Report for LAB 3-2: TCP

Name: Akshay Rathod	Student ID: 90	Date: 21-03-2023

Part I	
1	Socket addresses: source address: 103.230.151.202 Destination address: 192.168.101.128
2	Set flags: Acknowledgement (ACK)
3	Sequence number and acknowledgement number: seq :- 7960 ack number:- 1078
4	Window size: 64240

Part II	
1	Set flag in HTTP GET message: Push (PSH) , Acknowledgement (ACK)
2	Number of bytes transmitted by the HTTP GET message: 141 Bytes
3	Acknowledgement frequency: Corresponding rule:
4	Number of bytes transmitted by each packet: Relation to sequence and acknowledgement Number: When data is transmitted from one device to another, it is divided into segments or packets, each of which is assigned a unique sequence number. The sequence number is used to keep track of the order in which the packets were transmitted and to identify any missing or duplicate packets. The receiving device uses the acknowledgement number to inform the sender which packets it has successfully received. The acknowledgement number represents the next expected sequence number that the receiver is waiting for. By using sequence numbers and acknowledgement numbers, the sender and receiver can ensure that data is transmitted reliably and in the correct order, even in the presence of network errors or congestion.
5	Original window sizes: Are these numbers expected? How window sizes change?
6	How the window size is used in flow control? Flow control is accomplished by the receiver sending back a window to the sender. The size of this

	window, called the receive window, tells the sender how much data to send. Often, when the client is saturated, it might not be able to send back a receive window to the sender to signal it to slow down transmission. However, the *sliding windows* protocol is designed to let the sender know, before reaching a meltdown, to start slowing down transmission by a steadily decreasing window size. At the same time these flow control windows are going back and forth, the speed at which ACKs come back from the receiver to the sender provides additional information to the sender that caps the amount of data to send to the client. This is computed indirectly.
7	Purpose of the HTTP OK message: The HTTP 200 OK success status response code indicates that the request has succeeded.

Part III	
1	Number of TCP segments exchanged for connection termination: The termination of a TCP connection requires four exchanges of TCP segments. As a TCP connection is bidirectional (full duplex) , connection termination process should be made in both directions of the communication.
2	Which end point started the connection termination phase? The first FIN termination request is sent by the client to the server. It depicts the start of the termination process between the client and server.
3	Flags sets in each of the segments used for connection termination: The flags used in the segments exchanged during connection termination are as follows: 1. FIN (Finish): The FIN flag is set in the first segment sent by the endpoint that wants to initiate the connection termination process. This segment indicates that the endpoint has no more data to send and is ready to close the connection. 2. ACK (Acknowledgment): When the receiving endpoint receives the FIN segment, it sends an ACK segment to acknowledge that it has received the FIN segment. 3. FIN-ACK: After receiving the ACK segment from the other endpoint, the receiving endpoint sends its own FIN segment to indicate that it has no more data to send and is ready to close the connection. 4. ACK: Finally, the initiating endpoint responds with an ACK segment to acknowledge the receipt of the FIN segment from the other endpoint.

<table>
<tr><th colspan="3">Part IV</th></tr>
<tr><td rowspan="4">1</td><td>a. Source port number: 443</td><td>b. Destination port number: 56104</td></tr>
<tr><td>c. Sequence number:- 7960</td><td>d. Acknowledgement number:- 1078</td></tr>
<tr><td>e. Header length: 20 bytes</td><td>f. Set flags: ACK</td></tr>
<tr><td>g. Window size: 64240</td><td>h. Urgent pointer: 0</td></tr>
<tr><td>2</td><td colspan="2">Are answer in the question number 1 verified by the information in the detail pane lane?

Yes</td></tr>
<tr><td>3</td><td colspan="2">Does any of the TCP packet headers carry options? Yes

Explain:

The TCP (Transmission Control Protocol) packet header has an optional field called the "Options" field that can be used to provide additional information about the TCP connection.The Options field is used to include a variety of optional features and parameters, such as window scaling, selective acknowledgments (SACKs), timestamping, and maximum segment size (MSS). These options can be used to improve the performance, reliability, and security of TCP connections.</td></tr>
<tr><td>4</td><td colspan="2">Size of a TCP packet with no option: A TCP packet with no options consists of a 20-byte header followed by the data payload.

Size of a TCP packet with options:
The size of a TCP (Transmission Control Protocol) packet with options can vary depending on the number and type of options included in the packet. The Options field in the TCP header is variable in length and can contain one or more options, each of which can add additional bytes to the packet.

20 bytes (TCP header) + 40 bytes (maximum size of options field) + size of data payload</td></tr>
<tr><td>5</td><td colspan="2">Is window size in any of the TCP packet zero? Yes

Explain:
When the window size is set to zero, the sender should stop sending data and wait for the receiver to increase the window size before sending more data. This is an important flow control mechanism used in TCP to prevent overwhelming the receiver with data and potentially dropping packets.
If the sender continues to send data when the receiver's window size is zero, the receiver will send an acknowledgment packet with the Window field set to zero, which indicates that the sender should stop sending data until the receiver increases the window.</td></tr>
</table>