91     dynamic
156.178.1.255         ff-ff-ff-ff-ff-ff     static
167.0.0.11            05-00-9b-00-00-10     static
167.0.0.123           05-00-9b-00-00-ad     static
....."""

print(re.findall(('(?:[0-9a-fA-F]{1,}(?:\-|\:)){5}[0-9a-fA-F]{1,}'),addresses))
```

```
Output:
['5f-c9-6f-78-f9-91', 'ff-ff-ff-ff-ff-ff', '05-00-9b-00-00-10',
'05-00-9b-00-00-ad']
```

3.2.3.3

3_2_3_3.py

```
import subprocess
import re

addresses = subprocess.check_output(['arp', '-a'])

# print (addresses)

print(re.findall(('(?:[0-9a-fA-F]{1,}(?:\-|\:)){5}[0-9a-fA-F]{1,}'),str(addresses)))
```

```
Output:
['5f-c9-6f-78-f9-91', 'ff-ff-ff-ff-ff-ff', '05-00-9b-00-00-10',
'05-00-9b-00-00-ad']
```

3.2.3.4

3_2_3_4.py

```
import subprocess

addresses = subprocess.check_output(['arp', '-a'])

for item in str(addresses).split():
    if item.count('-') == 5:
        print (item)
```

```
Output:
e4-d5-3d-3c-36-a8
b0-48-7a-f6-a7-e0
ff-ff-ff-ff-ff-ff
01-00-5e-00-00-16
01-00-5e-00-00-fc
01-00-5e-7f-ff-fa
```

Tema aplicatii Python:

a. Documentarea tuturor programelor (instructiuni etc) - https://docs.python.org/3.7/genindex.html
b. Pentru aplicatiile MAC (3.2.2) – versiune pentru afisarea tuturor MAC-urilor (sistemele cu 2 sau mai multe placi de retea – cablata si wireless – vor avea cate o adresa MAC pentru fiecare din aceasta)
c. **Challenge:** Interfata grafica personalizata L3- pentru rularea tuturor scripturilor de la punctele 3.2.2 si 3.2.3.

Recomandare: Qt Designer , cu Designer din Anaconda prompt).
http://pythonforengineers.com/your-first-gui-app-with-python-and-pyqt/, https://www.codementor.io/deepaksingh04/design-simple-dialog-using-pyqt5-designer-tool-ajskrd09n, https://wiki.python.org/moin/PyQt/Tutorials

4. Tema (Partea practica: pag.3):

- **Toate punctele din sectiunea 3 "partea practica"** se vor relua de catre cursanti, folosind etapele de lucru indicate.
- Arhiva cu numele **L3_nume+prenume_info3.rar va contine**

L3_nume+prenume_Wireshark (folder – conform Tema Wireshark): Fisiere wireshark demonstrative, pe care s-a lucrat si care au fost analizate + .doc cu capturi (Snipping Tool) si comentarii pentru cele scenariile prezentate

L3_nume+prenume_Python (folder) (contine scripturile .py si .doc cu capturi (Snipping Tool) si comentarii pentru TOATE aplicatiile Python (pastrandu-se denumirile indicate) conform Tema aplicatii Python.

Arhiva cu numele **L3_nume+prenume_info3.rar** se va trimite prin e-mail la adresa retelecdsd@gmail.com precizandu-se la **Subject: L3_nume+prenume_info3**, pana pe data de **23 octombrie 2019, ora 8.00 a.m. (Atentie, gmail nu "prea vrea" .rar in .rar sau executabile http://www.makeuseof.com/tag/4-ways-email-attachments-file-extension-blocked/)**

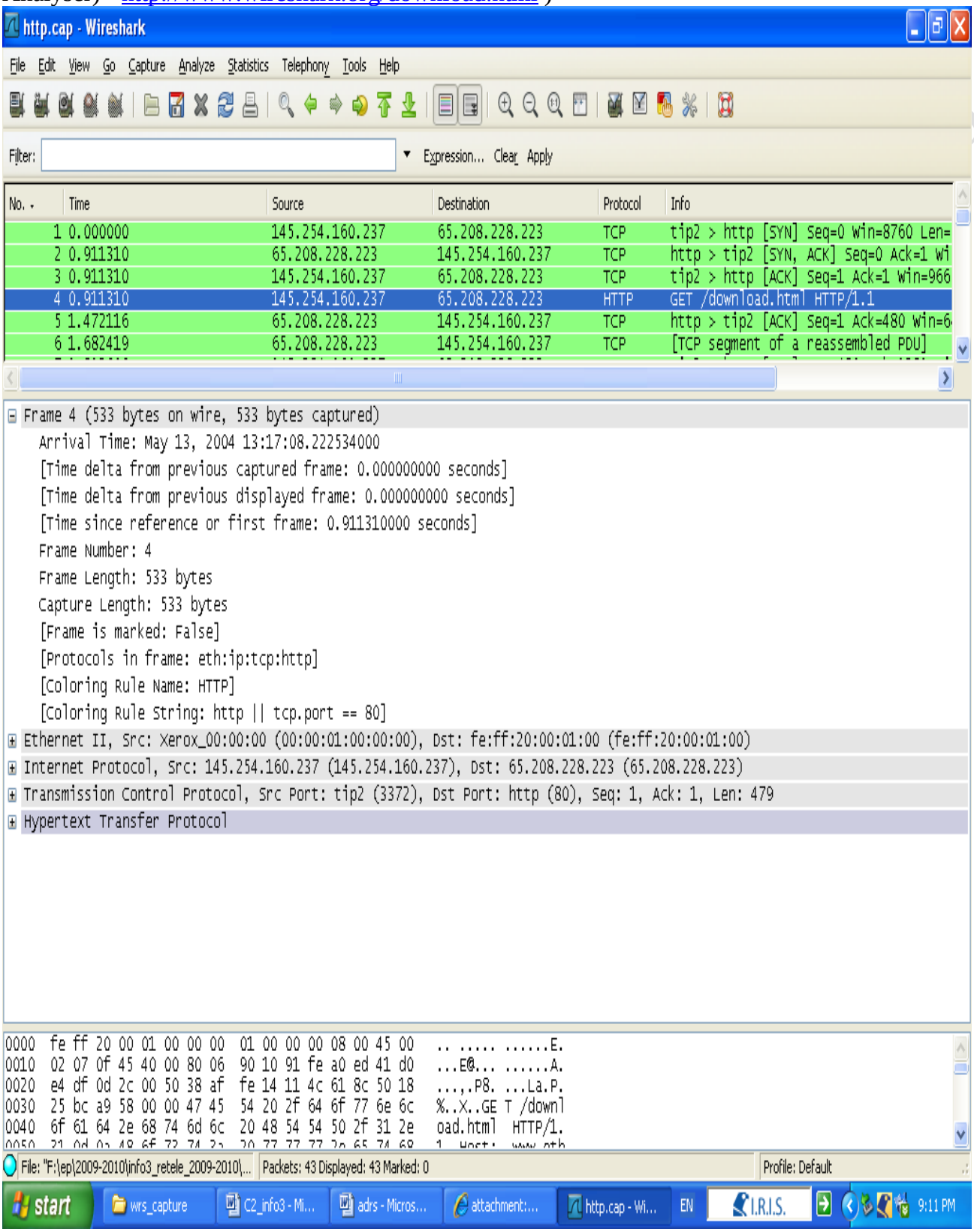
Cursantii sunt incurajati sa analizeze si sa comenteze rezultatele obtinute, studiind si materialele indicate in bibliografie si anexe.

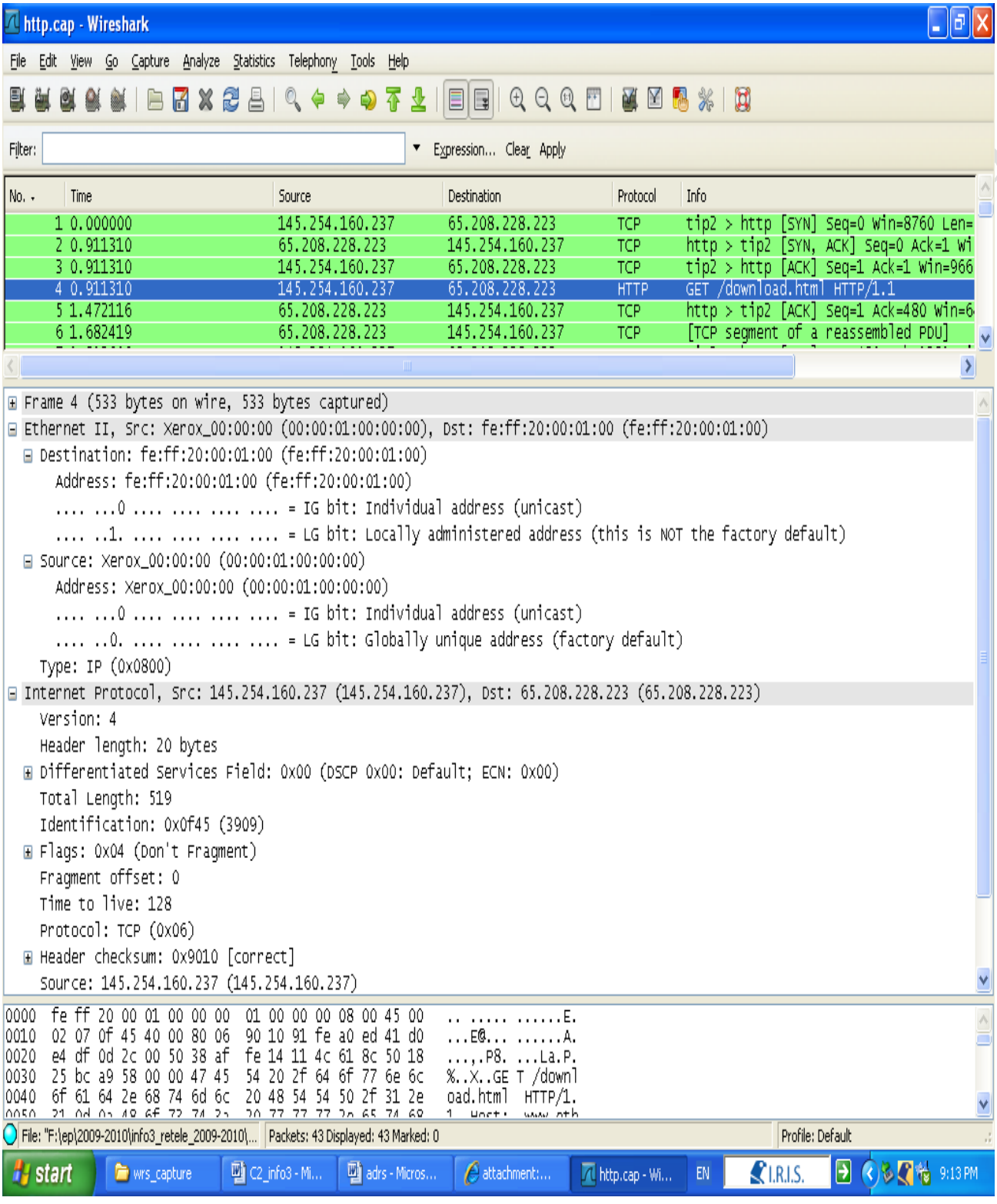
Obs:

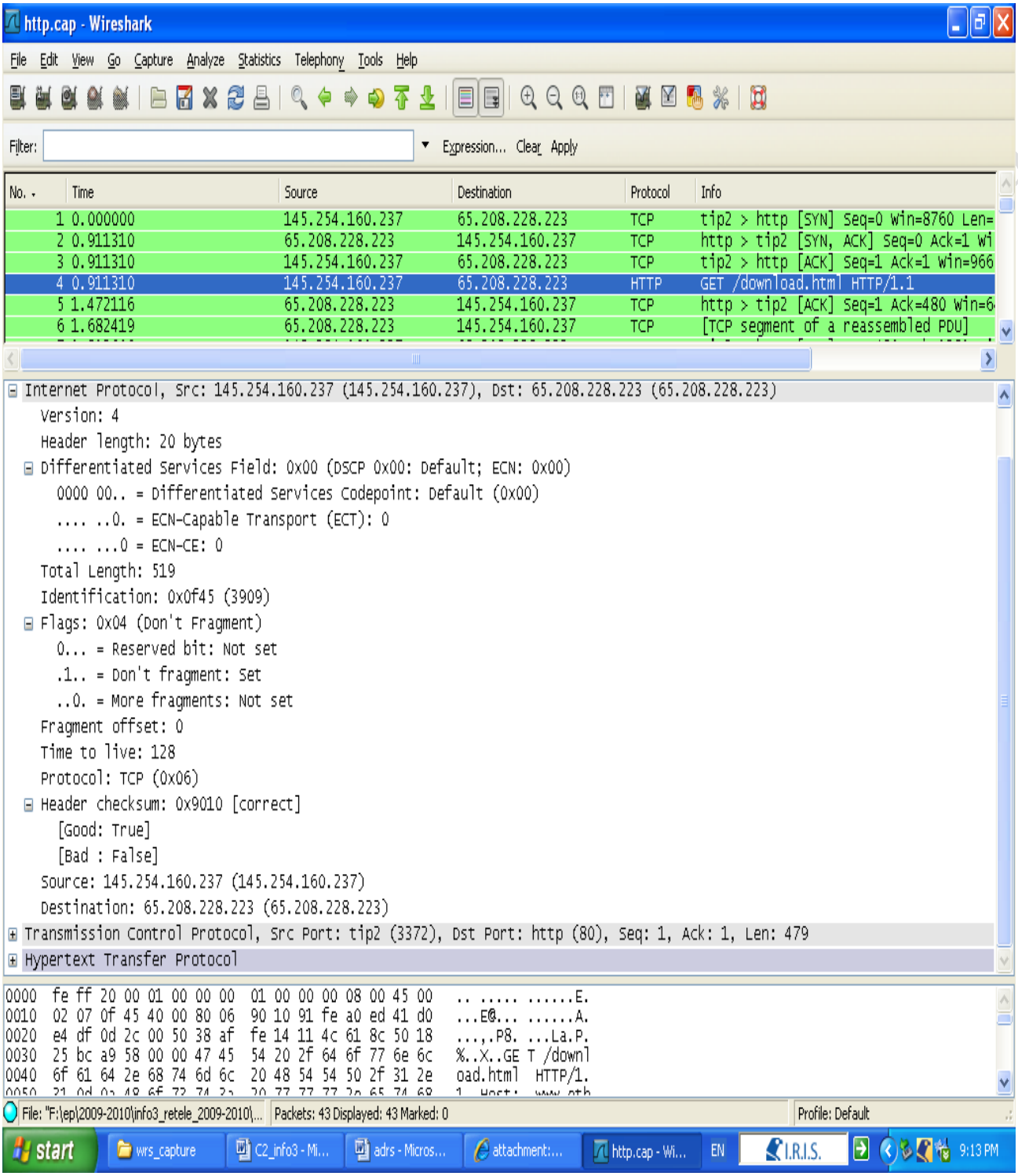
Punctaj maxim (Data trimiterii temei)			
<= 23.10. 2019	27.10. 2019	31.10.2019	4.11.2019
100 pct	80 pct	60 pct	50 pct

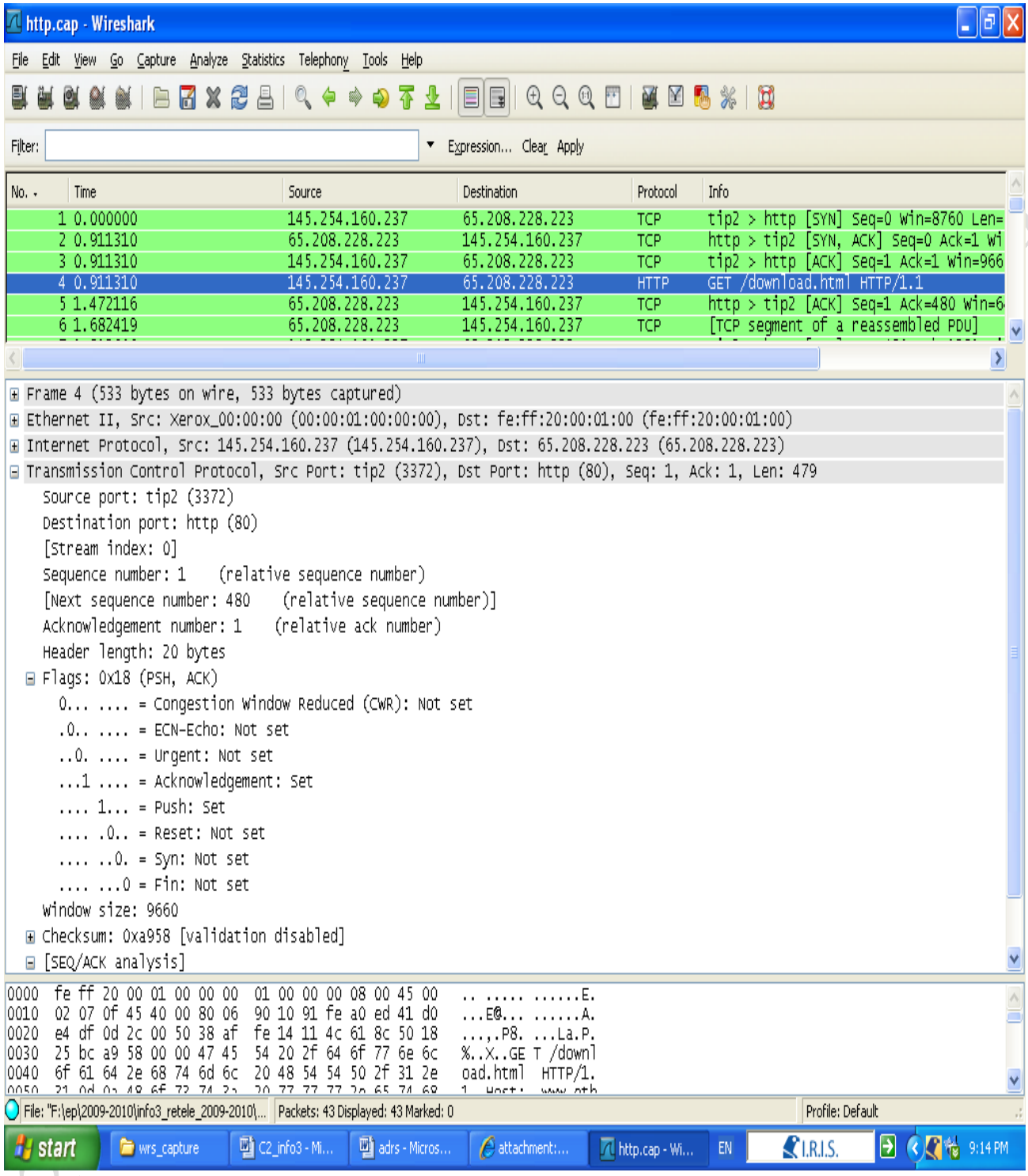
Anexa 1:

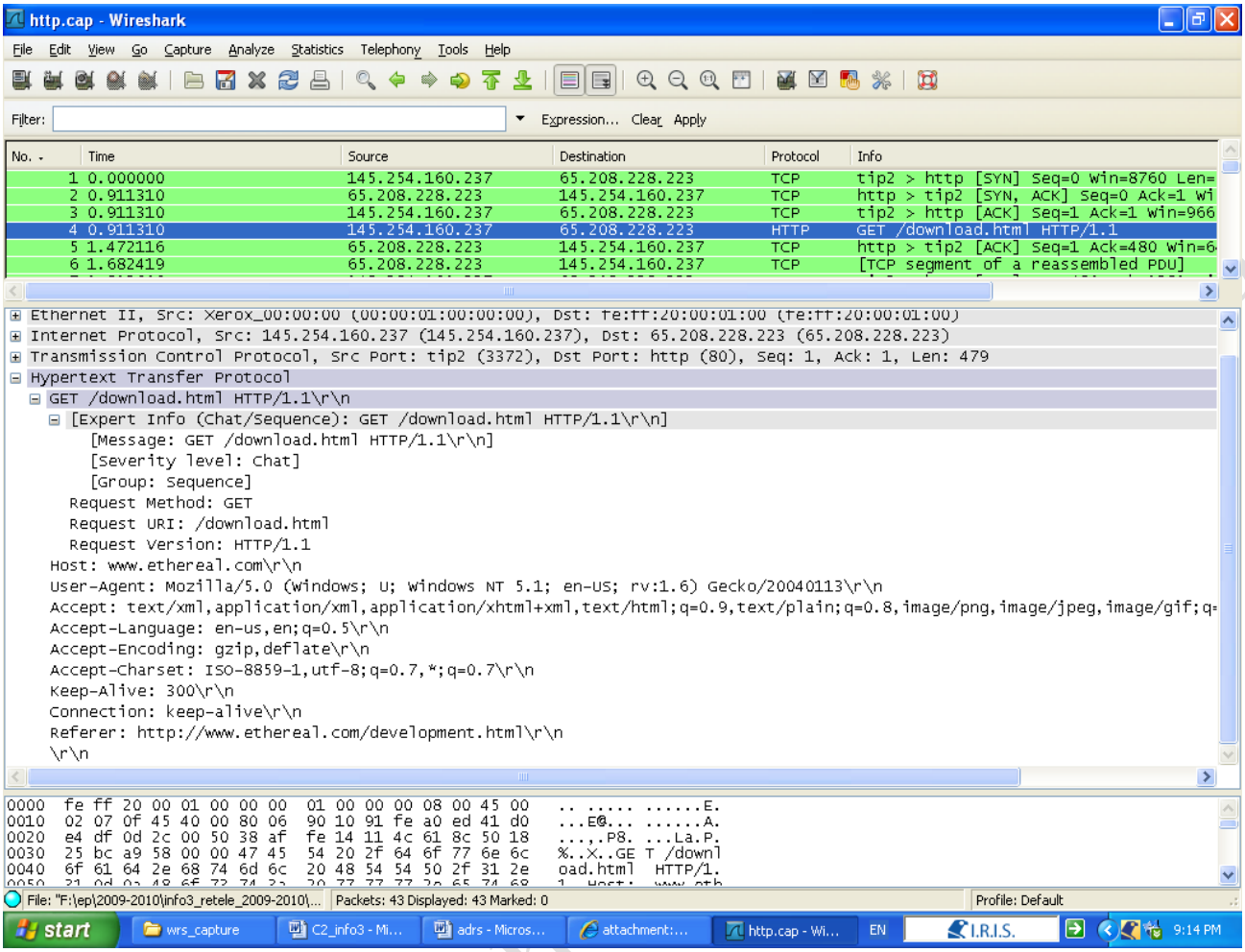
Analiza unui frame corespunzator protocolului http (Aplicatie Wireshark (Network Protocol Analyser) - http://www.wireshark.org/download.html)




http.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 145.254.160.237 65.208.228.223 TCP tip2 > http [SYN] Seq=0 Win=8760 Len=
2 0.911310 65.208.228.223 145.254.160.237 TCP http > tip2 [SYN, ACK] Seq=0 Ack=1 Wi
3 0.911310 145.254.160.237 65.208.228.223 TCP tip2 > http [ACK] Seq=1 Ack=1 Win=966
4 0.911310 145.254.160.237 65.208.228.223 HTTP GET /download.html HTTP/1.1
5 1.472116 65.208.228.223 145.254.160.237 TCP http > tip2 [ACK] Seq=1 Ack=480 Win=6
6 1.682419 65.208.228.223 145.254.160.237 TCP [TCP segment of a reassembled PDU]
Frame 4 (533 bytes on wire, 533 bytes captured)
Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
Destination: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
Address: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
....0 .... = IG bit: Individual address (unicast)
.... ..1. .... = LG bit: Locally administered address (this is NOT the factory default)
Source: Xerox_00:00:00 (00:00:01:00:00:00)
Address: Xerox_00:00:00 (00:00:01:00:00:00)
....0 .... = IG bit: Individual address (unicast)
.... ..0. .... = LG bit: Globally unique address (factory default)
Type: IP (0x0800)
Internet Protocol, Src: 145.254.160.237 (145.254.160.237), Dst: 65.208.228.223 (65.208.228.223)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 519
Identification: 0x0f45 (3909)
Flags: 0x04 (Don't Fragment)
Fragment offset: 0
Time to live: 128
Protocol: TCP (0x06)
Header checksum: 0x9010 [correct]
Source: 145.254.160.237 (145.254.160.237)
0000 fe ff 20 00 01 00 00 00 01 00 00 00 08 00 45 00 .. ..E.
0010 02 07 0f 45 40 00 80 06 90 10 91 fe a0 ed 41 d0 ...E@... ......A.
0020 e4 df 0d 2c 00 50 38 af fe 14 11 4c 61 8c 50 18 ...,.P8. ...La.P.
0030 25 bc a9 58 00 00 47 45 54 20 2f 64 6f 77 6e 6c %..X..GE T /downl
0040 6f 61 64 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e oad.html HTTP/1.
File: "F:\ep\2009-2010\info3_retele_2009-2010\... Packets: 43 Displayed: 43 Marked: 0 Profile: Default
start wrs_capture C2_info3 - Mi... adrs - Micros... attachment:... http.cap - Wi... EN I.R.I.S. 9:13 PM


http.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 145.254.160.237 65.208.228.223 TCP tip2 > http [SYN] Seq=0 Win=8760 Len=
2 0.911310 65.208.228.223 145.254.160.237 TCP http > tip2 [SYN, ACK] Seq=0 Ack=1 Wi
3 0.911310 145.254.160.237 65.208.228.223 TCP tip2 > http [ACK] Seq=1 Ack=1 Win=966
4 0.911310 145.254.160.237 65.208.228.223 HTTP GET /download.html HTTP/1.1
5 1.472116 65.208.228.223 145.254.160.237 TCP http > tip2 [ACK] Seq=1 Ack=480 Win=6
6 1.682419 65.208.228.223 145.254.160.237 TCP [TCP segment of a reassembled PDU]
Internet Protocol, Src: 145.254.160.237 (145.254.160.237), Dst: 65.208.228.223 (65.208.228.223)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
0000 00.. = Differentiated Services Codepoint: Default (0x00)
.... ..0. = ECN-Capable Transport (ECT): 0
....0 = ECN-CE: 0
Total Length: 519
Identification: 0x0f45 (3909)
Flags: 0x04 (Don't Fragment)
0... = Reserved bit: Not set
.1.. = Don't fragment: Set
..0. = More fragments: Not set
Fragment offset: 0
Time to live: 128
Protocol: TCP (0x06)
Header checksum: 0x9010 [correct]
[Good: True]
[Bad : False]
Source: 145.254.160.237 (145.254.160.237)
Destination: 65.208.228.223 (65.208.228.223)
Transmission Control Protocol, Src Port: tip2 (3372), Dst Port: http (80), Seq: 1, Ack: 1, Len: 479
Hypertext Transfer Protocol
0000 fe ff 20 00 01 00 00 00 01 00 00 00 08 00 45 00 .. ..E.
0010 02 07 0f 45 40 00 80 06 90 10 91 fe a0 ed 41 d0 ...E@... ......A.
0020 e4 df 0d 2c 00 50 38 af fe 14 11 4c 61 8c 50 18 ...,.P8. ...La.P.
0030 25 bc a9 58 00 00 47 45 54 20 2f 64 6f 77 6e 6c %..X..GE T /downl
0040 6f 61 64 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e oad.html HTTP/1.
File: "F:\ep\2009-2010\info3_retele_2009-2010\... Packets: 43 Displayed: 43 Marked: 0 Profile: Default
start wrs_capture C2_info3 - Mi... adrs - Micros... attachment:... http.cap - Wi... EN I.R.I.S. 9:13 PM


http.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 145.254.160.237 65.208.228.223 TCP tip2 > http [SYN] Seq=0 Win=8760 Len=
2 0.911310 65.208.228.223 145.254.160.237 TCP http > tip2 [SYN, ACK] Seq=0 Ack=1 Wi
3 0.911310 145.254.160.237 65.208.228.223 TCP tip2 > http [ACK] Seq=1 Ack=1 Win=966
4 0.911310 145.254.160.237 65.208.228.223 HTTP GET /download.html HTTP/1.1
5 1.472116 65.208.228.223 145.254.160.237 TCP http > tip2 [ACK] Seq=1 Ack=480 Win=6
6 1.682419 65.208.228.223 145.254.160.237 TCP [TCP segment of a reassembled PDU]
Frame 4 (533 bytes on wire, 533 bytes captured)
Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
Internet Protocol, Src: 145.254.160.237 (145.254.160.237), Dst: 65.208.228.223 (65.208.228.223)
Transmission Control Protocol, Src Port: tip2 (3372), Dst Port: http (80), Seq: 1, Ack: 1, Len: 479
Source port: tip2 (3372)
Destination port: http (80)
[Stream index: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 480 (relative sequence number)]
Acknowledgement number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x18 (PSH, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
...1 .... = Acknowledgement: Set
.... 1... = Push: Set
.... .0.. = Reset: Not set
.... ..0. = Syn: Not set
....0 = Fin: Not set
Window size: 9660
Checksum: 0xa958 [validation disabled]
[SEQ/ACK analysis]
0000 fe ff 20 00 01 00 00 00 01 00 00 00 08 00 45 00 .. ..E.
0010 02 07 0f 45 40 00 80 06 90 10 91 fe a0 ed 41 d0 ...E@... ......A.
0020 e4 df 0d 2c 00 50 38 af fe 14 11 4c 61 8c 50 18 ...,.P8. ...La.P.
0030 25 bc a9 58 00 00 47 45 54 20 2f 64 6f 77 6e 6c %..X..GE T /downl
0040 6f 61 64 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e oad.html HTTP/1.
File: "F:\ep\2009-2010\info3_retele_2009-2010\... Packets: 43 Displayed: 43 Marked: 0 Profile: Default
start wrs_capture C2_info3 - Mi... adrs - Micros... attachment:... http.cap - Wi... EN I.R.I.S. 9:14 PM



Anexa 2: Statistica, Notiuni de baza **Sursa:** Stefan Balint, Loredana Tanasie, Statistica - notite de curs, Universitatea de Vest,Timisoara

Definiţia 2.1. **Populaţia** este o colecţie (mulţime) de indivizi, obiecte sau date numerice obţinute prin măsurători ale cărei proprietăţi trebuiesc analizate.

Remarca 2.1. Populaţia este colecţia **completă** de indivizi, obiecte sau date numerice obţinute prin măsurători care prezintă interes (pentru cel care colectează eşantionul). Conceptul de populaţie este fundamental în statistică. Populaţia trebuie definită cu grijă şi se consideră complet definită dacă lista membrilor este specificată. Mulţimea studenţilor Facultăţii de Matematică şi Informatică este o populaţie bine definită.
Dacă auzim cuvântul populaţie de obicei ne gândim la o mulţime de oameni. În statistică populaţia poate fi o mulţime de animale, de obiecte fabricate sau de date numerice obţinute prin măsurători. De exemplu mulţimea "înălţimilor" studenţilor facultăţii de Matematică şi Informatică este o populaţie.

Definiţia 2.2. Eşantionul este o submulţime a unei populaţii.

Remarca 2.2. Un eşantion constă din indivizi, obiecte sau date măsurate selectate din populaţie (de către colectorul de eşantion).

Definiţia 2.3. O variabilă de răspuns (simplu **variabilă**) este o caracteristică (de obicei numerică) care prezintă interes în cazul fiecărui element (individ) al unei populaţii.

Remarca 2.3. Vârsta studentului, media lui, culoarea părului, înălţimea, greutatea ş.a.m.d. sunt variabile de răspuns în cazul populaţiei: studenţii de la Facultatea de Matematică şi Informatică.

Definiţia 2.4. O dată (la singular) este "valoarea" unei variabile de răspuns în cazul unui element al populaţiei sau eşantionului.

Exemplul 2.1. Popescu Nicolae are vîrsta de "19 ani", media 8.50, părul lui este "castaniu", înălţimea lui este "1 m şi 75 cm", iar greutatea lui este "65 kg". Aceste cinci "valori" ale celor cinci variabile de răspuns (Remarca 2.3) în cazul lui Popescu Nicolae sunt "cinci" date.

Definiţia 2.5. "Valorile" unei variabile de răspuns în cazul unei populaţii sau a unui eşantion constituie **un set de date** . Într-un set de date aceeaşi dată apare de atâtea ori de câte ori variabila are această "valoare".

Exemplul 2.2. Cele 25 de înălţimi în cazul unui eşantion de 25 de studenţi este un set de 25 de date nu neapărat diferite.

Definiţia 2.6. O activitate planificată în urma căreia se obţine un set de date se numeşte **experiment** sau **sondaj**.

Definiţia 2.7. Parametru este o caracteristică numerică a unei populaţii.

Exemplul 2.3. Procentul de studenţi de la Facultatea de Matematică şi Informatică care au promovat toate examenele la sesiunea din iarnă este un exemplu de parametru în cazul populaţiei: studenţii de la Facultatea de Matematică şi Informatică.

Remarca 2.4. Parametrul este o valoare numerică care se referă la întreaga populaţie. În statistică se obişnuieşte ca parametrul să fie notat cu literă grecească.

Definiţia 2.8. O statistică este o caracteristică numerică a unui eşantion

Exemplul 2.4. Înălţimea medie găsită folosind cele 25 de înălţimi în cazul unui eşantion de 25 de studenţi este un exemplu de statistică (de eşantion).

Remarca 2.5. O statistică este o valoare numerică care se referă la un eşantion. Statisticile (de eşantion) se notează cu literele alfabetului latin.