

TCP

네트워크 프로그래밍

휴먼지능정보공학과
201810776 소재휘

Wireshark_TCP_v6.0

Questions (1/4)

- 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`?
- What is the IP address of `gaia.cs.umass.edu`?
On what port number is it sending and receiving TCP segments for this connection?



#1 TCP

Network programming

- 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?
- 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?

무선 LAN 어댑터 Wi-Fi:

```
연결별 DNS 접미사. . . . . :  
설명. . . . . : Intel(R) Dual Band Wireless-AC 8260  
물리적 주소. . . . . : A0-C5-89-2B-BB-B4  
DHCP 사용. . . . . : 예  
자동 구성 사용. . . . . : 예  
링크-로컬 IPv6 주소. . . . : fe80::29e5:aaad:c498:e37%15(기본 설정)  
IPv4 주소. . . . . : 172.16.25.29(기본 설정)  
서브넷 마스크. . . . . : 255.255.248.0  
임대 시작 날짜. . . . . : 2019년 11월 11일 월요일 오후 4:20:04  
임대 만료 날짜. . . . . : 2019년 11월 11일 월요일 오후 5:05:04  
기본 게이트웨이. . . . . : 172.16.24.1  
DHCP 서버. . . . . : 172.16.10.5  
DHCPv6 IAID. . . . . : 144754057  
DHCPv6 클라이언트 DUID. . . : 00-01-00-01-23-78-8D-A1-98-83-89-30-09-51  
DNS 서버. . . . . : 164.124.101.2  
                              219.250.36.130  
Tcpip를 통한 NetBIOS. . . . : 사용
```

Client의 IP address

```
C:\Users\wasd>nslookup gaia.cs.umass.edu  
서버: ns.lgdacom.net  
Address: 164.124.101.2  
  
권한 없는 응답:  
이름: gaia.cs.umass.edu  
Address: 128.119.245.12
```

Gaia server의 IP address



#1 TCP

Network programming

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?
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?

317	2.054108	172.16.25.29	128.119.245.12	TCP	66	60041 → 80	[SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
324	2.255341	128.119.245.12	172.16.25.29	TCP	66	80 → 60041	[SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1
325	2.255464	172.16.25.29	128.119.245.12	TCP	54	60041 → 80	[ACK] Seq=1 Ack=1 Win=262144 Len=0
326	2.255712	172.16.25.29	128.119.245.12	TCP	558	60041 → 80	[PSH, ACK] Seq=1 Ack=1 Win=262144 Len=504 [TCP segment of a

Transmission Control Protocol, Src Port: 60041, Dst Port: 80, Seq: 1, Ack: 1, Len: 504

Source Port: 60041

Destination Port: 80

Client의 Port Address : 60041

Gaia server의 Port Address : 128.119.245.12

Wireshark_TCP_v6.0

Questions (2/4)

- 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?
- 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?
- 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.

TCP segment data (635 bytes)																	
0030	40	bf	1b	7a	00	00	50	4f	53	54	20	2f	77	69	72	65	@..z..PO ST wire
0040	73	68	61	72	6b	2d	6c	61	62	73	2f	6c	61	62	33	2d	Shark.ta us/ab3-
0050	31	2d	72	65	70	6c	79	2e	68	74	6d	20	48	54	54	50	1-reply. htm HTTP
0060	2f	31	2e	31	0d	0a	48	6f	73	74	3a	20	67	61	69	61	/1.1..Ho st: gaia



#1 TCP

Network programming

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?

```
317 2.054108      172.16.25.29      128.119.245.12    TCP      66 60041 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
```

```
Transmission Control Protocol, Src Port: 60041, Dst Port: 80, Seq: 0, Len: 0
```

```
Source Port: 60041
```

```
Destination Port: 80
```

```
[Stream index: 1]
```

```
[TCP Segment Len: 0]
```

```
Sequence number: 0      (relative sequence number)
```

```
Flags: 0x002 (SYN)
```

```
000. .... = Reserved: Not set
```

```
...0 .... = Nonce: Not set
```

```
.... 0... = Congestion Window Reduced (CWR): Not set
```

```
.... .0.. = ECN-Echo: Not set
```

```
.... ..0. = Urgent: Not set
```

```
.... ...0 = Acknowledgment: Not set
```

```
.... .... 0... = Push: Not set
```

```
.... .... .0.. = Reset: Not set
```

```
> .... .... ..1. = Syn: Set
```

가장 처음의 SYN flag가 포함된 segment를 조사하였다. Sequence number은 0이었고 포함되어 있는 SYN flag로 SYN segment를 식별 가능하다.



#1 TCP

Network programming

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?

```
324 2.255341 128.119.245.12 172.16.25.29 TCP 66 80 → 60041 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 W...
```

```
Transmission Control Protocol, Src Port: 80, Dst Port: 60041, Seq: 0, Ack: 1, Len: 0
```

```
Source Port: 80 Acknowledgment number: 1 (relative ack number)
```

```
Destination Port: 60041
```

```
[Stream index: 1]
```

```
[TCP Segment Len: 0]
```

```
Sequence number: 0 (relative sequence number)
```

```
Flags: 0x012 (SYN, 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
```

```
.... .... 0... = Push: Not set
```

```
.... .... .0.. = Reset: Not set
```

```
> .... .... ..1. = Syn: Set
```

```
.... .... ...0 = Fin: Not set
```

Gaia서버가 client에게 reply해준 SYNACK segment를 조사하였다.

Sequence number은 relativ로 0이었고 포함되어 있는 SYN flag와 ACK flag로 SYNACK segmen를 식별 가능하다.

3-way handshake 과정이므로 클라이언트가 SYN을 전달했으므로 그에 대한 답변인 ACK를 포함한다. 즉 전송한 SYN 번호 + 1을 ACK로 전송한다.

Acknowledgment numbe은 다음 패킷의 sequence numbe로 기대되는 1이다.



#1 TCP

Network programming

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.

▼ Transmission Control Protocol, Src Port: 60041, Dst Port: 80, Seq: 1, Ack: 1, Len: 504

Source Port: 60041

Destination Port: 80

[Stream index: 1]

[TCP Segment Len: 504]

Sequence number: 1 (relative sequence number)

[Next sequence number: 505 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

0101 = Header Length: 20 bytes (5)

▼ Flags: 0x018 (PSH, ACK)

000. = Reserved: Not set

...0 = Nonce: Not set

0000	08 5b 0e eb 64 b2 a0 c5 89 2b bb b4 08 00 45 00	·[·d···+···E·
0010	02 20 92 ac 40 00 80 06 2b 7a ac 10 19 1d 80 77	· ·@···+z····w
0020	f5 0c ea 89 00 50 6f da e3 82 a0 ad ef aa 50 18	····Po· ·····P·
0030	04 00 68 c7 00 00 50 4f 53 54 20 2f 77 69 72 65	··h··PO ST /wire

다음은 HTTP에서 POST명령을 포함하고 있는, 3-way handshake 이후의 첫 data를 포함하고 있는 segment이다. 이 packet의 sequence number은 1이다.

Wireshark_TCP_v6.0

Questions (3/4)

- 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)?
 - 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 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 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.



#1 TCP

Network programming

What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)?

Segment를
전송한 시간

At what time was each segment sent? When was the ACK for each segment received?

326	2.255712	172.16.25.29	128.119.245.12	TCP	558	60041 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=504 [TCP segment of a r...
327	2.256142	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=505 Ack=1 Win=262144 Len=1460 [TCP segment of a rea...
328	2.256143	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=1965 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
329	2.256144	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=3425 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
330	2.256146	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=4885 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
331	2.256147	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=6345 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
343	2.457946	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=505 Win=30336 Len=0
344	2.457946	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=4885 Win=39040 Len=0

Segment를
받은 시간

다음은 HTTP에서 POST명령 이후의 6개의 segment이다. 클라이언트가 서버에 보낸 data의 sequence number은 reply하는 ack 번호와 같은 것을 통해서 보낸 segment에 대한 reply 패킷을 조사할 수 있었고 이를 통해서 Segment를 받은 시간을 알 수 있었다.
각각의 segment의 sequence number은 1,505,1865,3425,4885,6345였다.



#1 TCP

Network programming

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?

326	2.255712	172.16.25.29	128.119.245.12	TCP	558 60041 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=504 [TCP segment of a r...
327	2.256142	172.16.25.29	128.119.245.12	TCP	1514 60041 → 80 [ACK] Seq=505 Ack=1 Win=262144 Len=1460 [TCP segment of a rea...
328	2.256143	172.16.25.29	128.119.245.12	TCP	1514 60041 → 80 [ACK] Seq=1965 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
329	2.256144	172.16.25.29	128.119.245.12	TCP	1514 60041 → 80 [ACK] Seq=3425 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
330	2.256146	172.16.25.29	128.119.245.12	TCP	1514 60041 → 80 [ACK] Seq=4885 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
331	2.256147	172.16.25.29	128.119.245.12	TCP	1514 60041 → 80 [ACK] Seq=6345 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
343	2.457946	128.119.245.12	172.16.25.29	TCP	60 80 → 60041 [ACK] Seq=1 Ack=505 Win=30336 Len=0
344	2.457946	128.119.245.12	172.16.25.29	TCP	60 80 → 60041 [ACK] Seq=1 Ack=4885 Win=39040 Len=0
352	2.458509	128.119.245.12	172.16.25.29	TCP	60 80 → 60041 [ACK] Seq=1 Ack=13645 Win=56576 Len=0

[SEQ/ACK analysis]

[This is an ACK to the segment in frame: 329]

[The RTT to ACK the segment was: 0.201802000 seconds]

[iRTT: 0.201356000 seconds]

[SEQ/ACK analysis]

[This is an ACK to the segment in frame: 326]

[The RTT to ACK the segment was: 0.202234000 seconds]

[iRTT: 0.201356000 seconds]

[This is an ACK to the segment in frame: 335]

[The RTT to ACK the segment was: 0.202360000 seconds]

[iRTT: 0.201356000 seconds]

RTT value는 segment가 전달되는 시간을 의미한다. 서버가 클라이언트에게 ACK를 보내는 패킷을 통해서 이 값을 확인할 수 있다. 6개의 segment들은 2개의 ack를 받을 수 있었는데 각각의 RTT를 segment 안에서 확인 가능했다.

첫번째 ack는 326번 segment에 대한 응답, 두번째 ack는 327,328,329번, 세번째 ack에서 330,331번에 대한 응답을 포함한다.

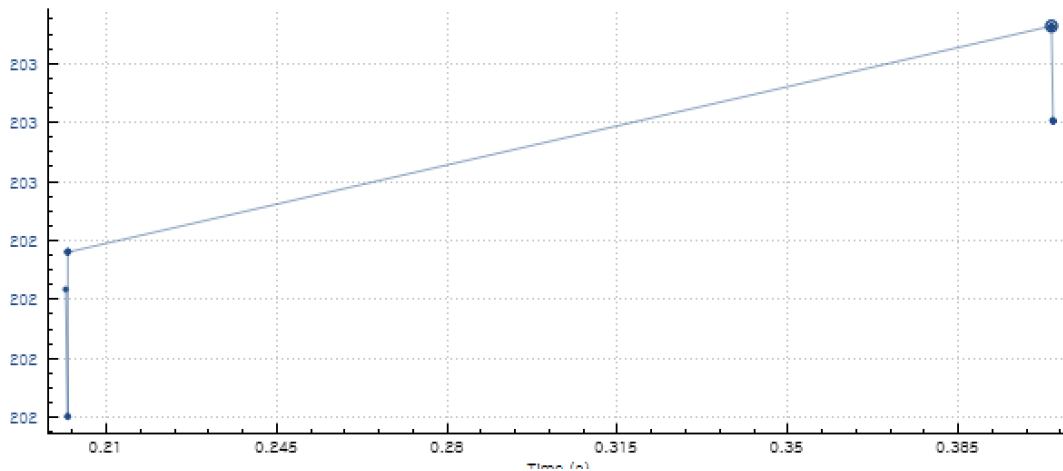
#1 TCP

Network programming



Round Trip Time for 172.16.25.29:60041 → 128.119.245.12:80

Wi-Fi



다음과 같이 Wireshark에서 제공하는 툴을
이용해서도 RRT를 확인할 수 있다.

Wireshark_TCP_v6.0

Questions (4/4)

- What is the length of each of the first six TCP segments?
- 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(차단 혹은 전송 rate 감소) the sender?
- Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
- 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 the table on next page).
- What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



#1 TCP

Network programming

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

326	2.255712	172.16.25.29	128.119.245.12	TCP	558	60041 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=504 [TCP segment of a r...
327	2.256142	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=505 Ack=1 Win=262144 Len=1460 [TCP segment of a rea...
328	2.256143	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=1965 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
329	2.256144	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=3425 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
330	2.256146	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=4885 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
331	2.256147	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=6345 Ack=1 Win=262144 Len=1460 [TCP segment of a re...

각각 6개의 segment의 length는 504,1460,1460,1460,1460,1460이다.

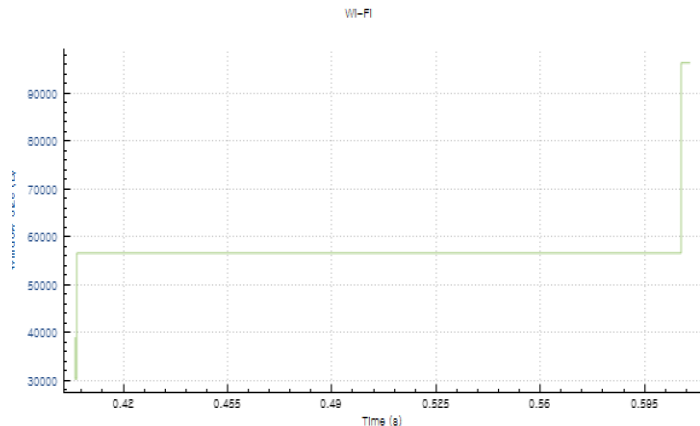


#1 TCP

Network programming

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(차단 혹은 전송 rate 감소) the sender?

317	2.054108	172.16.25.29	128.119.245.12	TCP	66	60041 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
324	2.255341	128.119.245.12	172.16.25.29	TCP	66	80 → 60041 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1
325	2.255464	172.16.25.29	128.119.245.12	TCP	54	60041 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
326	2.255712	172.16.25.29	128.119.245.12	TCP	558	60041 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=504 [TCP segment of a re...
327	2.256142	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=505 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
328	2.256143	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=1965 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
329	2.256144	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=3425 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
330	2.256146	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=4885 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
Window Scaling for 172.16.25.29:60041 → 128.119.245.12:80					1514	60041 → 80 [ACK] Seq=6345 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
					TCP	60 80 → 60041 [ACK] Seq=1 Ack=505 Win=30336 Len=0
					TCP	60 80 → 60041 [ACK] Seq=1 Ack=4885 Win=39040 Len=0
					TCP	60 80 → 60041 [ACK] Seq=1 Ack=13645 Win=56576 Len=0



할당되는 버퍼 사이즈, 즉 Window size는 계속해서 증가하는 추세이다. 즉 Congestion이나 Flow Control을 할 필요가 거의 없다는 것을 의미하고 lost된 패킷도 존재하지 않았다. 최소 Window size 크기는 29200이었다.

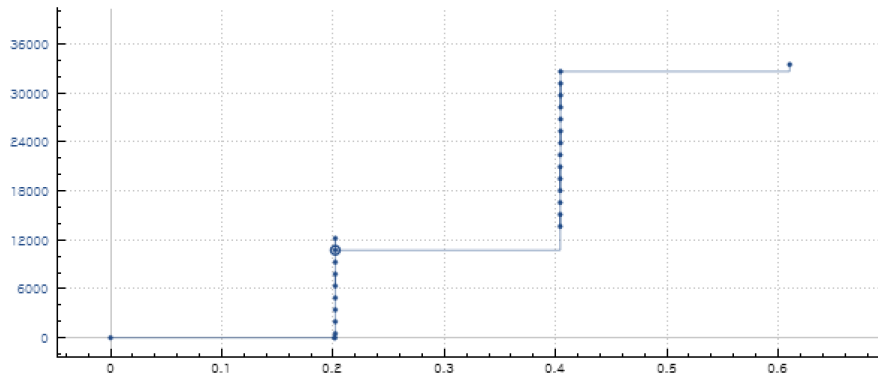


#1 TCP

Network programming

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

349	2.458304	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=19485 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
350	2.458304	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=20945 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
351	2.458305	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=22405 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
352	2.458509	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=13645 Win=56576 Len=0
353	2.458620	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=23865 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
354	2.458620	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=25325 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
355	2.458621	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=26785 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
356	2.458622	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=28245 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
357	2.458622	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=29705 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
358	2.458622	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=31165 Ack=1 Win=262144 Len=1460 [TCP segment of a r...
359	2.458623	172.16.25.29	128.119.245.12	HTTP	913	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (application/haansofthwp)
365	2.661427	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=33484 Win=96256 Len=0
366	2.664452	128.119.245.12	172.16.25.29	HTTP	831	HTTP/1.1 200 OK (text/html)
367	2.664560	172.16.25.29	128.119.245.12	TCP	54	60041 → 80 [ACK] Seq=33484 Ack=778 Win=261120 Len=0



다음을 확인하면 재전송되는 segment가 하나도 없는 것을 확인할 수 있다. 전송률이 계속해서 올라가는 것을 확인할 수 있다.



#1 TCP

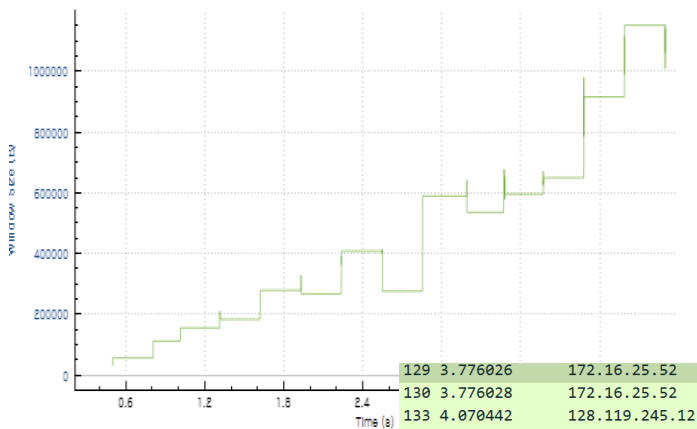
Network programming

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(차단 혹은 전송 rate 감소) the sender?

앞의 결과는 congestion과 flow control을 하는 과정을 관찰이 힘들어 가이아 서버에 큰 파일을 업로드 하였다.

Window Scaling for 172.16.25.52:54174 → 128.119.245.12:80

Wi-Fi



할당되는 버퍼 사이즈, 즉 Window size가 계속해서 조절되고 있다. Congestion이나 flow control을 하기 위해 Window size를 줄였다가 점점 늘이는 과정을 반복하며 버퍼 사이즈를 조절한다.

가장 작은 Window size는 SYN/ACK segment를 보낼 때 29200으로 가장 작았다.

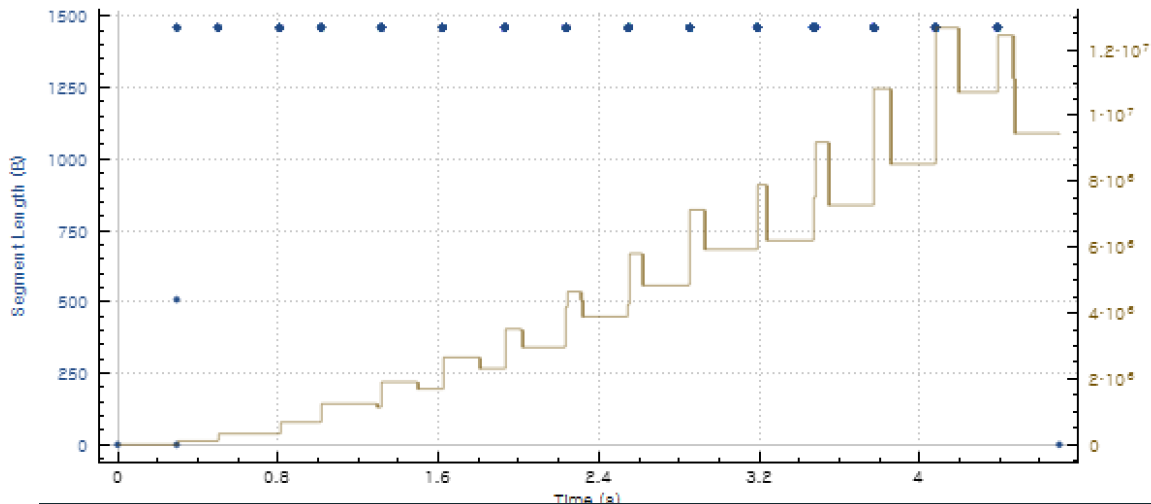
129	3.776026	172.16.25.52	128.119.245.12	TCP	66 54173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
130	3.776028	172.16.25.52	128.119.245.12	TCP	66 54174 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
133	4.070442	128.119.245.12	172.16.25.52	TCP	66 80 → 54173 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 W
134	4.070442	128.119.245.12	172.16.25.52	TCP	66 80 → 54174 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 W
135	4.070562	172.16.25.52	128.119.245.12	TCP	54 54173 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
136	4.070631	172.16.25.52	128.119.245.12	TCP	54 54174 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0



#1 TCP

Network programming

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



Throughput을 확인해 보면 처리량이 줄어들었다가 점점 늘어나는 과정을 반복하는 것을 확인할 수 있었다.

이 과정을 통해 Congestion control과 Flow control을 하고 있음을 확인할 수 있다. 그리고 이 과정에서 다음과 같이 버퍼의 용량이 꽉 차서 Segment가 lost된 것도 확인할 수 있다.

```
2652 8.472023 128.119.245.12 172.16.25.52 TCP 60 [TCP ACKed unseen segment] 80 → 54174 [ACK] Seq=1 Ack=3120529 Win=
2653 8.472023 128.119.245.12 172.16.25.52 TCP 60 [TCP ACKed unseen segment] 80 → 54174 [ACK] Seq=1 Ack=3216889 Win=
2654 8.472023 128.119.245.12 172.16.25.52 TCP 60 [TCP ACKed unseen segment] 80 → 54174 [ACK] Seq=1 Ack=3332182 Win=
2655 8.472024 128.119.245.12 172.16.25.52 HTTP 831 [TCP ACKed unseen segment] HTTP/1.1 200 OK (text/html)
2656 8.472134 172.16.25.52 128.119.245.12 TCP 54 [TCP Previous segment not captured] 54174 → 80 [ACK] Seq=3332182 A
```



#1 TCP

Network programming

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 the table on next page).

326	2.255712	172.16.25.29	128.119.245.12	TCP	558	60041 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=504 [TCP segment of a r...
327	2.256142	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=505 Ack=1 Win=262144 Len=1460 [TCP segment of a rea...
328	2.256143	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=1965 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
329	2.256144	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=3425 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
330	2.256146	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=4885 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
331	2.256147	172.16.25.29	128.119.245.12	TCP	1514	60041 → 80 [ACK] Seq=6345 Ack=1 Win=262144 Len=1460 [TCP segment of a re...
343	2.457946	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=505 Win=30336 Len=0
344	2.457946	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=4885 Win=39040 Len=0
352	2.458509	128.119.245.12	172.16.25.29	TCP	60	80 → 60041 [ACK] Seq=1 Ack=13645 Win=56576 Len=0

첫번째 ACK는 504byte의 data를 포함하고, 두번째 ACK는 4380byte의 data를 포함하고 세번째 ACK는 8760byte의 data를 포함한다. 앞에서 Window size가 점점 증가하는 것을 확인할 수 있었는데 이에 따라 한 번에 보내는 segment의 양이 증가하고, 그에 따른 ACK의 data가 증가하는 것을 확인했다.

다음의 경우에는 수신자가 다음의 ack가 어떤 segment에 대한 ack임을 식별하는 방법은 Sequence number과 Acknowledge number을 비교함으로써 알 수 있다. 다음의 경우는 각각의 Segment에 대한 각각의 ACK를 일단 보류하고 일정 수의 Segment들을 묶어서 Ack를 보내는 것을 알 수 있다.



#1 TCP

Network programming

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

137	4.070750	172.16.25.52	128.119.245.12	TCP	562	54174 → 80 [PSH, ACK] Seq=1 Ack=1 Win=262144 Len=508 [TCP segment
2651	8.472023	128.119.245.12	172.16.25.52	TCP	60	80 → 54174 [ACK] Seq=1 Ack=2938029 Win=1099008 Len=0

다음을 통해서 데이터의 양은 3-way handshake 이후 처음 보내는 sequence number 1의 segment부터 시작한다. 3-way handshake 과정을 포함시키지 않는 이유는 실질적으로 보내는 data가 없기 때문이다. 그리고 connection을 종료하기 전 마지막 data를 보내는 segment에 대한 ACK 번호가 2938029였다. 따라서 2938028byte만큼의 데이터를 전송하였다. 시간은 8.462023-4.070750=4.391273이므로 데이터량/시간을 하여 669060.657이 throughput임을 확인할 수 있다.