National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Sir Haasaan Khaliq **Dated: 2/24/2023**

Semester:6th **Section: C**

EE-357 Computer and Communication Networks
Experiment - 3

General introduction to Wireshark and networking

		PLO5/ CLO4	PLO9/ CLO5
Name	**Reg. No**	**Modern Tool Usage** **10 Marks**	**Individual and Team Work** **5 Marks**
Muhammad Ahmed Mohsin	**333060**		
Amina Bashir	**343489**		
Imran Haider	**332569**		

1 TABLE OF CONTENTS

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

EXPERIMENT NO 5

General introduction to Wireshark and networking

2 OBJECTIVE OF THIS LAB:

The basic purpose of this lab is to introduce you to Wireshark, a popular protocol analyzer. By the end of this lab you will be familiar to its environment and will know how to capture and interactively browse the traffic running on a computer network using it.

3 INSTRUCTIONS:

1) Read carefully before starting the lab.
2) These exercises are to be done individually.
3) You are supposed to provide the answers to the questions listed at the end of this document and upload this completed document to your course's LMS site.
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.

4 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.

4.1 Introduction to Networking:

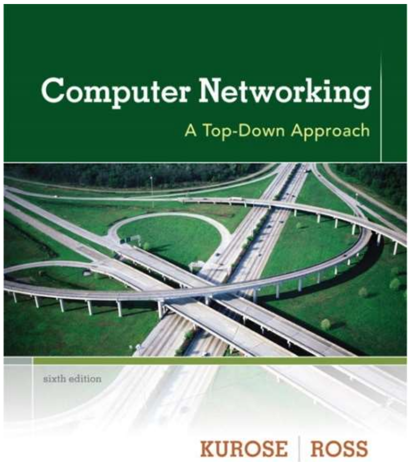
A **computer network**, often simply referred to as a network, is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information. In the world of computers, networking is the practice of linking two or more computing devices 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.

4.2 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.

4.3 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 shows the structure of a packet sniffer. At the right of Figure 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 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. 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, 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.

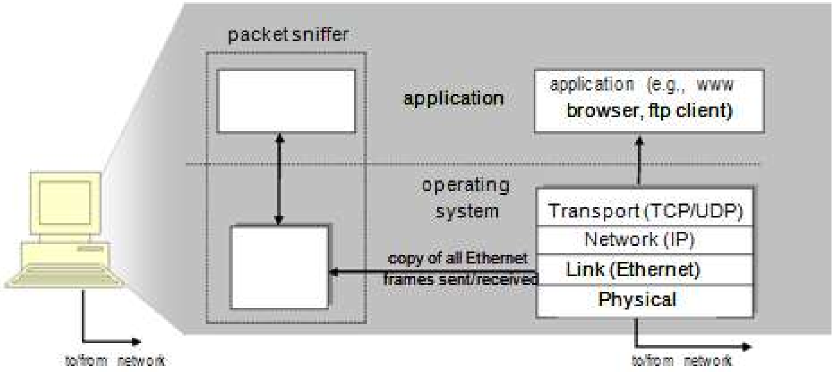


Figure 1: *Packet sniffer structure*

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. 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,”.

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 comprehensive user-guide (http://www.wireshark.org/docs/wsug_html_chunked/), man pages (http://www.wireshark.org/docs/man-pages/), and a 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).

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 Wireshark 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.

1.2 Running Wireshark

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 “*which wireshark*” that would show path /usr/bin/wireshark where wireshark is typically installed). When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will be displayed. Initially, no data will be displayed in the various windows.

The Wireshark interface has five major components:

- The **command menus** are standard pull down menus located at the top of the window. Of interest to us is 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 exits the Wireshark application. The Capture menu allows you to begin packet capture.

- **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.

- 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.

- The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

- 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.

5 STEPS FOR PERFORMING THIS LAB:

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 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 Options" window to be displayed, as shown in Figure 3.

4. ***Selecting the network interface on which packets would be captured:*** You can use most of the default values in this window, but uncheck "Hide capture info dialog" under Display Options. The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the Interface pull down menu at the top of the Capture Options window. 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 (most likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - Wireshark is now capturing all packets being sent/received from/ by your computer!

5. Once you begin packet capture, a packet capture summary window will appear, as shown in Figure 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.

6. ***Capturing an HTTP interaction on Wireshark:*** While Wireshark is running, enter the URL: http://seecs.nust.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://seecs.edu.pk, and exchange HTTP messages with the server in order to download this page, as discussed. Wireshark will capture the Ethernet frames containing these HTTP messages.

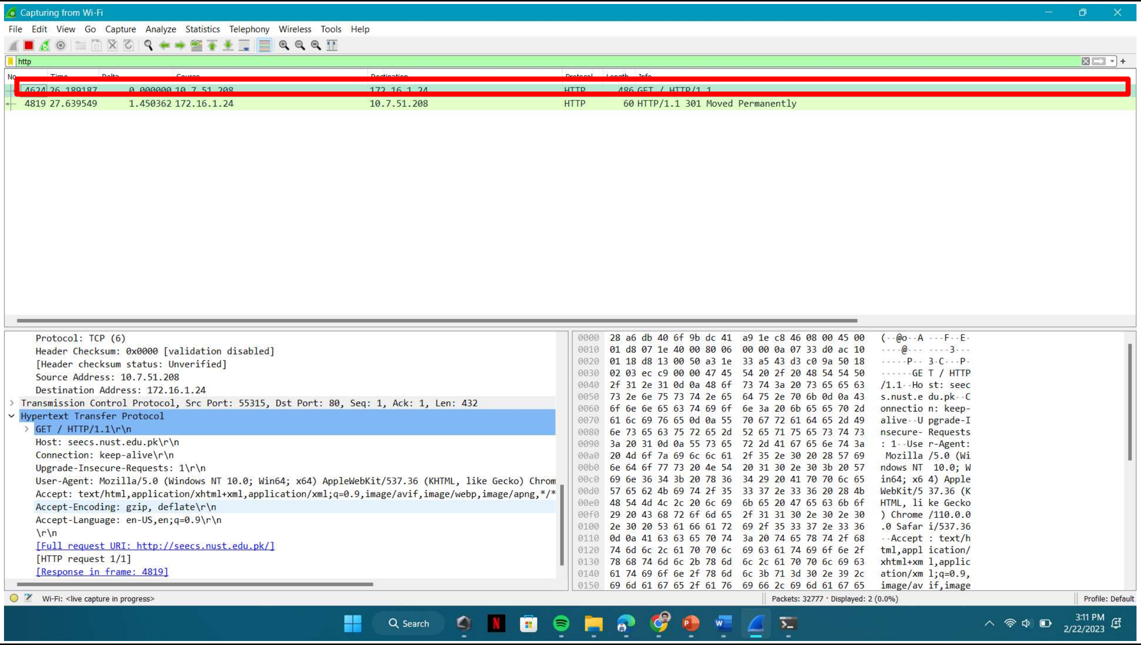
7. ***Stopping the capture and inspecting captured packets:*** After your browser has displayed the page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to Figure 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 seecs.nust.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 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".

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.

5.1 OUTPUT

The packet traced by wire shark is as shown:



Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

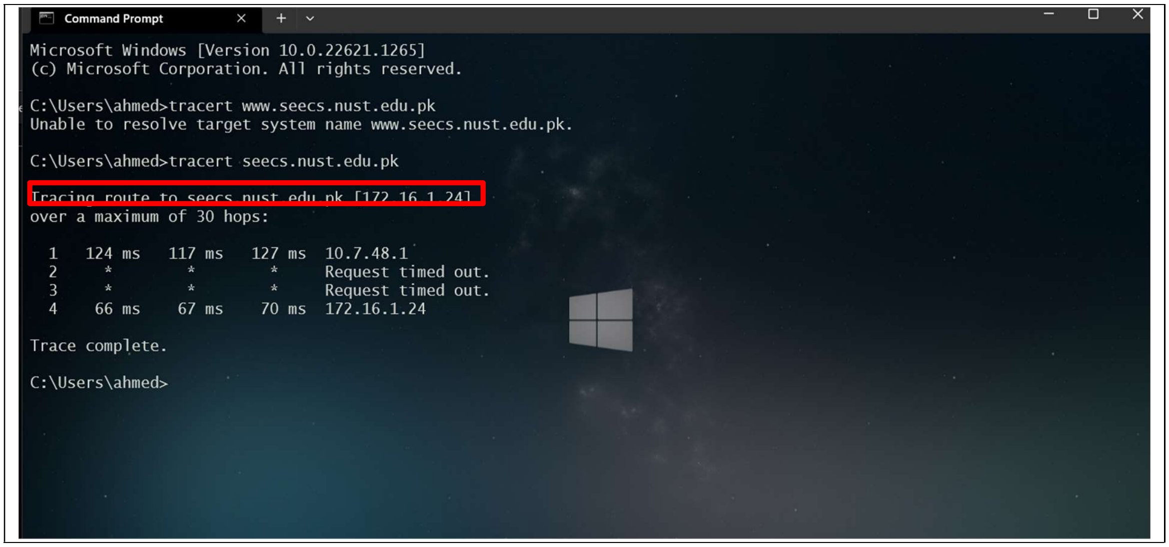
```
No.  Time       Delta    Source       Destination  Protocol Length Info
4624 26.189187  0.000000 10.7.51.208  172.16.1.24  HTTP     486 GET / HTTP/1.1
4819 27.639549  1.450362 172.16.1.24  10.7.51.208  HTTP      60 HTTP/1.1 301 Moved Permanently
```

```
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.7.51.208
    Destination Address: 172.16.1.24
> Transmission Control Protocol, Src Port: 55315, Dst Port: 80, Seq: 1, Ack: 1, Len: 432
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: seecs.nust.edu.pk\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://seecs.nust.edu.pk/]
    [HTTP request 1/1]
    [Response in frame: 4819]
```

```
0000  28 a6 db 40 6f 9b dc 41  a9 1e c8 46 08 00 45 00   (··@o··A ···F··E·
0010  01 d8 07 1e 40 00 80 06  00 00 0a 07 33 d0 ac 10   ····@··· ····3···
0020  01 18 d8 13 00 50 a3 1e  33 a5 43 d3 c0 9a 50 18   ·····P·· 3·C···P·
0030  02 03 ec c9 00 00 47 45  54 20 2f 20 48 54 54 50   ······GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 73 65 65 63   /1.1··Ho st: seec
0050  73 2e 6e 75 73 74 2e 65  64 75 2e 70 6b 0d 0a 43   s.nust.e du.pk··C
0060  6f 6e 6e 65 63 74 69 6f  6e 3a 20 6b 65 65 70 2d   onnectio n: keep-
0070  61 6c 69 76 65 0d 0a 55  70 67 72 61 64 65 2d 49   alive··U pgrade-I
0080  6e 73 65 63 75 72 65 2d  52 65 71 75 65 73 74 73   nsecure- Requests
0090  3a 20 31 0d 0a 55 73 65  72 2d 41 67 65 6e 74 3a   : 1··Use r-Agent:
00a0  20 4d 6f 7a 69 6c 6c 61  2f 35 2e 30 20 28 57 69    Mozilla /5.0 (Wi
00b0  6e 64 6f 77 73 20 4e 54  20 31 30 2e 30 3b 20 57   ndows NT  10.0; W
00c0  69 6e 36 34 3b 20 78 36  34 29 20 41 70 70 6c 65   in64; x6 4) Apple
00d0  57 65 62 4b 69 74 2f 35  33 37 2e 33 36 20 28 4b   WebKit/5 37.36 (K
00e0  48 54 4d 4c 2c 20 6c 69  6b 65 20 47 65 63 6b 6f   HTML, li ke Gecko
00f0  29 20 43 68 72 6f 6d 65  2f 31 31 30 2e 30 2e 30   ) Chrome /110.0.0
0100  2e 30 20 53 61 66 61 72  69 2f 35 33 37 2e 33 36   .0 Safar i/537.36
0110  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 68   ··Accept : text/h
0120  74 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   tml,appl ication/
0130  78 68 74 6d 6c 2b 78 6d  6c 2c 61 70 70 6c 69 63   xhtml+xm l,applic
0140  61 74 69 6f 6e 2f 78 6d  6c 3b 71 3d 30 2e 39 2c   ation/xm l;q=0.9,
0150  69 6d 61 67 65 2f 61 76  69 66 2c 69 6d 61 67 65   image/av if,image
```

Wi-Fi: <live capture in progress>
Packets: 32777 · Displayed: 2 (0.0%)
Profile: Default
Search
3:11 PM
2/22/2023

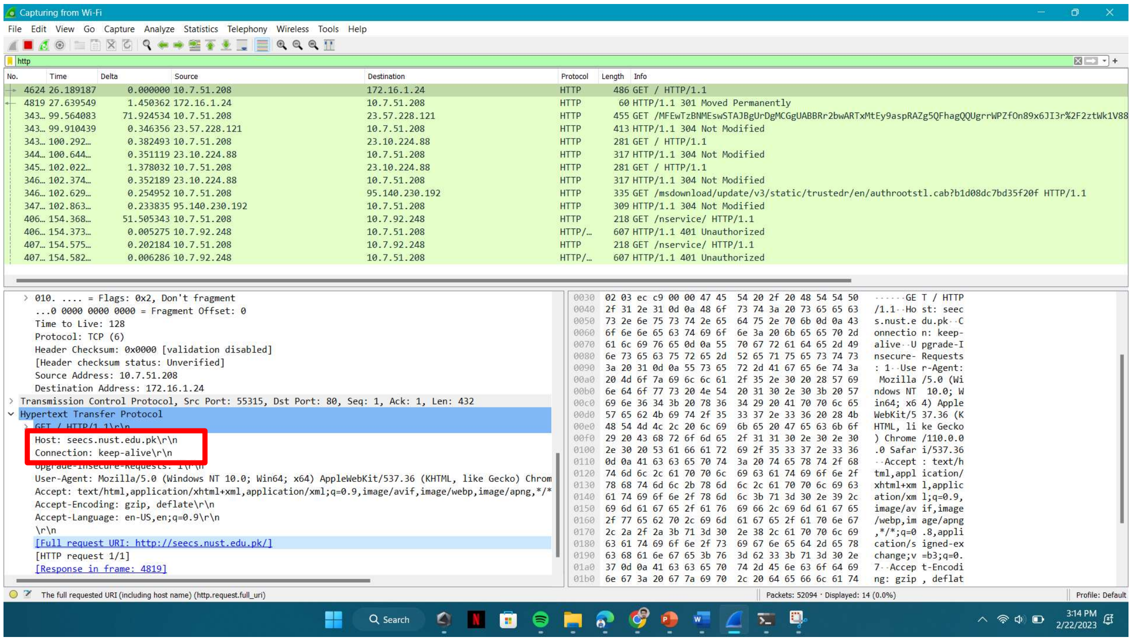
This is the same IP address of seecs.nust.edu.pk as observed by tracert command:



```
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ahmed>tracert www.seecs.nust.edu.pk
Unable to resolve target system name www.seecs.nust.edu.pk.

C:\Users\ahmed>tracert seecs.nust.edu.pk

Tracing route to seecs.nust.edu.pk [172.16.1.24]
over a maximum of 30 hops:

  1   124 ms   117 ms   127 ms  10.7.48.1
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4    66 ms    67 ms    70 ms  172.16.1.24

Trace complete.

C:\Users\ahmed>
```

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 seecs.nust.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 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).

5.2 OUPUT



Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time       Delta      Source          Destination     Protocol Length Info
4624 26.189187   0.000000 10.7.51.208     172.16.1.24     HTTP      486 GET / HTTP/1.1
4819 27.639549   1.450362 172.16.1.24     10.7.51.208     HTTP       60 HTTP/1.1 301 Moved Permanently
343… 99.564083  71.924534 10.7.51.208     23.57.228.121   HTTP      455 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBRr2bwARTxMtEy9aspRAZg5QFhagQQUgrrWPZfOn89x6JI3r%2F2ztWk1V88
343… 99.910439   0.346356 23.57.228.121   10.7.51.208     HTTP      413 HTTP/1.1 304 Not Modified
343… 100.292…    0.382493 10.7.51.208     23.10.224.88    HTTP      281 GET / HTTP/1.1
344… 100.644…    0.351119 23.10.224.88    10.7.51.208     HTTP      317 HTTP/1.1 304 Not Modified
345… 102.022…    1.378032 10.7.51.208     23.10.224.88    HTTP      281 GET / HTTP/1.1
346… 102.374…    0.352189 23.10.224.88    10.7.51.208     HTTP      317 HTTP/1.1 304 Not Modified
346… 102.629…    0.254952 10.7.51.208     95.140.230.192  HTTP      335 GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?b1d08dc7bd35f20f HTTP/1.1
347… 102.863…    0.233835 95.140.230.192  10.7.51.208     HTTP      309 HTTP/1.1 304 Not Modified
406… 154.368…   51.505343 10.7.51.208     10.7.92.248     HTTP      218 GET /nservice/ HTTP/1.1
406… 154.373…    0.005275 10.7.92.248     10.7.51.208     HTTP/…    607 HTTP/1.1 401 Unauthorized
407… 154.575…    0.202184 10.7.51.208     10.7.92.248     HTTP      218 GET /nservice/ HTTP/1.1
407… 154.582…    0.006286 10.7.92.248     10.7.51.208     HTTP/…    607 HTTP/1.1 401 Unauthorized
```

```
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.7.51.208
    Destination Address: 172.16.1.24
> Transmission Control Protocol, Src Port: 55315, Dst Port: 80, Seq: 1, Ack: 1, Len: 432
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: seecs.nust.edu.pk\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://seecs.nust.edu.pk/]
    [HTTP request 1/1]
    [Response in frame: 4819]
```

```
0030  02 03 ec c9 00 00 47 45  54 20 2f 20 48 54 54 50   ······GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 73 65 65 63   /1.1··Ho st: seec
0050  73 2e 6e 75 73 74 2e 65  64 75 2e 70 6b 0d 0a 43   s.nust.e du.pk··C
0060  6f 6e 6e 65 63 74 69 6f  6e 3a 20 6b 65 65 70 2d   onnectio n: keep-
0070  61 6c 69 76 65 0d 0a 55  70 67 72 61 64 65 2d 49   alive··U pgrade-I
0080  6e 73 65 63 75 72 65 2d  52 65 71 75 65 73 74 73   nsecure- Requests
0090  3a 20 31 0d 0a 55 73 65  72 2d 41 67 65 6e 74 3a   : 1··Use r-Agent:
00a0  20 4d 6f 7a 69 6c 6c 61  2f 35 2e 30 20 28 57 69    Mozilla /5.0 (Wi
00b0  6e 64 6f 77 73 20 4e 54  20 31 30 2e 30 3b 20 57   ndows NT  10.0; W
00c0  69 6e 36 34 3b 20 78 36  34 29 20 41 70 70 6c 65   in64; x6 4) Apple
00d0  57 65 62 4b 69 74 2f 35  33 37 2e 33 36 20 28 4b   WebKit/5 37.36 (K
00e0  48 54 4d 4c 2c 20 6c 69  6b 65 20 47 65 63 6b 6f   HTML, li ke Gecko
00f0  29 20 43 68 72 6f 6d 65  2f 31 31 30 2e 30 2e 30   ) Chrome /110.0.0
0100  2e 30 20 53 61 66 61 72  69 2f 35 33 37 2e 33 36   .0 Safar i/537.36
0110  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 68   ··Accept : text/h
0120  74 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   tml,appl ication/
0130  78 68 74 6d 6c 2b 78 6d  6c 2c 61 70 70 6c 69 63   xhtml+xm l,applic
0140  61 74 69 6f 6e 2f 78 6d  6c 3b 71 3d 30 2e 39 2c   ation/xm l;q=0.9,
0150  69 6d 61 67 65 2f 61 76  69 66 2c 69 6d 61 67 65   image/av if,image
0160  2f 77 65 62 70 2c 69 6d  61 67 65 2f 61 70 6e 67   /webp,im age/apng
0170  2c 2a 2f 2a 3b 71 3d 30  2e 38 2c 61 70 70 6c 69   ,*/*;q=0 .8,appli
0180  63 61 74 69 6f 6e 2f 73  69 67 6e 65 64 2d 65 78   cation/s igned-ex
0190  63 68 61 6e 67 65 3b 76  3d 62 33 3b 71 3d 30 2e   change;v =b3;q=0.
01a0  37 0d 0a 41 63 63 65 70  74 2d 45 6e 63 6f 64 69   7··Accep t-Encodi
01b0  6e 67 3a 20 67 7a 69 70  2c 20 64 65 66 6c 61 74   ng: gzip , deflat
```

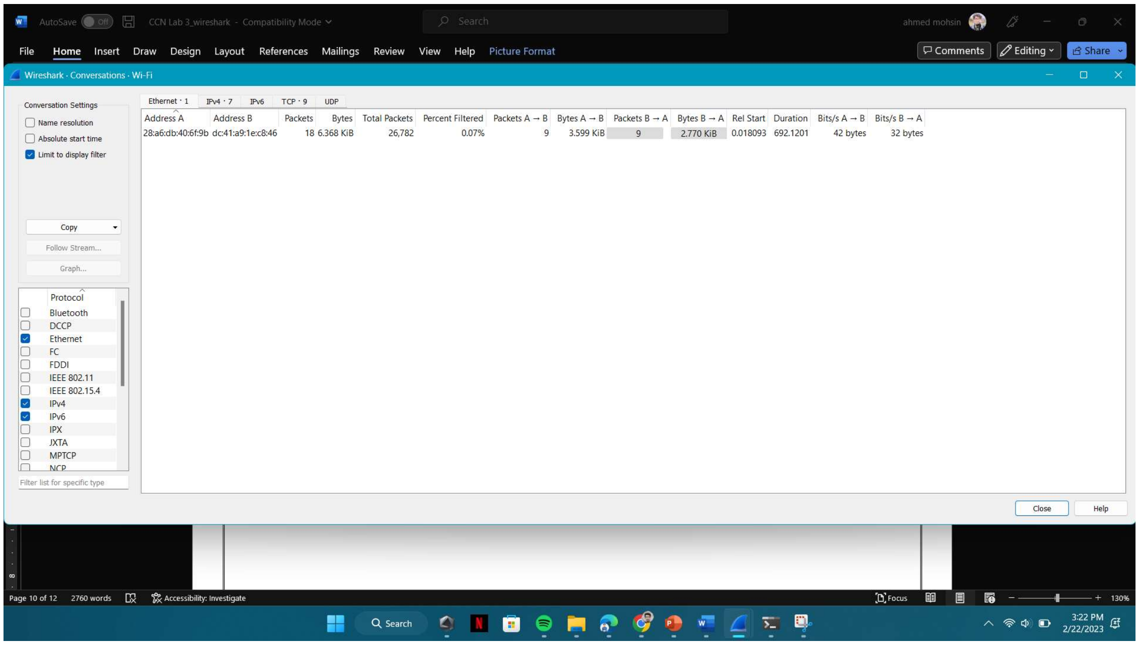
The full requested URI (including host name) (http.request.full_uri)

Packets: 52094 · Displayed: 14 (0.0%)

Profile: Default

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.

5.3 OUTPUT



AutoSave Off
CCN Lab 3_wireshark - Compatibility Mode
Search
ahmed mohsin
File Home Insert Draw Design Layout References Mailings Review View Help Picture Format
Comments
Editing
Share
Wireshark · Conversations · Wi-Fi
Copy
Follow Stream...
Graph...
Filter list for specific type
Close
Help
Page 10 of 12
2760 words
Accessibility: Investigate
Focus
130%
Search
3:22 PM
2/22/2023

Conversation Settings

☐ Name resolution

☐ Absolute start time

☑ Limit to display filter

Ethernet · 1 | IPv4 · 7 | IPv6 | TCP · 9 | UDP

Address A	Address B	Packets	Bytes	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
28:a6:db:40:6f:9b	dc:41:a9:1e:c8:46	18	6.368 KiB	26,782	0.07%	9	3.599 KiB	9	2.770 KiB	0.018093	692.1201	42 bytes	32 bytes

Protocol

☐ Bluetooth

☐ DCCP

☑ Ethernet

☐ FC

☐ FDDI

☐ IEEE 802.11

☐ IEEE 802.15.4

☑ IPv4

☑ IPv6

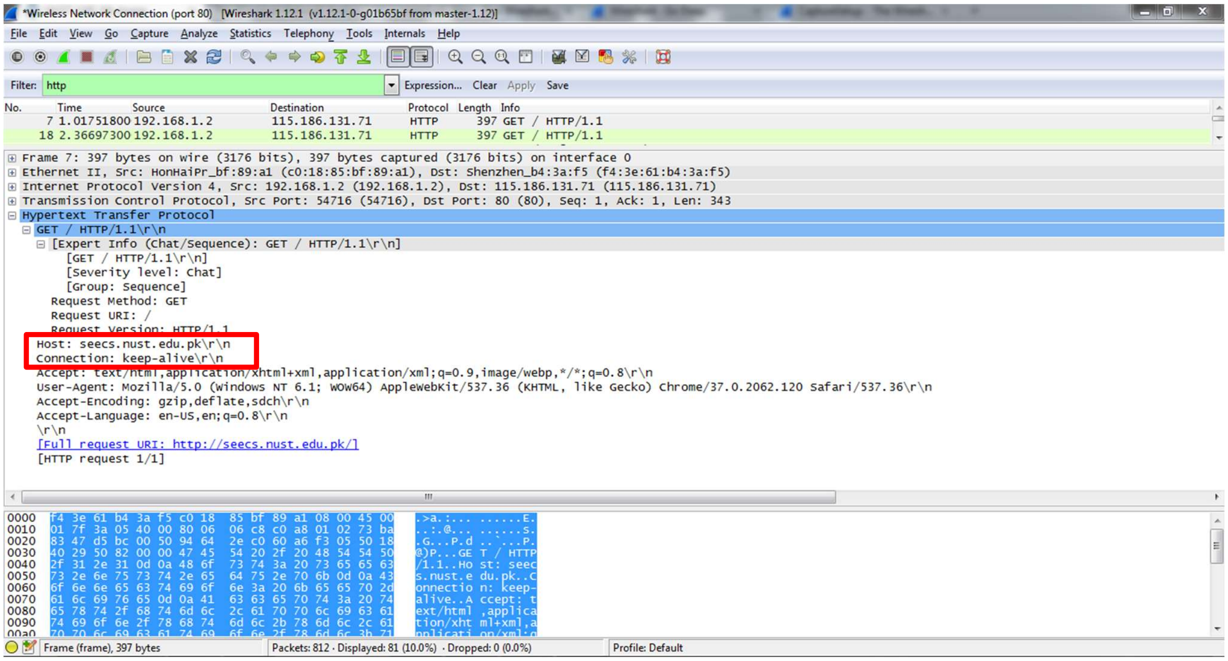
☐ IPX

☐ JXTA

☐ MPTCP

☐ NCP

11. ***Obtaining credit for this lab:*** Answer the questions listed at the end of this lab. Please note that this is an individual activity and every student must upload the answer file (after duly filling in the answers) through the appropriate link at your LMS course site for the specific date of your lab (an upload link would be made available) to obtain credit. Please clarify with your instructor/ lab engineer if you have any queries.


*Wireless Network Connection (port 80) [Wireshark 1.12.1 (v1.12.1-0-g01b65bf from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save
Profile: Default

Filter: http

```
No.     Time       Source          Destination      Protocol Length Info
      7 1.01751800 192.168.1.2     115.186.131.71   HTTP        397 GET / HTTP/1.1
     18 2.36697300 192.168.1.2     115.186.131.71   HTTP        397 GET / HTTP/1.1
```

```
⊞ Frame 7: 397 bytes on wire (3176 bits), 397 bytes captured (3176 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_bf:89:a1 (c0:18:85:bf:89:a1), Dst: Shenzhen_b4:3a:f5 (f4:3e:61:b4:3a:f5)
⊞ Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 115.186.131.71 (115.186.131.71)
⊞ Transmission Control Protocol, Src Port: 54716 (54716), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 343
⊟ Hypertext Transfer Protocol
  ⊟ GET / HTTP/1.1\r\n
    ⊟ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
        [GET / HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: seecs.nust.edu.pk\r\n
    Connection: keep-alive\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120 Safari/537.36\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    \r\n
    [Full request URI: http://seecs.nust.edu.pk/]
    [HTTP request 1/1]
```

```
0000  f4 3e 61 b4 3a f5 c0 18  85 bf 89 a1 08 00 45 00   .>a.:... ......E.
0010  01 7f 3a 05 40 00 80 06  06 c8 c0 a8 01 02 73 ba   ..:.@... ......s.
0020  83 47 d5 bc 00 50 94 64  2e c0 60 a6 f3 05 50 18   .G...P.d ..`...P.
0030  40 29 50 82 00 00 47 45  54 20 2f 20 48 54 54 50   @)P...GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 73 65 65 63   /1.1..Ho st: seec
0050  73 2e 6e 75 73 74 2e 65  64 75 2e 70 6b 0d 0a 43   s.nust.e du.pk..C
0060  6f 6e 6e 65 63 74 69 6f  6e 3a 20 6b 65 65 70 2d   onnectio n: keep-
0070  61 6c 69 76 65 0d 0a 41  63 63 65 70 74 3a 20 74   alive..A ccept: t
0080  65 78 74 2f 68 74 6d 6c  2c 61 70 70 6c 69 63 61   ext/html ,applica
0090  74 69 6f 6e 2f 78 68 74  6d 6c 2b 78 6d 6c 2c 61   tion/xht ml+xml,a
00a0  70 70 6c 69 63 61 74 69  6f 6e 2f 78 6d 6c 3b 71   pplicati on/xml;q
```

Frame (frame), 397 bytes | Packets: 812 · Displayed: 81 (10.0%) · Dropped: 0 (0.0%)

Figure 5: *Wireshark display after step 9*

6 QUESTIONS:

1. ***Finding IP address of your machine in Wireshark:*** What is the IP address of 'alibaba.com'? What is the IP address of your computer? How did you find it in Wireshark? Compare the IP address of your machine by using ipconfig command.

6.1 ANSWER

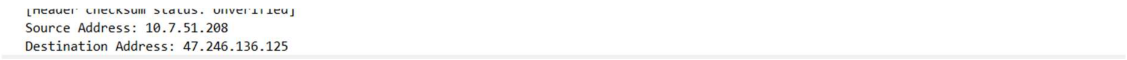
To find the IP address of 'alibaba.com':

Open the command prompt or terminal on your computer. Type 'nslookup alibaba.com' and press Enter. The IP address associated with the domain name 'alibaba.com' will be displayed.

To find the IP address of your computer:

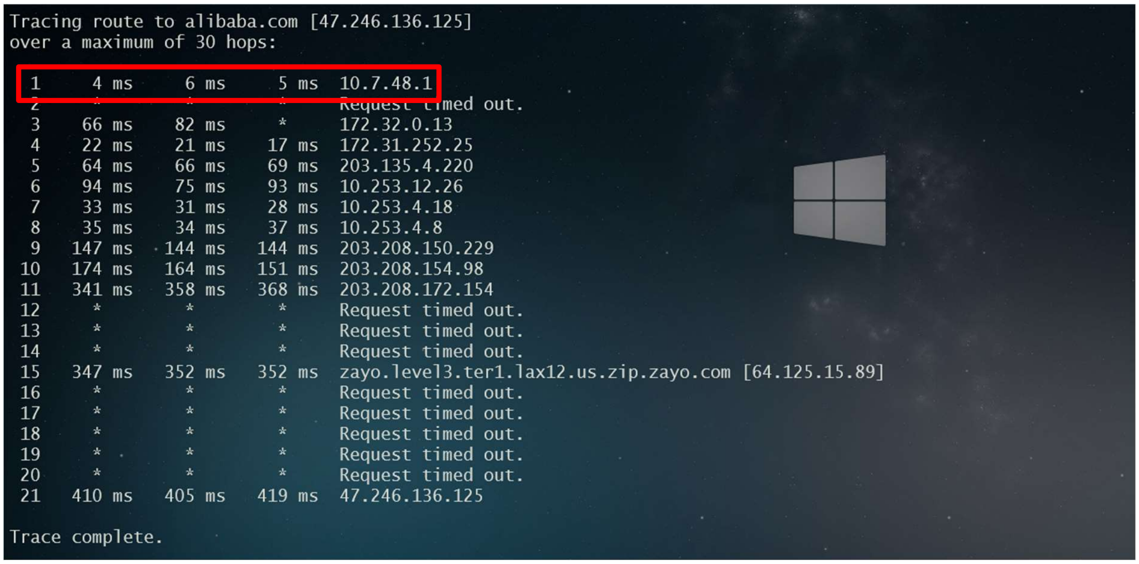
Open the command prompt or terminal on your computer. Type 'ipconfig' and press Enter. The IP address of your computer will be displayed under the 'IPv4 Address' or 'IP Address' field.

The IP address of my PC using wire shark is:



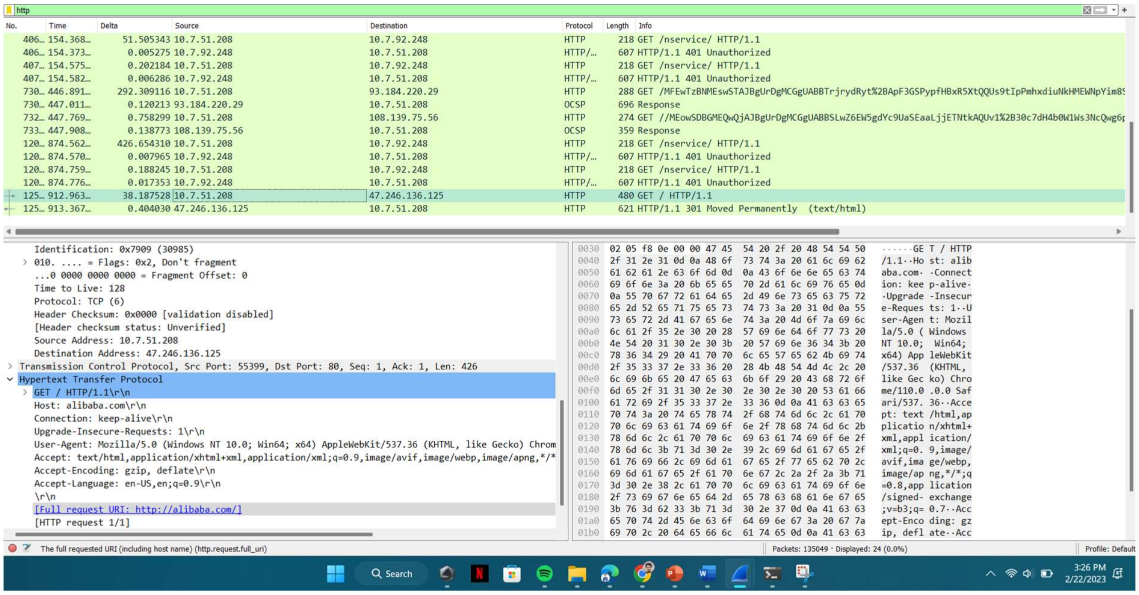
```
    [Header checksum status: Unverified]
    Source Address: 10.7.51.208
    Destination Address: 47.246.136.125
```

The IP address of my pc using tracert command is:



```
Tracing route to alibaba.com [47.246.136.125]
over a maximum of 30 hops:

  1     4 ms     6 ms     5 ms  10.7.48.1
  2     *        *        *     Request timed out.
  3    66 ms    82 ms     *     172.32.0.13
  4    22 ms    21 ms    17 ms  172.31.252.25
  5    64 ms    66 ms    69 ms  203.135.4.220
  6    94 ms    75 ms    93 ms  10.253.12.26
  7    33 ms    31 ms    28 ms  10.253.4.18
  8    35 ms    34 ms    37 ms  10.253.4.8
  9   147 ms   144 ms   144 ms  203.208.150.229
 10   174 ms   164 ms   151 ms  203.208.154.98
 11   341 ms   358 ms   368 ms  203.208.172.154
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15   347 ms   352 ms   352 ms  zayo.level3.ter1.lax12.us.zip.zayo.com [64.125.15.89]
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21   410 ms   405 ms   419 ms  47.246.136.125

Trace complete.
```

The first hop shows the source PC and for this case is our own PC;s IP address.



```
http

No.      Time        Delta       Source            Destination       Protocol  Length  Info
406…  154.368…   51.505343  10.7.51.208       10.7.92.248       HTTP       218 GET /nservice/ HTTP/1.1
406…  154.373…    0.005275  10.7.92.248       10.7.51.208       HTTP/…     607 HTTP/1.1 401 Unauthorized
407…  154.575…    0.202184  10.7.51.208       10.7.92.248       HTTP       218 GET /nservice/ HTTP/1.1
407…  154.582…    0.006286  10.7.92.248       10.7.51.208       HTTP/…     607 HTTP/1.1 401 Unauthorized
730…  446.891…  292.309116  10.7.51.208       93.184.220.29     HTTP       288 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTrjrydRyt%2BApF3GSPypfHBxR5XtQQUs9tIpPmhxdiuNkHMEWNpYim8S
730…  447.011…    0.120213  93.184.220.29     10.7.51.208       OCSP       696 Response
732…  447.769…    0.758299  10.7.51.208       108.139.75.56     HTTP       274 GET //MEowSDBGMEQwQjAJBgUrDgMCGgUABBSLwZ6EW5gdYc9UaSEaaLjjETNtkAQUv1%2B30c7dH4b0W1Ws3NcQwg6p
733…  447.908…    0.138773  108.139.75.56     10.7.51.208       OCSP       359 Response
120…  874.562…  426.654310  10.7.51.208       10.7.92.248       HTTP       218 GET /nservice/ HTTP/1.1
120…  874.570…    0.007965  10.7.92.248       10.7.51.208       HTTP/…     607 HTTP/1.1 401 Unauthorized
120…  874.759…    0.188245  10.7.51.208       10.7.92.248       HTTP       218 GET /nservice/ HTTP/1.1
120…  874.776…    0.017353  10.7.92.248       10.7.51.208       HTTP/…     607 HTTP/1.1 401 Unauthorized
125…  912.963…   38.187528  10.7.51.208       47.246.136.125    HTTP       480 GET / HTTP/1.1
125…  913.367…    0.404030  47.246.136.125    10.7.51.208       HTTP       621 HTTP/1.1 301 Moved Permanently  (text/html)
```

```
    Identification: 0x7909 (30985)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.7.51.208
    Destination Address: 47.246.136.125
> Transmission Control Protocol, Src Port: 55399, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: alibaba.com\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://alibaba.com/]
    [HTTP request 1/1]
```

```
0030  02 05 f8 0e 00 00 47 45  54 20 2f 20 48 54 54 50   ······GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 61 6c 69 62   /1.1··Ho st: alib
0050  61 62 61 2e 63 6f 6d 0d  0a 43 6f 6e 6e 65 63 74   aba.com· ·Connect
0060  69 6f 6e 3a 20 6b 65 65  70 2d 61 6c 69 76 65 0d   ion: kee p-alive·
0070  0a 55 70 67 72 61 64 65  2d 49 6e 73 65 63 75 72   ·Upgrade -Insecur
0080  65 2d 52 65 71 75 65 73  74 73 3a 20 31 0d 0a 55   e-Reques ts: 1··U
0090  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
00a0  6c 61 2f 35 2e 30 20 28  57 69 6e 64 6f 77 73 20   la/5.0 ( Windows 
00b0  4e 54 20 31 30 2e 30 3b  20 57 69 6e 36 34 3b 20   NT 10.0;  Win64; 
00c0  78 36 34 29 20 41 70 70  6c 65 57 65 62 4b 69 74   x64) App leWebKit
00d0  2f 35 33 37 2e 33 36 20  28 4b 48 54 4d 4c 2c 20   /537.36  (KHTML, 
00e0  6c 69 6b 65 20 47 65 63  6b 6f 29 20 43 68 72 6f   like Gec ko) Chro
00f0  6d 65 2f 31 31 30 2e 30  2e 30 2e 30 20 53 61 66   me/110.0 .0.0 Saf
0100  61 72 69 2f 35 33 37 2e  33 36 0d 0a 41 63 63 65   ari/537. 36··Acce
0110  70 74 3a 20 74 65 78 74  2f 68 74 6d 6c 2c 61 70   pt: text /html,ap
0120  70 6c 69 63 61 74 69 6f  6e 2f 78 68 74 6d 6c 2b   plicatio n/xhtml+
0130  78 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   xml,appl ication/
0140  78 6d 6c 3b 71 3d 30 2e  39 2c 69 6d 61 67 65 2f   xml;q=0. 9,image/
0150  61 76 69 66 2c 69 6d 61  67 65 2f 77 65 62 70 2c   avif,ima ge/webp,
0160  69 6d 61 67 65 2f 61 70  6e 67 2c 2a 2f 2a 3b 71   image/ap ng,*/*;q
0170  3d 30 2e 38 2c 61 70 70  6c 69 63 61 74 69 6f 6e   =0.8,app lication
0180  2f 73 69 67 6e 65 64 2d  65 78 63 68 61 6e 67 65   /signed- exchange
0190  3b 76 3d 62 33 3b 71 3d  30 2e 37 0d 0a 41 63 63   ;v=b3;q= 0.7··Acc
01a0  65 70 74 2d 45 6e 63 6f  64 69 6e 67 3a 20 67 7a   ept-Enco ding: gz
01b0  69 70 2c 20 64 65 66 6c  61 74 65 0d 0a 41 63 63   ip, defl ate··Acc
```

The full requested URI (including host name) (http.request.full_uri)

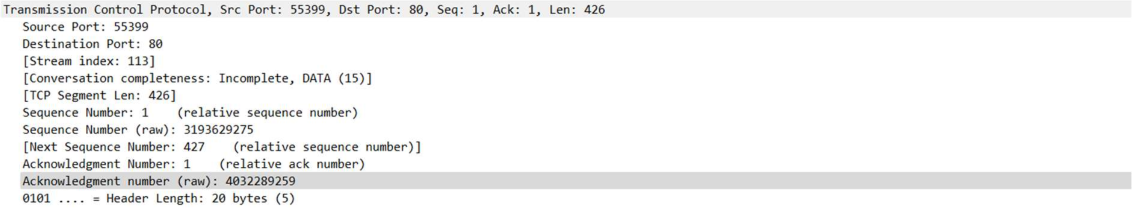
Packets: 135049 · Displayed: 24 (0.0%)

Profile: Default

2. What is the ***port number*** used by the HTTP server 'alibaba.com'. How did you note it in Wireshark?

6.2 Answer:

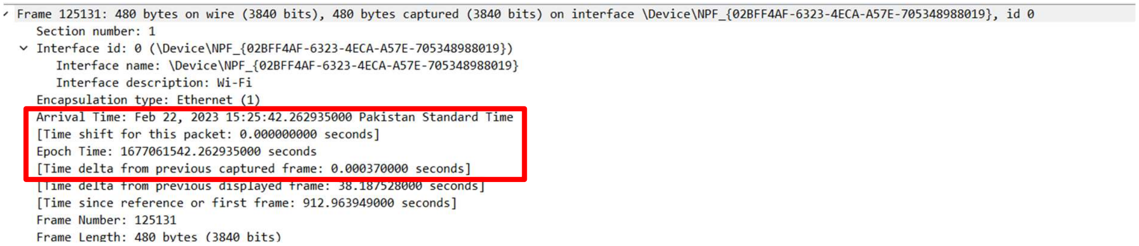
We can see the port number under the transmission control protocol as shown in the screenshot below:



```
Transmission Control Protocol, Src Port: 55399, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
   Source Port: 55399
   Destination Port: 80
   [Stream index: 113]
   [Conversation completeness: Incomplete, DATA (15)]
   [TCP Segment Len: 426]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 3193629275
   [Next Sequence Number: 427    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 4032289259
   0101 .... = Header Length: 20 bytes (5)
```

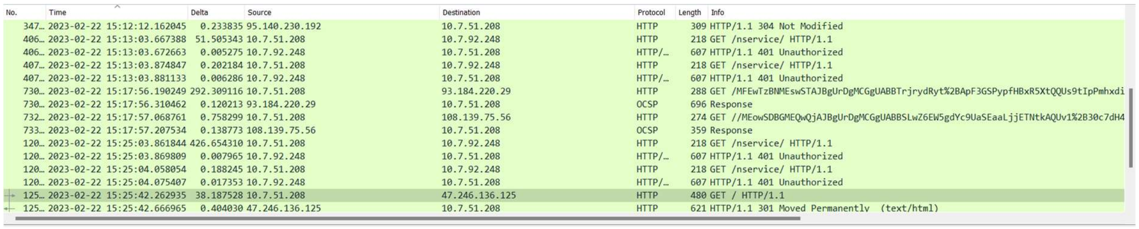
3. ***Delay between request and reply:*** How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field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

The time taken by the packet to be send and received is sown below in the screen shot as "time of day" format.



```
⌄ Frame 125131: 480 bytes on wire (3840 bits), 480 bytes captured (3840 bits) on interface \Device\NPF_{02BFF4AF-6323-4ECA-A57E-705348988019}, id 0
    Section number: 1
  ⌄ Interface id: 0 (\Device\NPF_{02BFF4AF-6323-4ECA-A57E-705348988019})
      Interface name: \Device\NPF_{02BFF4AF-6323-4ECA-A57E-705348988019}
      Interface description: Wi-Fi
    Encapsulation type: Ethernet (1)
    Arrival Time: Feb 22, 2023 15:25:42.262935000 Pakistan Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1677061542.262935000 seconds
    [Time delta from previous captured frame: 0.000370000 seconds]
    [Time delta from previous displayed frame: 38.187528000 seconds]
    [Time since reference or first frame: 912.963949000 seconds]
    Frame Number: 125131
    Frame Length: 480 bytes (3840 bits)
```

The time it took as displayed by wire shark time column is as:

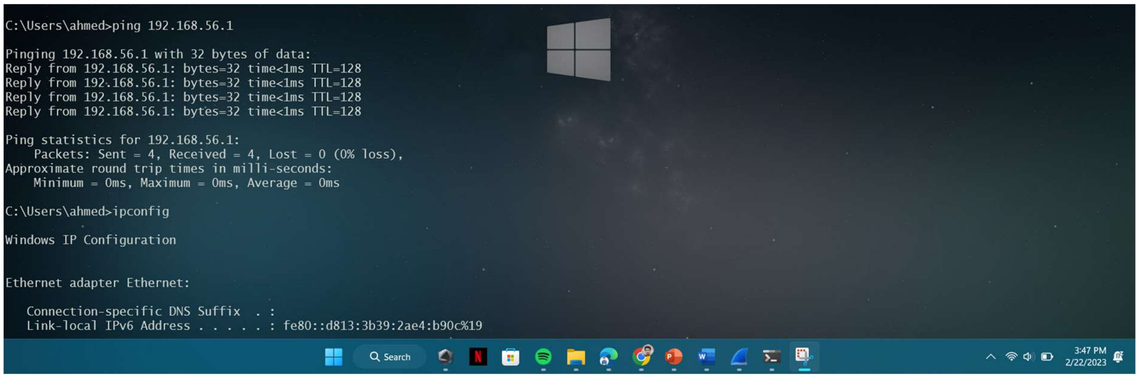


No.	Time	Delta	Source	Destination	Protocol	Length	Info
347…	2023-02-22 15:12:12.162045	0.233835	95.140.230.192	10.7.51.208	HTTP	309	HTTP/1.1 304 Not Modified
406…	2023-02-22 15:13:03.667388	51.505343	10.7.51.208	10.7.92.248	HTTP	218	GET /nservice/ HTTP/1.1
406…	2023-02-22 15:13:03.672663	0.005275	10.7.92.248	10.7.51.208	HTTP/…	607	HTTP/1.1 401 Unauthorized
407…	2023-02-22 15:13:03.874847	0.202184	10.7.51.208	10.7.92.248	HTTP	218	GET /nservice/ HTTP/1.1
407…	2023-02-22 15:13:03.881133	0.006286	10.7.92.248	10.7.51.208	HTTP/…	607	HTTP/1.1 401 Unauthorized
730…	2023-02-22 15:17:56.190249	292.309116	10.7.51.208	93.184.220.29	HTTP	288	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTrjrydRyt%2BApF3GSPypfHBxR5XtQQUs9tIpPmhxdi
730…	2023-02-22 15:17:56.310462	0.120213	93.184.220.29	10.7.51.208	OCSP	696	Response
732…	2023-02-22 15:17:57.068761	0.758299	10.7.51.208	108.139.75.56	HTTP	274	GET //MEowSDBGMEQwQjAJBgUrDgMCGgUABBSLwZ6EW5gdYc9UaSEaaLjjETNtkAQUv1%2B30c7dH4
733…	2023-02-22 15:17:57.207534	0.138773	108.139.75.56	10.7.51.208	OCSP	359	Response
120…	2023-02-22 15:25:03.861844	426.654310	10.7.51.208	10.7.92.248	HTTP	218	GET /nservice/ HTTP/1.1
120…	2023-02-22 15:25:03.869809	0.007965	10.7.92.248	10.7.51.208	HTTP/…	607	HTTP/1.1 401 Unauthorized
120…	2023-02-22 15:25:04.058054	0.188245	10.7.51.208	10.7.92.248	HTTP	218	GET /nservice/ HTTP/1.1
120…	2023-02-22 15:25:04.075407	0.017353	10.7.92.248	10.7.51.208	HTTP/…	607	HTTP/1.1 401 Unauthorized
125…	2023-02-22 15:25:42.262935	38.187528	10.7.51.208	47.246.136.125	HTTP	480	GET / HTTP/1.1
125…	2023-02-22 15:25:42.666965	0.404030	47.246.136.125	10.7.51.208	HTTP	621	HTTP/1.1 301 Moved Permanently (text/html)

7 ADDITIONAL LAB TASK

Pinging Imran Haider PC:

I Pinged Imran Haider's laptop using my PC by using the ping command and it worked.



```
C:\Users\ahmed>ping 192.168.56.1

Pinging 192.168.56.1 with 32 bytes of data:
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.56.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ahmed>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::d813:3b39:2ae4:b90c%19
```

8 CONCLUSION:

In conclusion, this lab provided us with valuable insights into the powerful network analysis tool **Wireshark**. Through this lab, we were able to learn how to apply filters to analyze network traffic, identify the packets sent and received during the communication process, and extract valuable information from those packets. By tracing the packets of the site alibaba.com, we gained a better understanding of how internet protocols work, and how information is transmitted and received over the network. We were able to observe the flow of data packets, their structure, and the information they carry. Moreover, by answering theoretical questions related to network protocols, we were able to enhance our understanding of the underlying mechanisms and principles that govern network communication.