

Project #1: Wireshark Lab 1: TCPProfessor: *Dr. Nasir Ghani*

Author: Juan Sebastián Caballero

Note: *All answers must contain a clear description related with the answer and as many screenshots of the process as possible.*

Performance Evaluation and Pulse Design for Ultra-Wideband Communications

This lab studies the behavior of the IP transport control protocol (TCP). Namely, this will be done by analyzing a trace of TCP segments that are sent and received in transferring a 150KB file from a client's computer to a remote server. The use of sequence and acknowledgement numbers will be investigated for providing reliable data transfer. The TCP slow-start and congestion avoidance algorithms will also be investigated as well as receiver-based flow control. TCP connection setup will also be briefly reviewed along with the performance (throughput and round-trip time) of the TCP connection between a client computer and the server. Note that this lab is presented within the context of the Microsoft Windows operating system.

Submission Guidelines

1. All submissions will be turned into Turnitin for the plagiarism check.
2. All the submissions need to have findings which highlight the work.
3. Student should observe that they should not include any questions and material from this file in the submission.
4. Students should just mention the question number and complete the answer and mention their findings.
5. Students with no proper findings evidence for the questions answered will be graded zero and subjected for the further proof for submission.
6. All the submissions are subjected to grading by the professor.
7. It is suggested that the students should provide their own work. If found plagiarized, the grade will be marked as F and will be notified for further proceedings to department.
8. Students are suggested to reach out to the TAs whenever needed and advised not to wait till the submission date.
9. All submissions must be done on Canvas. You are required to submit a single file in either word or pdf form. Answer all the questions, providing the appropriate screenshots where asked. Mention your Name and student number at the top right of the page in the header.

Lab Guidelines

1. Download the Wireshark file trace named "tcp_trace" from the Canvas course section and open it.
2. Before analyzing the behavior of the TCP connection in detail, take a further look at this trace. Filter the packets displayed in the Wireshark window by entering "tcp" into the display filter specification window.

This will show a series of TCP and HTTP messages between the client computer and the website server at `gaia.cs.umass.edu`. You should see the initial three-way handshake containing a SYN message. You should also see an HTTP POST message. Depending on the version of Wireshark, you might see a series of “HTTP Continuation” messages being sent from your computer to `gaia.cs.umass.edu`. You may see “[TCP segment of a reassembled PDU]” in the Info column to indicate this TCP segment contains data of an upper layer protocol message (HTTP). You should also see TCP ACK segments returned from `gaia.cs.umass.edu` to your computer.

Answer the following questions. Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printouts to explain your answers. To print a packet, use File->Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

Answer the following questions. Include screenshots of the packets.

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, note that it is easier to select a 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”).
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?

Since the focus of this lab is on TCP rather than HTTP, change Wireshark’s “listing of captured packets” window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To do this, select Analyze->Enabled Protocols and uncheck the HTTP box and select OK. You should now see a Wireshark window as follows (a series of TCP segments sent between a client computer and `gaia.cs.umass.edu`).

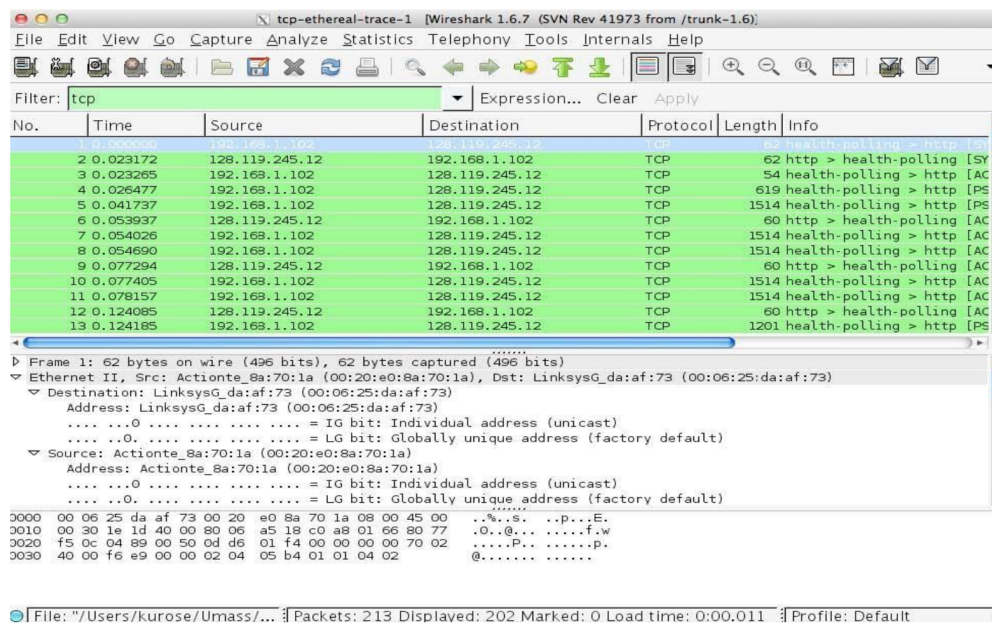


Figure 1.1: Wireshark capture example.

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?
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?
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 will need to check the packet content field at the bottom of the Wireshark window, i.e., for a segment with a “POST” within its DATA field.
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)? When 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? Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph.
5. What is the length of each of the first six TCP segments?
6. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
7. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
8. How much data does the receiver typically acknowledge in an ACK?
9. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

TCP Congestion Control in Action

The amount of data sent per unit time from the client to the server is now studied. Rather than (tediously) calculating this from the raw data in the Wireshark window, you can use one of Wireshark’s TCP graphing utilities to plot data, i.e., Time-Sequence-Graph(Stevens). Namely, select a TCP segment in the Wireshark’s “listing of captured-packets” window. Then select the Statistics->TCP Stream Graph->Time-Sequence-Graph(Stevens) option. You should see a plot that looks similar to the following plot:

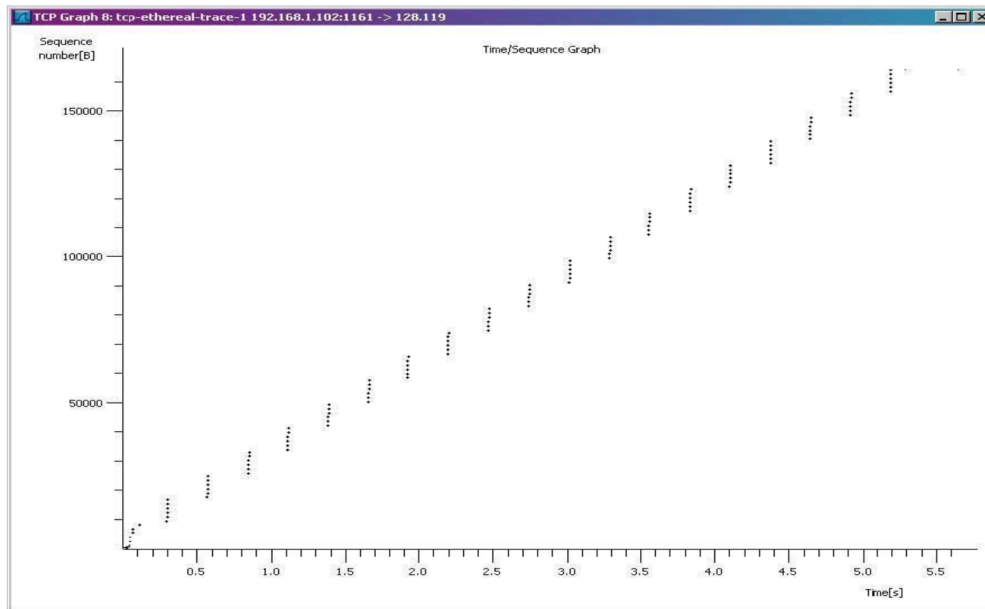


Figure 1.2: Time-Sequence Graph example

Here, each dot represents a TCP segment sent, plotting the sequence number of the segment versus the time at which it was sent. Note that a set of dots stacked above each other represents a series of packets that were sent back-to-back by the sender.

10. Use the Time-Sequence-Graph 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 slow start 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

Advanced TCP Concepts and Analysis

This section aims to deepen your understanding of TCP's mechanisms by exploring theoretical foundations and applying practical Wireshark analysis. Each question is designed to challenge your comprehension of TCP features and how they influence network communication reliability and efficiency.

1. **Theoretical:** TCP uses checksums to detect data corruption. Explain in detail the function and importance of the TCP checksum. How does it ensure reliable data transmission?
2. **Theoretical:** TCP communication relies on flags (URG, ACK, PSH, RST, SYN, FIN) for various operations. Discuss the significance of each flag, providing real-world scenarios where different flags would be appropriately used.
3. **Theoretical:** Nagle's algorithm is a key optimization in TCP. Describe how it works, and discuss scenarios where it can either enhance or degrade network performance.
4. **Practical:** Unexpected termination of TCP connections can impact network performance. Using Wireshark, simulate a scenario where a TCP connection is abruptly terminated with the RST flag. Provide screenshots and analyze the observed effects.
5. **Practical:** Window scaling is vital for efficient TCP data transmission over high-speed networks. Capture a TCP connection using Wireshark and analyze the impact of window scaling on throughput and overall TCP efficiency.

6. **Extra Credit:** TCP Fast Retransmit is an optimization technique used to recover from packet losses more quickly. Explain how TCP Fast Retransmit operates and illustrate its impact using a practical example captured in Wireshark.