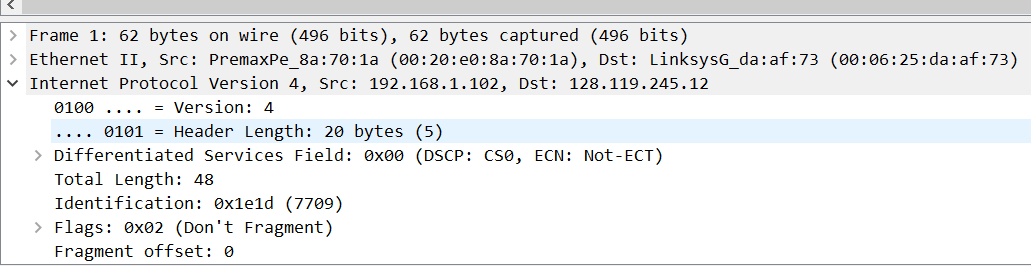
Lab08-Wireshark TCP Lab

Part 1 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”



IP address: 192.168.1.102

TCP port number: 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?

Destination computer: gaia.cs.umass.edu

IP address: 128.119.245.12

TCP port number: 80

Part 2 TCP Basics

***1. 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?***

Sequence number of the TCP SYN segment is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu. The value is 0 in this trace.

***2. 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?***

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.

***3. 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.***

No. 4 segment is the TCP segment containing the HTTP POST command. The sequence number of this segment has the value of 1.

***4. 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 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 242 for all subsequent segments.***

The HTTP POST segment is considered as the first segment. Segments 1 – 6 are No. 4, 5, 7, 8, 10, and 11 in this trace respectively. The ACKs of segments 1 – 6 are

No. 6, 9, 12, 14, 15, and 16 in this trace. Segment 1 sequence number: 1

Segment 2 sequence number: 566

Segment 3 sequence number: 2026

Segment 4 sequence number: 3486

Segment 5 sequence number: 4946

Segment 6 sequence number: 6406

The sending time and the received time of ACKs are tabulated in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sent Time | ACK Time | RTT(sceonds) |
| Segment1 | 0.026477 | 0.053937 | 0.02746 |
| Segment2 | 0.041737 | 0.077294 | 0.035557 |
| Segment3 | 0.054026 | 0.124085 | 0.070059 |
| Segment4 | 0.054690 | 0.169118 | 0.11443 |
| Segment5 | 0.077405 | 0.217299 | 0.13989 |
| Segment6 | 0.078157 | 0.267802 | 0.18964 |

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

Length of the first TCP segment (containing the HTTP POST): 565 bytes Length of each of the other five TCP segments: 1460 bytes (MSS)

***6. 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?***

The minimum amount of buffer space (receiver window) advertised at gaia.cs.umass.edu for the entire trace is 5840 bytes, which shows in the first acknowledgement from the server. This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. The sender is never throttled due to lacking of receiver buffer space by inspecting this trace.

***7. 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 segments in the trace file. We can verify this by checking the sequence numbers of the TCP segments in the trace file. In the Time-Sequence-Graph (Stevens) of this trace, all sequence numbers from the source (192.168.1.102) to the destination (128.119.245.12) are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighboring segments.

8. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment.

***9. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.***

The computation of TCP throughput largely depends on the selection of averaging time period. As a common throughput computation, in this question, we select

the average time period as the whole connection time. Then, the average throughput for this TCP connection is computed as the ratio between the total amount data and the total transmission time. The total amount data transmitted can be computed by the difference between the sequence number of the first TCP segment (i.e. 1 byte for No. 4 segment) and the acknowledged sequence number of the last ACK (164091 bytes for No. 202 segment). Therefore, the total data are 164091 - 1 = 164090 bytes. The whole transmission time is the difference of the time instant of the first TCP segment (i.e.,

0.026477 second for No.4 segment) and the time instant of the last ACK (i.e., 5.455830 second for No. 202 segment). Therefore, the total transmission time is 5.455830 - 0.026477 = 5.4294 seconds. Hence, the throughput for the TCP connection is computed as 164090/5.4294 = 30.222 KByte/sec.