**SOLUTION**

**AISHA MUHAMMAD NAWAZ  
L20-0921**

**BSCS – SECTION 5E1**

Lab Manual 08

Objective:

• Observing the structure and working of **TCP**, **UDP** and **ICMP** Protocols in Wireshark. Lab Statement 1: Analyzing TCP Packets using Wireshark (10)

• **Step 1:** Run Wireshark.

• **Step 2:** Load the trace file **tcp-ethereal-trace-1**

• **Step 3:** Now filter out all TCP packets by typing “tcp” (without quotes) in the filter field towards the top of the Wireshark window. You should see a series of TCP and HTTP messages between the host in MIT and gaia.cs.umass.edu. The first three packets of the trace consist of the initial *three-way handshake* containing the SYN, SYN ACK and ACK messages. You should see a series of “TCP Segment of Reassembled PDU” messages being sent from the host in MIT to gaia.cs.umass.edu. Recall from the previous lab that there is no such thing as an HTTP Continuation message – this is Wireshark’s way of indicating that there are multiple segments being used to carry a single HTTP message. You should also see TCP ACK segments being returned from gaia.cs.umass.edu to the host in MIT.

**Question 1:** What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

**Source Port: 1161 and Src: 192.168.1.102**

**Question 2:** What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

**Dst: 128.119.245.12 and Port 80**

**Question 3:** What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is in the segment that identifies the segment as a SYN segment?

**Sequence: 0**

**Flags: 0x002 (SYN) identifies**

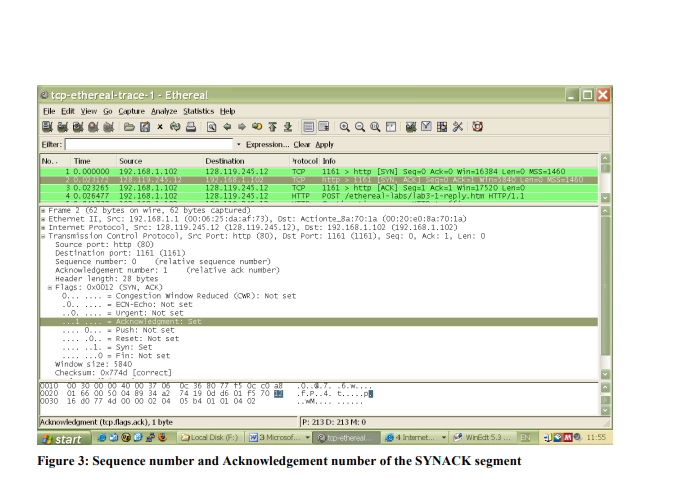
**Question 4:** What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? What is it in the segment that identifies the segment as a SYNACK segment?

**Sequence: 0**

**Ack:1**

**Flags: 0x012 (SYN, ACK) identifies**

Solution: Sequence number of the SYNACK segment from gaia.cs.umass.edu to the client computer in reply to the SYN has the value of 0 in this trace. The value of the ACKnowledgement field in the SYNACK segment is 1. The value of the ACKnowledgement field in the SYNACK segment is determined by gaia.cs.umass.edu by adding 1 to the initial sequence number of SYN segment from the client computer (i.e. the sequence number of the SYN segment initiated by the client computer is 0.). The SYN flag and Acknowledgement flag in the segment are set to 1 and they indicate that this segment is a SYNACK segment



Page **2** of **4**

**\*Question 5:** In packet 9, **Ack = 2026** and **Seq = 1**. Explain these values?

**Ack = 2026**

**Seq = 1**

**The previous tcp resembled packet (packet number 7) has sequence number =2026 and Ack=1**

**So ack is sent of it**

**\*Question 6:** In packet 16, **Ack = 7866** and **Seq = 1**. Explain these values?

**The previous tcp resembled packet (packet number 13) has sequence number =7866 and Ack=1**

**So ack is sent of it**

**Question 7:** Why Wireshark uses relative sequence and ack?

**so the user can identify the sequence of events**

**Lab Statement 2: Analyzing UDP Packets using Wireshark**

**(5)**

• **Step 1:** Run Wireshark

• **Step 2:** Load the trace file **dns-ethereal-trace-2.**

• **Step 3:** Now filter out all non-UDP packets by typing “udp” (without quotes) in the filter field towards the top of the Wireshark window

• **Step 4:** Analyze the UDP Packets and answer the following questions

**Question 1:** Select the first DNS packet in the trace. Determine, how many fields there are in the UDP header

**Four Fields**: **Source Port**, **Destination Port**, **Length, Checksum**

**Question 2:** From the packet content field (click on any header and observe the display in the Packet Bytes Window), determine the length (in bytes) of each of the UDP header fields.

**Seeing packet 16**

**Total 8 Bytes**

**Source port: 0e 9c = 2 bytes**

**Destination port: 00 35 = 2 bytes**

**Length: 00 34 = 2 bytes**

**Checksum: c4 93 = 2 bytes**

**Question 3:** The value in the Length field is the length of what? Verify your claim using the selected packet.

**For Packet Number 15**

**The length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next.**

**The length of UDP payload for selected packet is 44 bytes. 52 bytes - 8 bytes = 44 bytes**

**Question 4:** What is the port number to query the DNS Server?

**Port Number 53**

**Lab Statement 3: Analyzing ICMP Packets using Wireshark (5)**

• **Step 1:** Run Wireshark

• **Step 2:** Load the Session file **ICMP\_Session**

• **Step 3:** Now filter out all non-ICMP packets by typing “icmp” (without quotes) in the filter field towards the top of the Wireshark window

• **Step 4:** Analyze the ICMP Packets and answer the following questions

Page **3** of **4**

|  |  |
| --- | --- |
| **1-** Are ICMP messages sent over UDP or TCP? | **neither** |
| **2-** What is the link-layer (e.g., Ethernet) address of the host? | **Src: IntelCor\_55:7b:ac (60:67:20:55:7b:ac),** |
| **3-** Which kind of request is sent through these ICMP packets? | **Ping request**  **echo request and echo reply** (used for ping) and time to live exceeded in transit (used for traceroute). |
| **4-** How many requests are sent through the host? | **4** |
| **5-** What is the IP address of your host? What is the IP address of the destination host? | **Src: 192.168.33.110, Dst: 172.217.27.36** |
| **6-** Why is it that an ICMP packet does not have source and destination port numbers? | **It was designed to communicate network-layer information between hosts and routers, not between application layer processes**.  The ICMP packet does not have source and destination port numbers because it was designed to communicate network-layer information between hosts and routers, not between application layer processes. Each ICMP packet has a "Type" and a "Code". The Type/Code combination identifies the specific message being received. Since the network software itself interprets all ICMP messages, no port numbers are needed to direct the ICMP message to an application layer process. |
| **7-** What values in the ICMP request message  differentiate this message from the ICMP reply message? | **Type 8 for request**  **Type 0 for reply** |
| **8-** Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? | **For packet 48:**  **Type: 8 (Echo (ping) request)**  **Code: 0** |
| **9-** Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? | **For Packet 50 :**  **Type: 0 (Echo (ping) reply)**  **Code: 0**  **Other fields:**  **2 bytes**  **Checksum: 0x5539 [correct]**  **2 bytes**  **Identifier (BE): 1 (0x0001)**  **Identifier (LE): 256 (0x0100)**  **2 bytes**  **Sequence number (BE): 34 (0x0022)**  **Sequence number (LE): 8704 (0x2200)** |
| **10-**Examine the packet no 56. What are the ICMP type and code numbers? Why is the IP and TCP Header included in the ICMP Header? What does these headers depict? | **Type: 3 (Destination unreachable)**  **Code: 3 (Port unreachable)**  **To report errors**  **It depicts where to send error(Network level information)**  Solution:  ICMP messages are transmitted as datagrams and consist of an IP header that encapsulates the ICMP data. A datagram, much like a packet, is a self-contained independent entity of data. Think of it as a package carrying a piece of a bigger message across the network. ICMP packets are IP packets with ICMP in the IP data portion. ICMP messages also contain the entire IP header from the original message, so the end system knows which packet failed.  **ICMPv4 Error Messages**  Each error message includes the full IP header and the first 8 bytes of the payload. Since the beginning of the payload will contain the encapsulated higher-layer header, this means the ICMP message also carries either the full UDP header, or the first 8 bytes of the TCP header. In both cases, the [source and destination port numbers](http://www.tcpipguide.com/free/t_TCPIPPortsTransportLayerTCPUDPAddressing.htm) are part of what is included. |

Page **4** of **4**