Lab08: Working with UDP and TCP Protocols
CIS 457
Due by:11/8/ 2016
Total Points: 25 Points
Submission format: hardcopy

Lab Objective
In this lab, we'll explore several aspects of the TCP and UDP protocols. The lab is organized in three parts as follows:

Part I: Working with UDP Protocol (3 Points)
In this section, we'll take a quick look at the UDP transport protocol. As we saw in Chapter 3 of the text, UDP is a streamlined, no-frills protocol. You may want to re-read the UDP section in the text before doing this lab. Download the packet trace named "UDP" from Blackboard and answer the following questions:

1. Select the first UDP packet in the trace. From this packet, determine how many fields there are in the UDP header. Name these fields.
2. By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.
3. The value in the Length field is the length of what? Verify your claim with the captured UDP packet.
4. What is the maximum number of bytes that can be included in a UDP payload?
5. What is the largest possible source port number?
6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you'll need to look into the Protocol field of the IP datagram containing this UDP segment.

Part II: Retransmission in TCP (8 Points)

The **pcattcp_retrans_t** wireshark file is captured after having the client on the desktop machine **(192.168.0.100)** writes 50 buffers with each buffer contains 1000 Bytes to the laptop machine **(192.168.0.102).** Answer the following questions:

1. How many TCP segments are sent from port 4480 on the desktop machine to port 5001 on the laptop machine? Write a display filter to isolate this side of the connection and use it to answer the question for the **pcattcp_retrans_t.cap**.
2. Examine the suspected retransmissions that are identified by Wireshark in the **pcattcp_retrans_t.cap**. (Hint: Use the filter tcp.analysis.retransmission). List the retransmitted packets' assigned number by Wireshark and their associated sequence number. Use this given template:
   Packet Num in Pcattcp_retrans_t.cap → Sequence Numbers

3. For the first packet in the **pcattcp_retrans_t.cap** that is lost before reaching the receiver, identify the retransmitted packet(s). (Hint: Use the filter tcp.eeq == certain sequence number). Use the following template to record your answer:
   Original Packet's total #bytes Re-transmitted Packet # Associated Pkt's # of bytes
4. List the sequence number (beginning and ending of every packet) of the TCP data packets number 4, 5, 7, and 8. Investigate the packet Options field of packet # 9. Does the option field

contain a SACK value? If yes, what does this value represent and how the TCP's sending side will use and interpret this info?

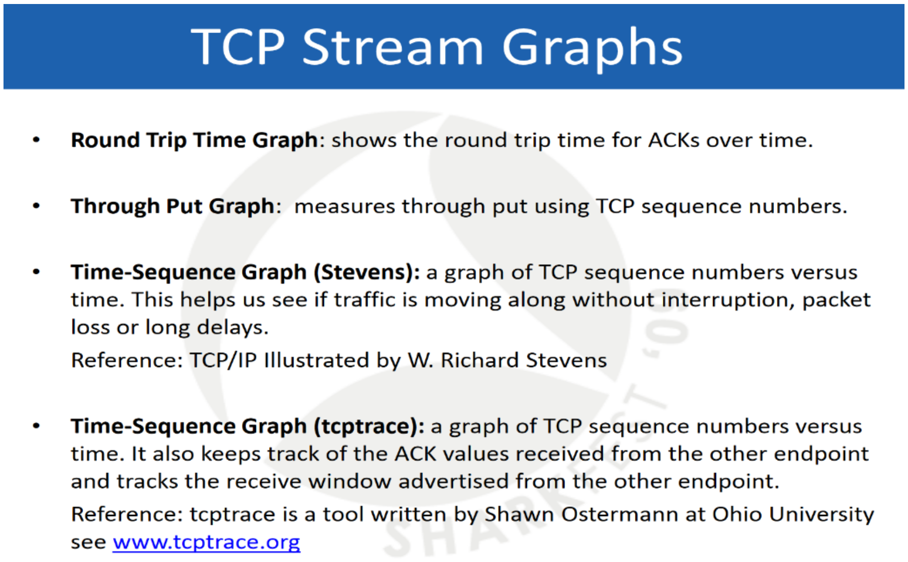
Part III: Working with TCP-TCP, TCP-UDP and UDP-UDP connections (14 pts)

In this section, two streams are sharing the network bandwidth and our objective is to investigate how these streams live together and share the bandwidth. Use, the files **tcp_tcp.cap**, **tcp_udp.cap** and **udp_udp.cap** to answer the following questions:

5. In **tcp_udp.cap**, how many TCP packets are sent during the time that the UDP transmission is actively sending data packets? Write display filters that help you identify these packets.
6. In **tcp_udp.cap**, how many bytes of data are transmitted by TCP during the time that the UDP transmission is active? How did you determine your answer?

7. In **tcp_tcp.cap**, examine the Time-Sequence Graph for the first TCP connection (4421→5001). You can obtain this graph by going to the main menu of statistics→TCP stream graphs→ Time sequence (Stevens). Please, use figure 1 to get more info about the possible TCP sequence graphs available. Focus on the point at which the slope changes (from a steeper line to a little bit flatter line). What time does this occur? Does this correspond exactly with the beginning of the second TCP stream? Why do you think this is happening? Include a screen capture for the Time-Sequence Graph for each tcp connection.
8. Do you see any retransmissions in the trace **tcp_tcp.cap**? How did you determine your answer? Why do you think this is?

9. In **udp_udp.cap**, how many UDP packets from port 4445 to 5001 occur after the second UDP transmission begins? How large is the gap between these packets and all of the other packets from 4445 to 5001? What do all of these packets have in common? Why do you think this is? (Hint: the last real data packet occurs at time 0.145140 (packet 1653) from port 4445 to 5001).
10. In **udp_udp.cap**, the second UDP transmission stream is significantly more successful in acquiring bandwidth than the first. Do you think that there is a fundamental advantage to the latecomer? Why or why not? Describe a set of experiments you could perform to test your answer.
11. Examine the traces and determine what we did to limit the size of the traces when capturing.



TCP Stream Graphs

- **Round Trip Time Graph**: shows the round trip time for ACKs over time.
- **Through Put Graph**: measures through put using TCP sequence numbers.
- **Time-Sequence Graph (Stevens):** a graph of TCP sequence numbers versus time. This helps us see if traffic is moving along without interruption, packet loss or long delays.

  Reference: TCP/IP Illustrated by W. Richard Stevens
- **Time-Sequence Graph (tcptrace):** a graph of TCP sequence numbers versus time. It also keeps track of the ACK values received from the other endpoint and tracks the receive window advertised from the other endpoint.

  Reference: tcptrace is a tool written by Shawn Ostermann at Ohio University see www.tcptrace.org

Figure 1:TCP Stream graphs