



Komunikasi Data

Dosen Pengampu

Adi Hermansyah, S.KOM., M.T.

Oleh

Sultan Zidan (09011382025119)

Kelas SKU B

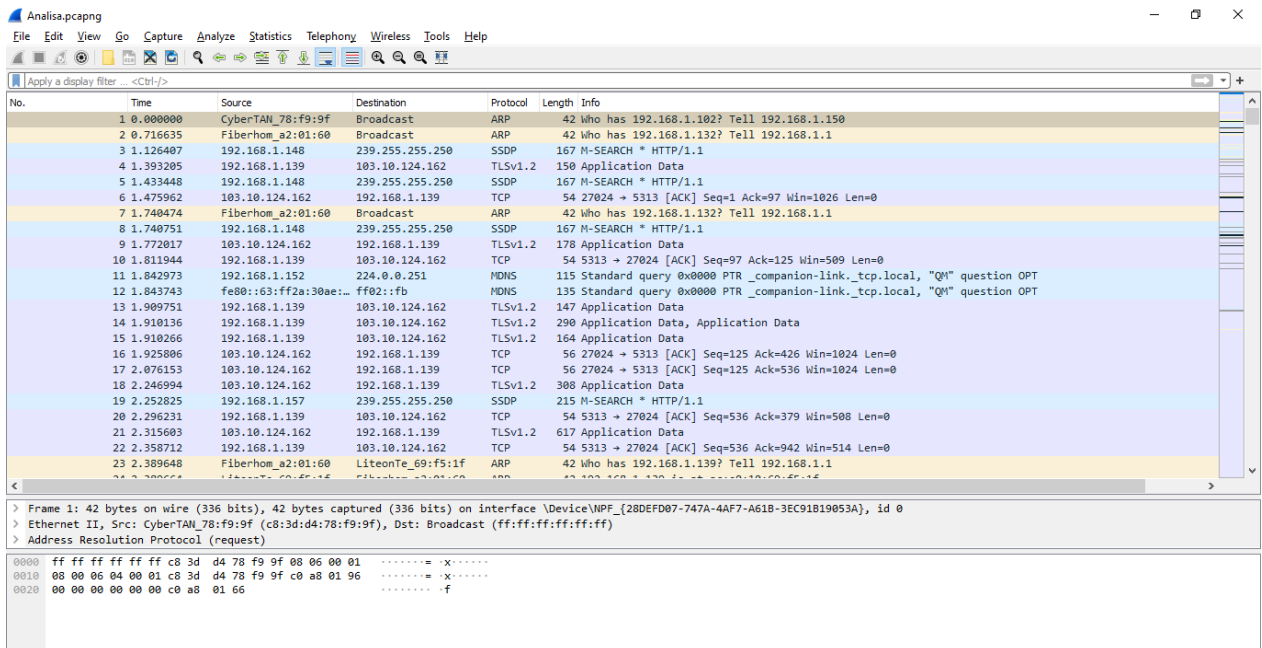
Jurusan Sistem Komputer

Fakultas Ilmu Komputer

Universitas Sriwijaya

2021

1. Menjalankan aplikasi sambal merecord jaringan pada wireshark



Pada proses ini saya menjalankan wireshark selama lebih kurang 10-11menit sambal memainkan aplikasi game online yang tersambung ke server online game itu sendiri. Dapat dilihat pada gambar dibawah ini bahwa wireshark merecord jaringan server yang tersambung pada game yang saya mainkan

1141	39.779637	192.168.1.139	51.89.97.102	UDP	126	53751 → 6021	Len=84
1142	39.810346	192.168.1.139	51.89.97.102	UDP	127	53751 → 6021	Len=85
1143	39.845853	192.168.1.139	51.89.97.102	UDP	127	53751 → 6021	Len=85
1144	39.875179	192.168.1.139	51.89.97.102	UDP	131	53751 → 6021	Len=89
1145	39.928045	192.168.1.139	192.168.1.1	DNS	86	Standard query 0x7c4f A public-ubiservices.ubi.com	
1146	39.954099	192.168.1.139	192.168.1.1	DNS	86	Standard query 0x7c4f A public-ubiservices.ubi.com	
1147	39.973184	192.168.1.1	192.168.1.139	DNS	377	Standard query response 0x7c4f A public-ubiservices.ubi.com CNAME live-public.prod.external.gateway.ubiserv...	
1148	39.974113	192.168.1.139	35.171.148.218	TCP	66	12679 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	
1149	39.974425	192.168.1.1	192.168.1.139	DNS	253	Standard query response 0x7c4f A public-ubiservices.ubi.com CNAME live-public.prod.external.gateway.ubiserv...	
1150	40.023187	192.168.1.139	51.89.97.102	UDP	125	53751 → 6021	Len=83

Saya memainkan game dengan developer Ubisoft sehingga akan terdapat informasi mengenai server Ubisoft pada hasil record wireshark seperti pada gambar diatas, public-ubiservice.ubi.com merupakan service dari server Ubisoft untuk game yang saya mainkan.

Selain itu saya juga melakukan perhitungan throughput, packet loss, jitter, dan delay yang terjadi pada saat proses wireshark tersebut sedang berlangsung. Untuk hasil dan gambar untuk pencarian throughput, packet loss dan jitter dapat dilihat pada halaman selanjutnya.

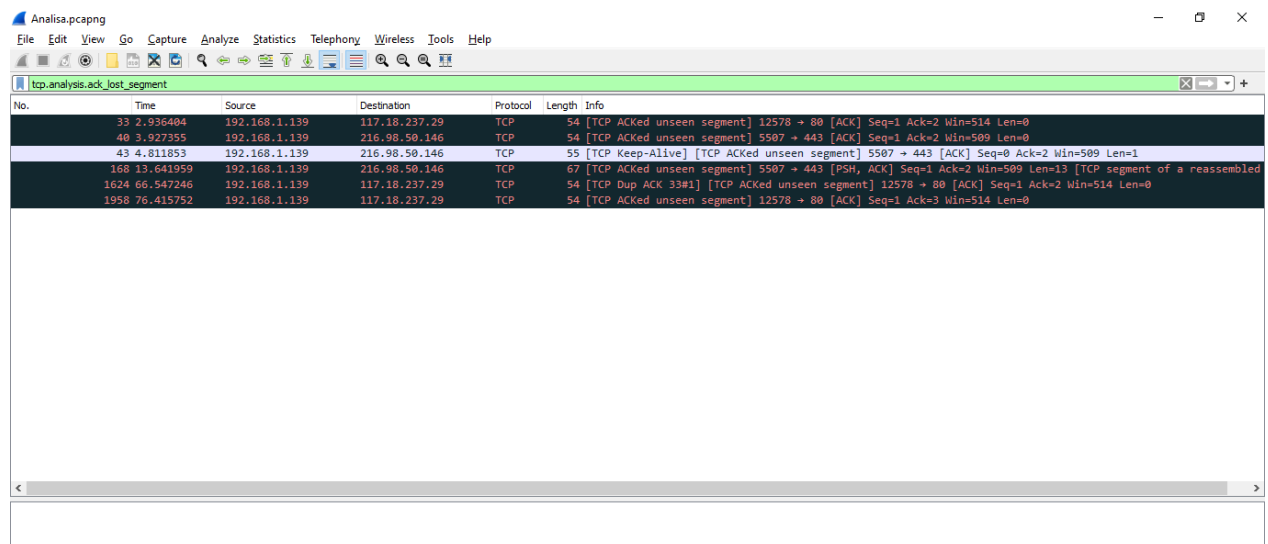
Mencari Throughput, Packet Loss, Jitter dan Delay

Pertama buka hasil statistic dari record pada wireshark tadi seperti pada gambar dibawah ini

Statistics			
Measurement	Captured	Displayed	Marked
Packets	40097	6 (0.0%)	—
Time span, s	665.762	73.479	—
Average pps	60.2	0.1	—
Average packet size, B	452	56	—
Bytes	18111916	338 (0.0%)	0
Average bytes/s	27 k	4	—
Average bits/s	217 k	36	—

Ini adalah hasil statistic dari record wireshark yang saya lakukan, terdapat 40097 paket yang diproses dengan rentang waktu 665,762 second atau bisa dibilang 11 menit.

Untuk mencari throughput bisa dilakukan dengan jumlah bytes data dibagi time span kemudian hasilnya yang bernilai bytes dikalikan dengan 8 bit(1 bytes = 8 bit). Sehingga didapat throughputnya sekitar 217k. Selanjutnya adalah mencari packet loss, untuk packet loss sendiri kita harus mencari tcp.analysis.segment untuk packet loss seperti pada gambar dibawah ini



The image shows a Wireshark packet capture window titled 'Analisa.pcapng'. The 'Statistics' pane is selected, showing a list of packets. The 'tcp.analysis.ack_lost_segment' filter is applied, displaying a list of lost segments. The table below represents the data shown in the packet list pane.

No.	Time	Source	Destination	Protocol	Length	Info
33	2.936404	192.168.1.139	117.18.237.29	TCP	54	[TCP ACKed unseen segment] 12578 → 80 [ACK] Seq=1 Ack=2 Win=514 Len=0
40	3.927355	192.168.1.139	216.98.50.146	TCP	54	[TCP ACKed unseen segment] 5507 → 443 [ACK] Seq=1 Ack=2 Win=509 Len=0
43	4.811853	192.168.1.139	216.98.50.146	TCP	55	[TCP Keep-Alive] [TCP ACKed unseen segment] 5507 → 443 [ACK] Seq=0 Ack=2 Win=509 Len=1
168	13.641959	192.168.1.139	216.98.50.146	TCP	67	[TCP ACKed unseen segment] 5507 → 443 [PSH, ACK] Seq=1 Ack=2 Win=509 Len=13 [TCP segment of a reassembled
1624	66.547246	192.168.1.139	117.18.237.29	TCP	54	[TCP Dup ACK 33#1] [TCP ACKed unseen segment] 12578 → 80 [ACK] Seq=1 Ack=2 Win=514 Len=0
1958	76.415752	192.168.1.139	117.18.237.29	TCP	54	[TCP ACKed unseen segment] 12578 → 80 [ACK] Seq=1 Ack=3 Win=514 Len=0

Pada gambar didapat ada 6 packet loss dari 40097 paket, sehingga untuk mencari persentasenya adalah $(6 * 100\%) / 40097$ sehingga didapat packet lossnya sekitar lebih kurang 0,015%.

Untuk jitter dan delay dapat dicari dengan mengexport hasil record wireshark menjadi CSV atau file dengan format excel seperti pada gambar dibawah

No.	Time	Source	Destination	Time2	Time1	Delay	Delay2	Delay1	Jitter	Total Delay	Rata-Rata Delay	Total Jitter	Rata-Rata Jitter
1	0	CyberTAN_78:f9:5f	Broadcast	0.716635	0	0.716635	0.409772	0.306863	0.102909	665.762056	0.016603787	664.583803	0.016574816
2	0.716635	Fiberhom_a2:01:6c	Broadcast	1.126407	0.716635	0.409772	0.266798	0.142974	0.123824				
3	1.126407	192.168.1.148	239.255.255.250	1.393205	1.126407	0.266798	0.040243	0.226555	-0.186312				
4	1.393205	192.168.1.139	103.10.124.162	1.433448	1.393205	0.040243	0.042514	-0.002271	0.044785				
5	1.433448	192.168.1.148	239.255.255.250	1.475962	1.433448	0.042514	0.264512	-0.221998	0.48651				
6	1.475962	103.10.124.162	192.168.1.139	1.740474	1.475962	0.264512	0.000277	0.264235	-0.263958				
7	1.740474	Fiberhom_a2:01:6c	Broadcast	1.740751	1.740474	0.000277	0.031266	-0.030989	0.062255				
8	1.740751	192.168.1.148	239.255.255.250	1.772017	1.740751	0.031266	0.039927	-0.008661	0.048588				
9	1.772017	103.10.124.162	192.168.1.139	1.811944	1.772017	0.039927	0.031029	0.008898	0.022131				
10	1.811944	192.168.1.139	103.10.124.162	1.842973	1.811944	0.031029	0.00077	0.030259	-0.029489				
11	1.842973	192.168.1.152	224.0.0.251	1.843743	1.842973	0.00077	0.066008	-0.065238	0.131246				
12	1.843743	fe80:63:ff2a:30ae:ff02::fb		1.909751	1.843743	0.066008	0.000385	0.065623	-0.065238				
13	1.909751	192.168.1.139	103.10.124.162	1.910136	1.909751	0.000385	0.00013	0.000255	-0.000125				
14	1.910136	192.168.1.139	103.10.124.162	1.910266	1.910136	0.00013	0.01554	-0.01541	0.03095				
15	1.910266	192.168.1.139	103.10.124.162	1.925806	1.910266	0.01554	0.150347	-0.134807	0.285154				
16	1.925806	103.10.124.162	192.168.1.139	2.076153	1.925806	0.150347	0.170841	-0.020494	0.191335				
17	2.076153	103.10.124.162	192.168.1.139	2.246994	2.076153	0.170841	0.005831	0.16501	-0.159179				
18	2.246994	103.10.124.162	192.168.1.139	2.252825	2.246994	0.005831	0.043406	-0.037575	0.080981				
19	2.252825	192.168.1.157	239.255.255.250	2.296231	2.252825	0.043406	0.019372	0.024034	-0.004662				
20	2.296231	192.168.1.139	103.10.124.162	2.315603	2.296231	0.019372	0.043109	-0.023737	0.066846				
21	2.315603	103.10.124.162	192.168.1.139	2.358712	2.315603	0.043109	0.030936	0.012173	0.018763				
22	2.358712	192.168.1.139	103.10.124.162	2.389648	2.358712	0.030936	1.60E-05	0.03092	-0.030904				

Mencar Delay sendiri adalah dengan pengurangan, yaitu waktu pertama dibagi waktu kedua kemudian ditotalkan, untuk rata ratanya dibagi 40097. Untuk jitter sendiri juga sama namun yang dikurangi adalah delay kedua dikurang delay pertama kemudian ditotalkan, untuk rata rata jitter dibagi dengan 40096 karena pada jitter total data berkurang 1.

Proses komunikasi data yang terlihat adalah dimana setiap jaringan memiliki ip source dan destinationnya. Ip yang sama kebanyakan adalah ip dari komputer atau device kita sendiri seperti pada record wireshark saya ip saya adalah 192.168.1.139, sementara untuk ip yang berbeda beda adalah ip dari server atau web yang saya kunjungi atau terjadi dalam proses penggunaan aplikasi yang menggunakan jaringan didalamnya. Selain itu, kesamaan ip tidak menjamin proses mengirim file yang sama aka nada port yang menjadi penentu arah transport dari data data yang diproses seperti pada gambar dibawah

```
Source Port: 53
Destination Port: 53081
Length: 205
Checksum: 0x0395 [unverified]
[Checksum Status: Unverified]
[Stream index: 54]
> [Timestamps]
UDP payload (197 bytes)
> Domain Name System (response)
```

Ip yang sama tidak selamanya mengirim data yang sama, port menjadi pembeda dalam proses transportasi data.

Link Repository GitHub : <https://github.com/zidansvltan/Network-Traffic-Analysis.git>