GSI023 – Redes de Computadores

Wireshark Lab - TCP

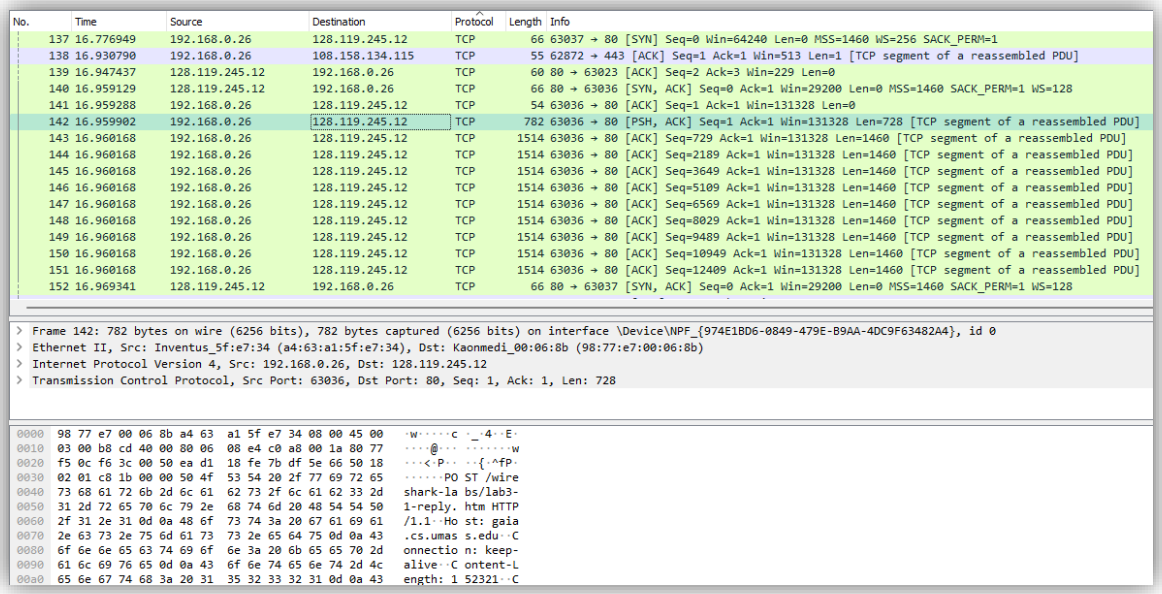
Murielly Oliveira Nascimento – 11921BSI222 – murielly.nascimento@ufu.br

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.

Computador do cliente (source), endereço IP: 192.168.0.26, número da porta TCP: 63036.

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?

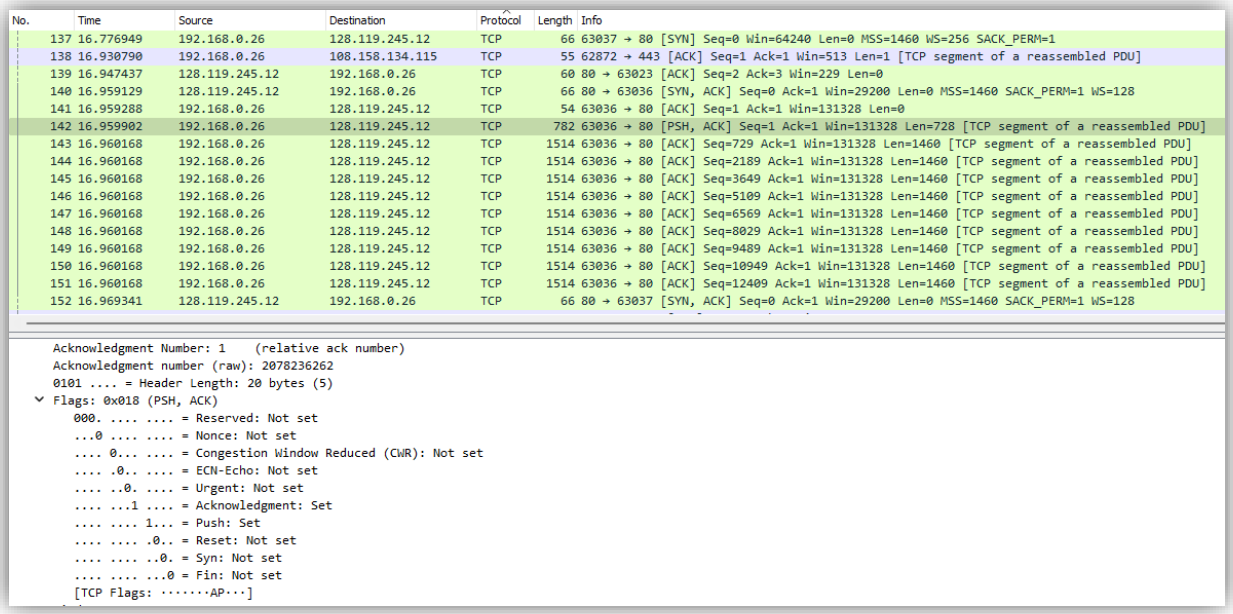
Computador de destino (gaia.cs.umass.edu), endereço IP: 128.119.245.12, número da porta TCP: 80.



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?

Computador do cliente (source), endereço IP: 192.168.0.26, número da porta TCP: 63036.

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?



No.	Time	Source	Destination	Protocol	Length	Info
137	16.776949	192.168.0.26	128.119.245.12	TCP	66	63037 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
138	16.930790	192.168.0.26	108.158.134.115	TCP	55	62872 → 443 [ACK] Seq=1 Ack=1 Win=513 Len=1 [TCP segment of a reassembled PDU]
139	16.947437	128.119.245.12	192.168.0.26	TCP	60	80 → 63023 [ACK] Seq=2 Ack=3 Win=229 Len=0
140	16.959129	128.119.245.12	192.168.0.26	TCP	66	80 → 63036 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
141	16.959288	192.168.0.26	128.119.245.12	TCP	54	63036 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
142	16.959902	192.168.0.26	128.119.245.12	TCP	782	63036 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=728 [TCP segment of a reassembled PDU]
143	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=729 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
144	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=2189 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
145	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=3649 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
146	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=5109 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
147	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=6569 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
148	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=8029 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
149	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=9489 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
150	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=10949 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
151	16.960168	192.168.0.26	128.119.245.12	TCP	1514	63036 → 80 [ACK] Seq=12409 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
152	16.969341	128.119.245.12	192.168.0.26	TCP	66	80 → 63037 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128

```
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 2078236262
0101 .... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 1... = Push: Set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
   .... .... ...0 = Fin: Not set
   [TCP Flags: ·····AP···]
```

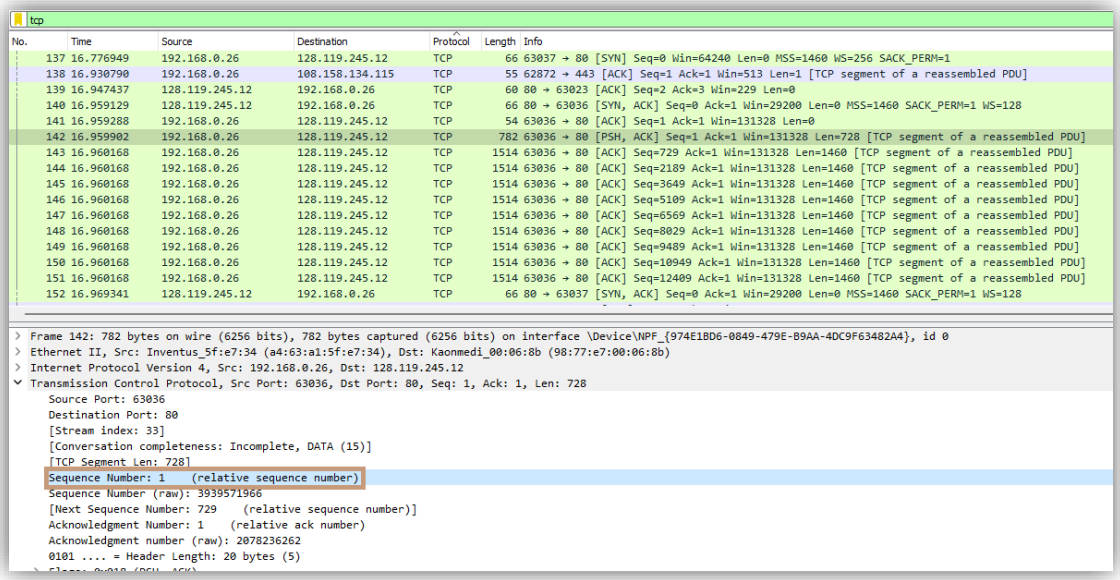
O número de sequência TCP SYN é usado para iniciar a conexão TCP entre o cliente e gaia.cs.umass.edu. O valor é 0 nesse caso.

A SYN flag em 1 indica que esse segmento é do tipo SYN.

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?

Número de sequência do segmento SYNACK de gaia.cs.umass.edu para o computador cliente em resposta ao SYN tem o valor de 0 neste rastreamento. O valor do campo de confirmação no segmento SYNACK é 1. O valor do campo de reconhecimento no segmento SYNACK é determinado por gaia.cs.umass.edu adicionando 1 ao número de sequência inicial do segmento SYN do computador cliente (ou seja, o número de sequência do segmento SYN iniciado pelo computador cliente é 0.). O sinalizador SYN e o sinalizador de confirmação no segmento são definidos como 1 e indicam que este segmento é um segmento SYNACK.

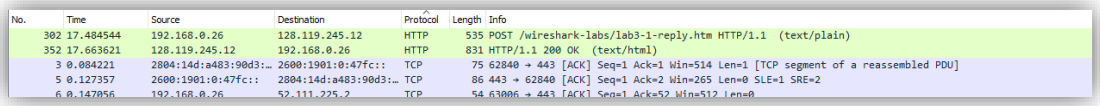
6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

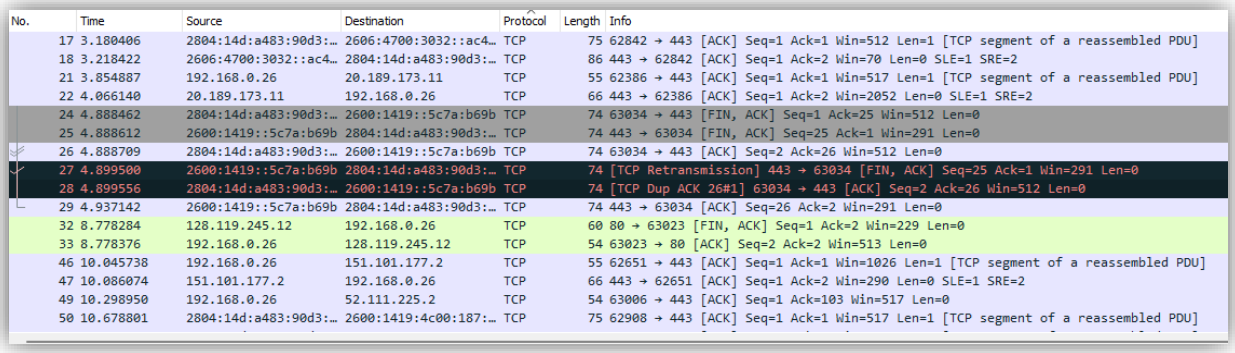


O segmento nº 4 é o segmento TCP que contém o comando HTTP POST. O número de sequência deste segmento tem o valor de 1.

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 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.**





Número de sequência do segmento 1: 302

Número de sequência do segmento 2: 352

EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

EstimatedRTT after the receipt of the ACK of segment 1:

EstimatedRTT = RTT for Segment 1 = 4.899500 second

EstimatedRTT after the receipt of the ACK of segment 2:

EstimatedRTT = 0.875 * 4.899500 + 0.125 * 0.035557 = 0.0285

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

Comprimento do primeiro segmento TCP (contendo o HTTP POST): 565 bytes Comprimento de cada um dos outros cinco segmentos TCP: 1460 bytes (MSS)

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?

A quantidade mínima de espaço de buffer (janela do receptor) anunciada em gaia.cs.umass.edu para o trace inteiro é de 5840 bytes, o que mostra na primeira confirmação do servidor. Esta janela do receptor cresce de forma constante até um máximo tamanho do buffer do receptor de 62780 bytes. O remetente nunca é estrangulado devido à falta de espaço de buffer do receptor inspecionando esse rastreamento.

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

Não há segmentos retransmitidos no arquivo de rastreamento. Podemos verificar isso por verificando os números de sequência dos segmentos TCP no arquivo de rastreamento. No TimeSequence-Graph (Stevens) deste trace, todos os números de sequência da fonte (192.168.1.102) para o destino (128.119.245.12) estão aumentando monotonamente com respeito ao tempo. Se houver um segmento retransmitido, o número de sequência deste segmento retransmitido deve ser menor do que os de seus segmentos vizinhos.

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).

ACK 1 566 566

ACK 2 2026 1460

ACK 3 3486 1460

ACK 4 4946 1460

ACK 5 6406 1460

ACK 6 7866 1460

ACK 7 9013 1147

ACK 8 10473 1460

ACK 9 11933 1460

ACK 10 13393 1460

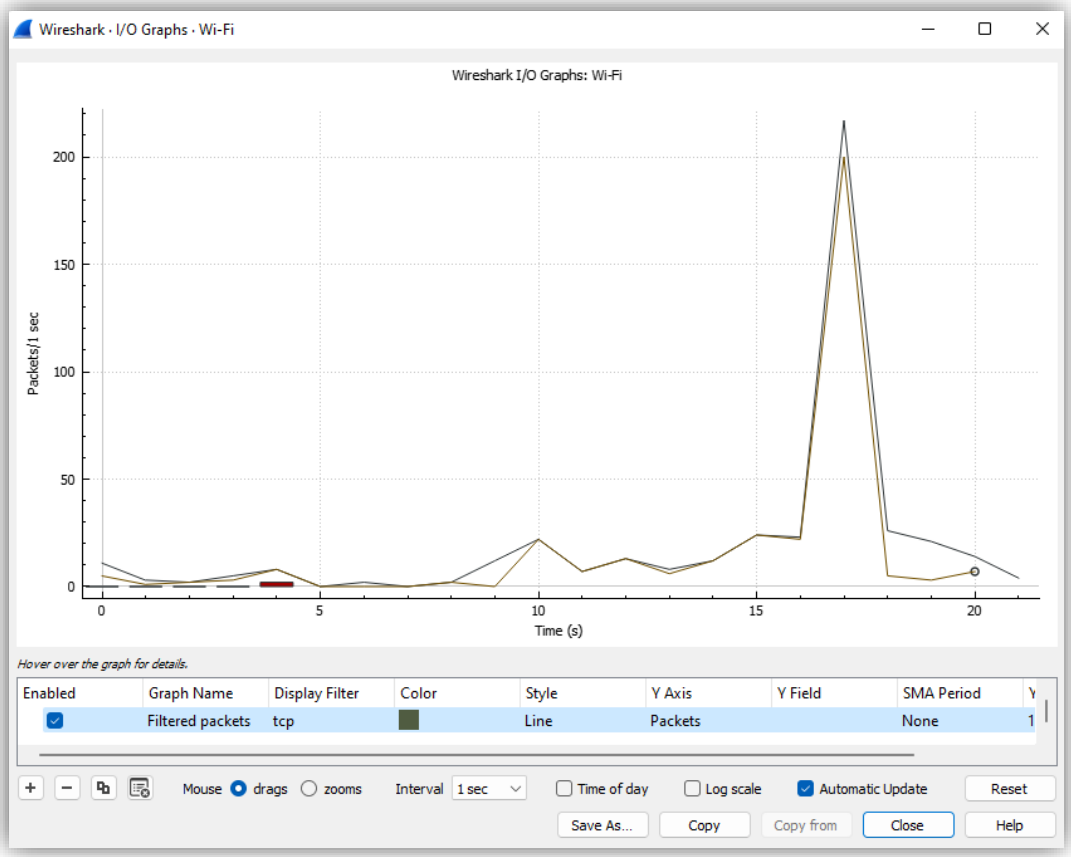
ACK 11 14853 1460

ACK 12 16313 1460

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

O cálculo da taxa de transferência TCP depende em grande parte da seleção de período de tempo médio. Como um cálculo de taxa de transferência comum, nesta questão, selecionamos o período de tempo médio como todo o tempo de conexão. Então, o rendimento médio para esta conexão TCP é calculada como a razão entre a quantidade total de dados e a quantidade total tempo de transmissão. A quantidade total de dados transmitidos pode ser calculada pela diferença entre o número de sequência do primeiro segmento TCP (ou seja, 1 byte para o segmento nº 4) e o número de sequência reconhecido do último ACK (164091 bytes para o nº 202 segmento). Portanto, os dados totais são 164.091 - 1 = 164.090 bytes. O todo tempo de transmissão é a diferença do instante de tempo do primeiro segmento TCP (ou seja, 0,026477 segundo para o segmento nº 4) e o instante de tempo do último ACK (ou seja, 5,455830 segundo para o segmento nº 202). Portanto, o tempo total de transmissão é 5,455830 - 0,026477 = 5,4294 segundos. Assim, a taxa de transferência para a conexão TCP é calculada como 164090/5,4294 = 30,222 KByte/s.

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.



O congestionamento começa no segundo 200 quando os pacotes são enviados.

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

O congestionamento começa quando é transferido o arquivo alice.txt