ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

# MẠNG MÁY TÍNH THỰC HÀNH - CO3094

**Báo cáo:**

# Lab3a

**Giảng viên hướng dẫn:** Vũ Thành Tài

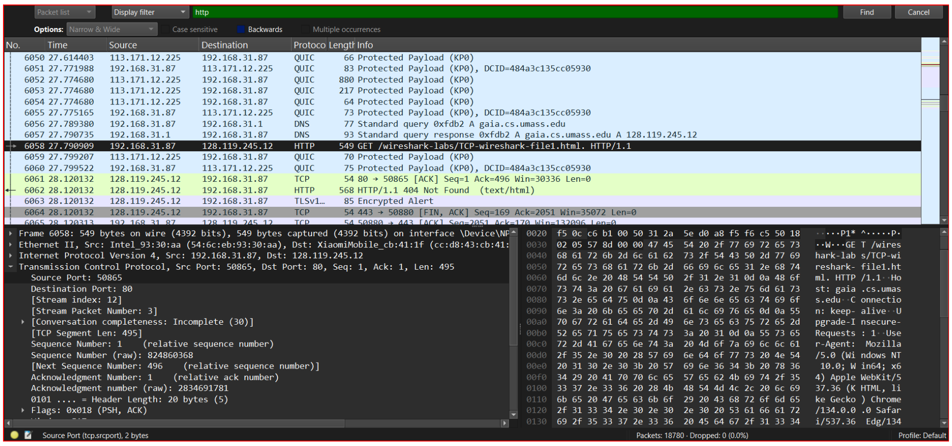
**Sinh viên:** Lê Đức Cường

Thành phố Hồ Chí Minh, tháng 3 năm 2025

# 1 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? 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.**

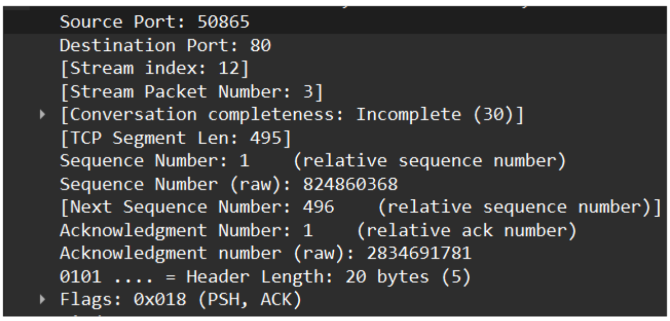
The source IP address was 192.168.31.87 using source port 50865.



# 2 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?**

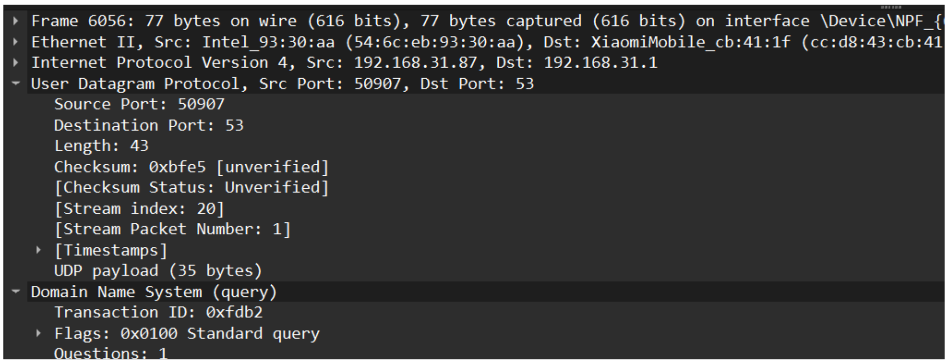
The destination IP address is 128.119.245.12 receiving on port 80



# 3 Question 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?**

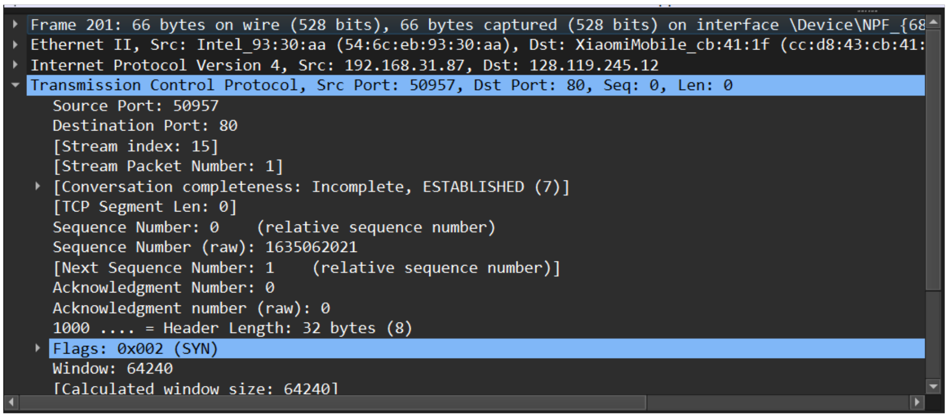
My IP address source is 192.168.31.87 sending on port 50907.




# 4 Question 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 of the segment used to initiate the TCP connection is 0.

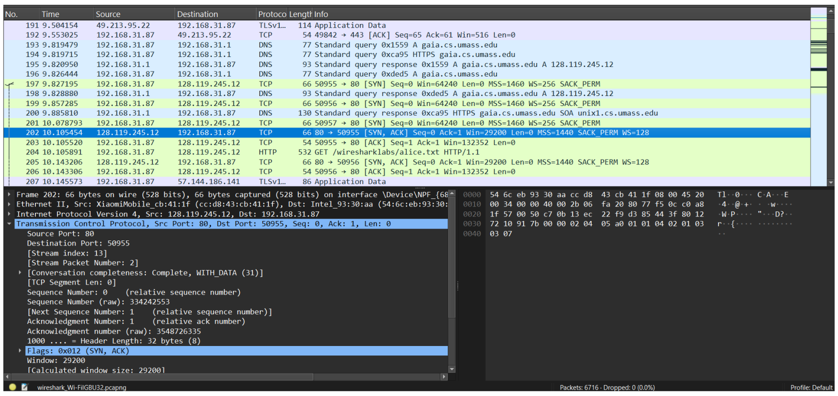


# 5 Question 5

**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 of the SYNACK segment is 0.

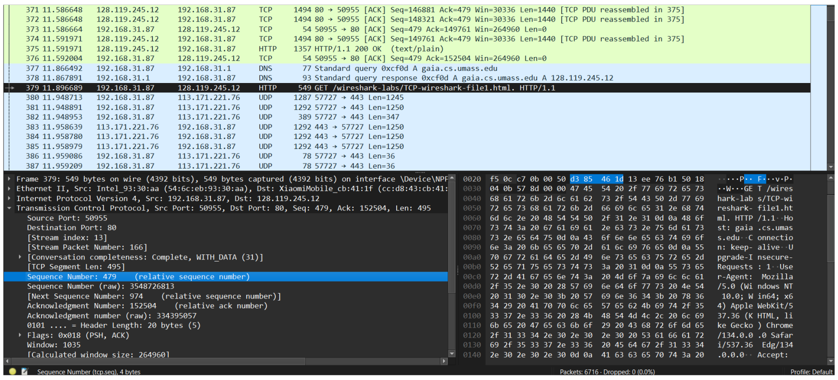
The value of the acknowledgement field is 1. This value is determined by the initial sequence number +1.



# 6 Question 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.**

The sequence number of the TCP segment containing the HTTP Post Command is 479.

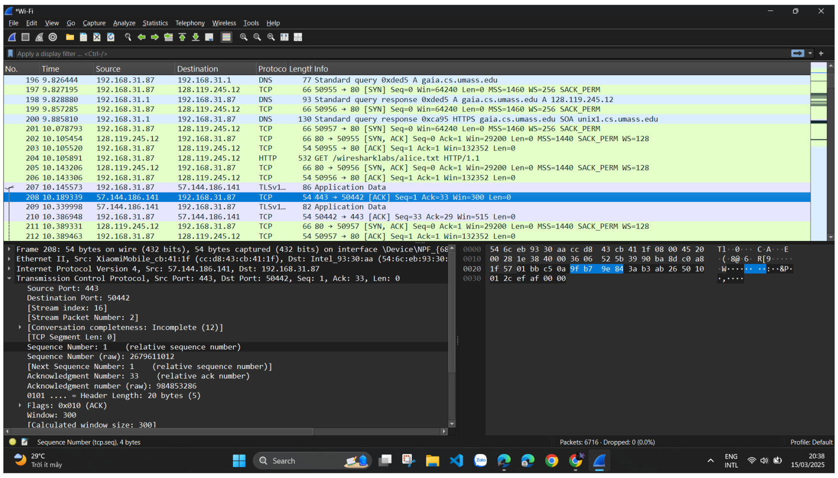


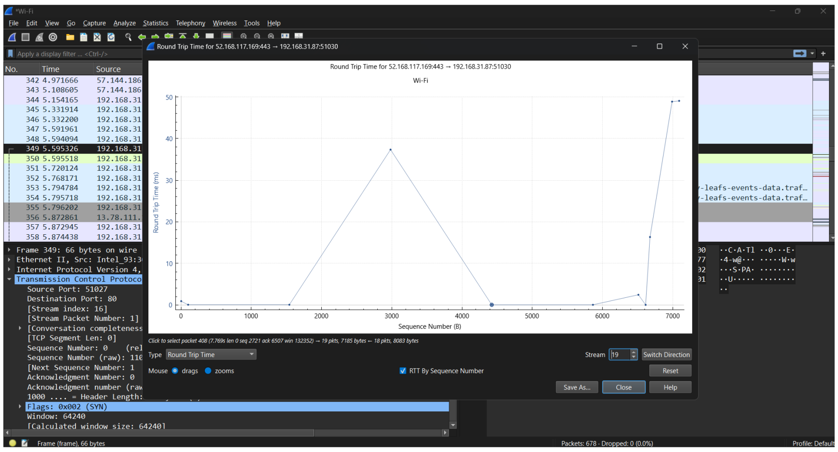
# 7 Question 7

**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 Section 3.5.3, page 242 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 242 for all subsequent 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.***





# 8 Question 8

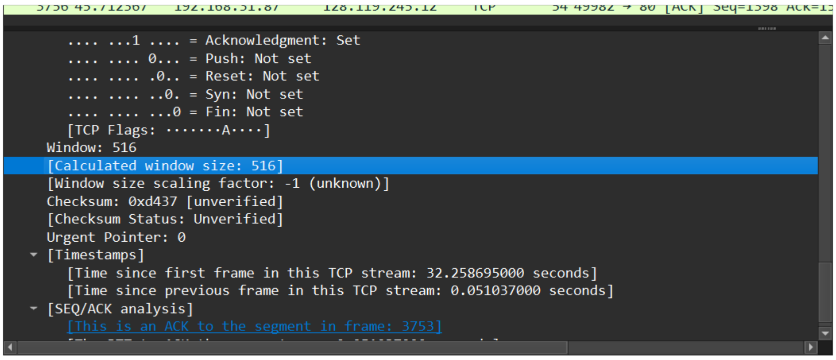
**What is the length of each of the first six TCP segments?**

The length of the first TCP segment is The length of each of the first TCP segment is 708 bytes.

# 9 Question 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?**

The minimum amount of available buffer space advertised at the received for the entire trace is indicated first ACK from the server, its value is 516 bytes (shown in above figure).




# 10 Question 10

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

No, there were never any retransmissions of segments. This is clear because an old acknowledgement number was never sent again to ask for the previous packets to be resent.

# 11 Question 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 250 in the text).**

The difference between the acknowledged sequence numbers of two consecutive ACKsindicates the data received by the server between these two ACKs.

# 12 Question 12

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

To calculate the throughput of a TCP connection, we use the following formula:

$$\text{Throughput} = \frac{\text{Total Bytes Transferred}}{\text{Total Time Taken}}$$

The alice.txt on the hard drive is 152,138 bytes, and the download time is 1.578736000 (First TCP segment) - 0.271257000 (last ACK) = 1.307479 second. Therefore, the throughput for the TCP connection is computed as 152,138/1.307479=116359.803867 bytes/s.

# 13 Question 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.**

Before the end of the slow start phase, the application already stops transmission temporally.

# 14 Question 14

**Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu**

The idealized behavior of TCP in the text assumes that TCP senders are aggressive in sending data. Too much traffic may congest the network; therefore, TCP senders should follow the AIMD algorithm so that when they detect network congestion (i.e., packet loss), their sending window size should drop down.