LAB WORK 06. WIRESHARK. NETWORK TRAFFIC CAPTURE AND ANALYZE

1. LAB WORK GUIDELINES

Disclaimer. This document was created on the basis of the textbook «Data Communication and Networking», 5th.Edition, 2011, -1269 pp., by Behrouz A. Forouzan.

We have created lab assignments for several layers of TCP/IP. We have only theoretical lab assignments for physical layer. We cannot use a standard packet sniffer, such as Wireshark, to capture bits. We cannot sniff management packets because we have normally no permission to act as a manager. In this document, we give an introduction to packet sniffing, introduce the Wireshark software, talk about the lab reports, and finally tell you what to do in this lab assignment.

1.1 PACKET SNIFFING

The purpose of lab assignments is to show how we can get a deeper understanding of the networking concepts by capturing and analysing the packets sent and received from our host. One way to do so is to use a packet sniffer. A packet sniffer is a piece of soft- ware that should be running in parallel with the application whose packets needed to be analysed. However, before running a packet sniffer, we need to interpret the term packet. As we discussed in Chapter 1 of the textbook, communication via the Internet is done using a five-layer suite. We can analyse the packets at four layers: application, transport, network, and data-link. There is no packet exchange at the physical layer; communication at this layer is done using bits.

Although it is useful to analyse the packets in each of the four upper layers of the TCP/IP protocol suite, should a packet sniffer software be designed to capture packets at each of these layers? The answer to this question can be found in Figure 2.8 in the textbook (encapsulation-decapsulation). In an outgoing situation, a packet created at any upper-layer is encapsulated in a frame (at the data-link layer); in an incoming situation, a packet intended for any layer is decapsulated from the received frame. This means we need to capture only outgoing or incoming frames; a packet-sniffer software can extract the packets at any layer desired to be analysed from these frames. For this reason, a packet-sniffer software is normally having two components: a packet-capturer and a packet-analyser. The packet-capturer captures a copy of all outgoing and incoming frames (at the data-link layer) and passes them to the packet-analyser. The packet- analyser can then extract different headers and the ultimate message for analysis.

Before we continue with our discussion, we need to make a point clear. Although Figure 2.8 in the textbook shows that the encapsulation starts or decapsulation ends at the application layer, a packet in the Internet can belong to any layer above the

data- link layer. As we will see in future chapters, protocols at the transport or network layer protocols also need to exchange packets. All of these packets are encapsulated in or decapsulated from the frames. A packet sniffer needs to capture all incoming and out- going frames and show the headers of all protocols used for communication. The source or the sink of a packet is not necessarily the application layer. Figure 1.1 shows two examples.

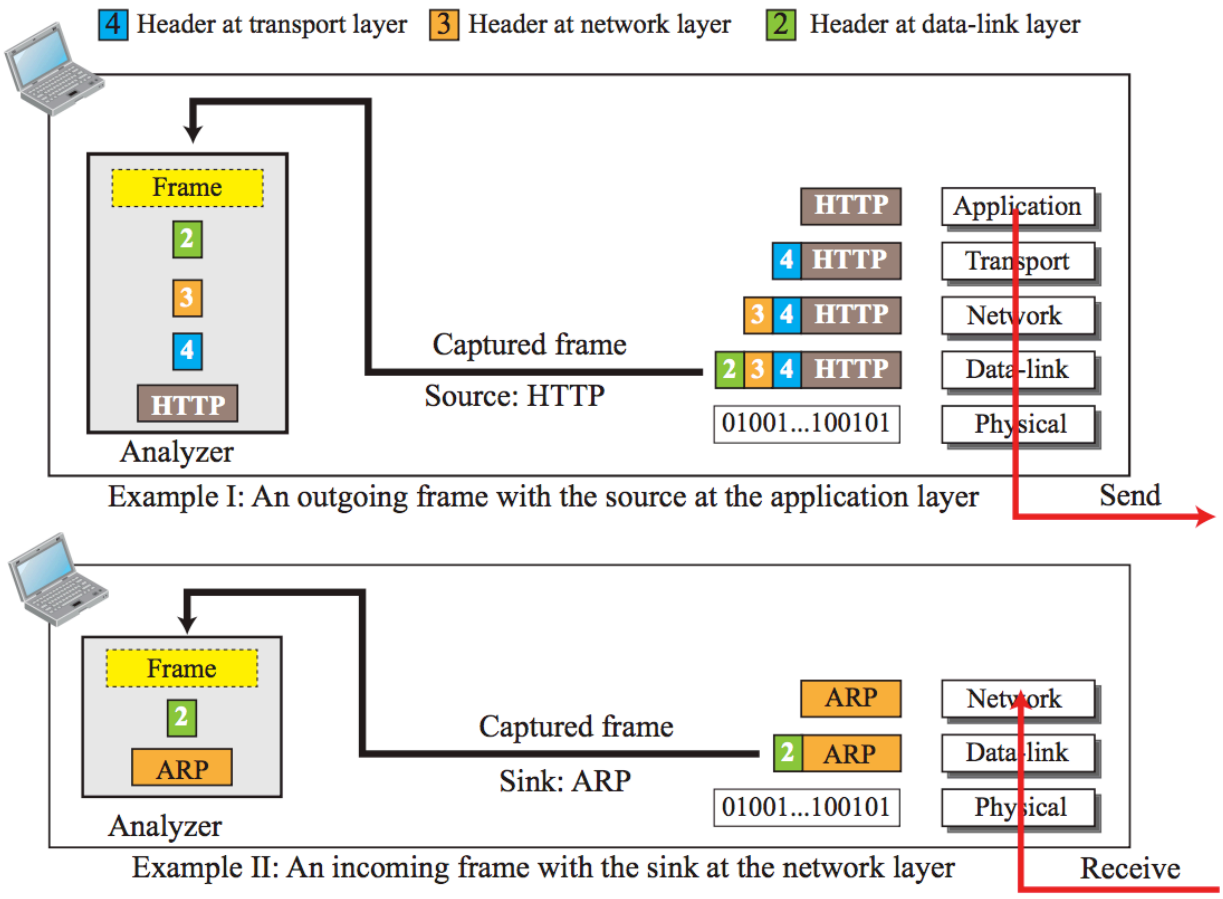

Figure 1.1 Role of frame capturing and packet analysing in a packet-sniffer

In Example I, an outgoing frames is captured. The source of the frame is the HTTP protocol at the application layer (discussed in Chapter 26 of the textbook). A copy of the frame is passed to the analyser. The analyser extracts the general information in the frame (the box marked frame), headers 2, 3, and 4, and the HTTP message for analysis. In Example II, an incoming frame is captured. The sink (final destination) is the ARP protocol at the network layer (discussed in Chapter 9 in the textbook). A copy of the frame is passed to the analyser. The analyser extracts the general information in the header (the box marked frame), header 2 and the ARP message for analysis.

1.2 WIRESHARK

In this and other lab assignments, we use a packet-sniffer called Wireshark. Wireshark (formerly known as ETHEREAL) is a free packet sniffer/analyser which is available for both UNIX-like (Unix, Linux, Mac OS X, BSD, and Solaris) and Windows operating systems. It captures packets from a network interface and displays them with detailed protocol information. Wireshark, however, is a passive analyser. It only captures packets without manipulate them; it neither sends packets to the network nor does other active operations. Wireshark is not an intrusion-detection tool either. It does not give warning about any network intrusion. It, nevertheless, can help network administrators to figure out what is going on inside a network and to troubleshoot network problems. In addition of being an indispensable tool for network administrators, Wire- shark is a valuable tool for protocol developers, who may use it to debug protocol implementations. It is also a great educational tool for computer-network students who can use it to see details of protocol operations in real time.

1.2.1 Main Window

The Wireshark main window is similar to other GUI tools as shown in Figure 1.2. The Wireshark window is made of seven sections: title bar, menu bar, filter bar, packet list pane, packet detail pane, packet byte pane, and status bar. We briefly discuss the functionality of each section below:

Title Bar

The title bar (like the one in any GUI) shows the title of the window, the closing, maximizing, and minimizing icons.

Menu Bar

The menu bar is made of several pull-down menus and tool bars used in most GUIs. We will use some of these menus in our lab assignments. We can use the File menu to perform some actions on the file itself such as saving and printing. The Capture menu is used to start and capturing frames. The View menu is useful to show or hide some of the sections in the window.

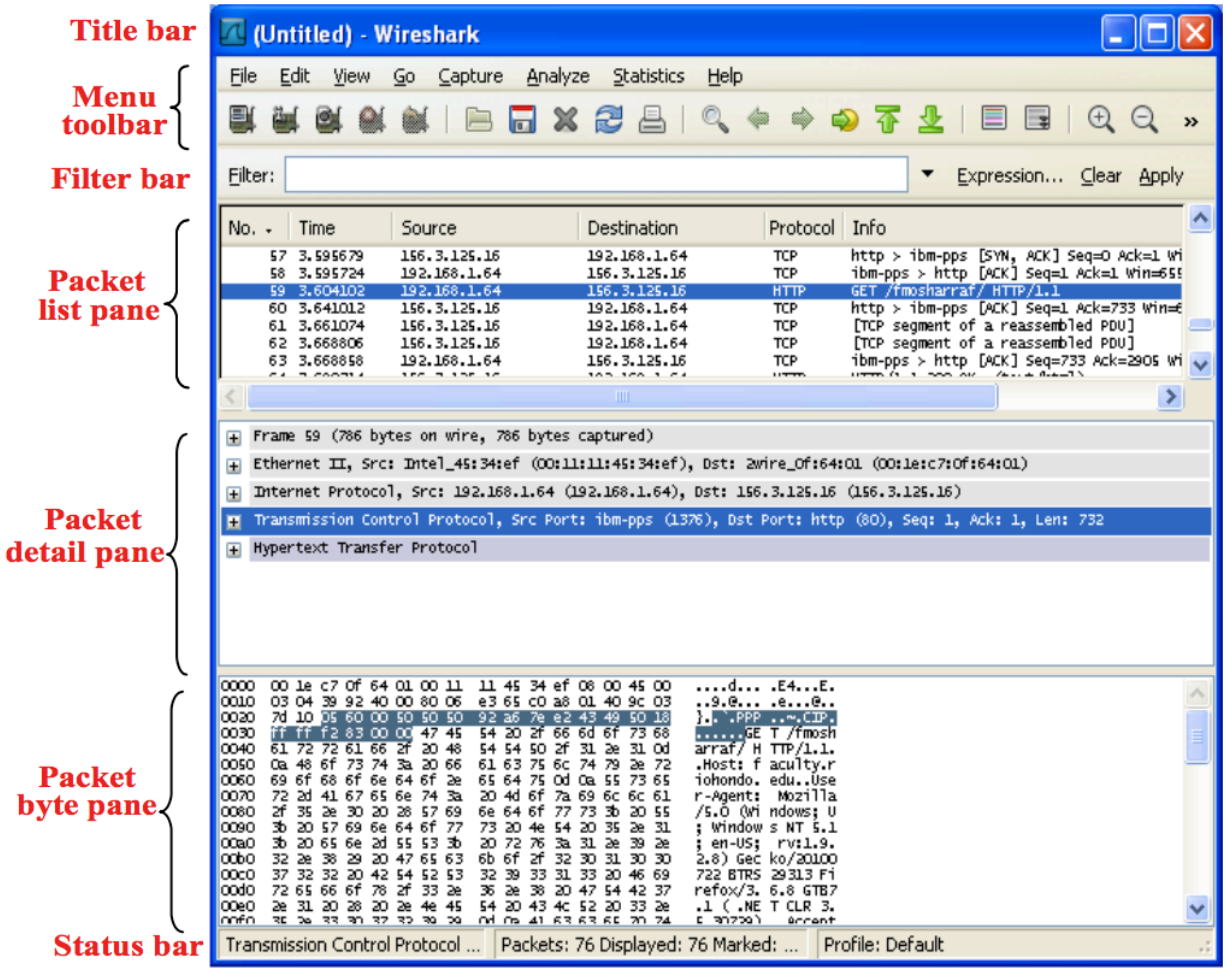

Figure 1.2 Main window of Wireshark

Filter Bar

The filter bar allows us to display packet we are interested in while hiding the rest. As we see later in this document, when we start capturing frames, Wireshark captures and analyse any outgoing and incoming frame no matter what is the source or sink protocol. Sometimes, this is not what we want. We may want to limit the analysis to a specific source or sink protocol. For example, we may want to analyse only packets sent or receive by the HTTP protocol at the application layer or the ARP protocol at the network layer. This is called filtering in the parlance of packet sniffing. After packets have been captured, we can type the name of the protocol in lowercase and click Apply.

Packet List Pane

The packet list pane displays a one-line summary for each captured packet (actually frame). The summary includes the packet (frame) number (added by the Wireshark and not part of the packet), the time when the packet was captured, the source and destination IP addresses of the packet (at the network layer), the packet source or sink proto- col, and the additional information about the packet contents. In other words, this pane shows the captured frames that will be passed for analysing to the packet analyser. For colouring packets use View → Colorize Packet List.

Packet Detail Pane

The packet detail pane shows the detailed analysis for each frame (Figure 1.3). The information is limited to one frame, which means we need to select one of the frames in the packet list pane for analysis. This can be done by clicking on the corresponding frame in the packet list pane. Clicking on any frame in the packet list pane highlights the frame and shows the details of the frame in the packet details pane. Information exhibited in this pane for each frame is made of a tree structure. However, each top branch of the tree is shown as one line as it is common in GUI trees.

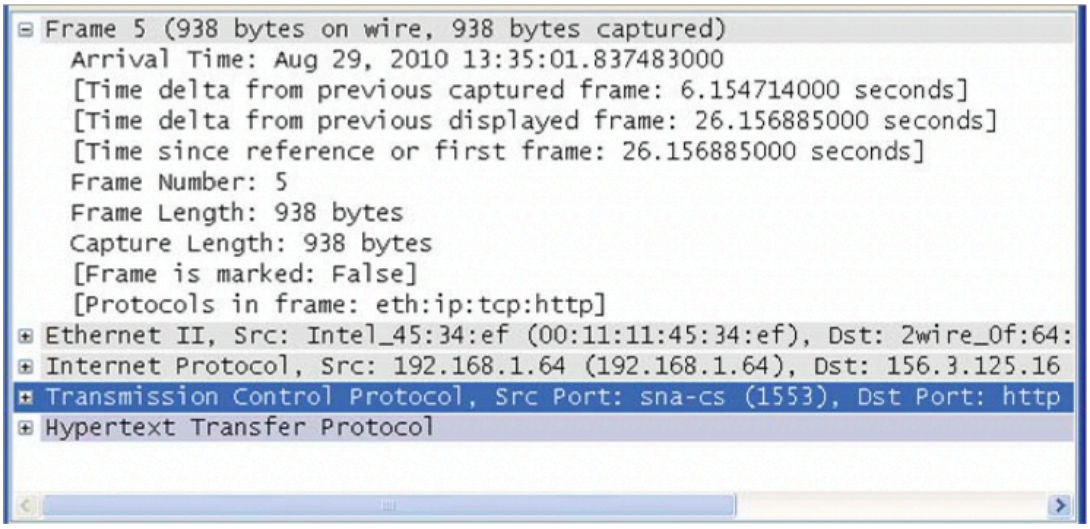

Figure 1.3 Packet detail pane

We can expand the branch (to see sub branches) by clicking on the plus box at the leftmost part of the line, which changes the plus sign to a minus sign; the branch can be collapsed again, which changes the minus sign to the plus sign. Note that the analyser first shows a general information at the data-link layer (frame). It then displays the information contained in each header from the data-link layer (H2) up to the source or sink protocol. It finally shows the whole message at the source or sink layer. Figure 1.3 shows an example of a packet details pane when the frame is expanded. It shows some general information and names of all protocols used in the frame (intermediate and source or sink).

Packet byte pane

The packet byte pane shows the entire current frame (selected in the packet list pane) in hex dump format (hexadecimal view of data) and ASCII format. The number in the left field shows the offset in the packet data; the hex dump of the packet is shown in the middle field; the corresponding ASCII characters are shown in the right field. If we need the byte (or ASCII equivalent) of any line in the packet detail pane, we can click on the line in the packet detail pane and the byte contents will be highlighted. Figure 1.4 shows an example of a packet byte pane. It shows all the bytes in the frame, but we can select the bytes in any protocol header by highlighting it in the packet detail pane section.

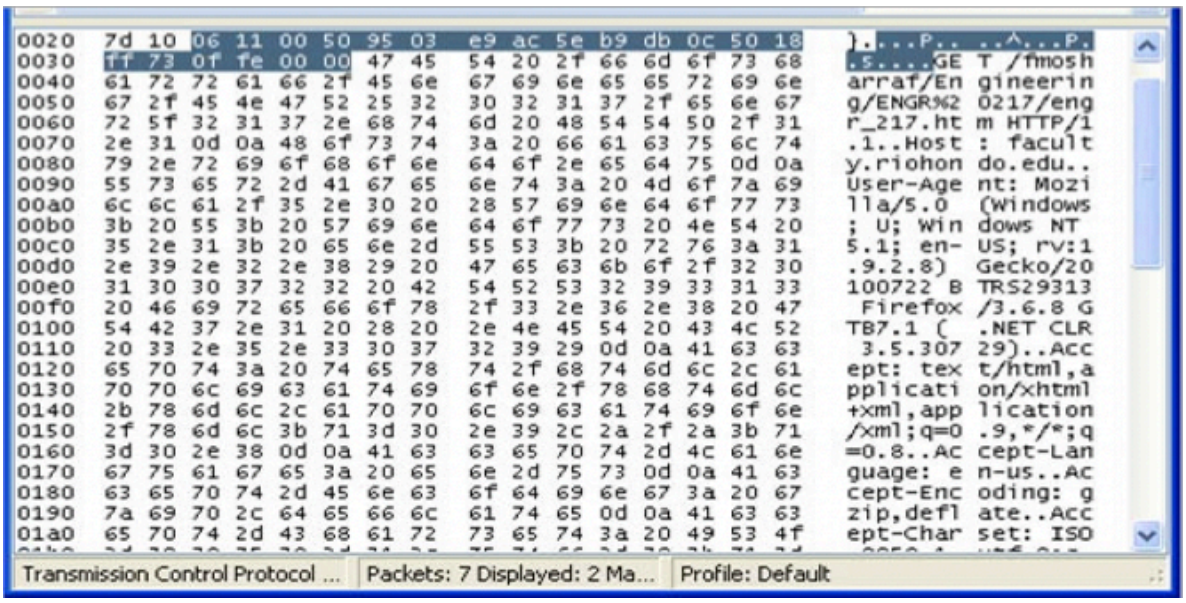

Figure 1.4 Packet byte pane

Status Bar

The last section of the window (at the bottom) is the status bar which shows the current protocol, the total number of packets captured, and so on.

1.2.3 Working with Wireshark

When we work with Wireshark in this and other labs, there are some actions that we need to repeat over and over. We mention the details of some of this action to avoid rementioning them.

Start Capturing

To begin capturing, select the Capture from the pull down menu and click Options… to open the Wireshark capture dialog box.

There are several steps that you need to follow before you start capturing:

1. The network interfaces are shown in the Interface list at the Input box. Select the network interface (or use the default interface chosen by Wireshark). If the IP address in the dialog box is unknown, you must select a different interface; otherwise, the Wireshark will not capture any packet (Figure 1.5).

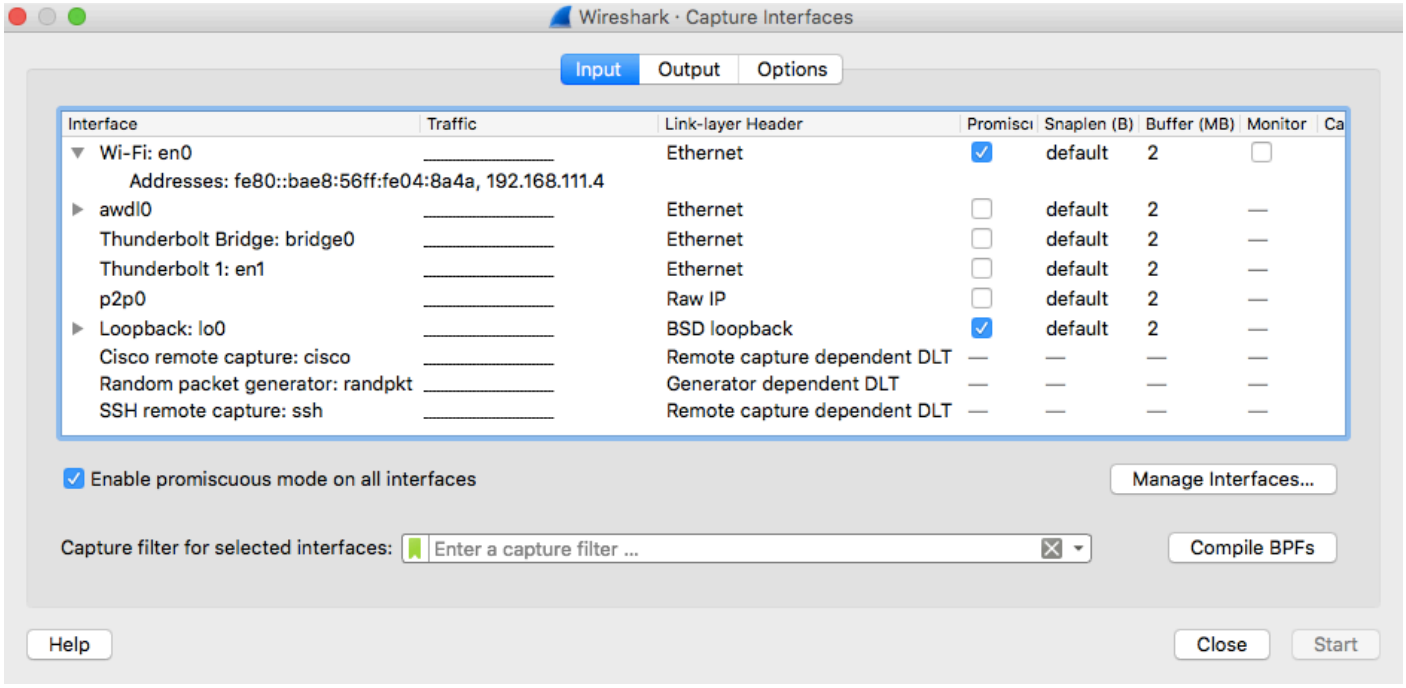

Figure 1.5 Capture Input window

2. It is possible to configure packet filtering using the window Capture Filters (Figure 1.6).

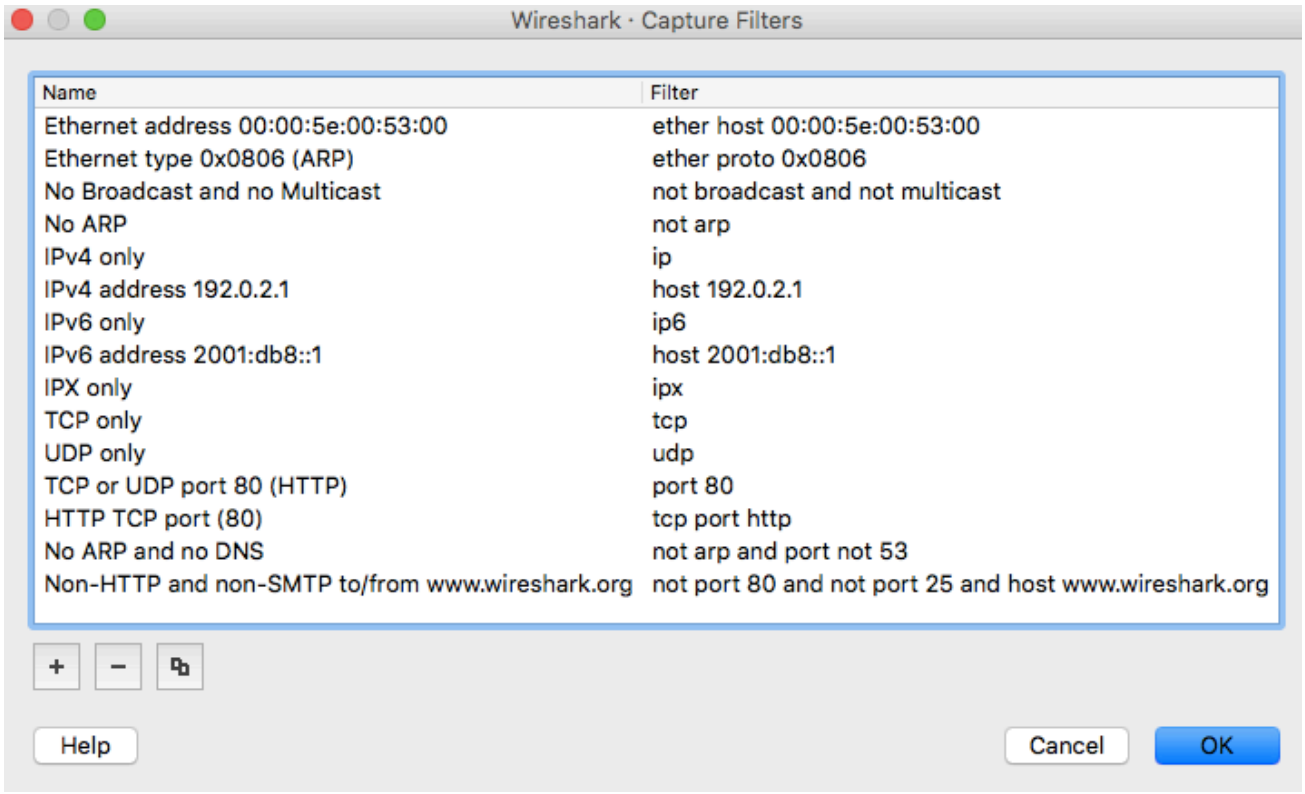

Figure 1.6 Capture Filters window

3. You normally will use the default values in the capture options dialog box, but there are some options that you may need to override the default (Figure 1.7).

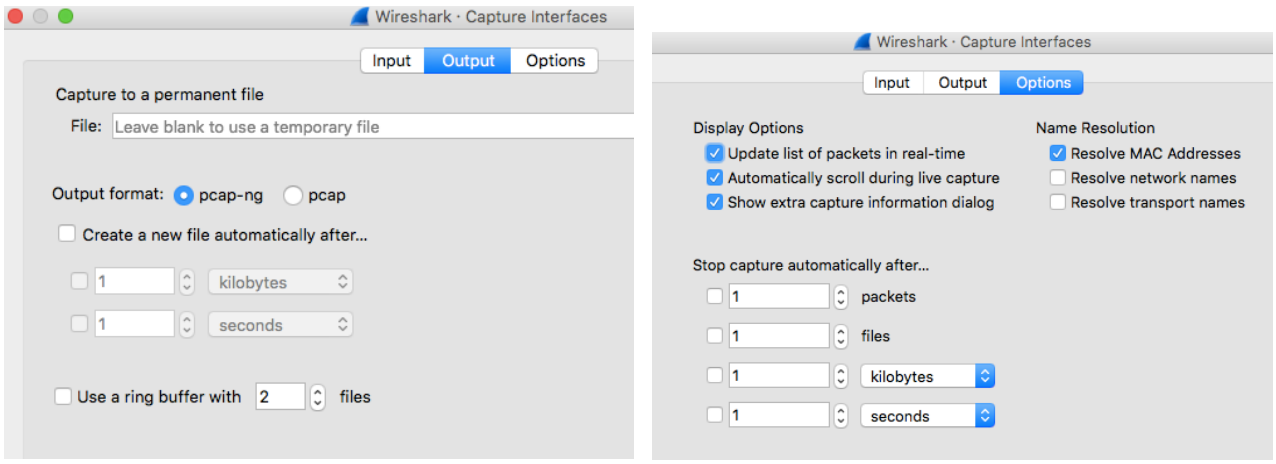

Figure 1.7 Capture Output and Options window

After the above three steps, click Start. Wireshark starts to captures packets that are exchanged between your computer and the network. If, after a minute, Wireshark does not capture any packet, there must be a problem; check for possible reason and troubleshooting.

Stop Capturing

Whenever you feel you have captured all the packets (frames) that you need to do your lab report, you can stop capturing. To do so, you need to use the Capture pull-down menu and click Stop. Wireshark stops capturing the frames.

Saving the Captured File

After you have stopped capturing, you may want to save the captured file for future use.

1.2.4 Incoming and Outgoing Frames

When we see the list of the captured frames, we often wonder which frames are the incoming and which ones are outgoing. This can be found by looking at the frame in packet list pane. The packet list pane shows the source and destination addresses of the frame (generated and inserted at the network layer). If the source address is the address of the host you are working with (shown on the Capture window when you start capturing), the frame is the outgoing frame; if the destination address is the address of your host, the frame is the incoming frame.

1.2.5 Analyze and Statistics

In addition, Wireshark has several convenient and useful functions. For example:

1. View → Coloring Rules (see Figure 1.8)

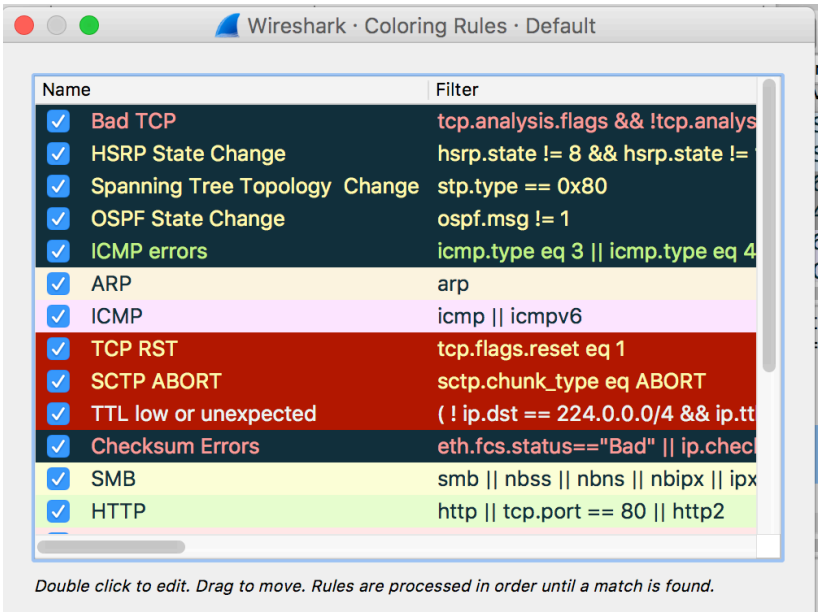

Figure 1.8 Coloring Rules window

2. Analyze → Expert Information (see Figure 1.9) will show a list of the main events that occurred during the capture - the opening of new sessions, not quite good protocol behaviour (repeated receipts in TCP, segment retransmissions, etc.).

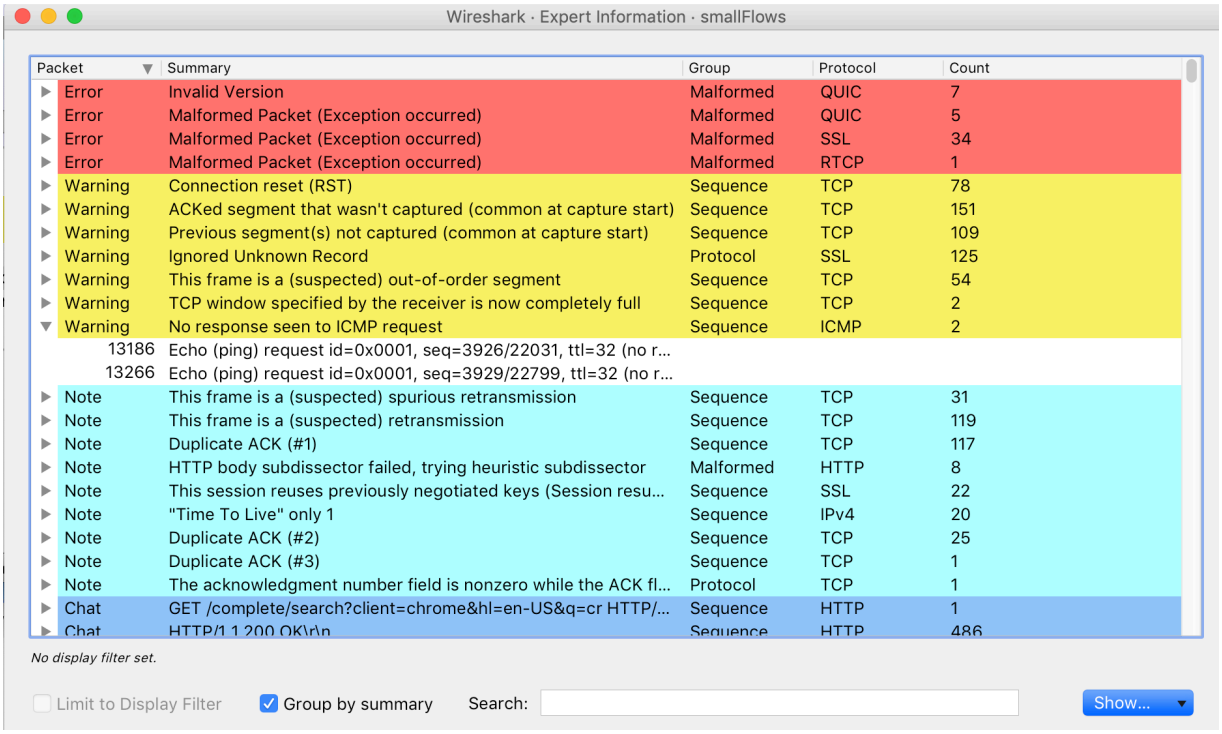

Figure 1.9 Expert Information window

3. Statistics → Capture File Properties allows you to view some statistics for the capture session in general - including the average number of packets per second and the amount of data transferred (Figure 1.10).

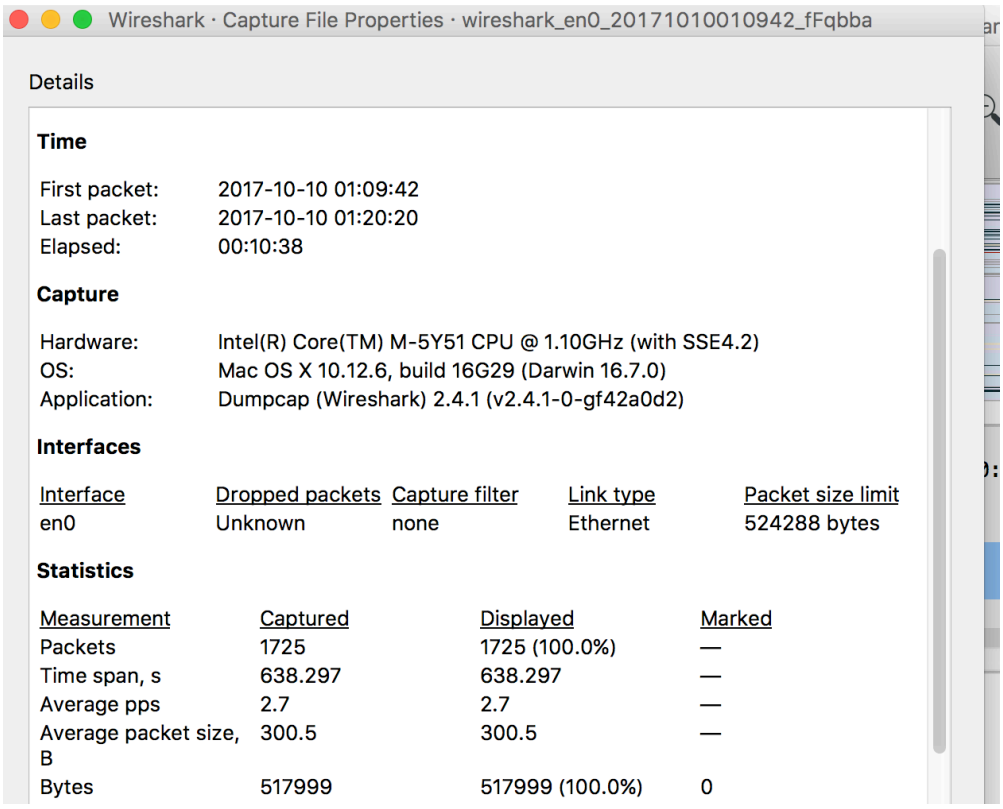

Figure 1.10 Capture File Properties window

4. Statistics → Protocol Hierarchy - statistics on the protocols used (Figure 1.11).

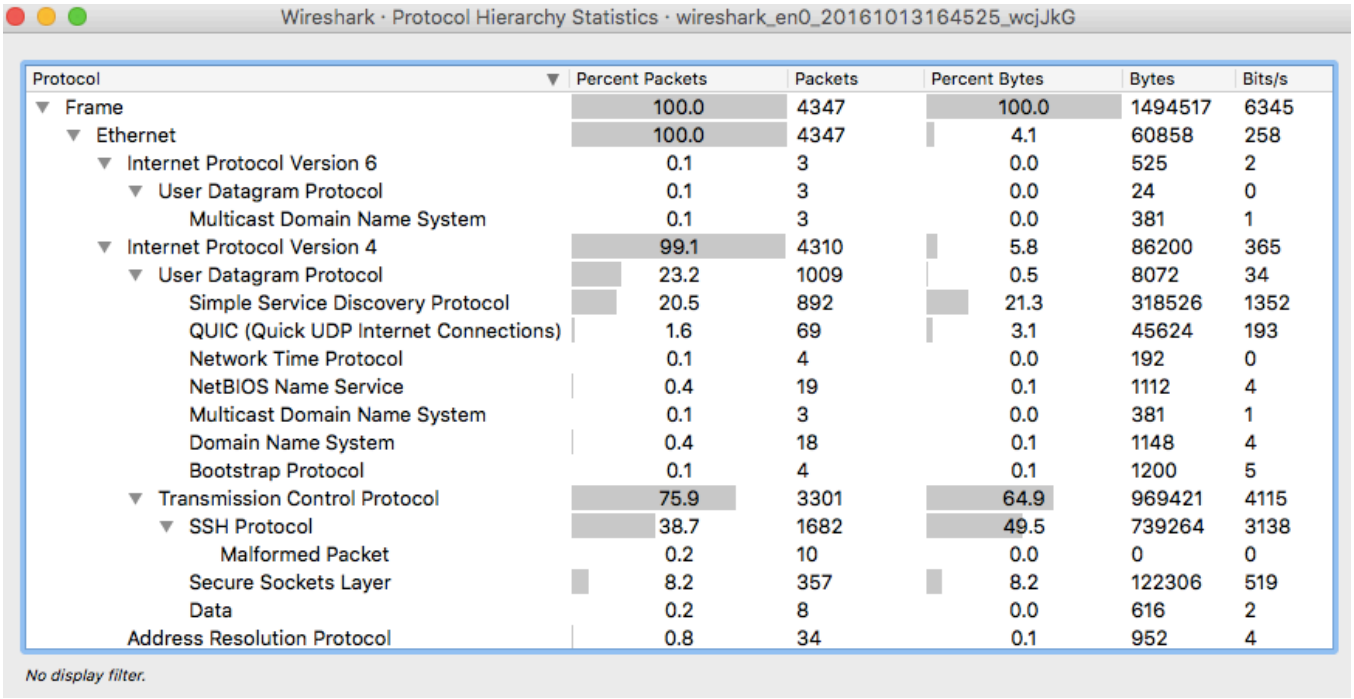

Figure 1.11 Protocol Hierarchy window

5. Statistics → Conversations shows information about the participants in the communication, who sent packets and data to whom to whom and how (Figure 1.12).

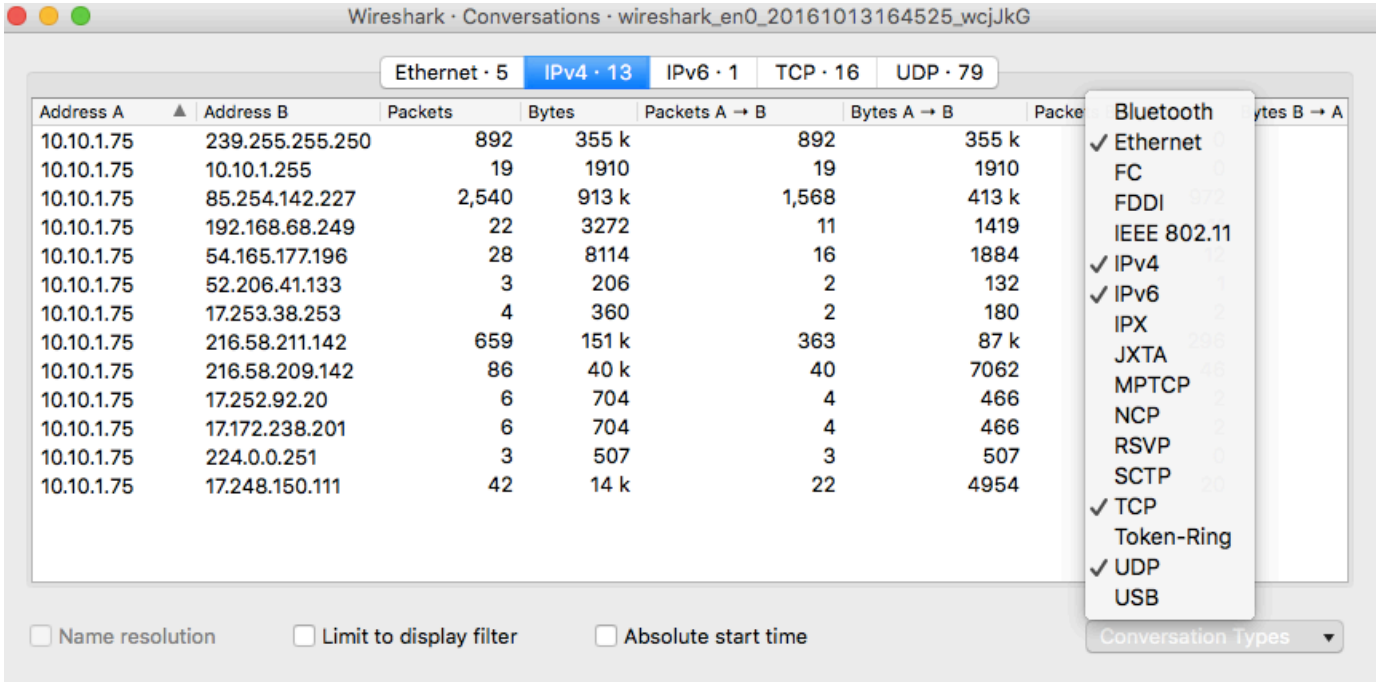

Figure 1.12 Conversations window

6. Statistics → IO Graphs allows to you build an almost arbitrary statistical graph of the captured data (Figure 1.13).

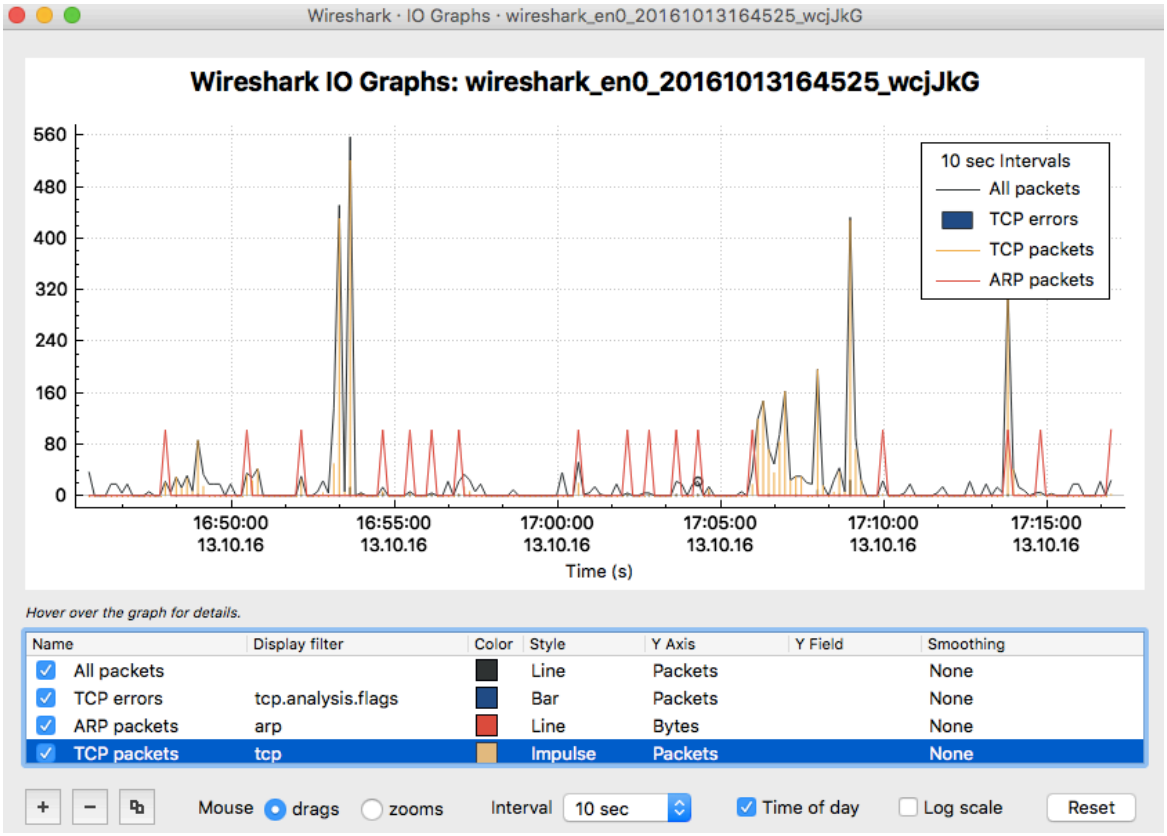

Figure 1.13 IO Graphs window

7. Statistics → Packet Lengths allows to you finding a very short and very long frames (Figure 1.14).

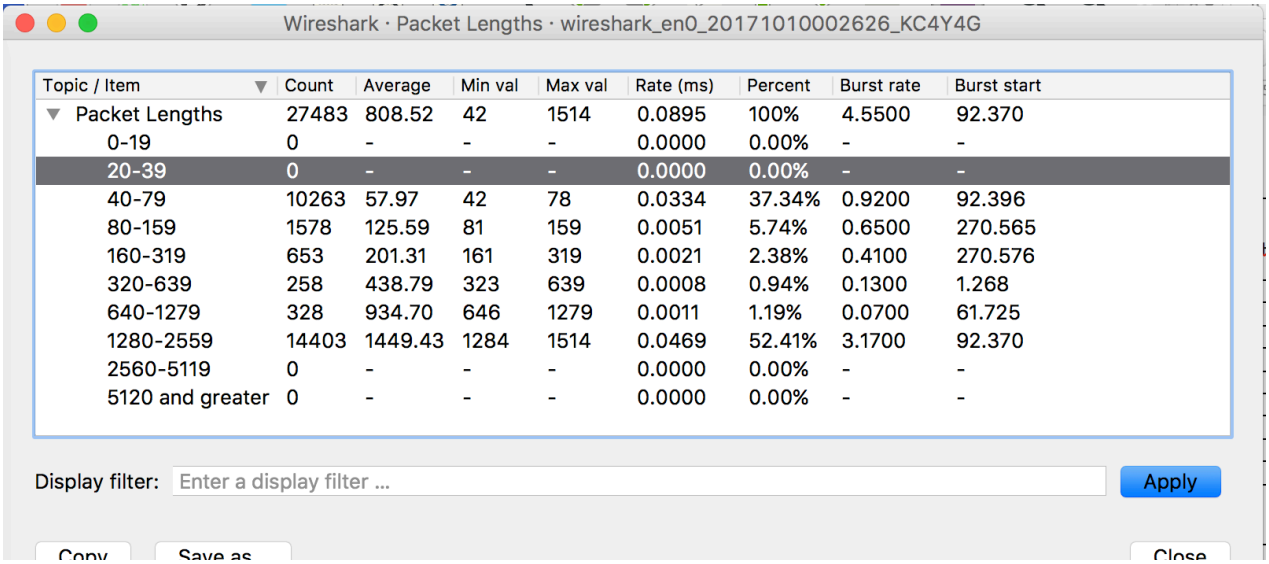

Figure 1.14 Packet Lengths window

8. File → Exporting Objects from HTTP, SMB, SMTP, TFTP Traffic. (Figure 1.15).

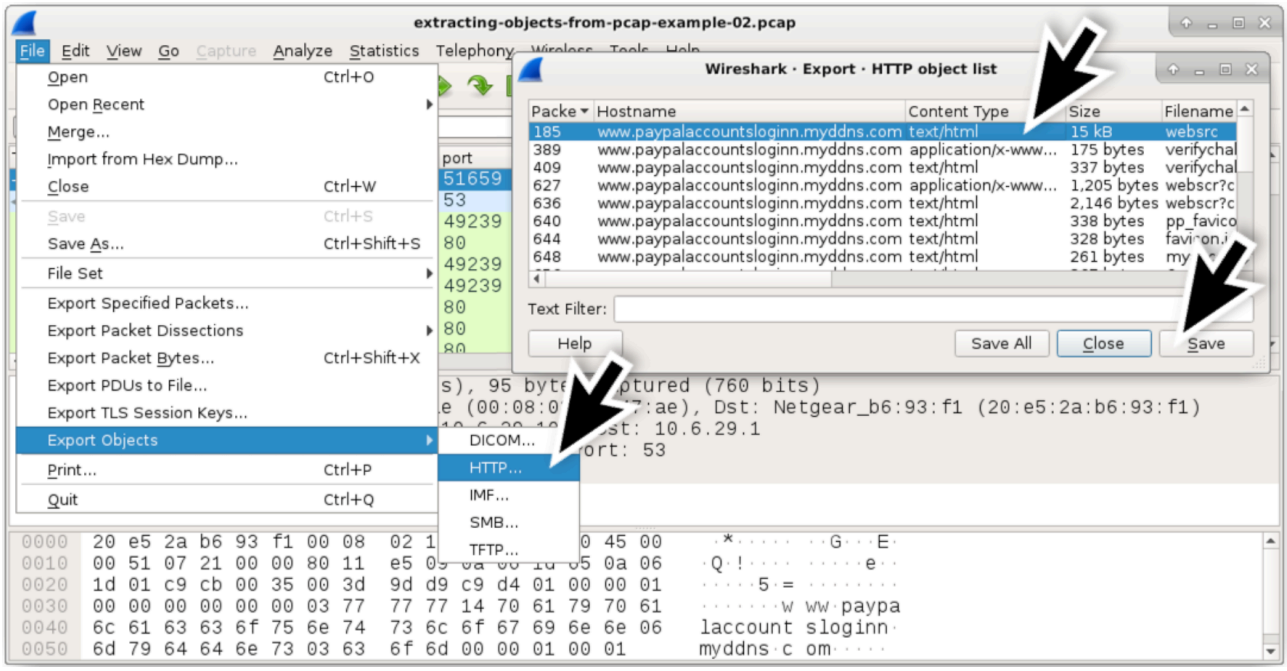

Figure 1.15 Exporting Objects window

9. Analyze → Follow (TCP | UDP | SSL) Stream - allows you to assemble the transfer session together and view its contents as a whole - until the restoration of the HTML page transmitted during the session.

2. LAB WORK ASSIGNMENT

2.1. Lab Target

Get the skills of monitoring and analysis of networks using sniffing programs Wireshark and Network Miner. The lab assignment for this chapter is a warm-up testing of the Wireshark software. In this lab, we retrieve a web page and then, using Wireshark, capture packets.

2.2. Lab Assignment

2.2.1. Downloading and Installing Wireshark

To download and install the Wireshark software, connect to the Internet using the site: http://www.wireshark.org/download.html

If you have any problem in downloading or installing, you can consult the following site: http://wiki.wireshark.org/CaptureSetup

We recommend use the Wireshark Portable Version from:
http://net.academy.lv/soft/WiresharkPortable.zip

2.2.2. Download and open example capturing file

Download the proposed .pcap file to Wireshark with the previously collected network traffic: from the archive http://net.academy.lv/soft/pcap.zip take the **smallFlows.pcap** file.

2.3. Lab-Report Sheets

To make the report of your observation easier and consistent, we have created lab report sheets for each lab assignment.

2.4. Printing & Saving the Captured Information

As a supporting document for each lab assignment, you need to turn in a printout of the captured information. You can do this by selecting the packet and expanding it in the packet detailed pane, selecting Print from the File menu or make **Screen Shots**.

2.5. Documents to Turn in

Turn in the following documents:

1. A copy of the Lab Works Report sheets that contains answered questions.
2. A printout (or Screen Shots) of the supporting captured information.
3. Report send to teachers e-mail.

2.6. Grade.

Grade on 10 points: correctly made of all 6 assignments.

3. LAB WORK REPORT

LAB WORK 06 REPORT:
WIRESHARK. NETWORK TRAFFIC CAPTURE AND ANALYZE.

Student Name Surname:	Student ID:	Date:
		DD.MM.YYYY

Use **smallFlows.pcap** from the archive http://net.academy.lv/soft/pcap.zip

3.1. Capture File Properties

Fill in the table. For initial data use the **Statistics/Capture File Properties**.

Nr	Parametr	Value
1	Time of capture, min	
2	Packets	
3	Bytes, MiB	
4	Average packet size, B	
5	Average packets per seconds, pps	
6	Average bytes per seconds, B/s	

7. Determine the relative network load L (in%) for the control period T by formula:
L = (Traffic [Mbits] / T [sec]) / (Bandwidth [Mbits/sec])
Bandwidth = 100 Mbits/sec
L = Your Ansver

3.2. Ethernet Traffic Distribution by Protocols

Fill in the table. For initial data use the **Statistics/Protocol Hierarchy**.

Nr	Protocol	Traffic, MiB	Traffic, %
1	IPv6		
2	IPv4		
3	--UDP		
4	--TCP		
5	--ICMP		
6	ARP		
7	802.1X		
	SUMM		**100**

8. What is the ratio of the numbers of application (http, mail, ftp, …) to numbers of service (dns, icmp, arp, …) protocols?

Anr / Snr = Your Ansver

3.3. Ethernet Traffic Distribution by Nodes

Fill in the table (for the 5 most active network nodes by Bytes). For initial data use the **Statistics/Endpoints/Ethernet**.

Nr	MAC-address	IP- address	Traffic					
			Rx input		Tx output		Overall	
			MiB	%	MiB	%	MiB	%
1.								
2.								
3.								
4.								
5.								
		SUM		**100**		**100**		**100**

6. Which IP nodes are the most loaded, given the direction of traffic?
Incoming –
Outgoing –
Overall –

3.4. Display Filters

Fill in the table. Write and test in Wireshark 5 simple search filters (Display Filters) using AND, OR, NO to display packets from (to) a specific node generated by ICMP, DNS, ARP requests (responses) when accessing any server of your choice.

Nr	Display Filter	Description
1		
2		
3		
4		
5		

3.5. Network Problem Analyze

Analyze the 5 note/warning/error problems existing on the network. Find and read information about network problems on the Internet.
For initial data use the **Analyze/Expert Information**.

Nr	Expert Information	Severity	Your Short Description (Problem Analyse)
1	Connection reset (RST)	Warning	
2	TCP keep-alive segment	Note	
3	...	Error	
4			
5			

3.6. Exporting File from Traffic Stream

Tasks.
You need to find and export the N-th (by size) JPEG file, where N is your number in the class list (the 1st by number takes the largest file, the 2nd takes the next, etc.).
Determine the starting packet number, source IP, destination IP, Jpeg file size.
Paste this image into your Report

Instruction.
For initial data use **File / Export Object / HTTP ...** / sort information by file size (large to small) / find the N-th file content-jpeg / read interesting information / save this file.

Answer.
1. Your variant Nr: ??
2. Starting packet number:
3. Source IP:
4. Destination IP, jpeg file size:
5. Jpeg file size:
6. Picture

Jpeg-file Picture

4. EXTEND LAB WORK ASSIGNMENT

Homework assignment (for funs only).

1. Get to know the capabilities of Network Miner. Download the .pcap file proposed by the teacher with the previously collected network traffic to Network Miner. Analyze the collected traffic in Network Miner.
2. Install Wireshark on your home computer.
3. Launch Wireshark in the mode of capturing traffic passing through an interface connected to the local network (usually eth0).
4. Emulate network activity for 10-15 minutes from various home nodes. To do this, you can perform, for example, some of the following actions.

    - Open the website http: // ...;
    - Connect to the ftp server;
    - Connect to the mail server;
    - Connect to the ssh server;
    - Ping any nodes;
    - Connect to one of the available Windows network drives (if such resources are available on the network);
    - Perform other actions that require a network connection.

5. Stop capture, save the pcap file and attach it to the report (if the file is larger than 10 MiB, then have it on a flash drive while protecting the laboratory work.
6. The remaining items are the same as 1 to 5 for Core Lab Work.
7. Make a report in electronic form.