

## Lab 5 - TCP

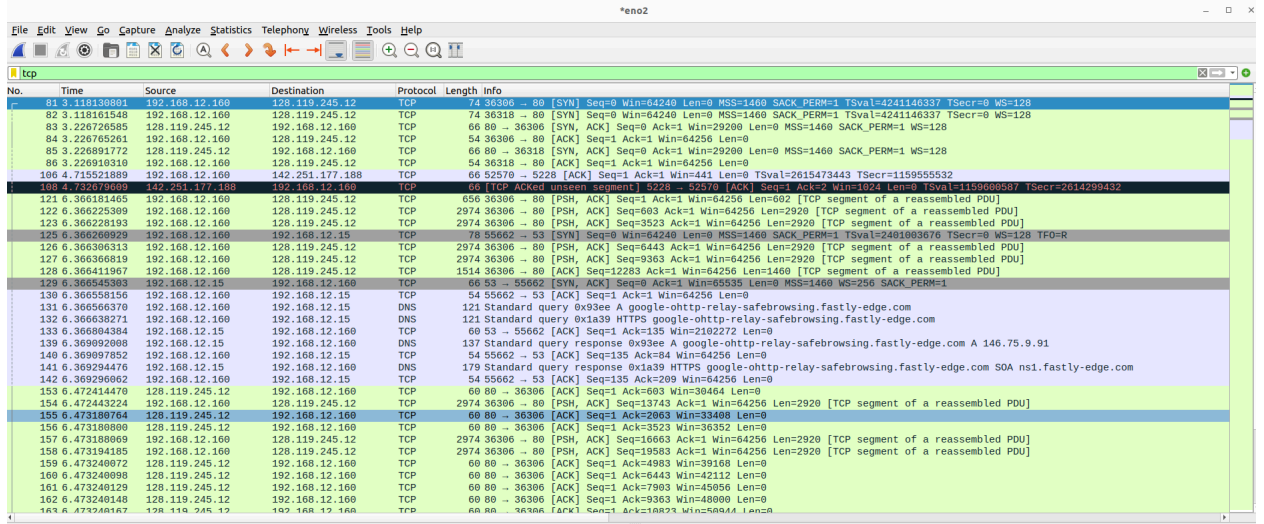
```

+ Frame 270: 2777 bytes on wire (22216 bits), 2777 bytes captured (22216 bits) on interface eno2, id 0
+ Ethernet II, Src: 10:7c:61:3e:d4:2c (10:7c:61:3e:d4:2c), Dst: 1a:c2:41:30:b5:90 (1a:c2:41:30:b5:90)
+ Internet Protocol Version 4, Src: 192.168.12.160, Dst: 128.119.245.12
+ Transmission Control Protocol, Src Port: 36306, Dst Port: 80, Seq: 146603, Ack: 1, Len: 2723
  Source Port: 36306
  Destination Port: 80
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 2723] (relative sequence number)
  Sequence Number: 146603 (relative sequence number)
  Sequence Number (raw): 919048031
  [Next Sequence Number: 149026 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 3423406165
  0101 ... = Header Length: 20 bytes (5)
  + Flags: 0x010 (PSH, ACK)
  Window: 502
  [Calculated window size: 64256]
  [Window size scaling factor: 128]
  Checksum: 0x0db8 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  + [Timestamps]
  + [SEQ/ACK analysis]
  TCP payload (2723 bytes)
  TCP segment data (2723 bytes)
+ [55 Reassembled TCP Segments (149325 bytes): #121(602), #122(2920), #123(2920), #126(2920), #127(2920), #128(1460), #154(2920), #157(2920), #158(2920), #166(1460), #167(2920), #168(2920), #169(2920), #170(2920)]
+ Hypertext Transfer Protocol
  POST /wirespark-labs/lab3-1-reply.htm HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  Content-Length: 148723\r\n
  Cache-Control: max-age=0\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36\r\n
  Origin: null\r\n

```

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the alice.txt 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 IP of the client computer is 192.168.12.160 and the TCP port number is 36306. ✓
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 IP address of gaia.cs.umass.edu is 128.119.245.12 and the port number it is sending and receiving TCP segment is 80. ✓
3. 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? (Note: this is the "raw" sequence number carried in the TCP segment itself; it is NOT the packet # in the "No." column in the Wireshark window. Remember there is no such thing as a "packet number" in TCP or UDP; as you know, there are sequence numbers in TCP and that's what we're after here. Also note that this is not the relative sequence number with respect to the starting sequence number of this TCP session.). What is it in this TCP segment that

identifies the segment as a SYN segment? Will the TCP receiver in this session be able to use Selective Acknowledgments (allowing TCP to function a bit more like a “selective repeat” receiver, see section 3.4.5 in the text)?



Frame 81: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eno2, id 0

Ethernet II, Src: 10:7c:61:3e:d4:2c (10:7c:61:3e:d4:2c), Dst: 1a:c2:41:30:b5:99 (1a:c2:41:30:b5:99)

Internet Protocol Version 4, Src: 192.168.12.160, Dst: 128.119.245.12

Transmission Control Protocol, Src Port: 36306, Dst Port: 80, Seq: 0, Len: 0

Source Port: 36306

Destination Port: 80

[Stream Index: 0]

[Conversation completeness: Incomplete, DATA (15)]

[TCP Segment Len: 0]

Sequence Number: 0 (relative sequence number)

Sequence Number (raw): 918901428

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment Number: 0

Acknowledgment number (raw): 0

1010 ... = Header Length: 40 bytes (10)

Flags: 0x002 (SYN)

Window: 64240

[Calculated window size: 64240]

Checksum: 0x42fb [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

[Timestamps]

Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

- TCP Option - Maximum segment size: 1460 bytes
- TCP Option - SACK permitted
- TCP Option - Timestamps: TSval 4241146337, TSecr 0
- TCP Option - No-Operation (NOP)
- TCP Option - Window scale: 7 (multiply by 128)

[Timestamps]

- The raw sequence number of the TCP SYN segment is **918901428** and the relative sequence number of the TCP SYN segment is **0**. The TCP segment identifies the segment as a SYN segment because it has the SYN flag at the header and it is set to 1. The TCP receiver in this session will be able to use Selective Acknowledgments because it says SACK is permitted in the TCP Option.



No.	Time	Source	Destination	Protocol	Length	Info
81	3.118130801	192.168.12.160	128.119.245.12	TCP	74	36306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4241146337 TSecr=0 WS=128
82	3.118161548	192.168.12.160	128.119.245.12	TCP	74	36318 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4241146337 TSecr=0 WS=128
83	3.122070509	192.168.12.160	128.119.245.12	TCP	54	36306 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
84	3.226705291	192.168.12.160	128.119.245.12	TCP	54	36306 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
85	3.226891772	128.119.245.12	192.168.12.160	TCP	66	80 → 36318 [SYN, ACK] Seq=0 Ack=1 Win=29208 Len=0 MSS=1460 SACK_PERM=1 WS=128
86	3.226919310	192.168.12.160	128.119.245.12	TCP	54	36318 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
166	4.715521889	192.168.12.160	142.251.177.188	TCP	66	52578 → 5228 [ACK] Seq=1 Ack=1 Win=441 Len=0 TSval=2615473443 TSecr=1159555532
167	4.72402009	192.168.12.160	128.119.245.12	TCP	66	52578 → 5228 [ACK] Seq=1 Ack=1 Win=441 Len=0 TSval=2615473443 TSecr=1159555532
121	6.366181465	192.168.12.160	128.119.245.12	TCP	656	36306 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=682 [TCP segment of a reassembled PDU]
122	6.366252389	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=683 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
123	6.366278193	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=3523 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
125	6.366299299	192.168.12.160	128.119.245.12	TCP	78	55662 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=26149106476 TSecr=0 WS=128 TFO-R
126	6.366390313	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=6443 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
127	6.366396819	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=9363 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
128	6.366411967	192.168.12.160	128.119.245.12	TCP	1514	36306 → 80 [ACK] Seq=12283 Ack=1 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
129	6.366435983	192.168.12.15	192.168.12.160	TCP	66	53 → 55662 [SYN, ACK] Seq=0 Ack=1 Win=64256 Len=0 MSS=1460 WS=256 SACK_PERM=1
130	6.366558156	192.168.12.160	192.168.12.15	TCP	54	55662 → 53 [ACK] Seq=1 Ack=1 Win=64256 Len=0
131	6.366566379	192.168.12.160	192.168.12.15	DNS	121	Standard query 0x93ee A google-ohhttp-relay-safebrowsing.fastly-edge.com
132	6.366638271	192.168.12.160	192.168.12.15	DNS	121	Standard query response 0x1a39 HTTPS google-ohhttp-relay-safebrowsing.fastly-edge.com
133	6.366894384	192.168.12.15	192.168.12.160	TCP	60	53 → 55662 [ACK] Seq=1 Ack=135 Win=2192272 Len=0
139	6.368929808	192.168.12.15	192.168.12.160	TCP	137	Standard query response 0x93ee A google-ohhttp-relay-safebrowsing.fastly-edge.com A 146.75.9.91
140	6.368987852	192.168.12.160	192.168.12.15	TCP	54	55662 → 53 [ACK] Seq=135 Ack=84 Win=64256 Len=0
141	6.369294476	192.168.12.15	192.168.12.160	DNS	179	Standard query response 0x1a39 HTTPS google-ohhttp-relay-safebrowsing.fastly-edge.com SOA ns1.fastly-edge.com
142	6.369295952	192.168.12.160	192.168.12.15	TCP	54	55662 → 53 [ACK] Seq=135 Ack=289 Win=64256 Len=0
153	6.472414470	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=603 Win=38464 Len=0
154	6.472443224	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=13743 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
155	6.473189764	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=2963 Win=33408 Len=0
156	6.473188860	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=3523 Win=36352 Len=0
157	6.473188069	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=16663 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
158	6.473194195	192.168.12.160	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=19583 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
159	6.473248072	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=4983 Win=38168 Len=0
160	6.473249098	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=6443 Win=42112 Len=0
161	6.473240129	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=7903 Win=45896 Len=0
162	6.473240148	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=9363 Win=48800 Len=0
163	6.473240167	128.119.245.12	192.168.12.160	TCP	60	80 → 36306 [ACK] Seq=1 Ack=10873 Win=50844 Len=0

Transmission Control Protocol, Src Port: 80, Dst Port: 36306, Seq: 0, Ack: 1, Len: 0	
Source Port:	80
Destination Port:	36306
[Stream index: 0]	
[Conversation completeness: Incomplete, DATA (15)]	
[TCP Segment Len: 0]	
Sequence Number: 0 (relative sequence number)	
Sequence Number (raw): 3423406164	
[Next Sequence Number: 1 (relative sequence number)]	
Acknowledgment Number: 1 (relative ack number)	
Acknowledgment number (raw): 918901429	
1000 .... = Header Length: 32 bytes (0)	
Flags: 0x012 (SYN, ACK)	
000. .... = Reserved: Not set	
...0. .... = Nonce: Not set	
...0... .... = Congestion Window Reduced (CWR): Not set	
....0... .... = ECH-Echo: Not set	
....0... .... = Urgent: Not set	
....1... .... = Acknowledgment: Set	
....0... .... = Push: Not set	
....0... .... = Reset: Not set	
...xxxxxx. .... = SYN: Set	
....0... .... = FIN: Not set	
[TCP Flags: .....A..S.]	
Window: 29200	
[Calculated window size: 29200]	
Checksum: 0xc2e5 [unverified]	
[Checksum Status: Unverified]	
Urgent Pointer: 0	
Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale	
> TCP Option - Maximum segment size: 1460 bytes	
> TCP Option - No-Operation (NOP)	
> TCP Option - No-Operation (NOP)	
> TCP Option - SACK permitted	
> TCP Option - No-Operation (NOP)	
> TCP Option - Window scale: 7 (multiply by 128)	

- What is the sequence number of the SYN ACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is it in the segment that identifies the segment as a SYNACK segment? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value?
  - The raw sequence number of the SYN ACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is **3423406164** and the relative sequence number is **0**. The TCP segment identifies the segment as a SYN ACK segment because it has the SYN and ACK flag at the header and both are set to 1. The value of the Acknowledgement field is 1. gaia.cs.umass.edu determined that value by adding 1 to the initial sequence value.
- What is the sequence number of the TCP segment containing the header of the HTTP POST command? Note that in order to find the POST message header, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with the ASCII text "POST" within its DATA field4,5. How many bytes of data are contained in the payload (data) field of this TCP segment? Did all of the data in the transferred file alice.txt fit into this single segment?

eno2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
81	3.118130801	192.168.12.100	128.119.245.12	TCP	74	36306 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4241146337 TSecr=0 WS=128
82	3.118161548	192.168.12.100	128.119.245.12	TCP	74	36318 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4241146337 TSecr=0 WS=128
83	3.226726585	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [SYN, ACK] Seq=0 Ack=1 Win=29208 Len=0 MSS=1460 SACK_PERM=1 WS=128
84	3.226765261	192.168.12.100	128.119.245.12	TCP	54	36306 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
85	3.226891772	128.119.245.12	192.168.12.100	TCP	66	80 → 36318 [SYN, ACK] Seq=0 Ack=1 Win=29208 Len=0 MSS=1460 SACK_PERM=1 WS=128
86	3.226910310	192.168.12.100	128.119.245.12	TCP	54	36318 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
106	4.715521889	192.168.12.100	142.251.177.188	TCP	66	52578 → 5228 [ACK] Seq=1 Ack=1 Win=441 Len=0 TSval=2615473443 TSecr=1159555532
108	4.722074097	192.168.12.100	192.168.12.100	TCP	66	52578 → 5228 [ACK] Seq=1 Ack=1 Win=441 Len=0 TSval=2615473443 TSecr=1159555532
121	6.366181465	192.168.12.100	128.119.245.12	TCP	656	36306 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=602 [TCP segment of a reassembled PDU]
122	6.366225389	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=083 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
123	6.366228193	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=3523 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
125	6.366299929	192.168.12.100	192.168.12.15	TCP	78	55662 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4241146337 TSecr=0 WS=128 TFO-R
126	6.366306313	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=6443 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
127	6.366366819	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=9363 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
128	6.366411967	192.168.12.100	128.119.245.12	TCP	1514	36306 → 80 [ACK] Seq=12283 Ack=1 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
129	6.366453983	192.168.12.15	192.168.12.100	TCP	66	53 → 55662 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
130	6.366558156	192.168.12.100	192.168.12.15	TCP	54	55662 → 53 [ACK] Seq=1 Ack=1 Win=64256 Len=0
131	6.366566370	192.168.12.100	192.168.12.15	DNS	121	Standard query 0x93ee A google-http-relay-safebrowsing.fastly-edge.com
132	6.366582721	192.168.12.100	192.168.12.15	DNS	121	Standard query 0x1a39 HTTPS google-http-relay-safebrowsing.fastly-edge.com
133	6.366604384	192.168.12.15	192.168.12.100	TCP	60	53 → 55662 [ACK] Seq=1 Ack=135 Win=2182272 Len=0
139	6.366982988	192.168.12.15	192.168.12.100	DNS	137	Standard query response 0x93ee A google-http-relay-safebrowsing.fastly-edge.com A 146.75.9.91
140	6.366987852	192.168.12.100	192.168.12.15	DNS	54	55662 → 53 [ACK] Seq=135 Ack=84 Win=64256 Len=0
141	6.369294476	192.168.12.15	192.168.12.100	TCP	179	Standard query response 0x1a39 HTTPS google-http-relay-safebrowsing.fastly-edge.com SOA ns1.fastly-edge.com
142	6.369296962	192.168.12.100	192.168.12.15	TCP	54	55662 → 53 [ACK] Seq=135 Ack=289 Win=64256 Len=0
153	6.472414470	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=603 Win=38464 Len=0
154	6.472432224	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=13743 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
155	6.473180784	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=2063 Win=33408 Len=0
156	6.473180808	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=3523 Win=36352 Len=0
157	6.473188869	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=16663 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
158	6.473194185	192.168.12.100	128.119.245.12	TCP	2974	36306 → 80 [PSH, ACK] Seq=19583 Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
159	6.473248972	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=4983 Win=39168 Len=0
160	6.473248998	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=6443 Win=42112 Len=0
161	6.473240129	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=7983 Win=45056 Len=0
162	6.473240148	128.119.245.12	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=9363 Win=48896 Len=0
163	6.473248167	192.168.12.100	192.168.12.100	TCP	66	80 → 36306 [ACK] Seq=1 Ack=16873 Win=58848 Len=0

Frame 121: 656 bytes on wire (5248 bits), 656 bytes captured (5248 bits) on interface eno2, id 0

Ethernet II, Src: 18:7c:61:3e:d4:2c (18:7c:61:3e:d4:2c), Dst: 1a:c2:41:30:b5:90 (1a:c2:41:30:b5:90)

Internet Protocol Version 4, Src: 192.168.12.100, Dst: 128.119.245.12

Transmission Control Protocol, Src Port: 36306, Dst Port: 80, Seq: 1, Ack: 1, Len: 602

Source Port: 36306

Destination Port: 80

[Stream index: 0]

[Conversation completeness: Incomplete, DATA (15)]

[TCP Segment Len: 602]

Sequence Number: 1 (relative sequence number)

Sequence Number (raw): 918981425

[Next Sequence Number: 603 (relative sequence number)]

Acknowledgment Number: 1 (relative ack number)

Acknowledgment number (raw): 3423486165

RDP ... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window: 502

[Calculated window size: 64256]

[Window size scaling factor: 128]

Checksum: 0x4541 [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

[Timestamps]

[Time since first frame in this TCP stream: 3.248059664 seconds]

[Time since previous frame in this TCP stream: 3.139416204 seconds]

[SEQ/ACK analysis]

TCP payload (602 bytes)

[Reassembled PDU in frame: 270]

TCP segment data (602 bytes)

0030 01 f6 45 41 00 00 00 4f 53 54 20 2f 77 09 72 05 --EAs0 St /wire

0040 73 68 63 72 6b 2b 6c 61 62 73 2f 6c 61 62 33 2d shark-la bs/lab3

0050 01 2d 72 05 70 6c 79 2e 68 74 6d 20 48 54 54 56 i-reply.htm HTTP

0060 2f 31 2e 31 6d 0a 48 6f 73 7a 3a 2b 6f 61 69 63 i:1: Ho st: gals

0070 2e 63 73 2e 75 6d 61 73 73 2e 65 44 75 8d 0a 43 .cs.umass.edu: C

0080 6f 6e 6e 63 74 69 6f 6e 3a 29 6b 65 65 70 2d onnection: keep

0090 61 6c 69 76 65 6d 0a 43 6f 6e 74 65 6e 74 2d 4c alive-Content-L

00a0 65 6f 74 6b 3a 2b 31 34 30 37 32 33 8d 0a 43 length: 149225-C

00b0 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 2b 6d 61 ache-Contr: ma

00c0 78 2d 61 67 65 3d 30 6d 0a 55 70 67 72 61 64 65 x-age=0-Upgrade

00d0 2d 49 6e 72 6b 63 75 72 65 2d 52 65 71 75 65 73 -Insecur e-Request

00e0 74 73 3a 2b 31 6d 0a 55 73 65 72 72 2d 41 67 65 6a ts: 1-U ser-Agen

00f0 74 3a 2b 4d 6f 7a 69 6c 6e 01 2f 35 2e 30 20 28 ti: Mozilla/5.0 (

0100 50 31 31 3b 20 4c 69 6e 75 70 7b 3b 30 5f 38 615: Lin ux x86\_6

0110 34 2b 2b 41 70 7b 6c 65 57 69 62 4b 49 74 2f 35 a) AppleWebKit/5

0120 33 37 2e 33 36 2b 28 4b 48 54 4d 4c 2c 20 6c 69 37.36 (KHTML, li

0130 6b 65 2b 47 65 63 6b 6f 29 2b 43 68 72 6f 6d 65 ke Gecko ) Chrome

0140 2f 31 32 3b 2b 2e 30 2e 30 2b 20 53 61 66 61 72 /229.0.0.0 Safar

0150 69 2f 35 33 37 2e 33 36 0a 4f 72 69 67 69 6a i/537.36 .Origin

0160 3a 2b 6e 75 6c 6c 0d 0a 43 6f 6e 74 65 6e 74 2d : null- Content-

0170 64 79 70 65 3a 2b 6d 75 6e 74 6b 70 61 72 74 2f Type: multipart/

0180 60 6f 72 6d 2d 64 61 74 61 3b 2b 62 6f 75 6e 64 form-dat a; bound

0190 61 72 79 3d 2d 2d 2d 2d 57 65 62 4b 69 74 46 6f ary=--- WebKitFo

01a0 72 6d 42 6f 75 6e 64 61 72 79 3b 65 45 49 36 4a rmbounds ry6E163

01b0 53 5a 77 6f 4f 4b 67 74 62 4c 0d 0a 41 63 63 65 32bQmpz bl-Acce

01c0 70 74 3a 2b 74 65 78 74 2f 68 74 6d 6c 2c 61 70 pt: text /html,ap

01d0 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 61ication/xhtml

01e0 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f xal:appl ication/

01f0 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 67 65 2f xal;q=0.9,image/

0200 61 76 69 6e 2c 69 6d 61 67 65 2f 77 65 62 70 2c avif,image/webp,

0210 6d 61 67 65 2f 61 70 6e 6f 2c 2a 2f 2a 3b 71 image/ap ng,\*/?;q

0220 39 2a 3b 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 9.5,app lication

0230 2f 73 69 6f 6e 65 64 2d 65 78 63 68 61 6e 67 65 /signed- exchange

0240 3b 70 3d 62 33 3b 71 3d 3b 2f 6d 0a 41 63 63 65vB3;q=0.7-Acc

0250 65 70 74 2d 65 6a 63 6f 64 69 6e 67 3a 2b 67 74 ept-Enco ding: g

0260 69 70 2c 2b 64 65 60 6c 61 74 65 6d 0a 41 63 63 ip, defl ate-Acc

0270 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 2b 65 66 ept-Lang uage: en

0280 2d 55 53 2c 65 6e 3b 71 3d 3b 2e 39 6d 0a 6d 6a -US,en;q=0.9---

- The sequence number of the TCP segment containing the header of the HTTP POST command is **1**. The amount of bytes of data contained in the payload field of this TCP segment are **602** bytes. No, the data in the transferred file did not fit into this single segment.

6. Consider the TCP segment containing the HTTP “POST” as the first segment in the data transfer part of the TCP connection.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcpLen=>

No.	Time	Source	Destination	Protocol	Length	Info
121	0.360181465	192.168.12.160	128.119.245.12	TCP	656	36396 -> 80 [PSH, ACK] Seq=1-Ack=1 Win=64256 Len=602 [TCP segment of a reassembled PDU]
122	0.360233919	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=603-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
123	0.360280313	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=3523-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
126	0.360363019	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=6443-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
127	0.360366819	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=9363-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
128	0.360411907	192.168.12.160	128.119.245.12	TCP	1514	36396 -> 80 [ACK] Seq=12283-Ack=1 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
131	0.360565603	192.168.12.160	192.168.12.15	DNS	121	Standard query 0x393e A google-http-relay-safeforwising.fastly-edge.com
132	0.360663871	192.168.12.160	192.168.12.15	DNS	121	Standard query 0x1a39 HTTPS google-http-relay-safeforwising.fastly-edge.com
139	0.360902688	192.168.12.15	192.168.12.160	DNS	137	Standard query response 0x1a39 HTTPS google-http-relay-safeforwising.fastly-edge.com SOA ns1.fastly-edge.com
141	0.360924476	192.168.12.15	192.168.12.160	DNS	179	Standard query response 0x1a39 HTTPS google-http-relay-safeforwising.fastly-edge.com SOA ns1.fastly-edge.com
154	0.472443224	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=13743-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
156	0.473380809	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=16663-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
158	0.473191485	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=19583-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
160	0.473245340	192.168.12.160	128.119.245.12	TCP	1514	36396 -> 80 [ACK] Seq=22563-Ack=1 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
161	0.473290754	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=23963-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
166	0.473423609	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=26883-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
168	0.473479279	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=29803-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
169	0.473660121	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=32723-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
171	0.473853948	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=35643-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
172	0.474840844	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=38563-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
173	0.474859022	192.168.12.160	128.119.245.12	TCP	1514	36396 -> 80 [ACK] Seq=41483-Ack=1 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
176	0.574223958	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=42943-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
177	0.574241235	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=45863-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
179	0.574856338	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=48783-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
188	0.579483812	192.168.12.160	128.119.245.12	TCP	1514	36396 -> 80 [ACK] Seq=51703-Ack=1 Win=64256 Len=1460 [TCP segment of a reassembled PDU]
189	0.579496784	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=55163-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
190	0.579565175	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=58083-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
200	0.579614131	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=61003-Ack=1 Win=64256 Len=2920 [TCP segment of a reassembled PDU]
201	0.579619355	192.168.12.160	128.119.245.12	TCP	2974	36396 -> 80 [PSH, ACK] Seq=64323-Ack=1 Win=64256 Len=2920 [TCP segment

- At what time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent?
  - The time for the first segment is **6.366181465**.
- At what time was the ACK for this first data-containing segment received?
  - The ACK for the first data-containing segment was received at **6.366225309**.
- What is the RTT for this first data-containing segment?
  - The round trip time (RTT) for the first data-containing segment is **0.108634460** seconds.



- What is the RTT value the second data-carrying TCP segment and its ACK?
    - The round trip time (RTT) for the second data-containing segment is **0.108634460 seconds** and the ACK value is **1**. ✓
  - What is the EstimatedRTT value (see Section 3.5.3, in the text) after the ACK for the second data-carrying segment is received? Assume that in making this calculation after the received of the ACK for the second segment, that the initial value of EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242, and a value of  $\alpha = 0.125$ .
    - $\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}$
    - $\alpha = 0.125$
    - The initial EstimatedRTT is equal to the measured RTT for the first segment, which is 0.108634460 seconds.
    - The SampleRTT for the second segment is also 0.108634460 seconds.
    - $\text{New estimatedRTT} = (.875) * .108634460 + .125 * .108634460$  ✓
    - **New estimatedRTT = .10863446**
7. What is the length (header plus payload) of each of the first four data-carrying TCP segments?
- The length of each of the first four data-carrying TCP segments are **2920**. ✓
8. What is the minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among these first four data-carrying TCP segments? Does the lack of receiver buffer space ever throttle the sender for these first four data carrying segments?
- The minim amount of available buffer space advertised to the client by the gaia.cs.umass.edu is **Window:30464**. No, it did not because the the window size increased over time.  
 where did you find that? the screen capture doesn't show it.
9. 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 segment in the trace file. The fact that an old Acknowledgement number was never resent in order to re-request previous packets demonstrates this. ✓  
 you should also check duplicate sequence numbers from the sender in case of duplicate packets. Not only based on duplicate acks.
10. How much data does the receiver typically acknowledge in an ACK among the first ten data-carrying segments sent from the client to gaia.cs.umass.edu? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 in the text) among these first ten data-carrying segments?
- The number of data the receiver acknowledges is 2920 bytes of payload data for each acknowledgement. No, each of the first 10 ACKs acknowledgements exactly 2920 bytes of payload data.

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

- The start time is 6.366411967
- The finished time is 6.789928743
- Difference:  $6.789928743 - 6.366411967 = 0.423516776$
- Calculated window size: 256256
- Throughput:  $256256 \div 0.423516776 = 604994.110033304$  bits/sec

12. 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. Consider the “fleets” of packets sent around  $t = 0.025$ ,  $t = 0.053$ ,  $t = 0.082$  and  $t = 0.1$ . Comment on whether this looks as if TCP is in its slow start phase, congestion avoidance phase or some other phase. Figure 6 shows a slightly different view of this data.

- TCP is in its slow start phase. We start off with 3 packets, then 6, and then even more after that. show couple numbers (doubled numbers) for every RTT

13. These “fleets” of segments appear to have some periodicity. What can you say about the period?

- We can say that the period corresponds to the Round Trip Time between the sender and the receiver.

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



- TCP is in its slow start phase. We start off with 6, then we get 11, and this continues. Just like number 13, the period corresponds to the Round Trip Time between the sender and the receiver.