

**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

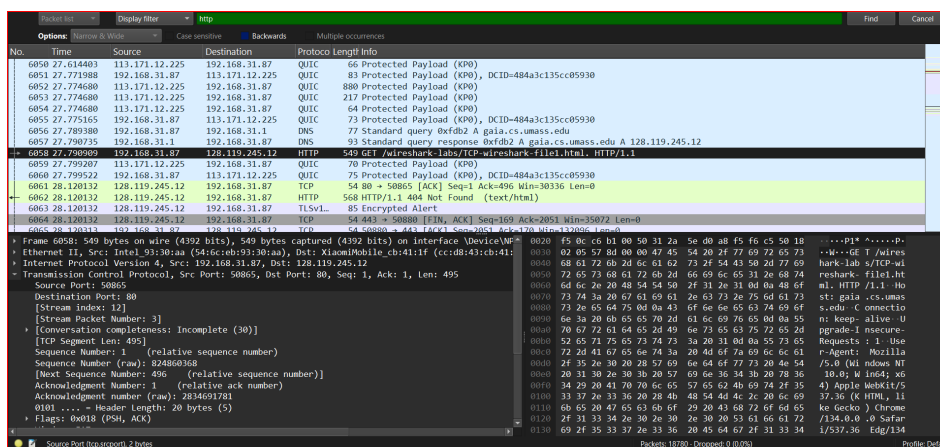
Thành phố Hồ Chí Minh, tháng 3 năm 2025



## 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

```
Source Port: 50865
Destination Port: 80
[Stream index: 12]
[Stream Packet Number: 3]
▶ [Conversation completeness: Incomplete (30)]
[TCP Segment Len: 495]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 824860368
[Next Sequence Number: 496 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2834691781
0101 .... = Header Length: 20 bytes (5)
▶ Flags: 0x018 (PSH, ACK)
```



### 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.

```
Frame 6056: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \Device\NPF_{...}
Ethernet II, Src: Intel_93:30:aa (54:6c:eb:93:30:aa), Dst: XiaomiMobile_cb:41:1f (cc:d8:43:cb:41:1f)
Internet Protocol Version 4, Src: 192.168.31.87, Dst: 192.168.31.1
User Datagram Protocol, Src Port: 50907, Dst Port: 53
  Source Port: 50907
  Destination Port: 53
  Length: 43
  Checksum: 0xbfe5 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 20]
  [Stream Packet Number: 1]
  [Timestamps]
  UDP payload (35 bytes)
Domain Name System (query)
  Transaction ID: 0xfdb2
  Flags: 0x0100 Standard query
  Questions: 1
```

### 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.

```
Frame 201: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{68...}
Ethernet II, Src: Intel_93:30:aa (54:6c:eb:93:30:aa), Dst: XiaomiMobile_cb:41:1f (cc:d8:43:cb:41:1f)
Internet Protocol Version 4, Src: 192.168.31.87, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 50957, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 50957
  Destination Port: 80
  [Stream index: 15]
  [Stream Packet Number: 1]
  [Conversation completeness: Incomplete, ESTABLISHED (7)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 1635062021
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 ... = Header Length: 32 bytes (8)
  Flags: 0x002 (SYN)
  Window: 64240
  [Calculated window size: 64240]
```



## 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.

No.	Time	Source	Destination	Protocol	Length	Info
191	9.584154	49.213.95.22	192.168.31.87	HTTP	114	Application data
192	9.553025	192.168.31.87	49.213.95.22	TCP	54	49042 → 443 [ACK] Seq=61 Min=516 Len=0
193	9.519479	192.168.31.87	192.168.31.1	DNS	77	Standard query 0x1559 A gaia.cs.umass.edu
194	9.819715	192.168.31.87	192.168.31.1	DNS	93	Standard query 0xca95 HTTPS gaia.cs.umass.edu
195	9.820950	192.168.31.1	192.168.31.87	DNS	93	Standard query response 0x1559 A gaia.cs.umass.edu A 128.119.245.12
196	9.826444	192.168.31.87	192.168.31.1	DNS	77	Standard query 0xede5 A gaia.cs.umass.edu
197	9.827195	192.168.31.87	128.119.245.12	TCP	66	58955 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
198	9.828880	192.168.31.1	192.168.31.87	DNS	93	Standard query response 0xede5 A gaia.cs.umass.edu A 128.119.245.12
199	9.857285	192.168.31.87	128.119.245.12	TCP	66	58956 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
200	9.885810	192.168.31.1	192.168.31.87	DNS	130	Standard query response 0xca95 HTTPS gaia.cs.umass.edu 500 unkl.cs.umass.edu
201	10.070293	192.168.31.87	128.119.245.12	TCP	66	58957 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
202	10.105454	128.119.245.12	192.168.31.87	TCP	66	80 → 58955 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440 SACK_PERM WS=128
203	10.105520	192.168.31.87	128.119.245.12	TCP	54	58955 → 80 [ACK] Seq=1 Ack=1 Win=13252 Len=0
204	10.105891	192.168.31.87	128.119.245.12	HTTP	516	GET /Airessharklab/aiics-test HTTP/1.1
205	10.143206	128.119.245.12	192.168.31.87	TCP	66	80 → 58956 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440 SACK_PERM WS=128
206	10.143306	192.168.31.87	128.119.245.12	TCP	54	58956 → 80 [ACK] Seq=1 Ack=1 Win=13252 Len=0
207	10.145573	192.168.31.87	57.144.186.141	TLSv1	86	Application Data

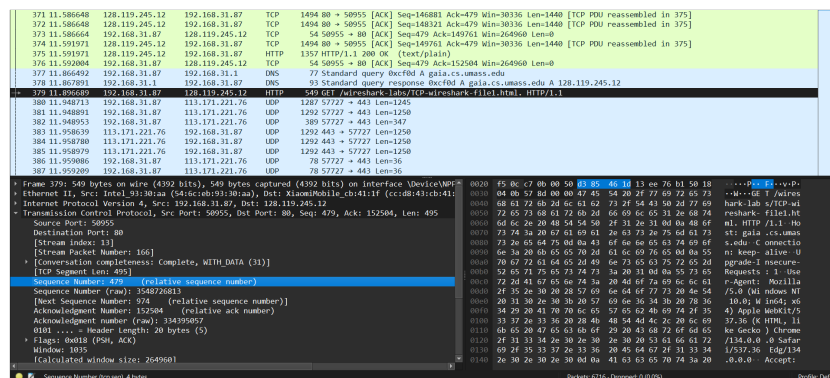
Frame 202: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface DeviceVNF [68]  
Ethernet II, Src: XilinxVNF000000000000, Dst: Intel\_E81B:20:AA (5A:6C:EB:93:20:AA)  
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.31.87  
Transmission Control Protocol, Src Port: 80, Dst Port: 58955, Seq: 0, Ack: 1, Len: 0  
Source Port: 80  
Destination Port: 58955  
[Stream index: 12]  
[Stream Packet Number: 2]  
[Conversation completeness: Complete, WITH\_DATA (31)]  
[TCP Segment Len: 0]  
Sequence Number: 0 (relative sequence number)  
Sequence Number (raw): 58424559  
[Next Sequence Number: 1 (relative sequence number)]  
Acknowledgment Number: 1 (relative ack number)  
Acknowledgment Number (raw): 3580726325  
1000 ..... Header length: 32 bytes (8)  
[Flags: SYN, ACK]  
Window: 29200  
[Calculated window size: 29200]



## 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.



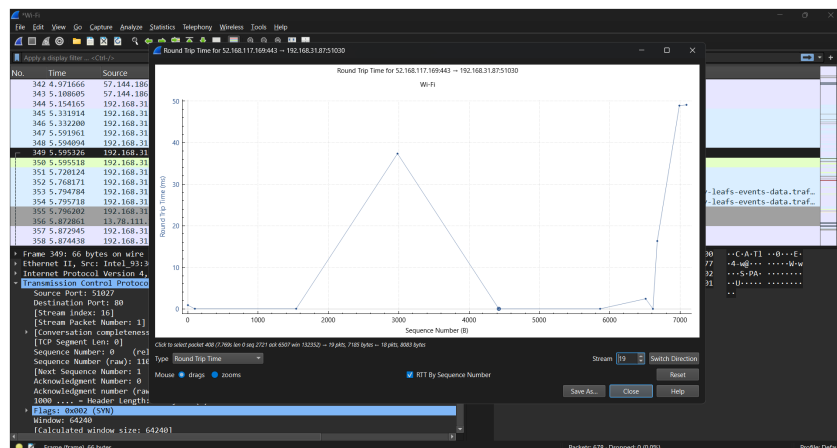
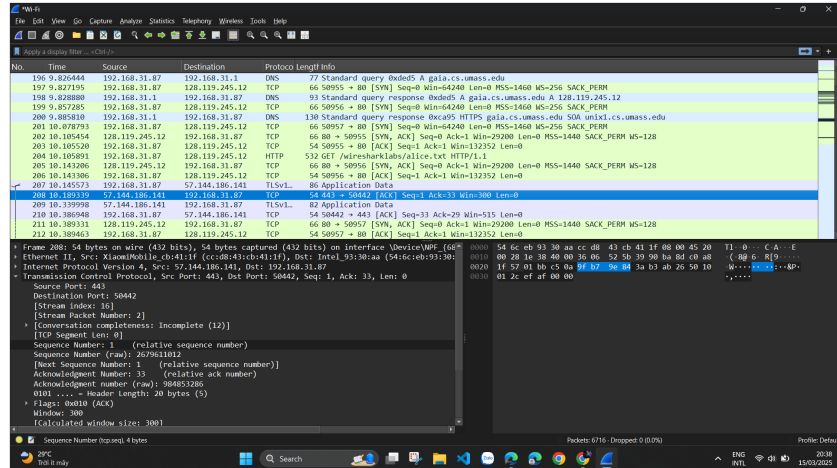
## 7 Question 7

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

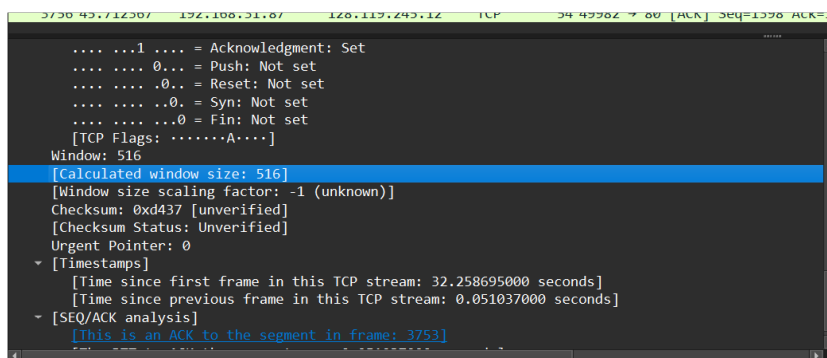
What is the length of each of the first six TCP segments?

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).



```
3730 45.712307 192.168.21.8/ 128.112.242.12 TCP 34 45782 → 80 [ACK] Seq=1326 Ack=1326
... .. = Acknowledgment: Set
... .. = Push: Not set
... .. = Reset: Not set
... .. = Syn: Not set
... .. = Fin: Not set
[TCP Flags: .....A....]
Window: 516
[Calculated window size: 516]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xd437 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
  [Timestamps]
    [Time since first frame in this TCP stream: 32.258695000 seconds]
    [Time since previous frame in this TCP stream: 0.051037000 seconds]
  [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 3753]
```

## 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 ACKs indicates 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:

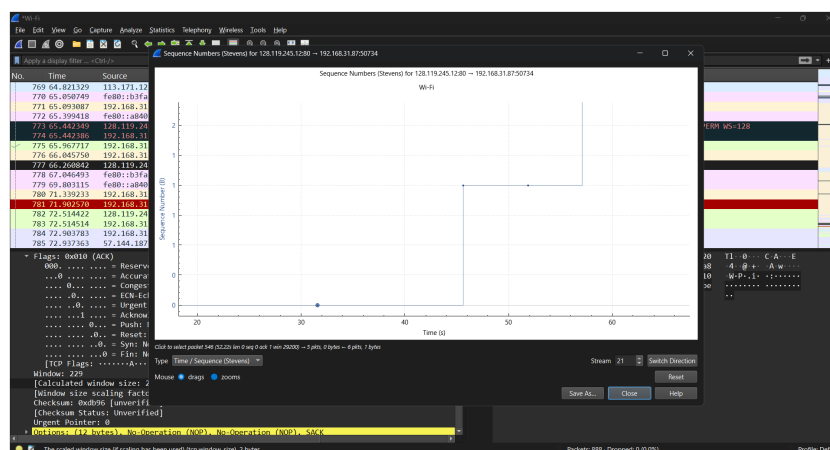
$$\text{Throughput} = \frac{\text{Total Bytes Transferred}}{\text{Total Time Taken}}$$

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.