Joshua Larkin

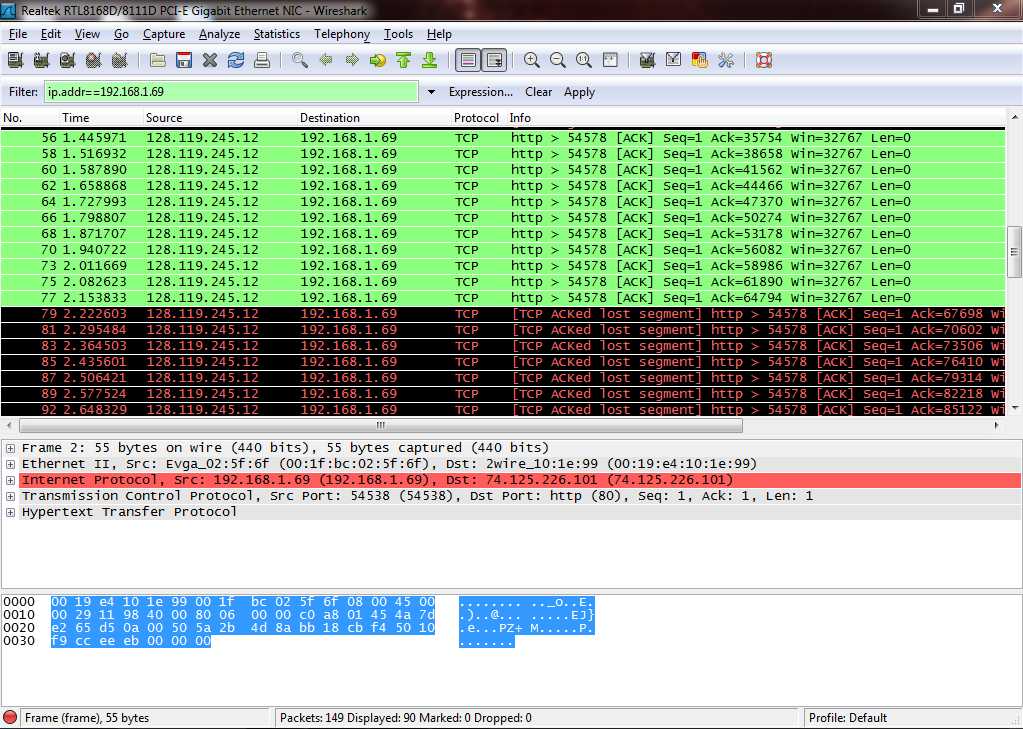
CSC 251

Net-Centric

Spring 2012

**Wireshark Lab 4: TCP**

**1. Capturing a bulk TCP transfer from your computer to a remote server:**

****

**2. A first look at the captured trace:**

**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? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows.**

*Client IP address: 192.168.1.102*

*Client port: 1161*

**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?**

*gaia.cs.umass.edu IP address: 128.119.245.12*

*paia.cs.umass.edu port: 80*

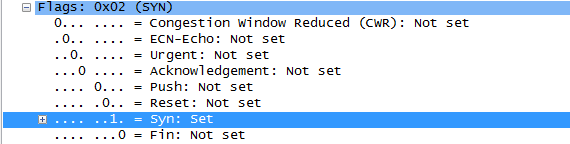
**3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?**

*Client IP address: 192.168.1.69*

*Client port: 54583*

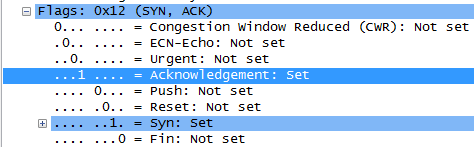
**4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?**

*The sequence number is 0. Under flags, the SYN flag is set to 1.*

****

**5. 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? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?**

*The sequence number is 0. The ACKnowledgement field i 1. gaia.cs.umass.edu determined the value by comparing checksums. The flags again determine the kind of segment it is. This time both SYN and ACK are set to 1:*

****

**6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.**

*Sequence number 1.*

**7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see page 249 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 249 for all subsequent segments.**

*Segment 1:*

*Sequence number: 1, Time Sent: 0.026477 ACK received: 0.053937*

*RTT: 0.027460 EstimatedRTT: 0.02746*

*Segment 2:*

*Sequence number: 566, Time Sent: 0.041737 ACK received: 0.077294*

*RTT: 0.035557 EstimatedRTT: 0.035557*

*Segment 3:*

*Sequence number: 2026, Time Sent: 0.054026 ACK received: 0.124085*

*RTT: 0.070059 EstimatedRTT: 0.070059*

*Segment 4:*

*Sequence number: 3486, Time Sent: 0.054690 ACK received: 0.169118*

*RTT: 0.114428 EstimatedRTT: 0.114428*

*Segment 5:*

*Sequence number: 4946, Time Sent: 0.077405 ACK received: 0.217299*

*RTT: 0.139894 EstimatedRTT: 0.139894*

*Segment 6:*

*Sequence number: 6406, Time Sent: 0.078157 ACK received: 0.267802*

*RTT: 0.189645 EstimatedRTT: 0.189645*

**8. What is the length of each of the first six TCP segments?**

*Segment 1: 565*

*Segment 2: 1460*

*Segment 3: 1460*

*Segment 4: 1460*

*Segment 5: 1460*

*Segment 6: 1460*

**9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?**

*5840 bytes. The sender is never throttled because of the size of the receiver buffer space. The packet lengths are limited to 1460 bytes and the receive buffer for the server is able to keep up with processing the incoming packets.*

**10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?**

*There are no retransmitted packets. I checked for any duplicate ACKS from the server.*

**11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 257 in the text).**

*It typically acknowledges every 1460 bytes. If the next in-order segment arrives while the server is waiting to send its ACK from the last packet, it will send one ACK for both. This occurs in the wireshark trace:*

****

*Each of these three packets is responding to two from the client representing 2920 bytes sent, as can be seen from the ACKS.*

**12. What is the throughput (bytes transferred per unit time) for the TCP connection?**

**Explain how you calculated this value.**

*23.4 KB/s. This is calculated by wireshark's statistics page but is based off of the equation of page 289. average throughput =(0.75W)/RTT where W is the receive window and RTT is the round trip time.*

**4. TCP congestion control in action:**

**13. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP’s slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we’ve studied in the text.**

*The slowstart phase begins at the beginning of transmission and levels out at about 0.3 s. Ideally the slowstart would continue with no packet loss until the TCP connection reached its full utilization, but in real world scenarios this level of efficiency is not possible to achieve.*