Department of Electrical Engineering, UET Lahore
EE432: Computer Networks

Course Instructor: Dr. Naveed Nawaz	**Dated:**
Session: Fall 2024	**Semester: 7th**

LAB 1 Wireshark: A Network Protocol Analyzer

Name	Roll. No.	Report Marks (10)	Viva Marks (5)	Total Marks (15)
Ayesha Ahmad	2021-EE-052			

Checked on: _____

Signature: _____

Introduction to Wireshark

1.1. Objectives

This lab will enable the students to achieve the following:

- Familiarize themselves with the Wireshark environment
- Learn how to capture packets of network tranffic
- Browse interactively the traffic running on a computer network

1.2. Instructions

1. Read manual carefully before starting lab.
2. All exercises are individual exercises
3. You are supposed to provide the answers to the questions listed at the end of this manual in text. Paste screenshots/images in the textboxes where required. You will be required to submit your complete manual on Google classroom.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/text but you must paraphrase the original work. Your submitted work should be written by yourself.
5. You must aim to complete the lab half an hour before the lab time ends.
6. At the end of the lab, a viva will be conducted to evaluate your understanding.

1.3. Background

A protocol analyzer is a tool that can be used to inspect what exactly is happening on a network with respect to traffic flow. For example, if your TCP/IP sessions are "hanging", a protocol analyzer can show which system sent the last packet, and which system failed to respond. If you are experiencing slow screen updates, a protocol analyzer can display delta time stamps and show which system is waiting for packets, and which system is slow to respond.

A protocol analyzer can show runaway traffic (broadcast or multicast storms) and its origin, system errors and retries, and whether a station is sending, trying to send, or only seeming to communicate. You will get information that is otherwise unavailable, which results in more efficient troubleshooting and better LAN health.

2.3.1. Computer network

A computer network, often simply referred to as a network, is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information. In the world of computers, networking is the practice of linking two or more computing devices together for the purpose of sharing data. In networking, the communication language used by computer devices is called the protocol. Yet another way to classify computer networks is by the set of protocols they support. Networks often implement multiple protocols to support specific applications.

2.3.2. What is a protocol analyzer?

Protocol analyzers capture conversations between two or more systems or devices. A protocol analyzer not only captures the traffic, it also decodes (interprets) the traffic. Decoding allows you to view the conversation in English, as opposed to binary language. A sophisticated protocol analyzer will also provide statistics and trend information on the captured traffic. Protocol analyzers provide information about the traffic flow on your local area network (LAN), from which you can view device-specific information.

2.3.3. Introduction to Wireshark

Wireshark is a free and open-source packet analyzer, used for network troubleshooting, analysis, software and communications protocol development, and education.

The basic tool for observing the messages exchanged between executing protocol entities is called a packet sniffer. As the name suggests, a packet sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/ received from/by application and protocols executing on your machine.

Figure 1.1 shows the structure of a packet sniffer. At the right of Figure 1.1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in Figure 1.1 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. Recall from the discussion from Section 1.5 in the textbook (Figure 1.20) that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In Figure 1.1, the assumed physical media is an Ethernet, and so all upper layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

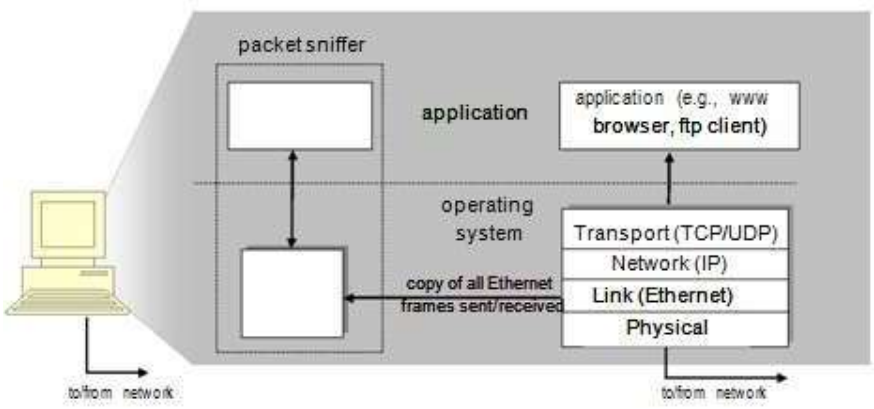


Figure 1.1: Packet sniffer structure

The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 1.1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string "GET," "POST," or "HEAD," as shown in Figure 2.8 in the textbook.

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers. It's an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a user-guide (http://www.wireshark.org/docs/wsug_html_chunked/), man pages (http://www.wireshark.org/docs/man-pages/), and a detailed FAQ (http://www.wireshark.org/faq.html), rich functionality that includes the capability to analyze more than 500 protocols, and a well-designed user interface. It operates in computers using Ethernet, Token-Ring, FDDI, serial (PPP and SLIP), 802.11 wireless LANs and ATM connections (if the OS on which it's running allows Wireshark to do so).

1.3.3.1. Getting Wireshark

In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the libpcap or WinPCap packet capture library. The libpcap software will be installed for you alongside Wireshark automatically. See http://www.wireshark.org/download.html for a list of supported operating systems and download sites

Download and install the Wireshark software:

- Go to http://www.wireshark.org/download.html and download and install the stable release Wireshark 3.0.3 binary for your computer. Wireshark can be installed on both Windows and Linux. See the documentation page of Wireshark for more details.
- Download the Wireshark user guide.

The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.

1.3.3.2. Running on Windows

On Windows, you should be able be able to find the link by clicking on the Start option of the Windows taskbar and thereby finding the wireshark program in All Programs.

On Linux machines, wireshark can be run by typing "wireshark" at the command prompt (in case there is a problem with your path, type /usr/bin/wireshark which is where wireshark is typically installed).

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 1.2 will be displayed. Initially, no data will be displayed in the various windows.

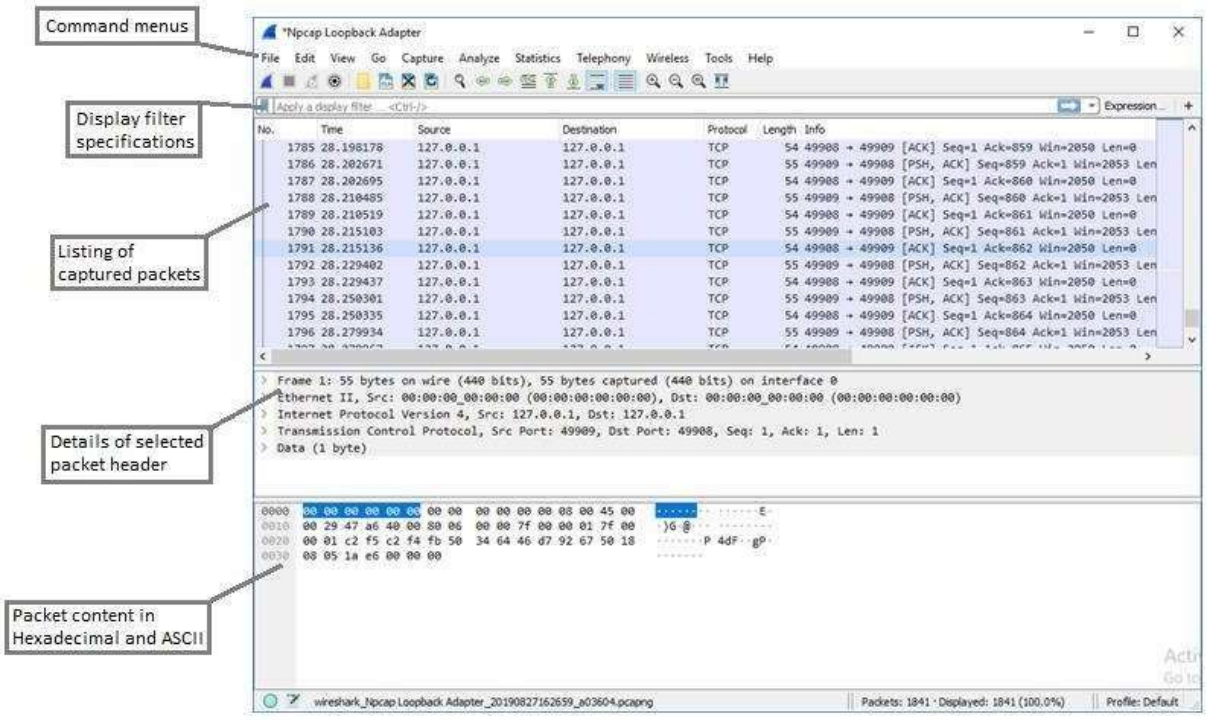


Figure 1.2: Wireshark GUI

The Wireshark interface has five major components:

1. The **command menus** are standard pull down menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.
2. The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information

contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

3. The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.in Wireshark
4. The **packet-contents** window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
5. Towards the top of the Wireshark graphical user interface, is the **packet display filter field**, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

1.4. Lab Procedure

The best way to learn about any new piece of software is to try it out! We'll assume that your computer is connected to the Internet via a wired Ethernet interface. Do the following:

1. **Start up your favorite web browser**, which will display your selected homepage.
2. **Start up the Wireshark software**. You will initially see a window similar to that shown in Figure 1.2 except that no packet data will be displayed in the packet-listing, packet-header, or packet-contents window, since Wireshark has not yet begun capturing packets..
3. **To begin packet capture**, select the Capture pull down menu and select Options. This will cause the "Wireshark·Capture Interfaces" window to be displayed, as shown in Figure 1.3.

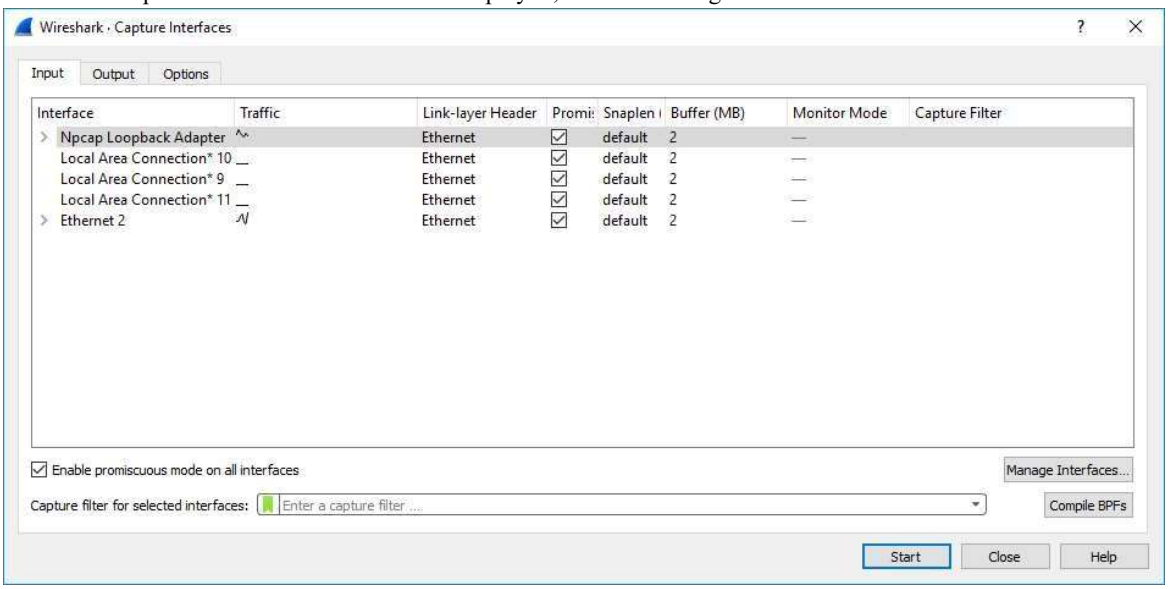


Figure 1.3: Wireshark capture interfaces

4. **Selecting the network interface on which packets would be captured:** Under the "Options" tab, you can use most of the default values, but check "Show capture information during live capture" under Display Options. The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the "Input" tab inside the "Interface" panel. In case your computer has more than one active network interface (e.g., if you have both a

wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (mostly likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin – all packets being sent/received from/by your computer are now being captured by Wireshark!

5. Once you begin packet capture, a packet capture summary window will appear, as shown in Figure 1.4. This window summarizes the number of packets of various types that are being captured, and (importantly!) contains the Stop button that will allow you to stop packet capture. Don't stop packet capture yet.

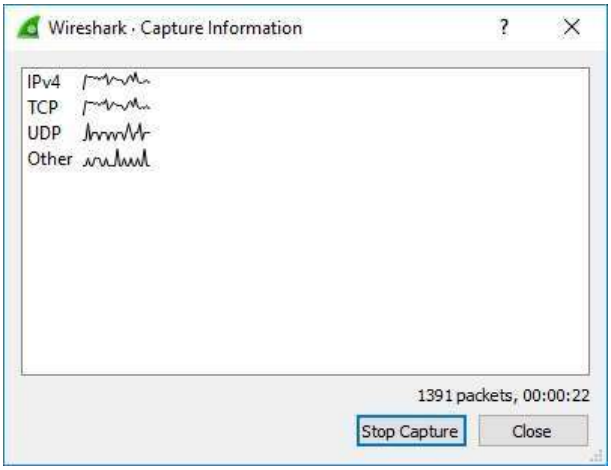


Figure 1.4: Wireshark capture information

6. **Capturing an HTTP interaction on Wireshark:** While Wireshark is running, enter the URL: http://uet.edu.pk/ and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at http://uet.edu.pk, and exchange HTTP messages with the server in order to download this page, as discussed in section 2.2 of the text. The Ethernet frames containing these HTTP messages will be captured by Wireshark.
7. **Stopping the capture and inspecting captured packets:** After your browser has displayed the page, stop Wireshark packet capture by clicking "Stop Capture" in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to Figure 1.2. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the uet.edu.pk web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the Protocol column in Figure 1.2). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user. We'll learn much more about these protocols as we progress through the text! For now, you should just be aware that there is often much more going on than "meets the eye".
   Note: You can answer question 1 of the "Questions" section now.
8. **Filtering:** Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply (to the right of where you entered "http"). This will cause only HTTP message to be displayed in the packet-listing window.
9. **Details of a packet:** Select the first http message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the uet.edu.pk HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window. By clicking on right-pointing and down-pointing arrows heads to the left side of the packet details window, minimize the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximize the amount information displayed about the HTTP protocol. Your Wireshark display should now look roughly as shown in Figure 1.5. (Note, in particular, the minimized amount of protocol information for all protocols except HTTP, and the maximized amount of protocol information for HTTP in the packet-header window).
   Note: You can answer questions 2 and 3 of the "Questions" section now.

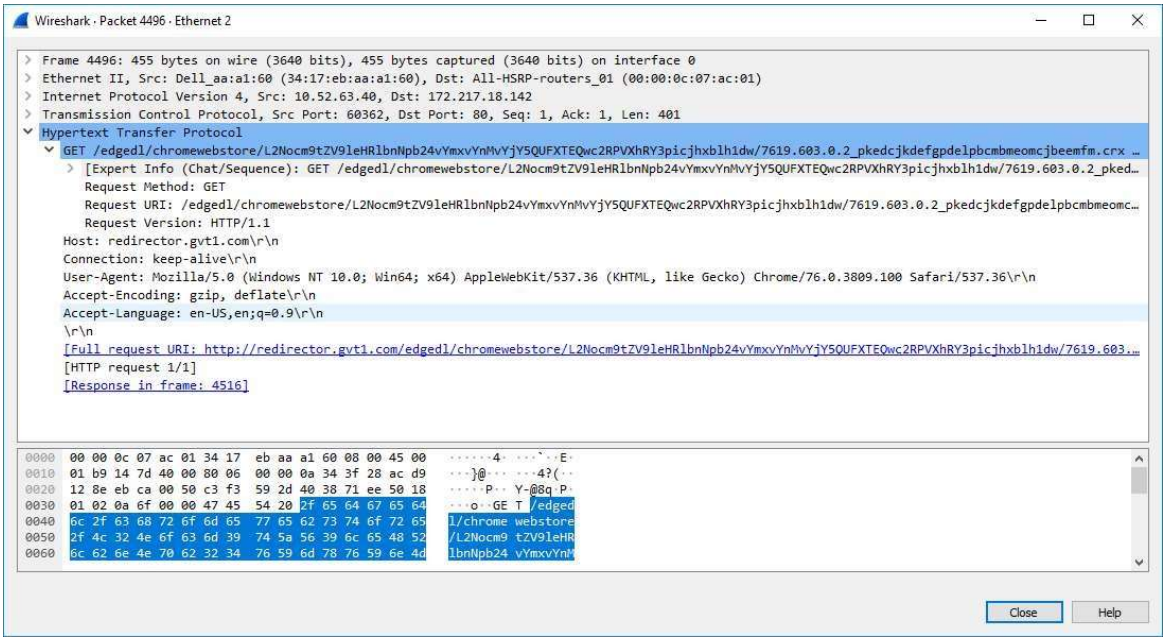


Figure 1.5: Wireshark display after step 9

10. **Statistics of packet captured:** Click on the "Statistics" option on the upper toolbar of Wireshark to explore the various ways in which statistics may be obtained about network traffic.
    Explore specifically the 'Conversation' options in 'Statistics' option on the upper toolbar of Wireshark. We shall be using it to track a conversation of an HTTP flow in future labs.
    Note: You can answer question 5 of the "Questions" section now.
11. **Obtaining credit for this lab:** Now, please proceed to the questions section to answer the questions. You must note down your answers in this file itself. Please note that every student must upload this file (after duly filling in the answers) to classroom. Please clarify with your instructor/ lab engineer if you have any queries.

1.5. Questions

1. List the different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.

[3 marks]

1.	DNS	1.	ICMPv6
2.	MDNS	2.	IGMPv3
3.	TCP	3.	ARP
4.	DHCP	4.	LLMNR
5.	DHCPv6	5.	SSDP
6.	TLSv1.2 and v1.3	6.	HTTP

2. **Finding IP address of your machine in Wireshark:** What is the Internet address (or the IP address) of the 'uet.edu.pk'? What is the Internet address of your computer? How did you find in Wireshark?

[2 marks]

IP Address of computer : 192.168.221.252 IP Address of 'uet.edu.pk' : 59.103.73.225
The IP address of the source and destination are shown in the HTTP GET message.
The HTTP GET message is sent from my computer to the website so the source IP is of my computer and the Destination IP is of 'uet.edu.pk'

No.	Time	Source	Destination	Protocol	Length	Info
81	3.554172	192.168.221.252	59.103.73.225	HTTP	208	GET /connecttest.txt HTTP/1.1
86	3.583064	59.103.73.225	192.168.221.252	HTTP	241	HTTP/1.1 200 OK (text/plain)

3. **Finding IP address of your machine without Wireshark:** Note the IP address of your machine manually by typing ipconfig on the DOS prompt or by typing the command ifconfig on linux machines. Is the IP address of your machine the same as noted in question 2?

[2 marks]

The IP Address of my computer shown with and without Wireshark is the same: 192.168.221.252

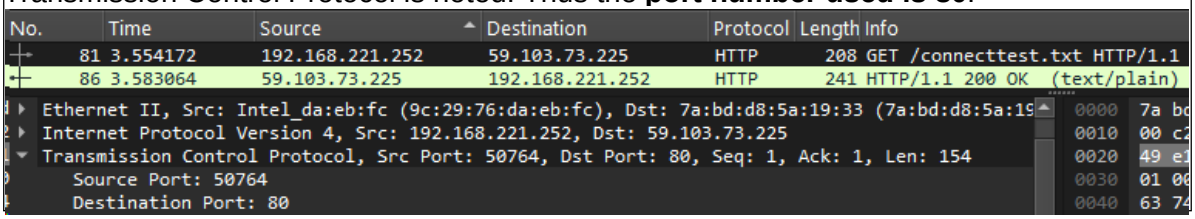
```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::d3f5:33d5:43e6:f979%10
   IPv4 Address. . . . . . . . . . . : 192.168.221.252
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.221.116
```

4. What is the port number used by the HTTP server "uet.edu.pk". How did you note in Wireshark?

[2 marks]

The Destination of the HTTP GET message is "uet.edu.pk" so the Destination Port from the Transmission Control Protocol is noted. Thus the **port number used is 80**.



5. **Delay between request and reply:** How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

[2 marks]

The Time-of-Day for the HTTP GET message and the OK received have the same hour and minute but the second is different.
HTTP OK Time (reply) - 08:53:05.511858
HTTP GET Time (request) - 08:53:05.482966
Delay between request and reply - 00:00:00.028892 which is equal to **28.892ms delay**.

No.	Time	Source	Destination	Protocol	Length	Info
81	08:53:05.482966	192.168.221.252	59.103.73.225	HTTP	208	GET /connecttest.txt HTTP/1.1
86	08:53:05.511858	59.103.73.225	192.168.221.252	HTTP	241	HTTP/1.1 200 OK (text/plain)

6. **Capturing conversations:** Document your interaction with the Conversations option of the Statistics tab on the upper toolbar on Wireshark. Were you able to capture the network conversation you had with UET's HTTP server?

[3 marks]

Yes, the network conversation wth the UET's HTTP server was captured ais shown below.

IPv4 · 1

Address A	Address B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Abs Start	Duration	Bits/s A → B	Bits/s B → A
192.168.221.252	59.103.73.225	2	449 bytes	10	12	16.67%	1	208 bytes	1	241 bytes	08:53:05.443	0.0959	17 kbps	20 kbps

Assessment Rubrics for EE432: Computer Networks Lab 1

Student Name: _____ Roll Number: _____

Method:

Lab report evaluation and instructor observation during lab sessions.

Outcomes Assessed:

a. Ability to condut experiments as well as to analyze and interpret data
b. Ability to adhere to safety and disciplinary rules
c. Ability to use the techniques, skills and modern engineering tools necessary for engineering practice

Performance	Exceeds expectation (5-4)	Meets expectation (3-2)	Does not meet expectation (1)	Marks
Realization of experiment (a)	Downloads and installs required software and sets up the system according to the experiment requirements	Needs guidance to set up the system according to the experiment requirements	Incapable of selecting relevant software to the experiment and unable to setup the system with required software tools	
Conducting experiment (a, c)	Carries out each procedural step in a satisfactory manner and studies outputs of the software application rigorously	Needs assistance or guidance to proceed through experiment steps, studies outputs with minor errors in interpretation	Unable to carry out procedural steps and make any useful observations of outputs	
Laboratory safety and disciplinary rules (b)	Observes lab safety rules; adheres to the lab disciplinary guidelines aptly	Observes safety rules and disciplinary guidelines with minor deviations	Disregards lab safety and disciplinary rules	
Data collection (c)	Completes data collection from the experiment setup by following procedural steps, ensures that the data is entered in the lab manual according to the specified instructions	Completes data collection with minor error and enters data in lab manual with slight deviation from guidelines	Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab manual properly	
Data analysis (a, c)	Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with sound explanation	Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from theory	Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data	