Group 9
Analysis of TCP Variants

Tanmay Rajore(2019118)

Tanya Gupta (2019119)

Utkrisht Sikka (2019215)

Taral Jain (2019392)

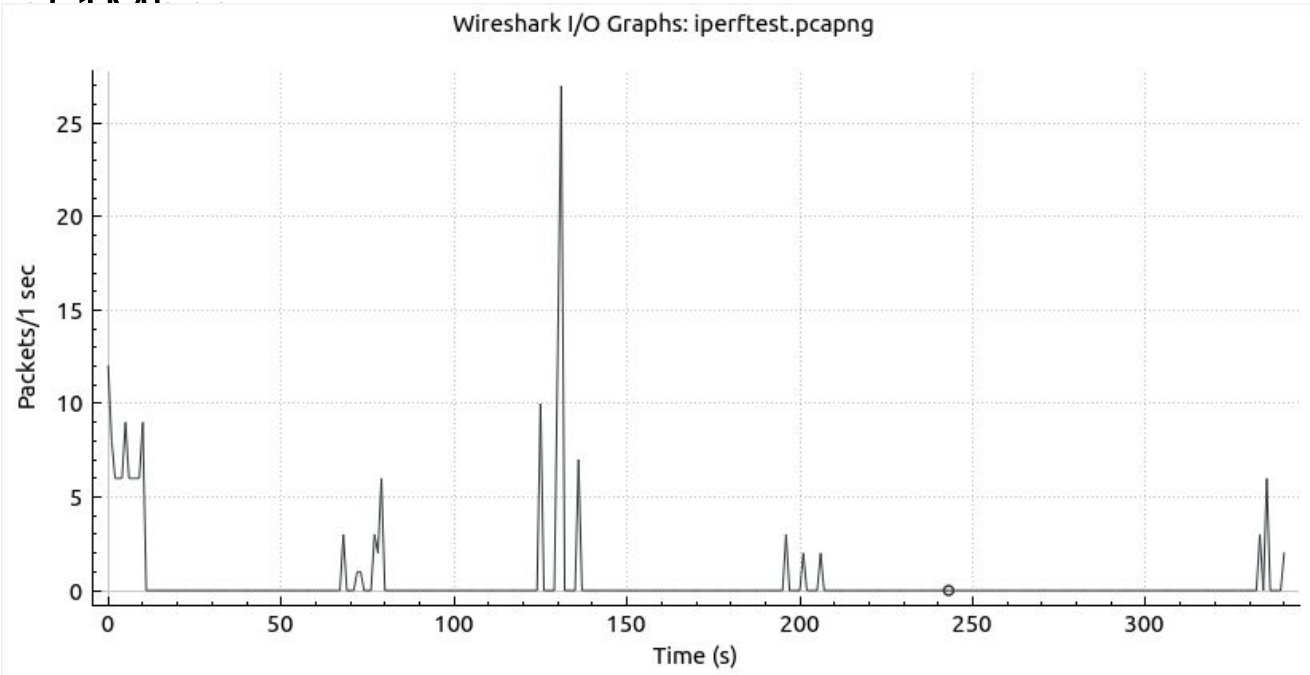
INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

A Brief History of Project

- TCP is Transmission Control Protocol
- UDP is User Datagram Protocol

Iperf

We first generated pcap for a rudimentary server and client on same machine using iperf.Client established TCP connection once to server, over a bandwidth



Real Time TCP Analysis

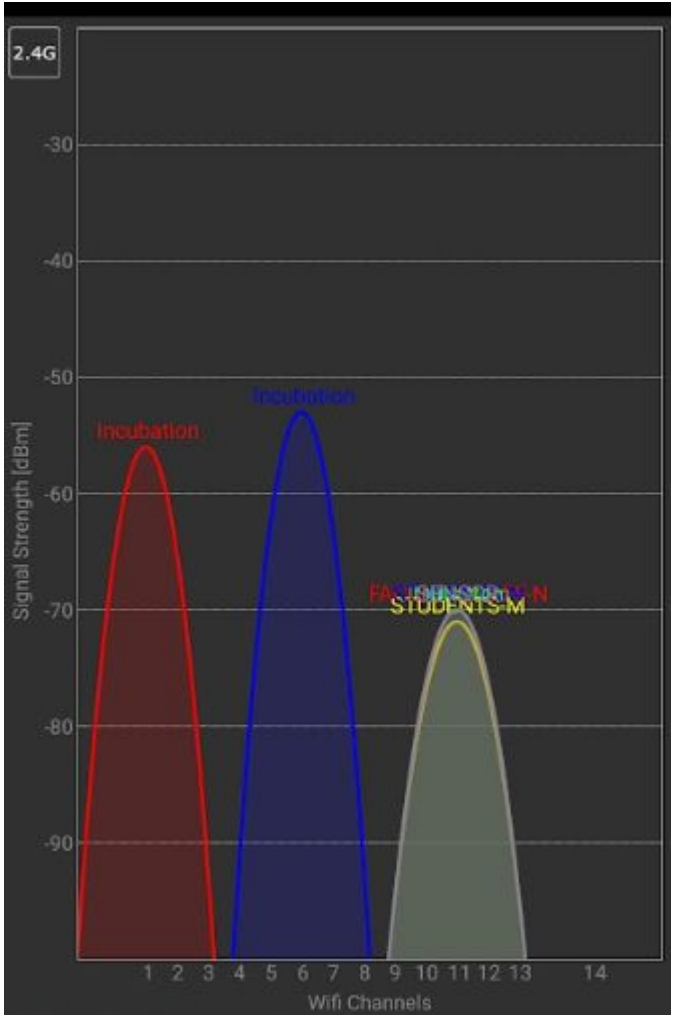
- Application: YouTube video of length 4:30 min
- Played on Mozilla Firefox 75
- Variants used:
  - TCP Veno
  - TCP Cubic
  - TCP Vegas

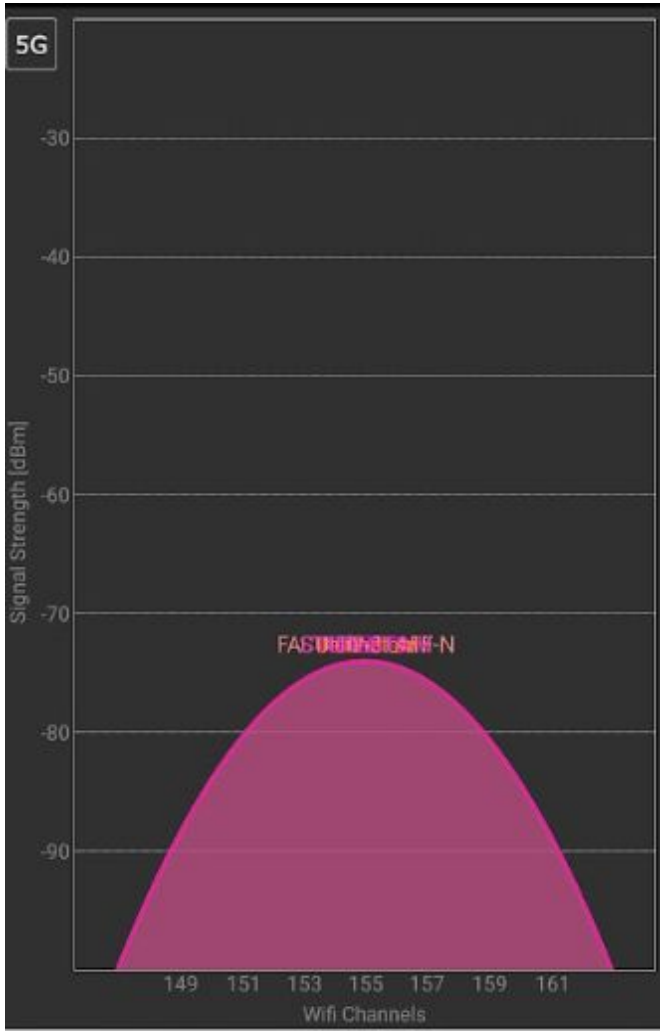
Background

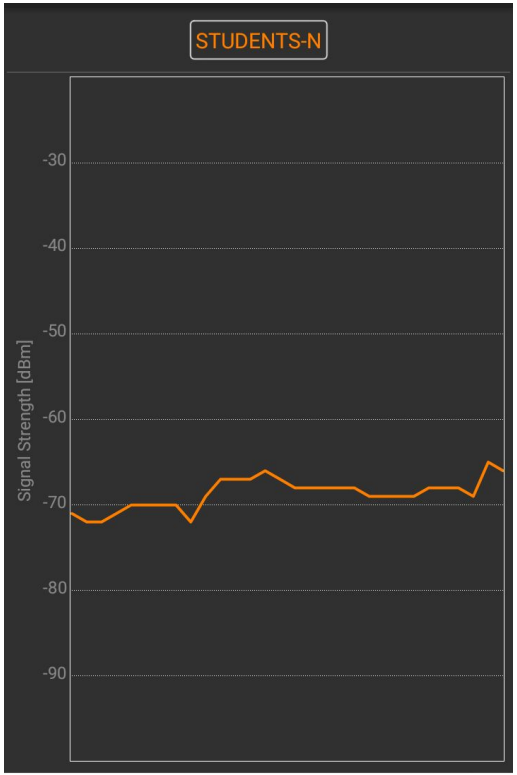
- Youtube predominantly uses QUIC protocol on both Firefox (ver >88) and chrome
- Baseline network performance was measured using veno, vegas and cubic over the QUIC protocol for comparative study of the performance over TCP variants.

Part – I: WiFi

Network Conditions


2.4G
-30
-40
-50
-60
-70
-80
-90
Signal Strength [dBm]
Incubation
Incubation
STUDENTS-M
1 2 3 4 5 6 7 8 9 10 11 12 13 14
Wifi Channels

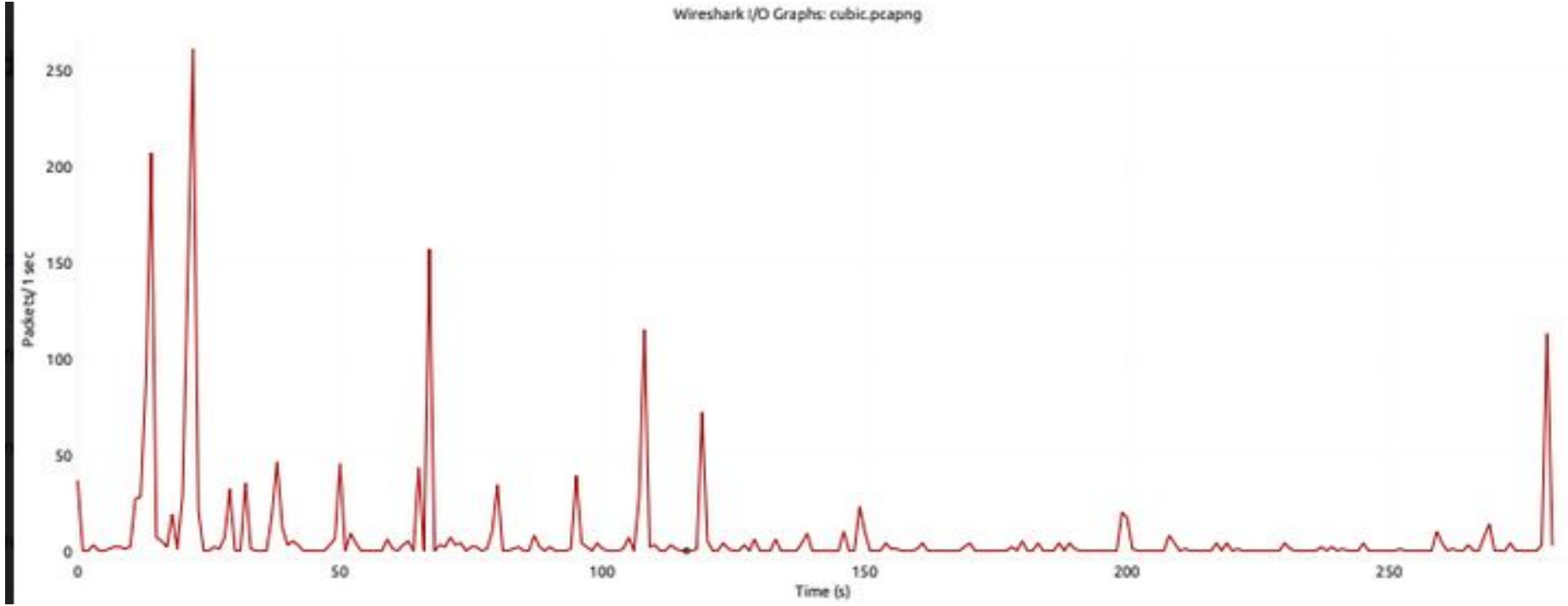

5G
-30
-40
-50
-60
-70
-80
-90
Signal Strength [dBm]
149 151 153 155 157 159 161
Wifi Channels



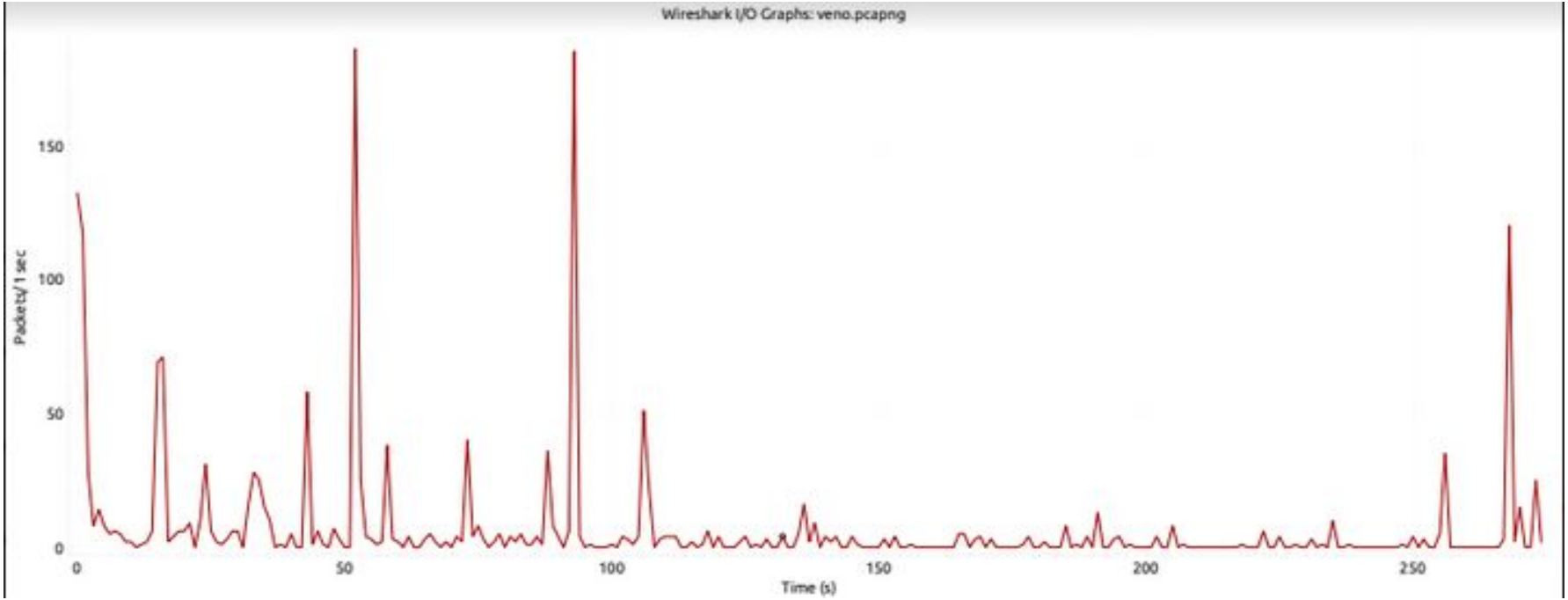
Network signal strength

Network speed:44.03 Mbps

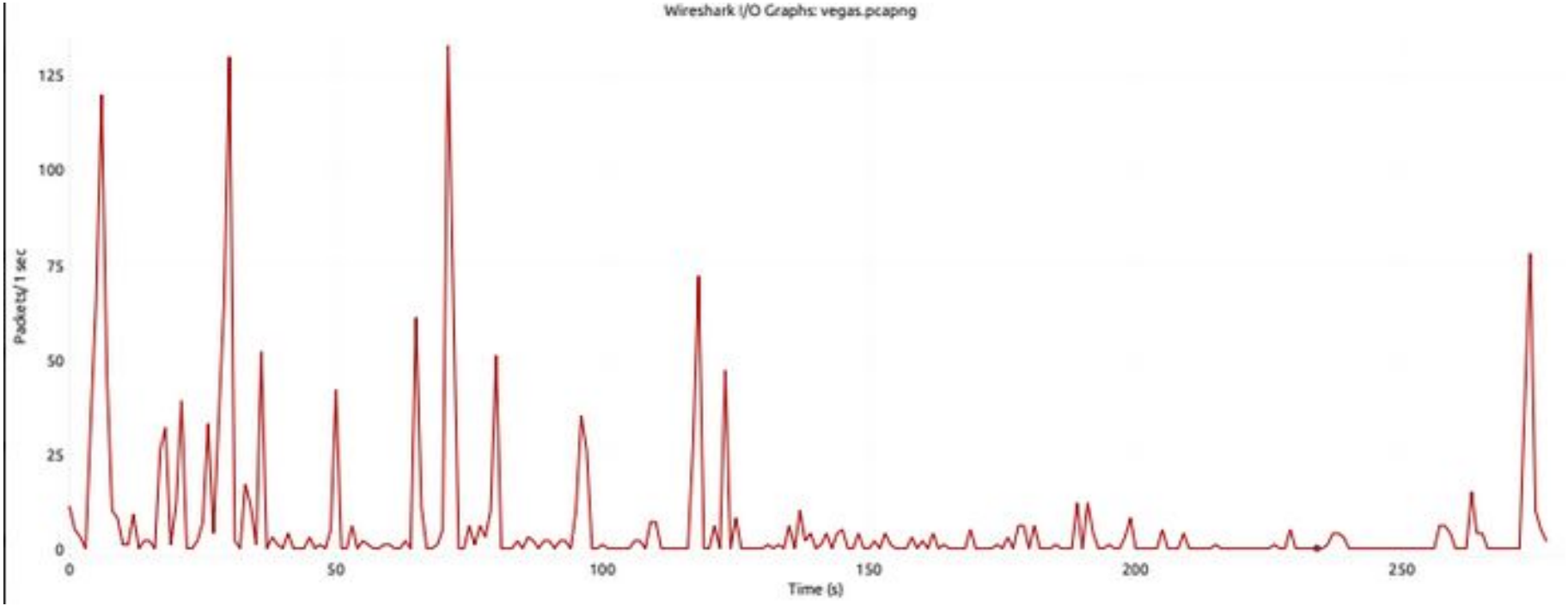
Throughput graphs


Wireshark I/O Graphs: cubic.pcapng
Packets/1 sec
250
200
150
100
50
0
0
50
100
150
200
250
Time (s)

TCP cubic


Wireshark I/O Graphs: veno.pcapng
Packets/1 sec
150
100
50
0
0
50
100
150
200
250
Time (s)

TCP veno


Wireshark I/O Graphs: vegas.pcapng
Packets/1 sec
125
100
75
50
25
0
0
50
100
150
200
250
Time (s)

Vegas

Throughput observations

- Video load time:
  - vegas: 124 sec
  - veno - 109 sec
  - cubic - 122 sec
- We notice that the video loaded the fastest in TCP Veno.
- Vegas and Cubic were almost equal with Vegas greater by 2 sec.
- These results are in line with the general design of these variants where TCP veno is particularly designed for improved performance on Wireless networks.

Spikes

- We define an instantaneous throughput to be a spike when it is >=50 bps.
- Intuitively, it is a throughput much greater than its neighbouring throughputs.
- Total spikes:
  - cubic - 5
  - vegas - 8
  - veno - 6
- Max spike:
  - veno - 186 bps
  - vegas - 133 bps
  - cubic - 261 bps

Spikes

- Thus veno has the least number of spikes, but they are greater in magnitude than cubic and vegas.
- I.e. cubic and vegas are more uniform than veno.
- Explanation:

The veno is designed with wireless networks as a main goal and to achieve this the veno variant makes use of the maximum network queue length available to achieve this , moreover it uses the queue value to determine the difference between the congestive and non-congestive loss.

Therefore due to achieve maximum utilization of the network the veno is relatively more aggressive in achieving the peak network capacity.

Packet lengths

Packet lengths over QUIC (vegas)

Wireshark · Packet Lengths · Trace_WiFi_Vegas.pcap

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	18199	991.20	42	16304	0.0613	100%	2.3300	292.032
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	5516	71.40	42	79	0.0186	30.31%	0.6200	253.193
80-159	488	101.42	80	158	0.0016	2.68%	0.1600	291.265
160-319	119	229.96	161	312	0.0004	0.65%	0.0500	10.475
320-639	126	477.10	323	636	0.0004	0.69%	0.0500	293.079
640-1279	137	966.11	641	1270	0.0005	0.75%	0.0500	291.386
1280-2559	11617	1414.08	1290	2554	0.0391	63.83%	2.0700	292.032
2560-5119	162	3981.20	2710	5054	0.0005	0.89%	0.2400	253.771
5120 and greater	34	8923.41	6304	16304	0.0001	0.19%	0.0500	253.345

Packet length over QUIC (veno)

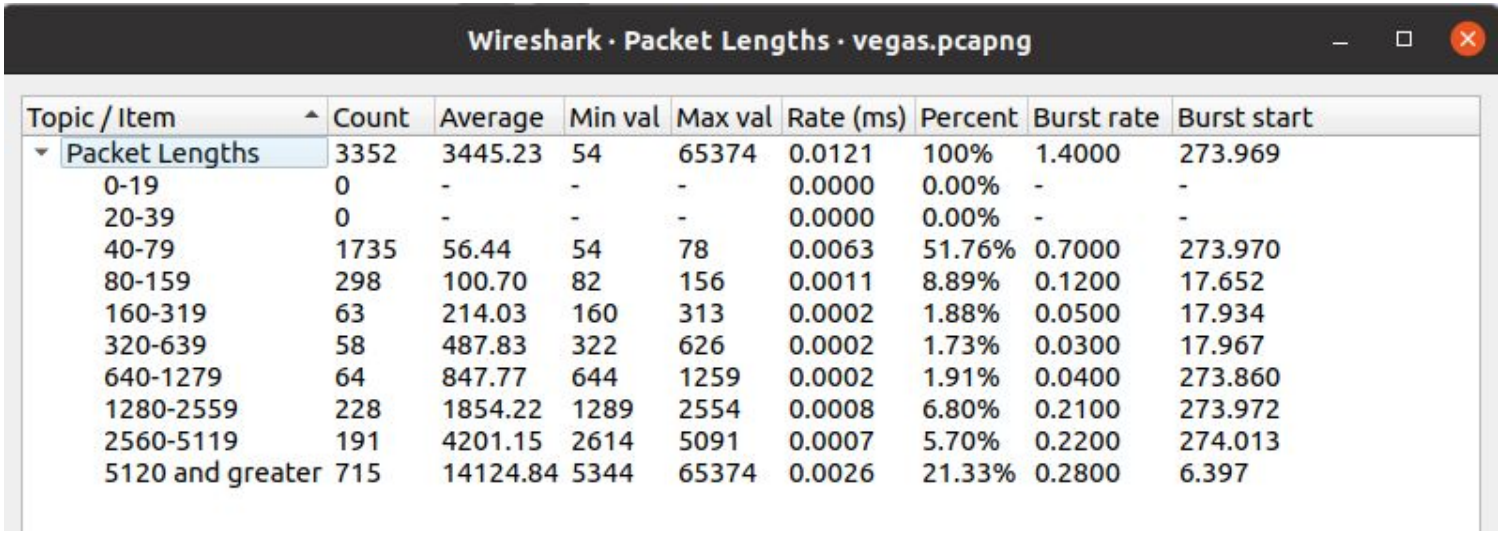
Wireshark · Packet Lengths · tcp_veno.pcap

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	16230	926.71	42	7354	0.0575	100%	4.1900	279.100
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	4837	72.37	42	79	0.0171	29.80%	0.8300	279.100
80-159	849	93.00	80	158	0.0030	5.23%	1.1200	280.833
160-319	66	215.02	160	313	0.0002	0.41%	0.0500	37.551
320-639	50	468.76	322	634	0.0002	0.31%	0.0400	268.345
640-1279	115	970.36	645	1279	0.0004	0.71%	0.0700	278.373
1280-2559	10300	1399.24	1285	2100	0.0365	63.46%	3.4600	279.120
2560-5119	10	3149.90	2914	3863	0.0000	0.06%	0.0200	28.140
5120 and greater	3	6246.00	5583	7354	0.0000	0.02%	0.0200	37.551

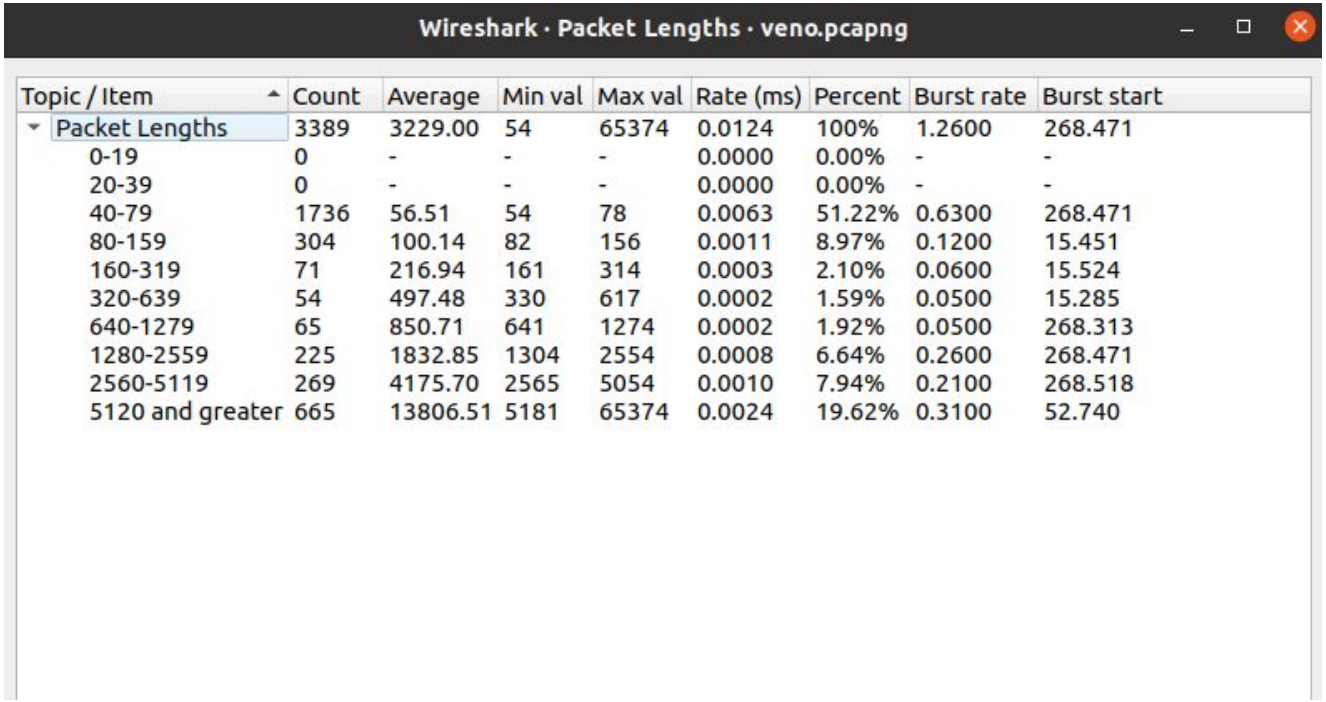
Packet lengths over QUIC(cubic)

Wireshark · Packet Lengths · tcp_baseline.pcap

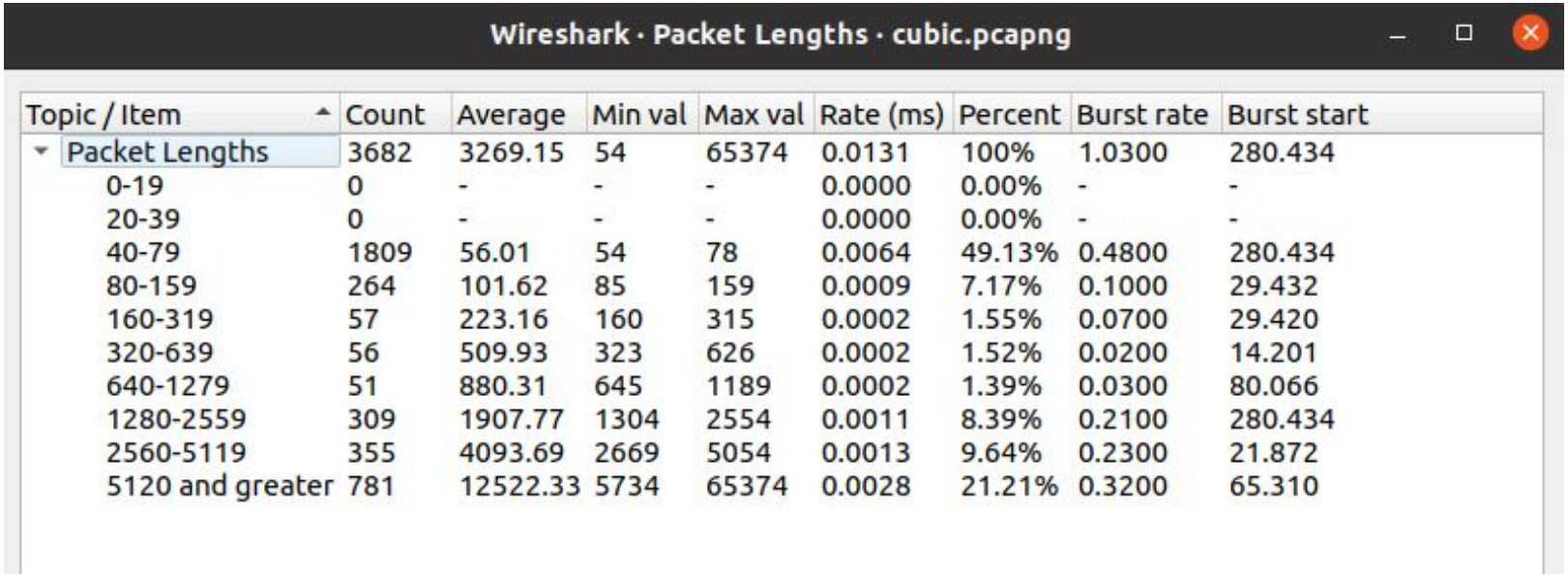
Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	19576	917.71	42	18367	0.0675	100%	3.8100	288.475
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	6293	70.83	42	79	0.0217	32.15%	0.6300	288.469
80-159	986	96.90	80	155	0.0034	5.04%	0.6200	288.547
160-319	138	223.02	161	319	0.0005	0.70%	0.0600	12.204
320-639	126	484.35	320	637	0.0004	0.64%	0.0600	3.495
640-1279	208	892.24	649	1266	0.0007	1.06%	0.0600	287.882
1280-2559	11690	1400.27	1282	2463	0.0403	59.72%	2.9800	288.470
2560-5119	77	3304.30	2707	4877	0.0003	0.39%	0.1800	287.248
5120 and greater	58	9015.86	5774	18367	0.0002	0.30%	0.1800	287.252



Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	3352	3445.23	54	65374	0.0121	100%	1.4000	273.969
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1735	56.44	54	78	0.0063	51.76%	0.7000	273.970
80-159	298	100.70	82	156	0.0011	8.89%	0.1200	17.652
160-319	63	214.03	160	313	0.0002	1.88%	0.0500	17.934
320-639	58	487.83	322	626	0.0002	1.73%	0.0300	17.967
640-1279	64	847.77	644	1259	0.0002	1.91%	0.0400	273.860
1280-2559	228	1854.22	1289	2554	0.0008	6.80%	0.2100	273.972
2560-5119	191	4201.15	2614	5091	0.0007	5.70%	0.2200	274.013
5120 and greater	715	14124.84	5344	65374	0.0026	21.33%	0.2800	6.397



Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	3389	3229.00	54	65374	0.0124	100%	1.2600	268.471
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1736	56.51	54	78	0.0063	51.22%	0.6300	268.471
80-159	304	100.14	82	156	0.0011	8.97%	0.1200	15.451
160-319	71	216.94	161	314	0.0003	2.10%	0.0600	15.524
320-639	54	497.48	330	617	0.0002	1.59%	0.0500	15.285
640-1279	65	850.71	641	1274	0.0002	1.92%	0.0500	268.313
1280-2559	225	1832.85	1304	2554	0.0008	6.64%	0.2600	268.471
2560-5119	269	4175.70	2565	5054	0.0010	7.94%	0.2100	268.518
5120 and greater	665	13806.51	5181	65374	0.0024	19.62%	0.3100	52.740

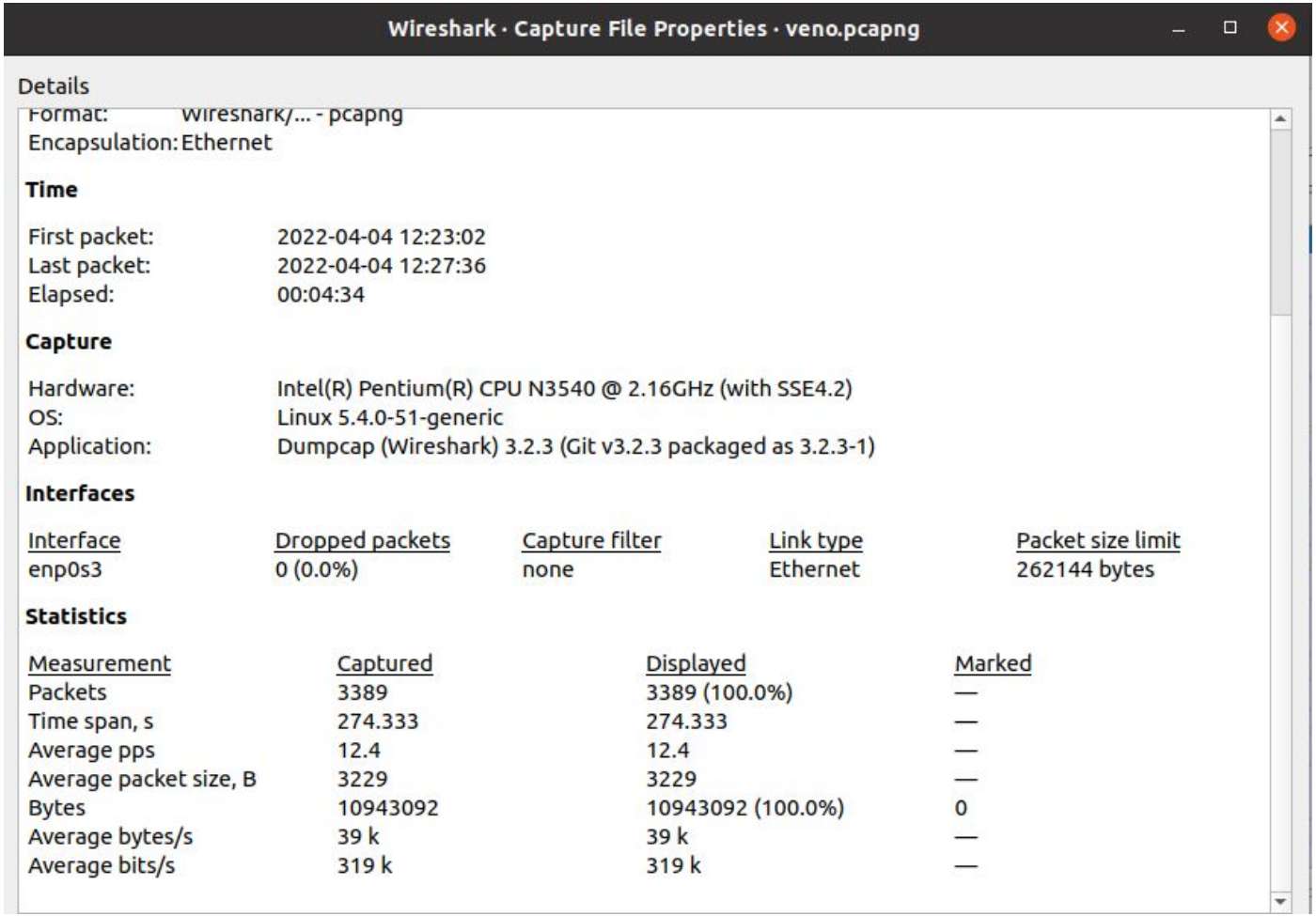


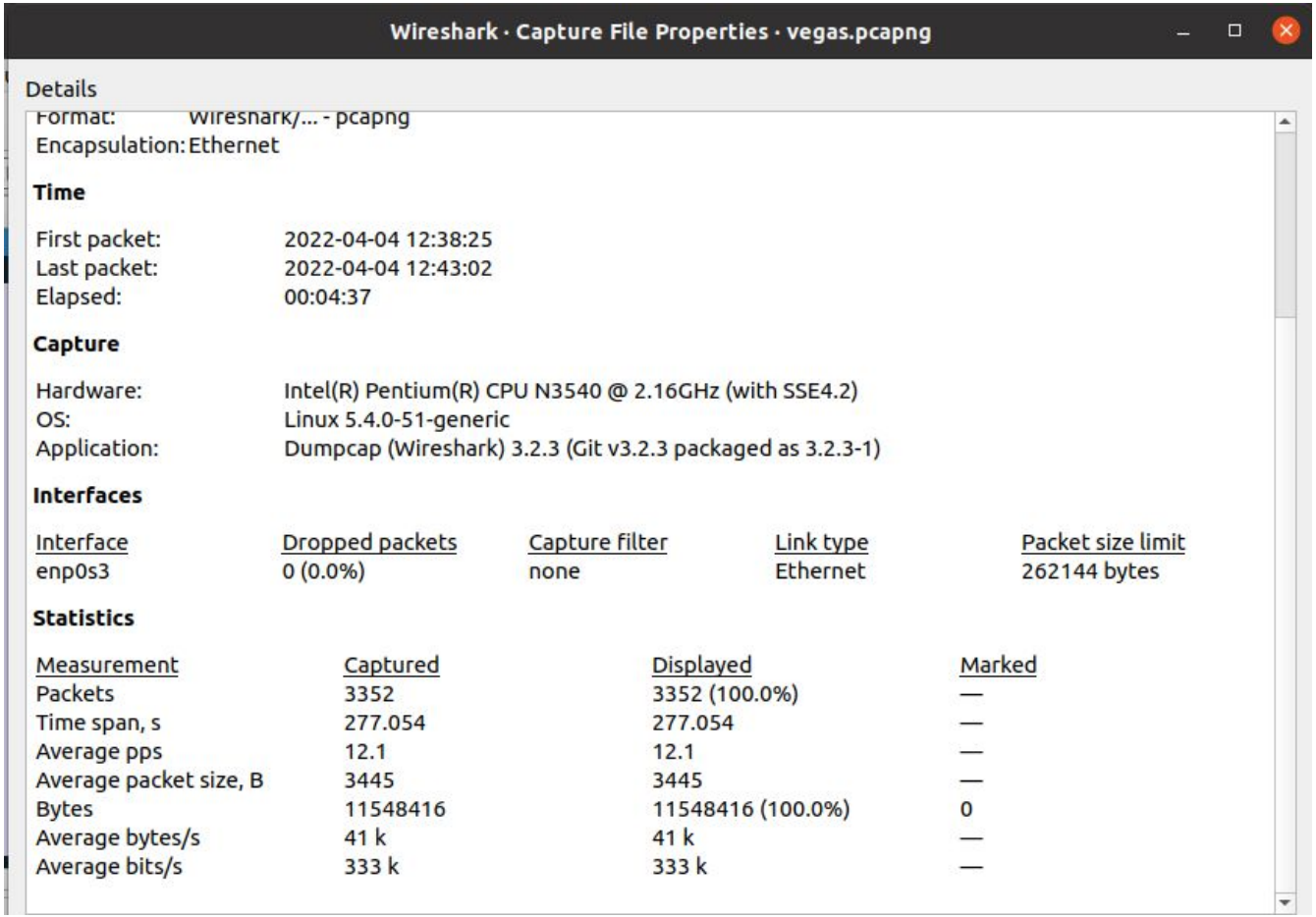
Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	3682	3269.15	54	65374	0.0131	100%	1.0300	280.434
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	1809	56.01	54	78	0.0064	49.13%	0.4800	280.434
80-159	264	101.62	85	159	0.0009	7.17%	0.1000	29.432
160-319	57	223.16	160	315	0.0002	1.55%	0.0700	29.420
320-639	56	509.93	323	626	0.0002	1.52%	0.0200	14.201
640-1279	51	880.31	645	1189	0.0002	1.39%	0.0300	80.066
1280-2559	309	1907.77	1304	2554	0.0011	8.39%	0.2100	280.434
2560-5119	355	4093.69	2669	5054	0.0013	9.64%	0.2300	21.872
5120 and greater	781	12522.33	5734	65374	0.0028	21.21%	0.3200	65.310

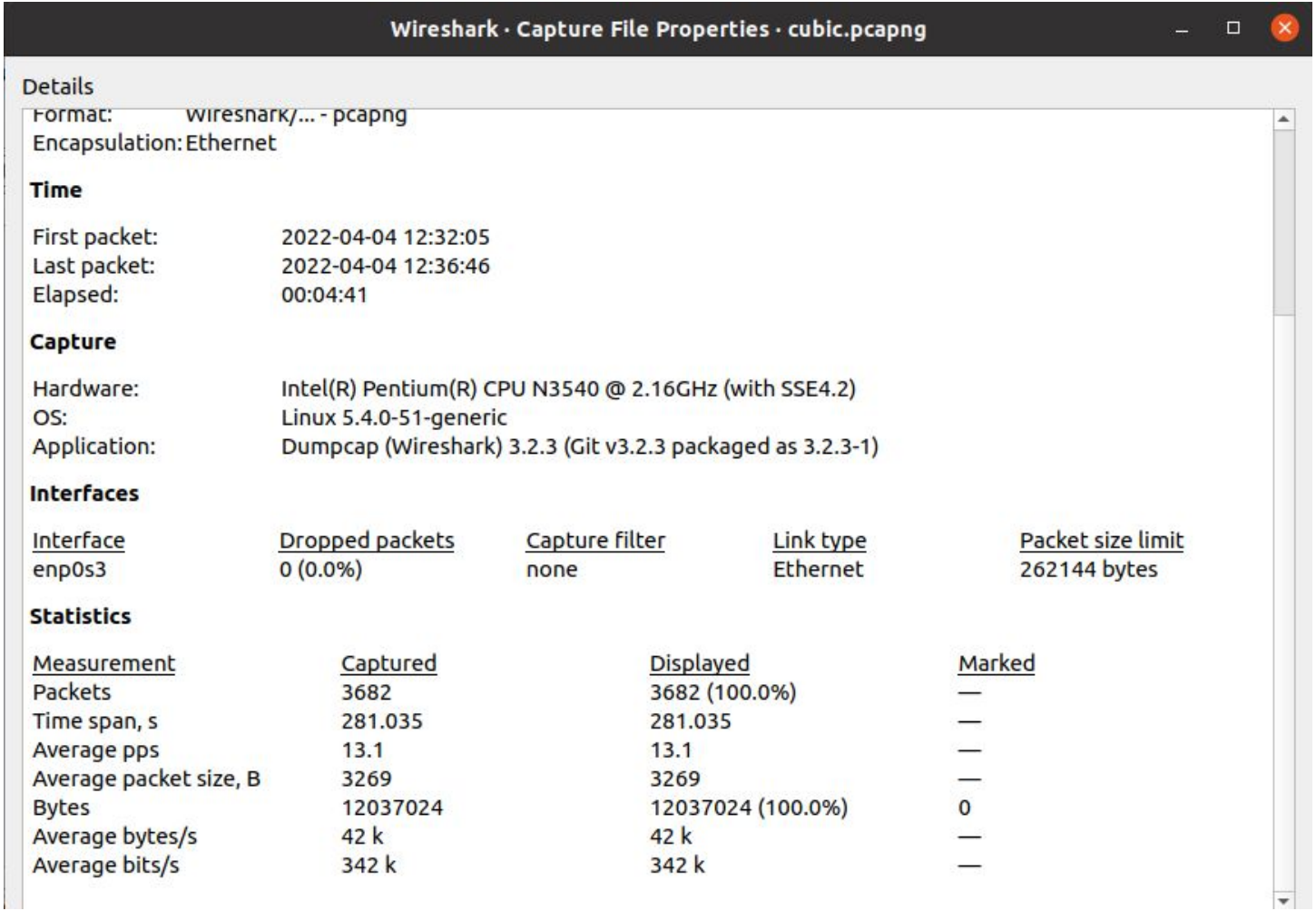
Analysis based on packet length

- Packets over any TCP variants are relatively very large when compared to packet size over QUIC protocol.
- One notable difference is that over a sufficiently high bandwidth there was no perceptible difference in performance of video playback.
- **NOTE**:
  - Since quic protocol sends a large amount of medium sized packets it may be suitable for low bandwidth connections as recovering from packet loss will be faster but with increased buffering time for the video playback.
  - Whereas in TCP protocol sends large sized packets thus reducing the number of packets that needs to be transmitted but the cost of packet loss is high when compared to QUIC but may result in lower buffering as a relatively large chunk of data is already received for playback.

General


Wireshark · Capture File Properties · veno.pcapng
Details
Format: Wireshark/... - pcapng
Encapsulation: Ethernet
Time
First packet: 2022-04-04 12:23:02
Last packet: 2022-04-04 12:27:36
Elapsed: 00:04:34
Capture
Hardware: Intel(R) Pentium(R) CPU N3540 @ 2.16GHz (with SSE4.2)
OS: Linux 5.4.0-51-generic
Application: Dumpcap (Wireshark) 3.2.3 (Git v3.2.3 packaged as 3.2.3-1)
Interfaces
Interface Dropped packets Capture filter Link type Packet size limit
enp0s3 0 (0.0%) none Ethernet 262144 bytes
Statistics
Measurement Captured Displayed Marked
Packets 3389 3389 (100.0%) —
Time span, s 274.333 274.333 —
Average pps 12.4 12.4 —
Average packet size, B 3229 3229 —
Bytes 10943092 10943092 (100.0%) 0
Average bytes/s 39 k 39 k —
Average bits/s 319 k 319 k —


Wireshark · Capture File Properties · vegas.pcapng
Details
Format: Wireshark/... - pcapng
Encapsulation: Ethernet
Time
First packet: 2022-04-04 12:38:25
Last packet: 2022-04-04 12:43:02
Elapsed: 00:04:37
Capture
Hardware: Intel(R) Pentium(R) CPU N3540 @ 2.16GHz (with SSE4.2)
OS: Linux 5.4.0-51-generic
Application: Dumpcap (Wireshark) 3.2.3 (Git v3.2.3 packaged as 3.2.3-1)
Interfaces
Interface Dropped packets Capture filter Link type Packet size limit
enp0s3 0 (0.0%) none Ethernet 262144 bytes
Statistics
Measurement Captured Displayed Marked
Packets 3352 3352 (100.0%) —
Time span, s 277.054 277.054 —
Average pps 12.1 12.1 —
Average packet size, B 3445 3445 —
Bytes 11548416 11548416 (100.0%) 0
Average bytes/s 41 k 41 k —
Average bits/s 333 k 333 k —

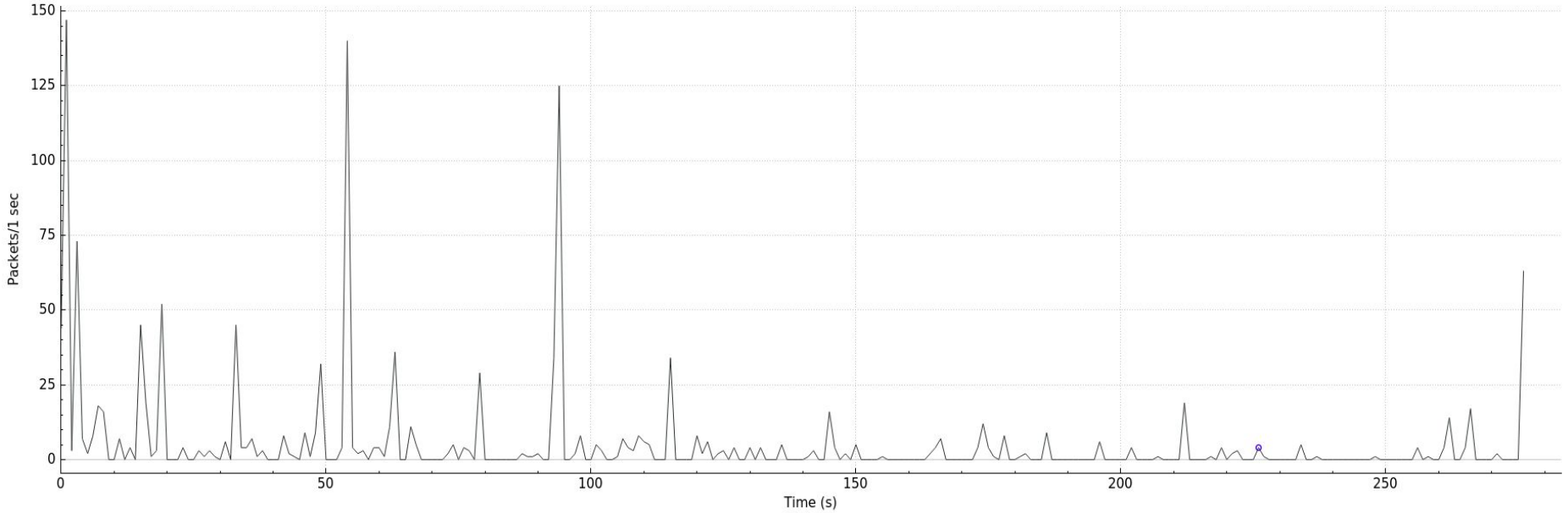

Wireshark · Capture File Properties · cubic.pcapng
Details
Format: Wireshark/... - pcapng
Encapsulation: Ethernet
Time
First packet: 2022-04-04 12:32:05
Last packet: 2022-04-04 12:36:46
Elapsed: 00:04:41
Capture
Hardware: Intel(R) Pentium(R) CPU N3540 @ 2.16GHz (with SSE4.2)
OS: Linux 5.4.0-51-generic
Application: Dumpcap (Wireshark) 3.2.3 (Git v3.2.3 packaged as 3.2.3-1)
Interfaces
Interface Dropped packets Capture filter Link type Packet size limit
enp0s3 0 (0.0%) none Ethernet 262144 bytes
Statistics
Measurement Captured Displayed Marked
Packets 3682 3682 (100.0%) —
Time span, s 281.035 281.035 —
Average pps 13.1 13.1 —
Average packet size, B 3269 3269 —
Bytes 12037024 12037024 (100.0%) 0
Average bytes/s 42 k 42 k —
Average bits/s 342 k 342 k —

Observations

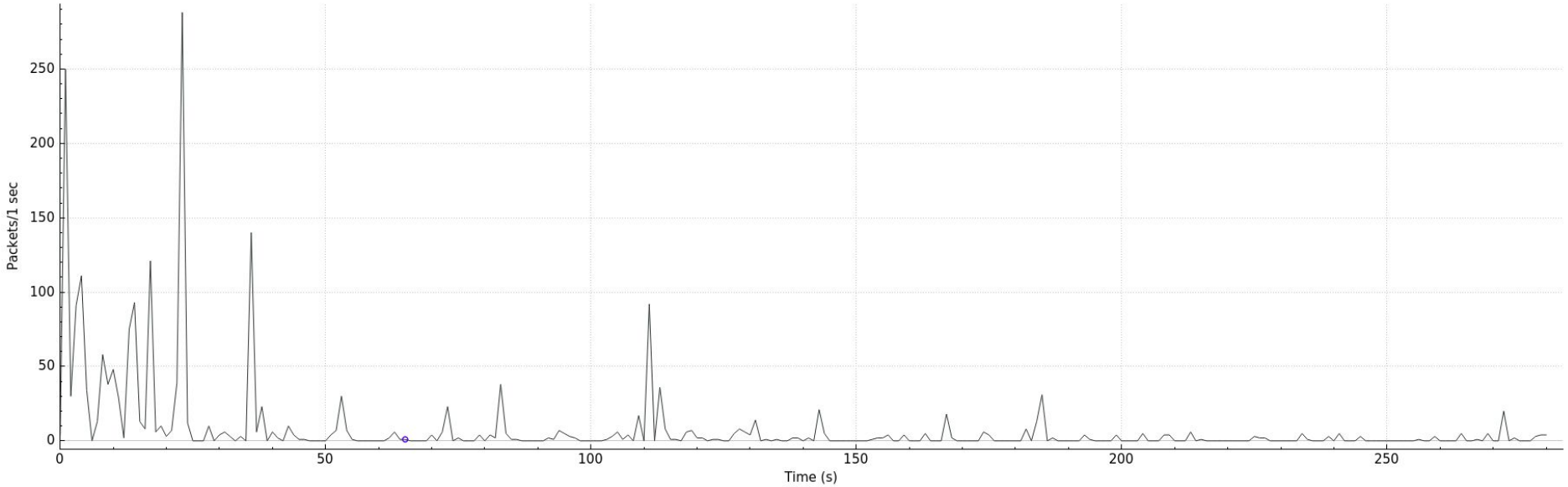
- Average packet size:
  - veno<cubic <vegas
- Average throughput:
  - veno<vegas<cubic
-

Ethernet

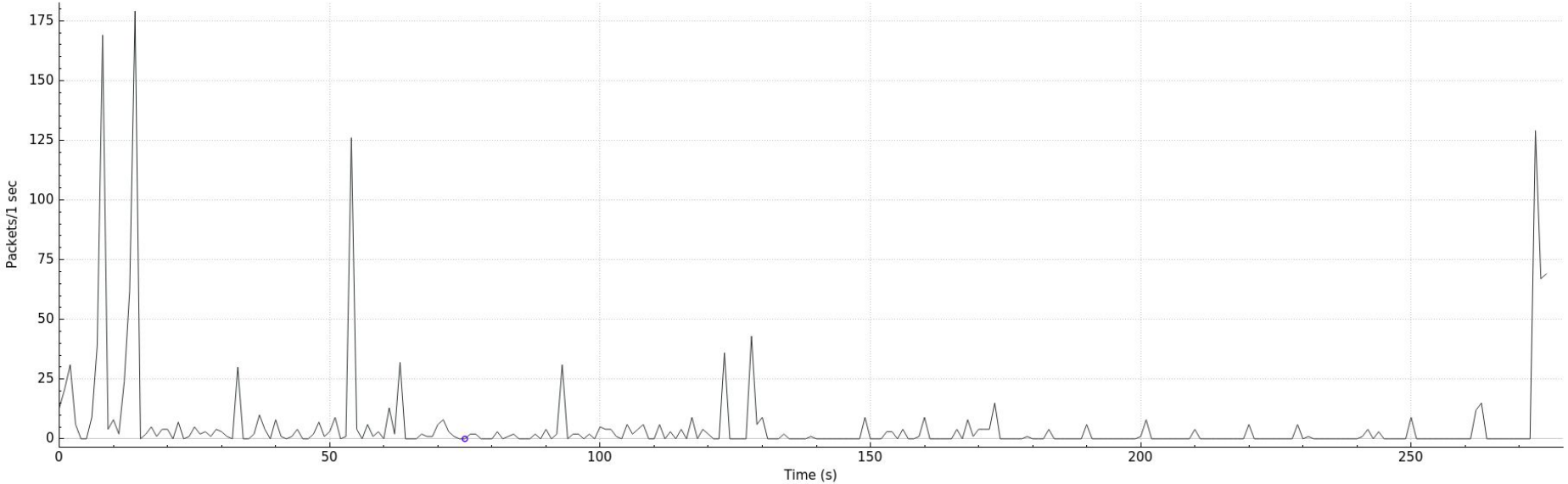
Throughput graphs



TCP cubic



TCP Veno


Packets/1 sec
175
150
125
100
75
50
25
0
0
50
100
150
200
250
Time (s)

TCP Vegas

Throughput observations

- Video load time:
  - vegas - 131 sec
  - veno - 112 sec
  - cubic - 116 sec
- We notice that the video loaded the fastest in TCP Veno.
- Veno and Cubic were almost equal with Veno lower by 4 sec.
- On comparison with the Wi-Fi throughput observations we can see that the wifi network conditions being less obstructive. The readings for the current gen Wi-Fi 5Ghz are comparable to ethernet but testing over congested network is yet to be studied.

Spikes

- We define an instantaneous throughput to a spike when it is >=50 p/s.
- Intuitively, it is a throughput much greater than its neighbouring throughputs.
- Total spikes:
  - cubic - 5
  - vegas - 3
  - veno - 8
- Max spike:
  - cubic - 147 p/s
  - vegas - 179 p/s
  - veno - 288 p/s
- Vegas has the least number of spikes, but they are greater in magnitude than cubic and lesser than veno.

Further work (Recommendations)

- Experiment for difference with network congestion
  - As the main advantage of TCP variants is exploited mostly in the case of network congestion and recovery.
- Carry out multiple iterations of each experiment and take the median values

Final Analysis

Real Time TCP Analysis

- Application: YouTube video of length 4:30 min
- Played on Mozilla Firefox 75
- Variants used:
  - TCP Veno
  - TCP Cubic
  - TCP Vegas

AGENDA

- Analysing the 3 variants w.r.t.
  - Congested Ethernet network
  - Congested WiFi network
  - High and low WiFi signal strength

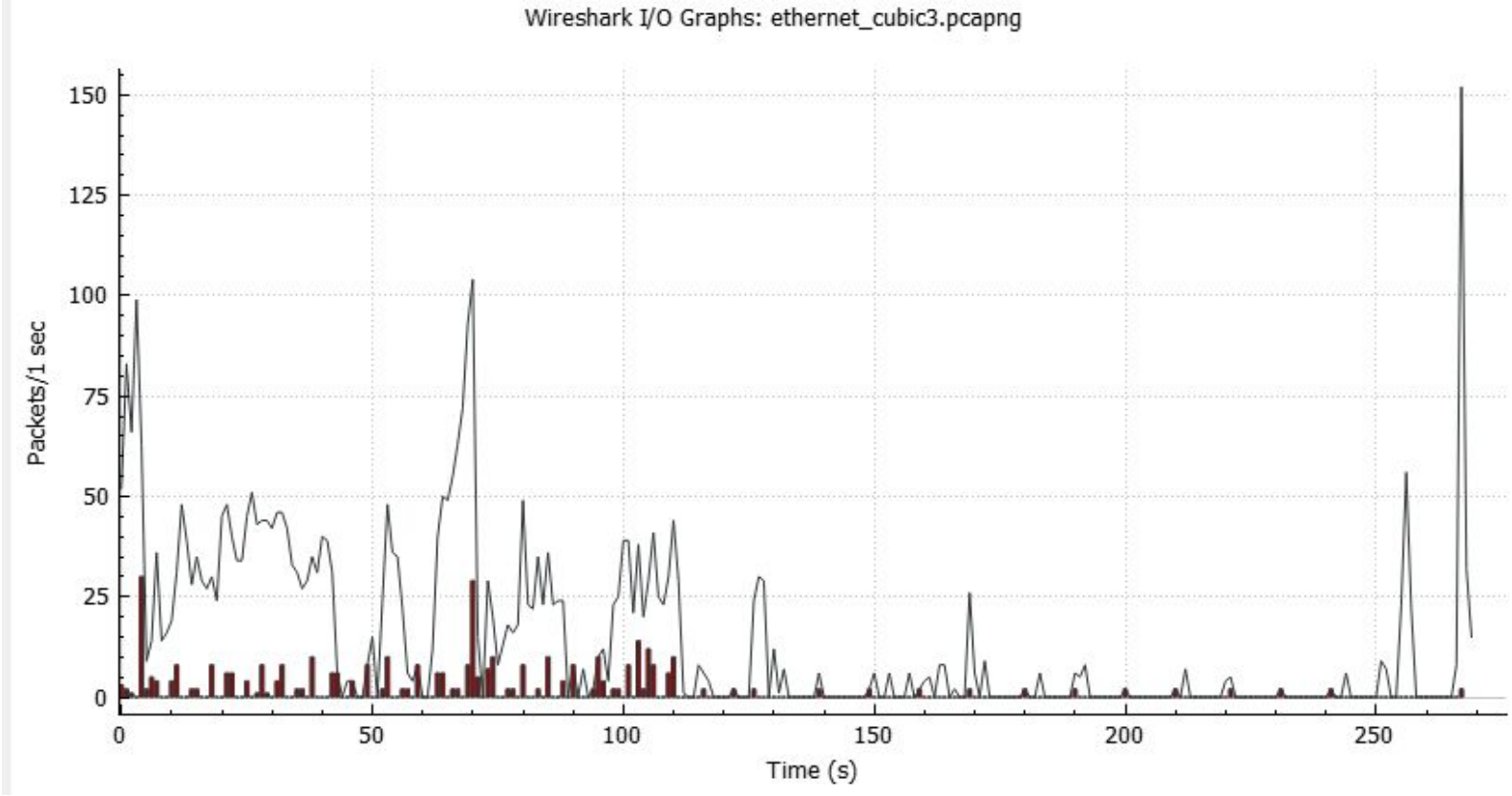
TCP variants performance based on Network congestion (Ethernet)

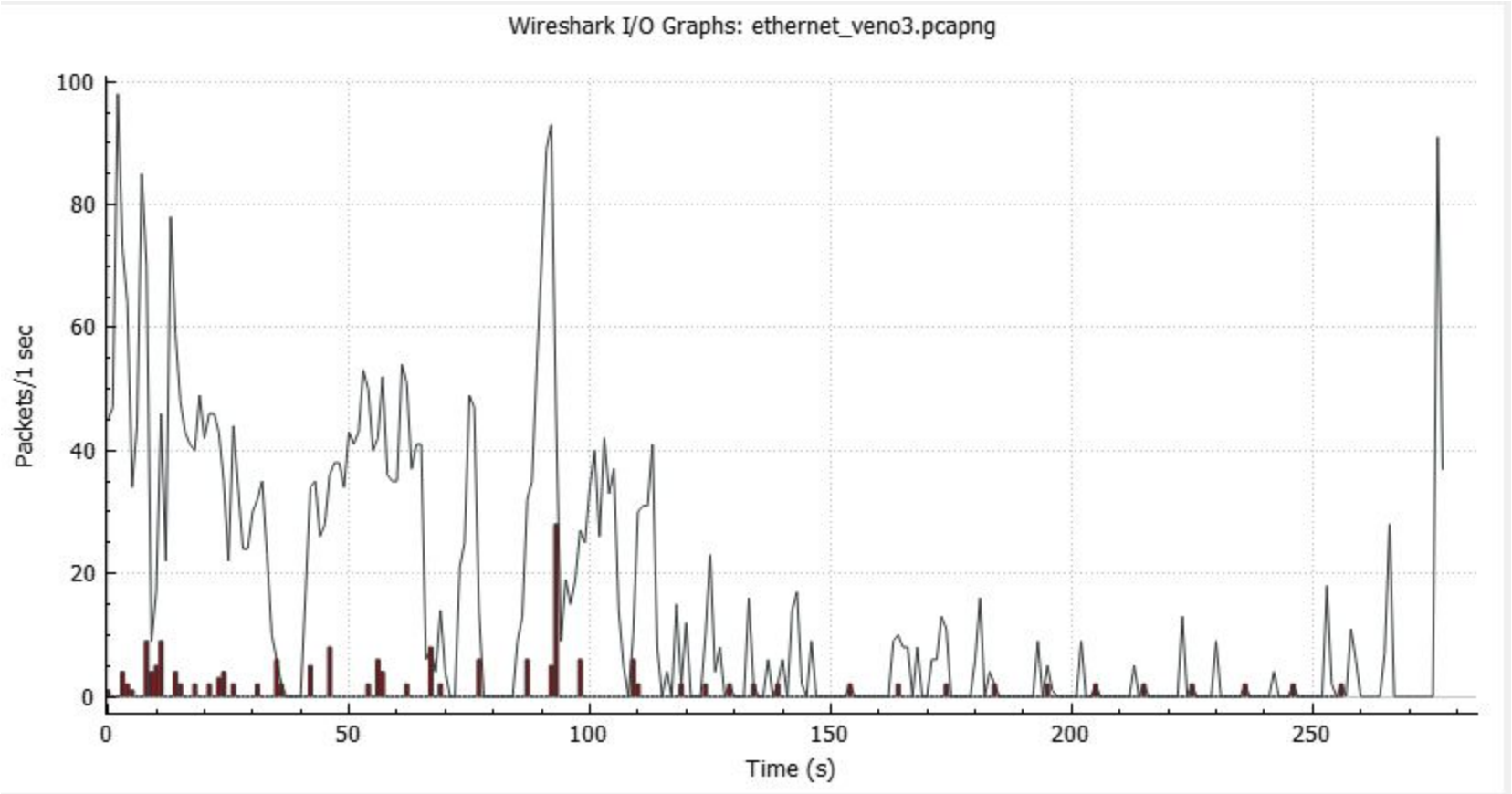
Setup: Ethernet cable connecting laptop to Airtel broadband router. “Wondershaper” tool uses bash scripts on linux to limit upload and download datarate on any application.

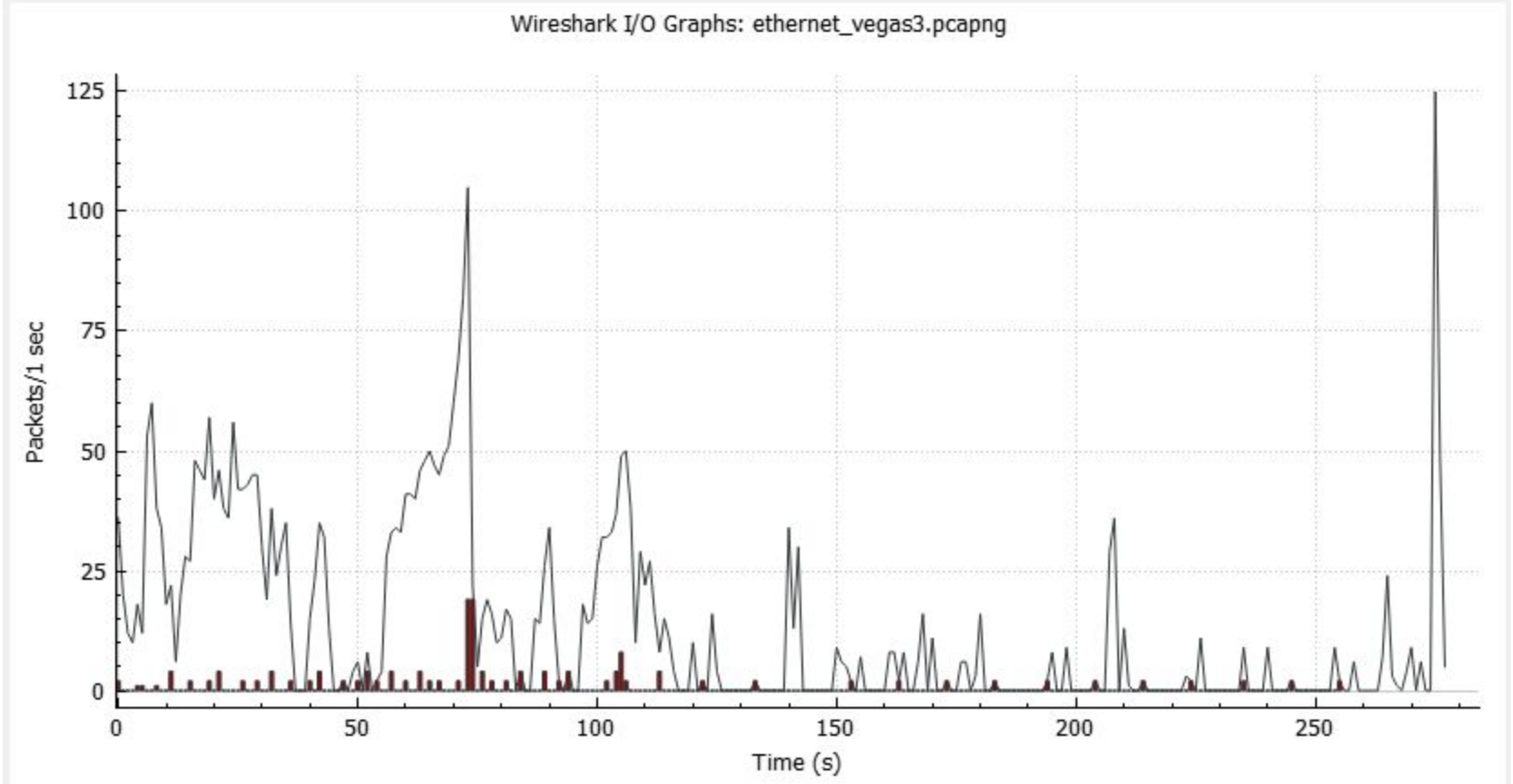
Upload rate set: 512 Kbps

Download rate set: 1024 Kbps

Reference: https://github.com/magnific0/wondershaper


Wireshark I/O Graphs: ethernet_cubic3.pcapng
Packets/1 sec
150
125
100
75
50
25
0
0
50
100
150
200
250
Time (s)


Wireshark I/O Graphs: ethernet_veno3.pcapng
Packets/1 sec
Time (s)
0
20
40
60
80
100
0
50
100
150
200
250


Wireshark I/O Graphs: ethernet_vegas3.pcapng
Packets/1 sec
125
100
75
50
25
0
0
50
100
150
200
250
Time (s)

Throughput

Units:KB/s

	1	2	3	4	5	median
TCP cubic	39	40	41	43	41	41
TCP veno	41	41	40	41	44	41
TCP vegas	39	39	40	41	41	40

Number of Retransmissions

Retransmissions here refer to total bytes of TCP retransmitted packets

	1	2	3	4	5	median
TCP cubic	40212	56851	49587	47146	107703	49587
TCP veno	1913	46985	38535	44277	15791	38535
TCP vegas	11621	43003	16852	37958	14992	16852

Most of the retransmissions bytes were occurred in cubic, then in veno, then vegas.

Number of Dupacks

No. of duplicate acks

	1	2	3	4	5	median
TCP cubic	193	222	277	371	249	249
TCP veno	142	188	105	134	115	134
TCP vegas	121	196	113	104	148	121

Number of Out of Order Packets

	1	2	3	4	5	median
TCP cubic	1	0	0	4	0	0
TCP veno	0	1	0	0	0	0
TCP vegas	0	0	0	1	1	0

TCP variants performance based on Network Congestion(WiFi)

Congested Network Conditions

- Real world (college Lecture Hall Wi-Fi AP)

  - signal strength: -79 to -81 dBm

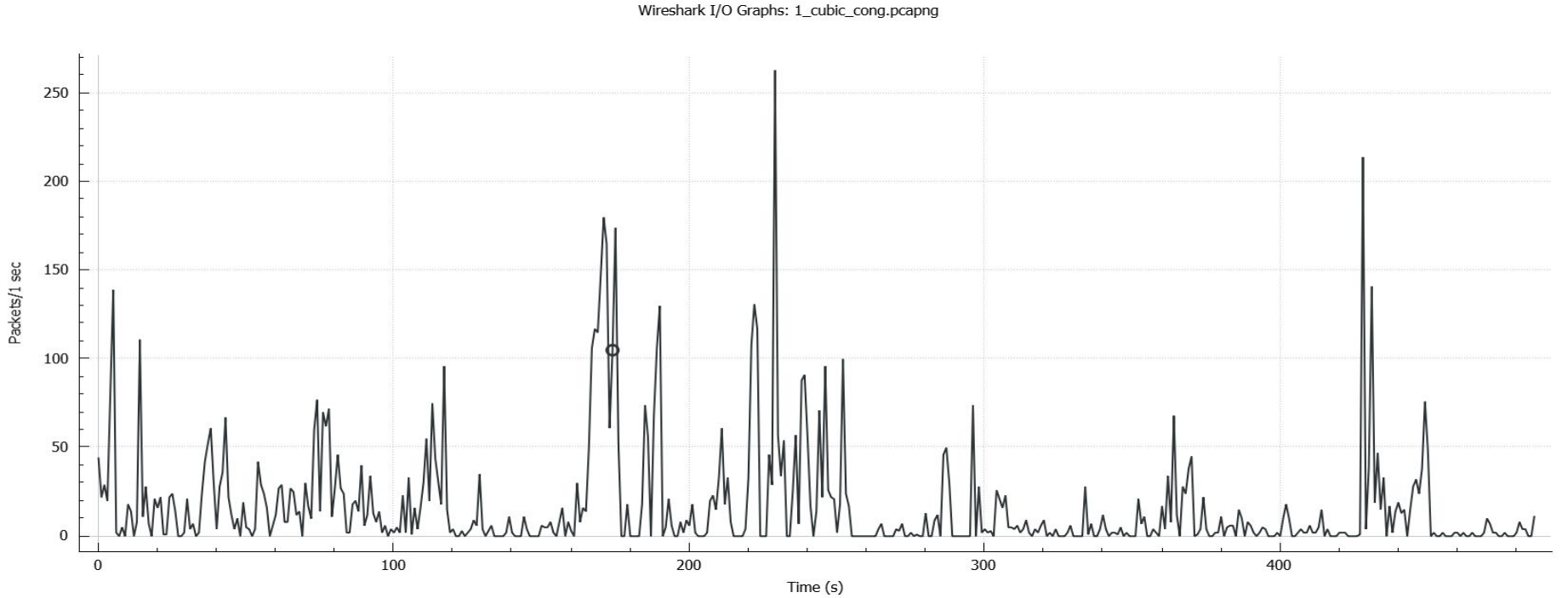
  - Download: 1.89 Mbps , Upload: 0.01 Mbps

  - In the following slides we will see the higher error rates compared to a stable environment and network conditions(Slide 76).

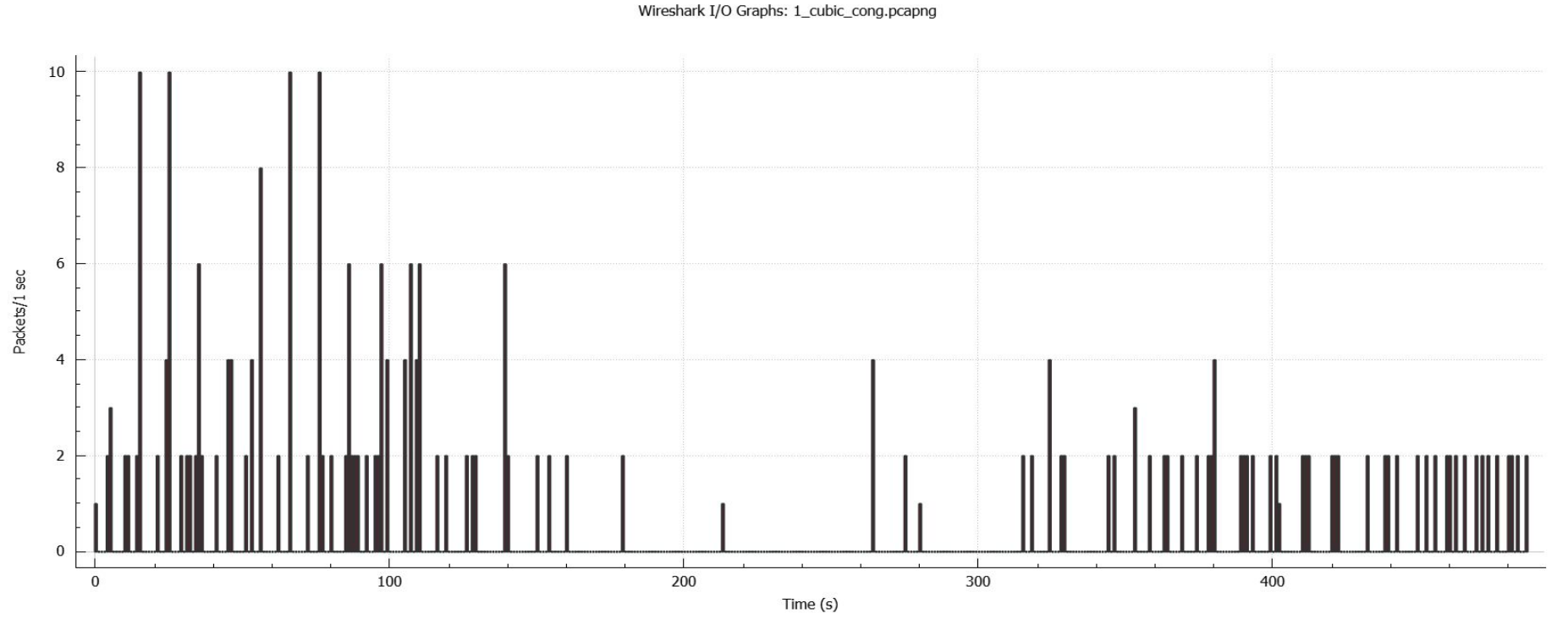
  NOTE: To avoid caching and cookie storage at the client level all test were done in an incognito session/private session.

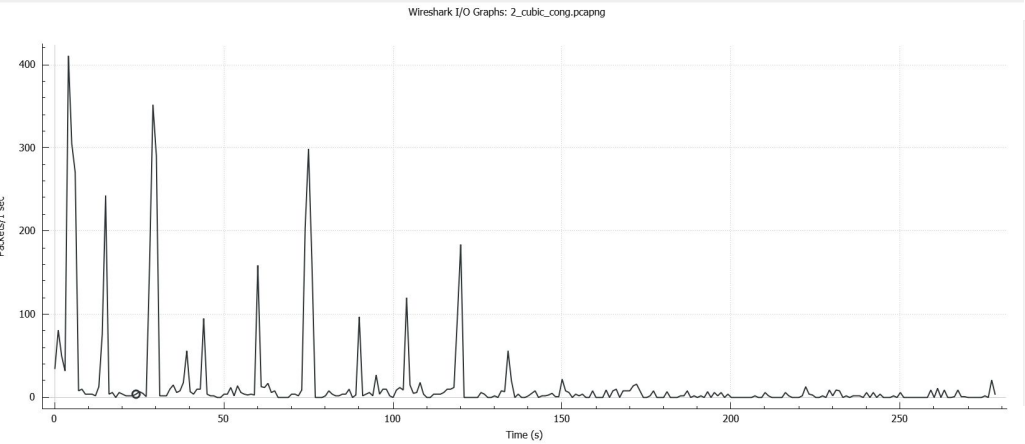
Figure: WiFi Network Conditions at Lecture Hall

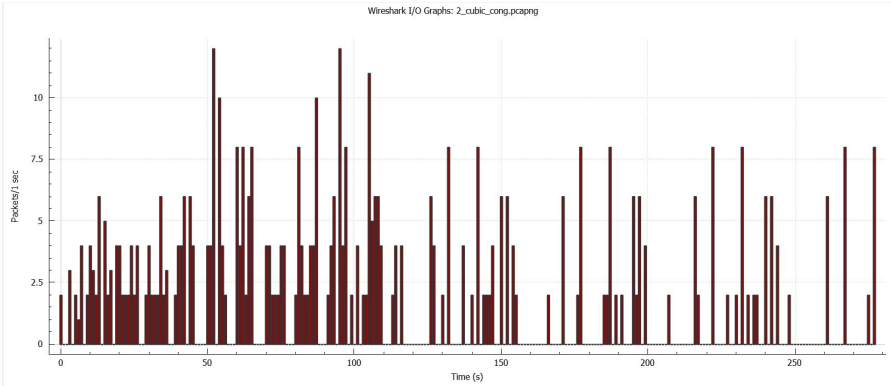
TCP cubic

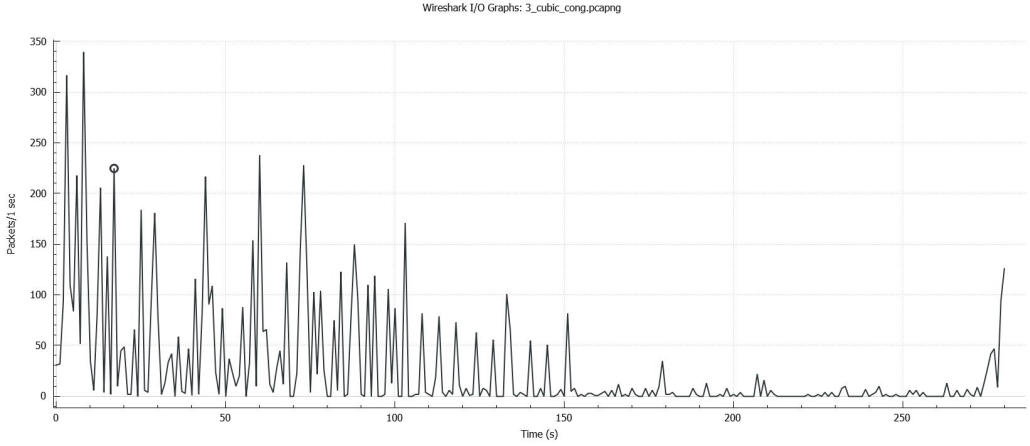

Wireshark I/O Graphs: 1_cubic_cong.pcapng
Packets/1 sec
250
200
150
100
50
0
0
100
200
300
400
Time (s)

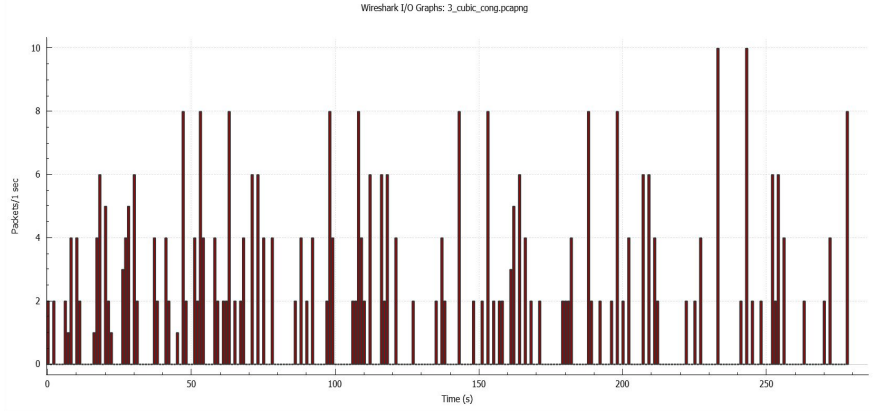
Errors


Wireshark I/O Graphs: 1_cubic_cong.pcapng
Packets/1 sec
10
8
6
4
2
0
0
100
200
300
400
Time (s)

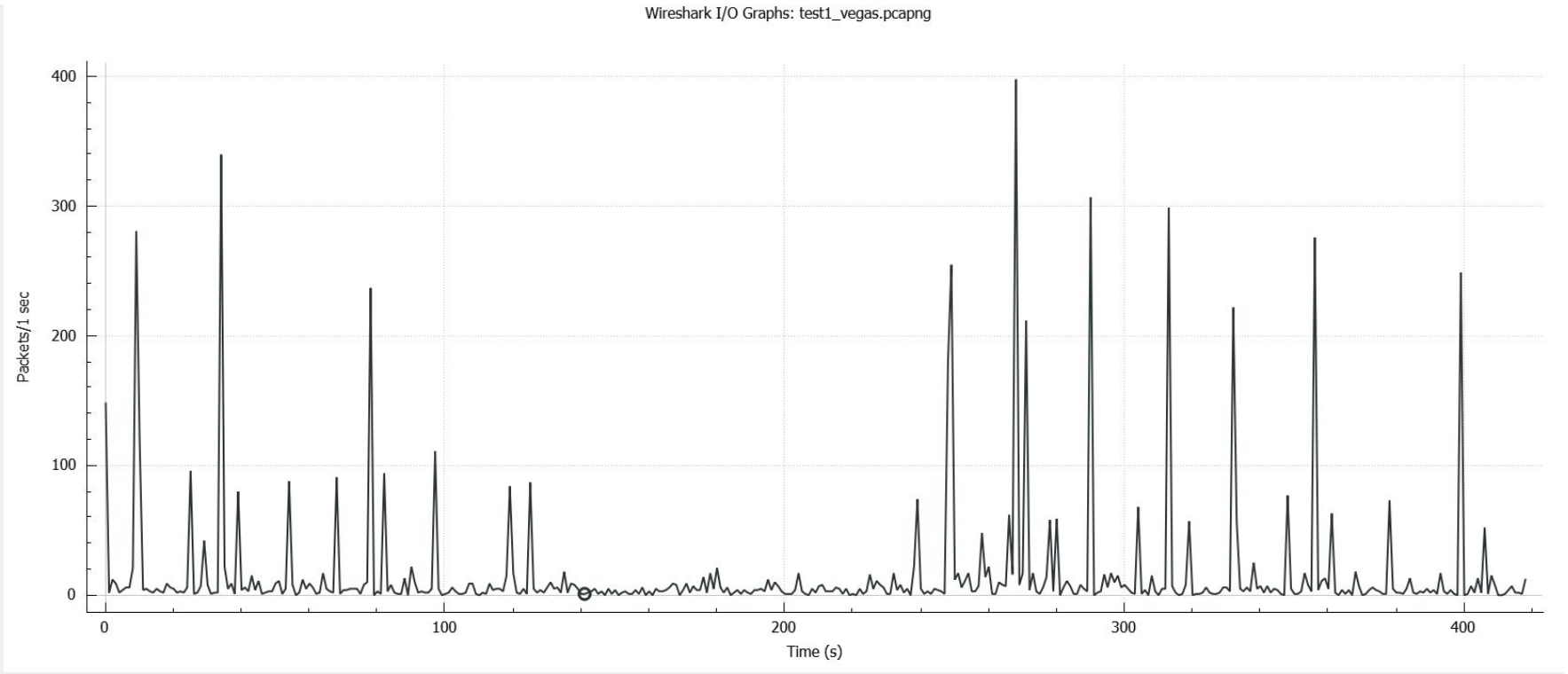

Wireshark I/O Graphs: 2_cubic_cong.pcapng
Packets/1 sec
Time (s)


Wireshark I/O Graphs: 2_cubic_cong.pcapng
Packets/1 sec
Time (s)

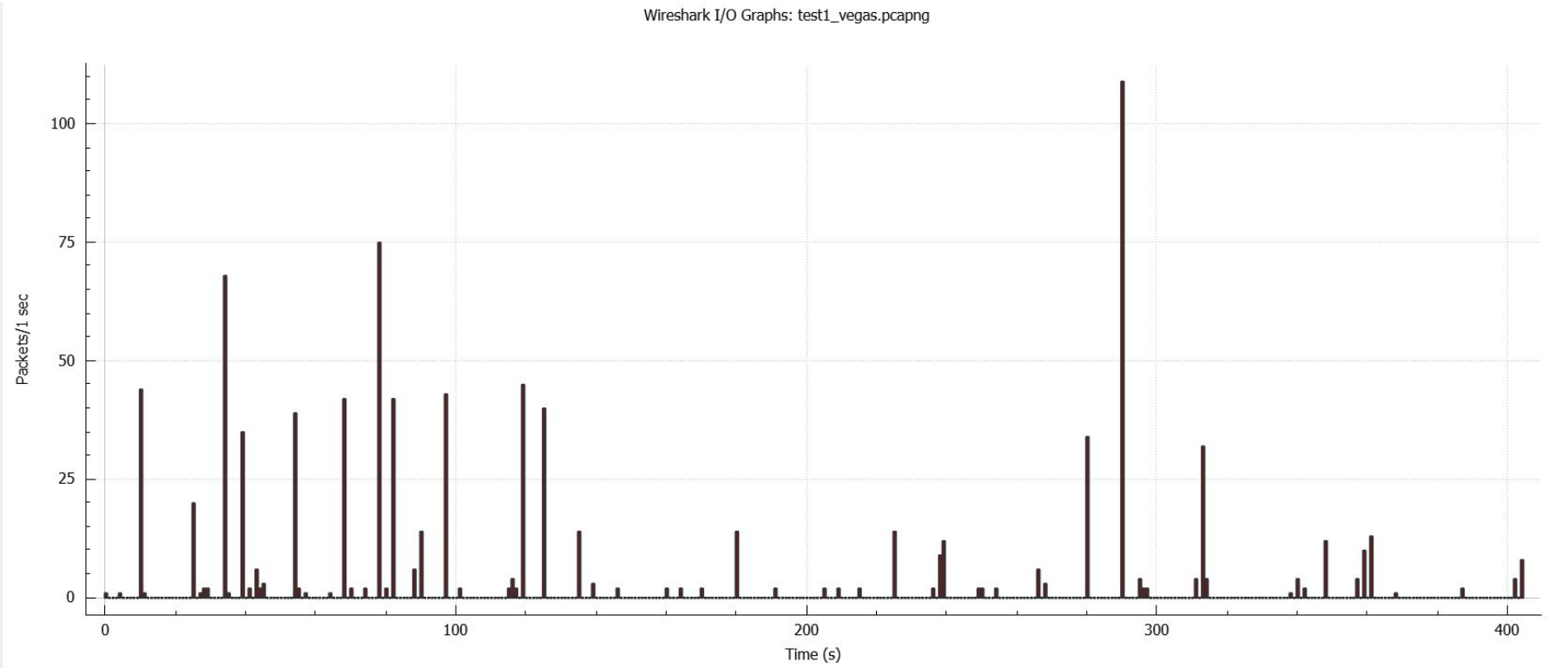

Wireshark I/O Graphs: 3_cubic_cong.pcapng
Packets/1 sec
Time (s)

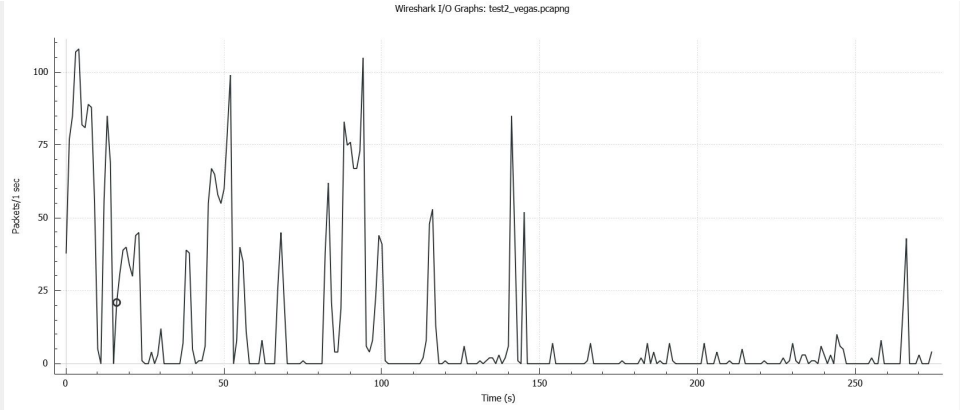

Wireshark I/O Graphs: 3_cubic_cong.pcapng
Packets/1 sec
Time (s)

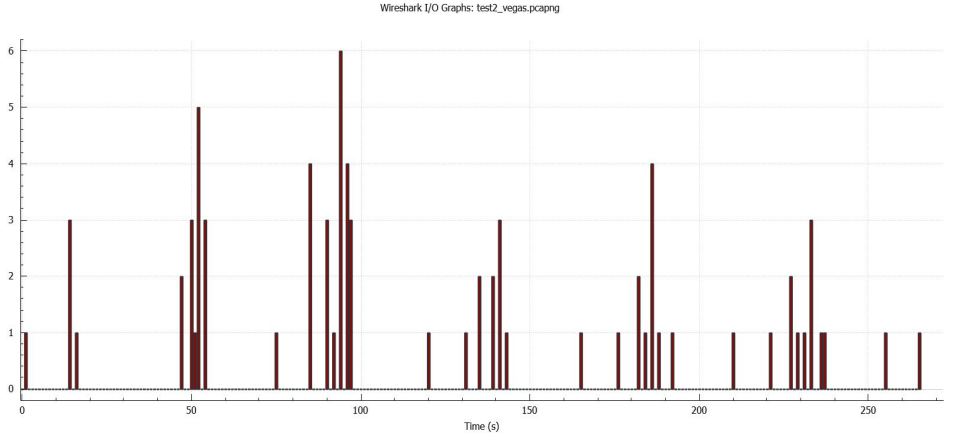
TCP vegas

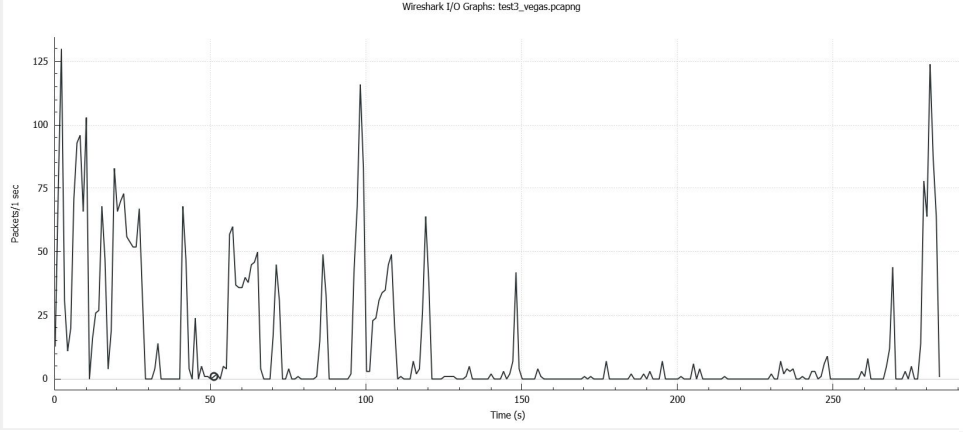

Wireshark I/O Graphs: test1_vegas.pcapng
Packets/1 sec
400
300
200
100
0
0
100
200
300
400
Time (s)

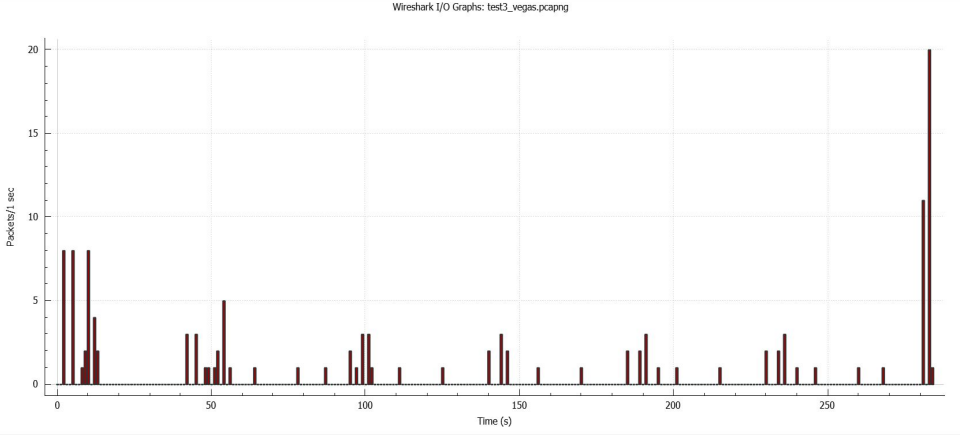
Errors


Wireshark I/O Graphs: test1_vegas.pcapng
Packets/1 sec
100
75
50
25
0
0
100
200
300
400
Time (s)

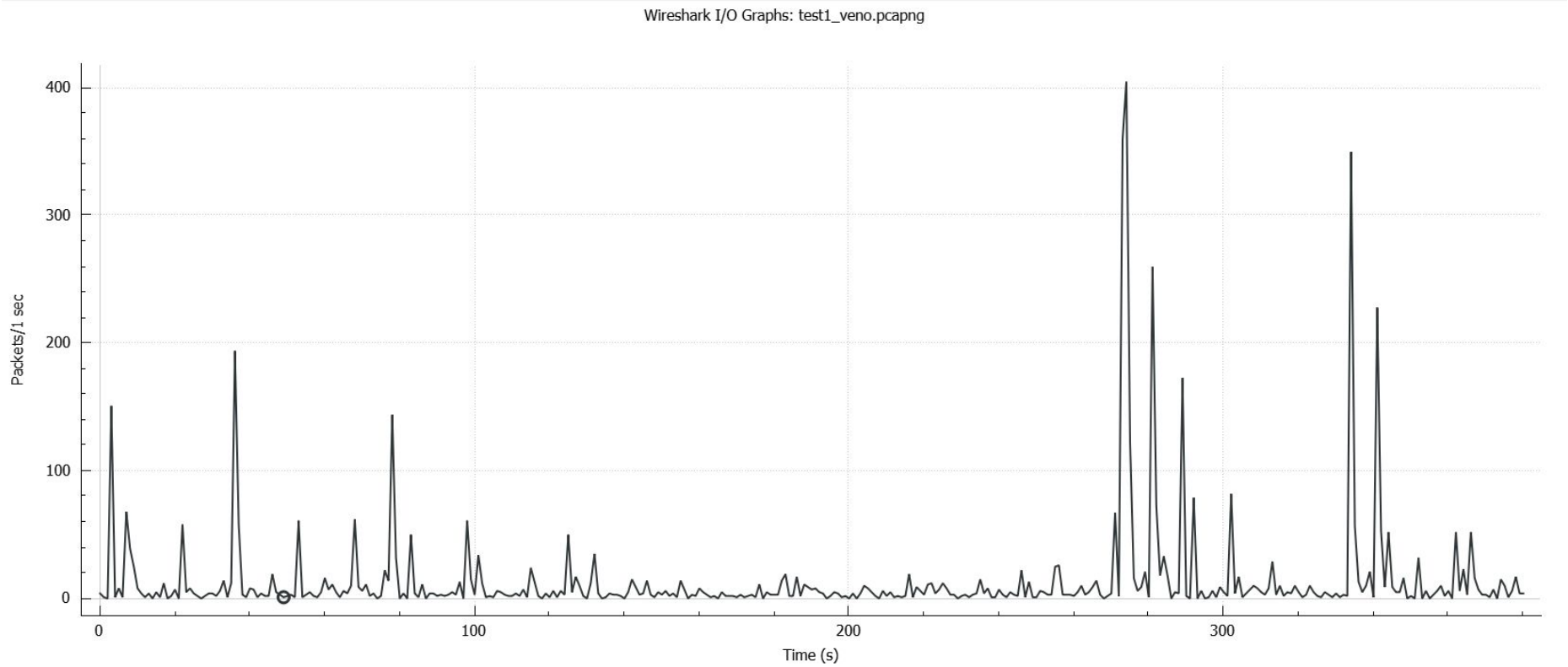

Wireshark I/O Graphs: test2_vegas.pcapng
Packets/1 sec
Time (s)


Wireshark I/O Graphs: test2_vegas.pcapng
Time (s)

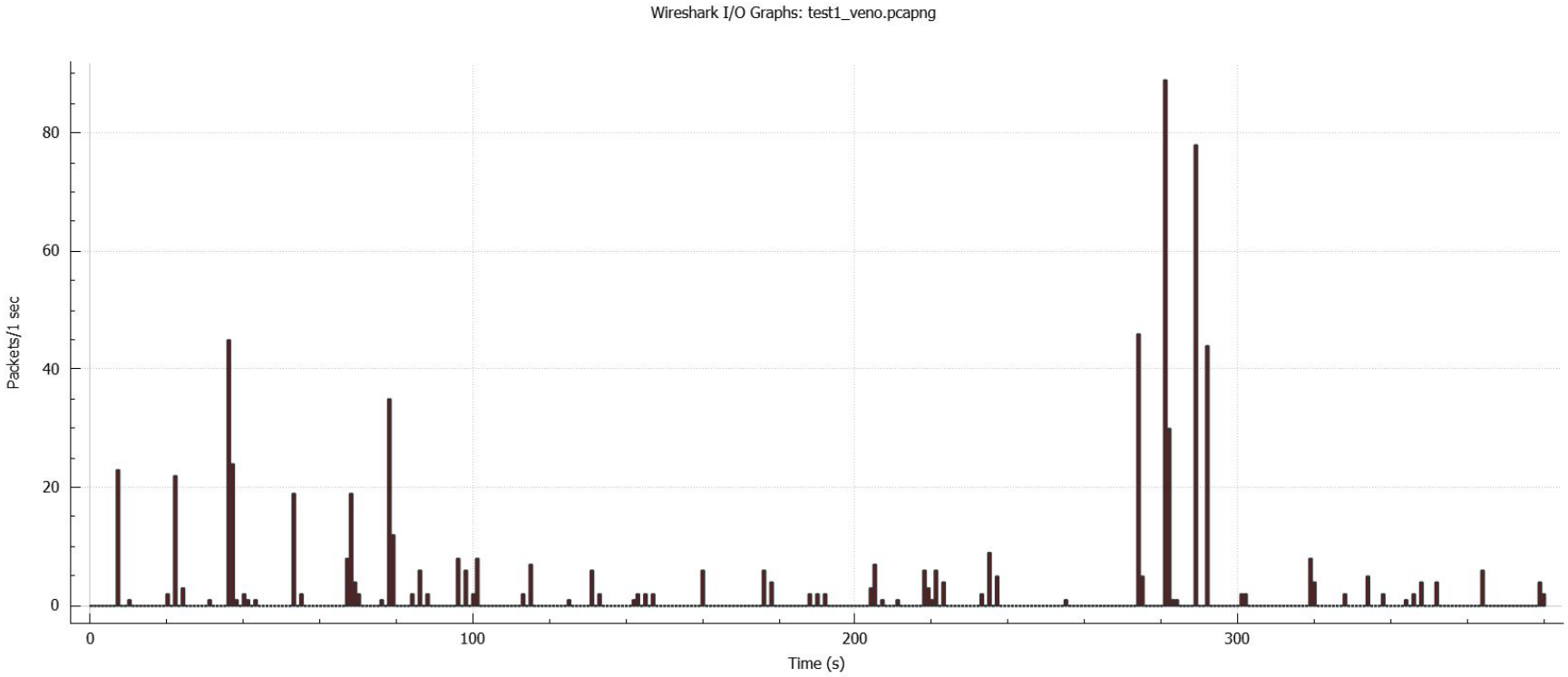

Wireshark I/O Graphs: test3_vegas.pcapng
Packets/1 sec
Time (s)

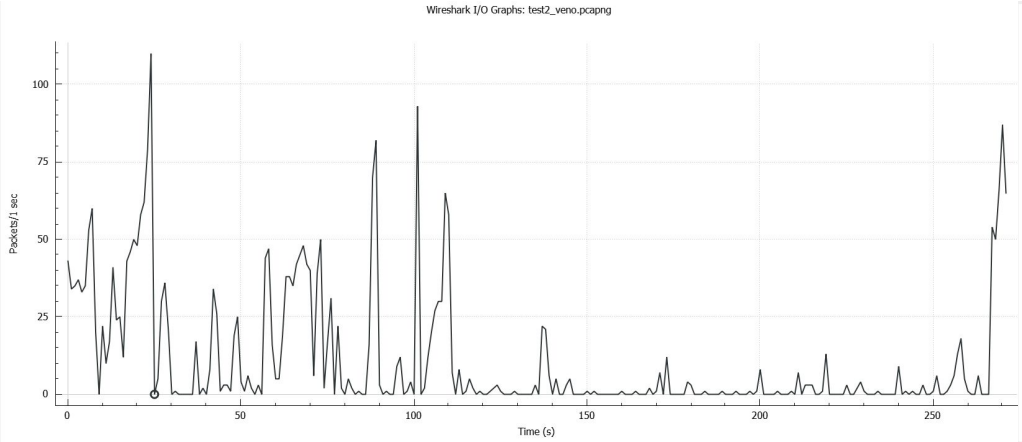

Wireshark I/O Graphs: test3_vegas.pcapng
Packets/1 sec
Time (s)

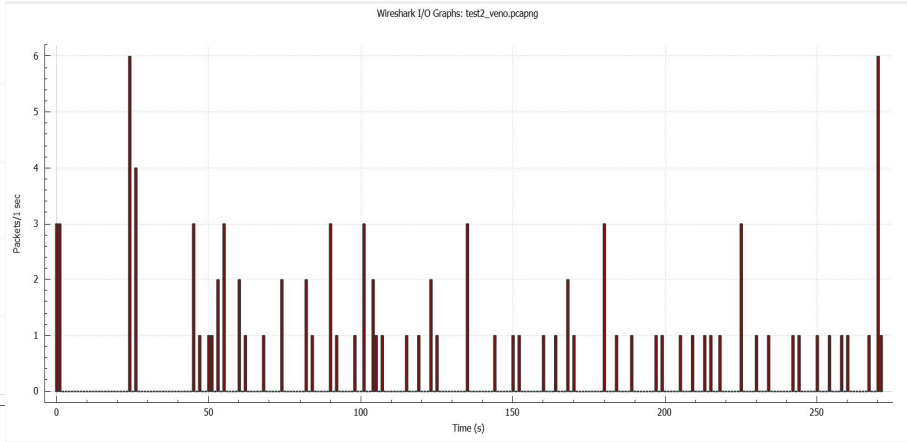
TCP Veno

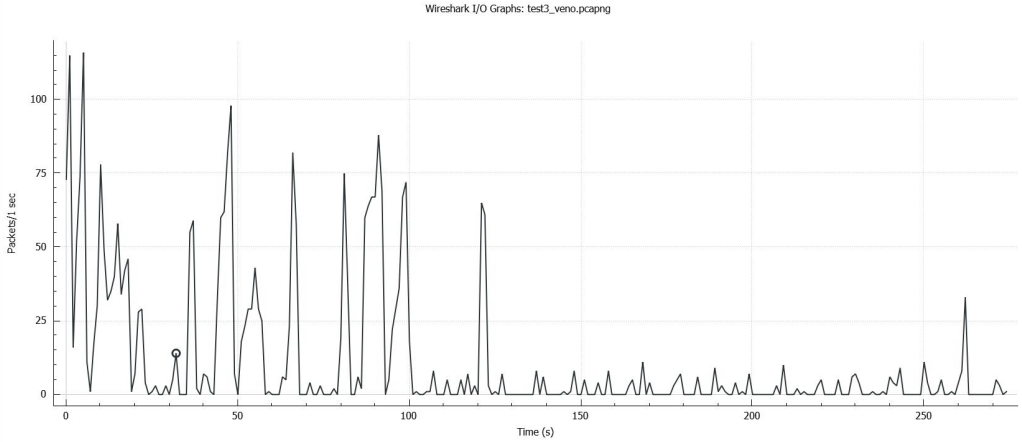

Wireshark I/O Graphs: test1_veno.pcapng
Packets/1 sec
400
300
200
100
0
0
100
200
300
Time (s)

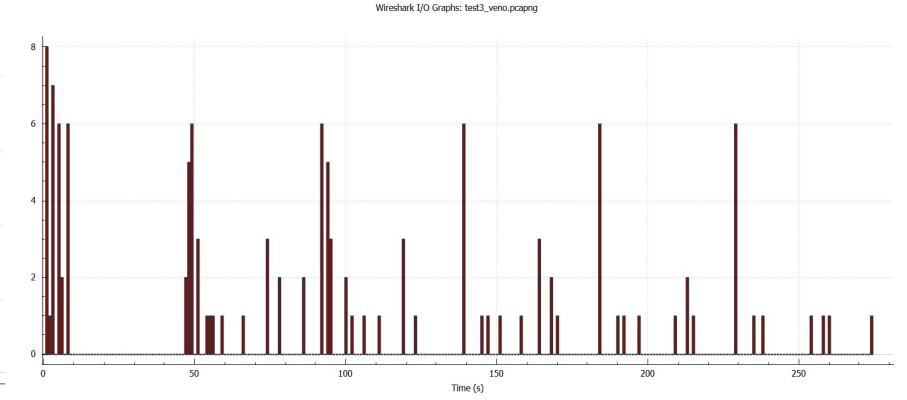
Errors


Wireshark I/O Graphs: test1_veno.pcapng
Packets/1 sec
0
20
40
60
80
0
100
200
300
Time (s)


Wireshark I/O Graphs: test2_veno.pcapng
Packets/1 sec
Time (s)


Wireshark I/O Graphs: test2_veno.pcapng
Packets/1 sec
Time (s)


Wireshark I/O Graphs: test3_veno.pcapng
Packets/1 sec
Time (s)


Wireshark I/O Graphs: test3_veno.pcapng
Time (s)

Comparison among TCP variants

TCP cubic showed highest errors in congested environment over Wi-Fi (2.4GHz)

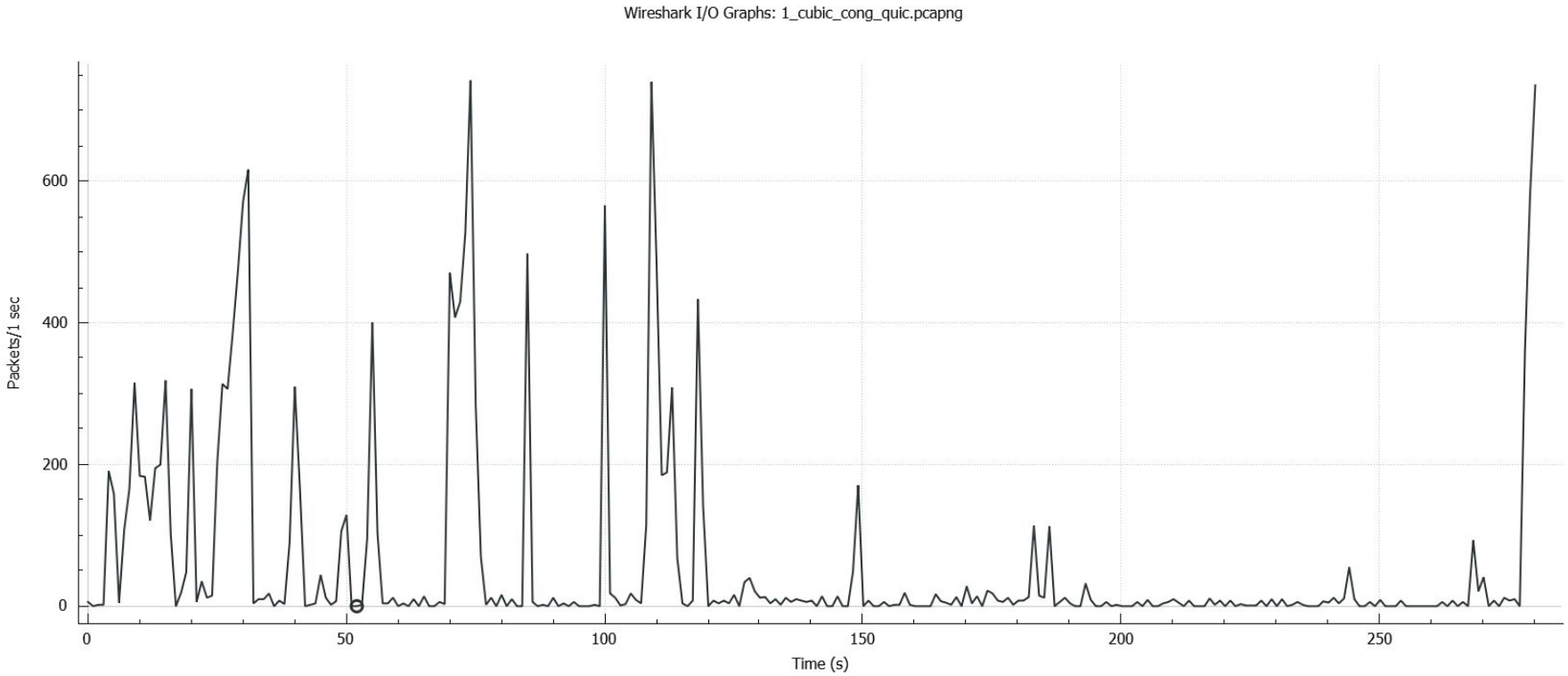
TCP Veno showed the best performance overall when compared on the basis of QoE (minimum re-buffering).

In case of All the TCP user page load time were on average in the range of 1-2 mins before any video even started playing and re-buffering was observed in all cases ,but minimum in TCP veno.

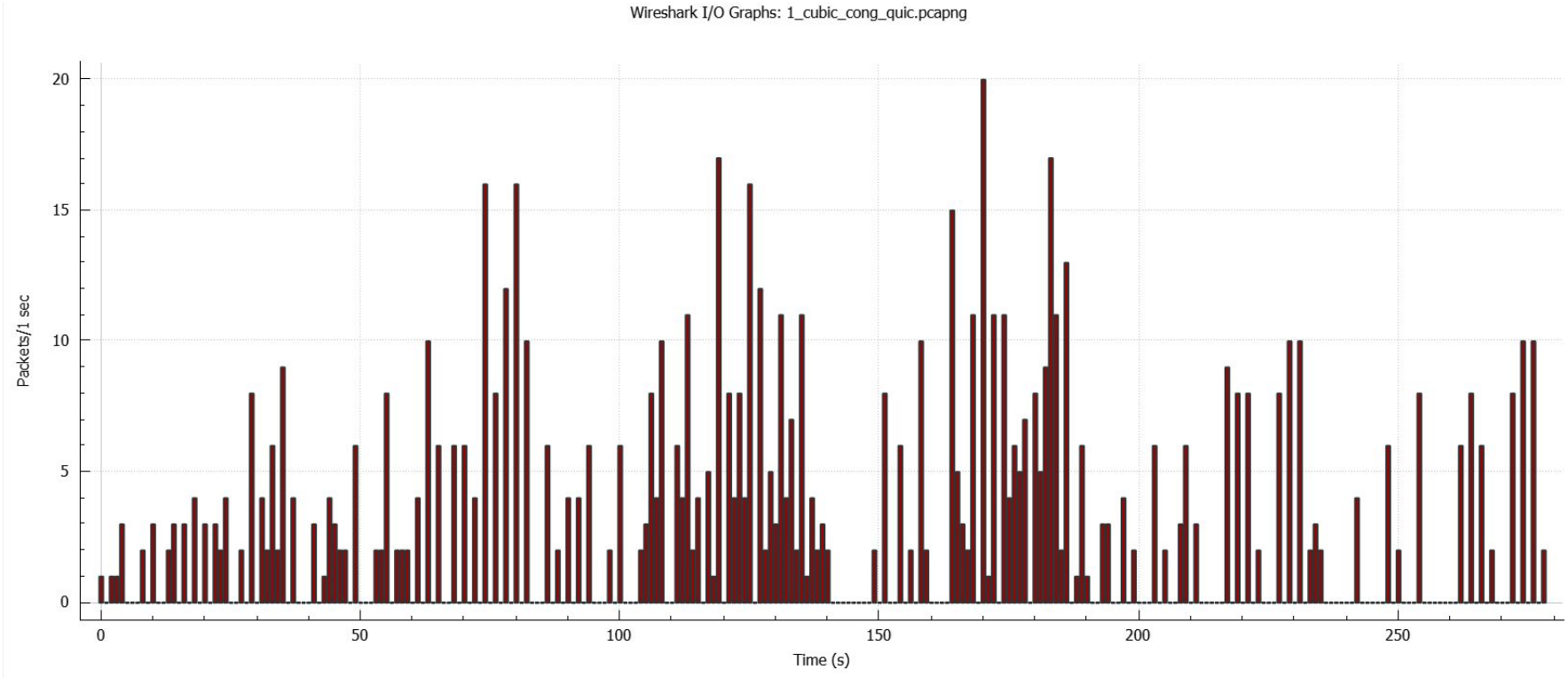
“TCP Veno has generally been presented as an option to address TCP’s lossy-link problem, rather than the high-bandwidth problem per se”.[1]

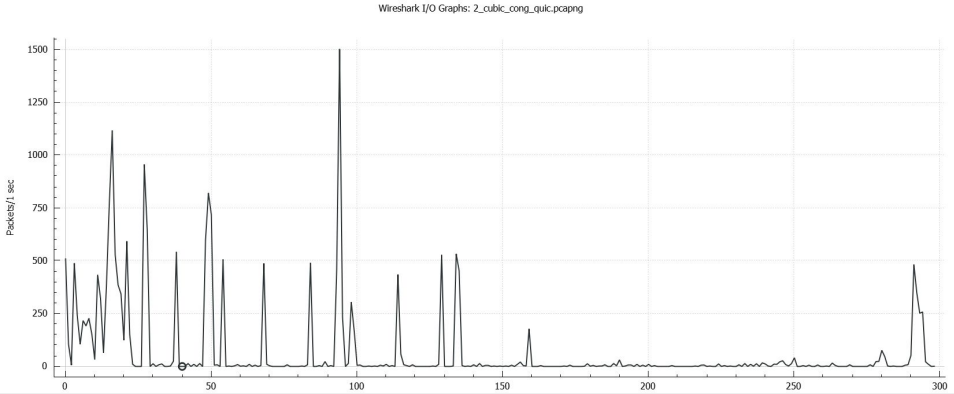
reference[1]:http://intronetworks.cs.luc.edu/1/html/newtcps.html

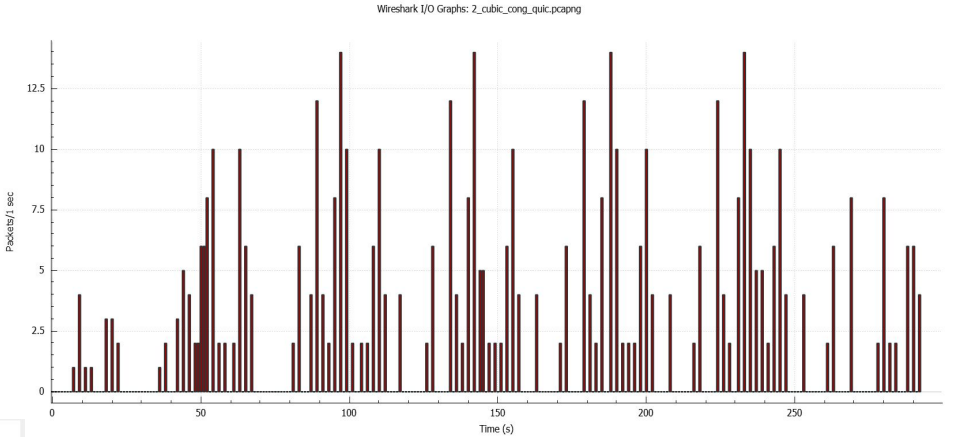
QUIC on congested Networks

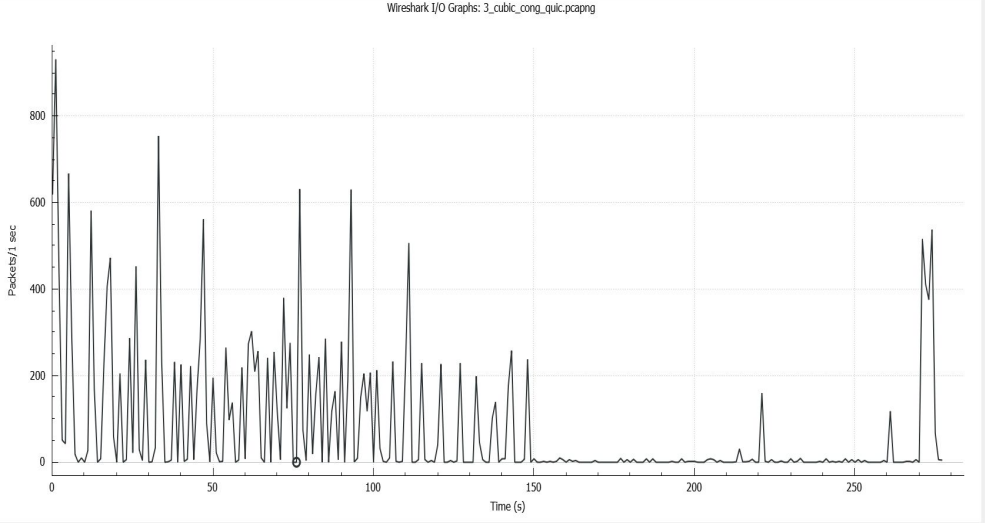


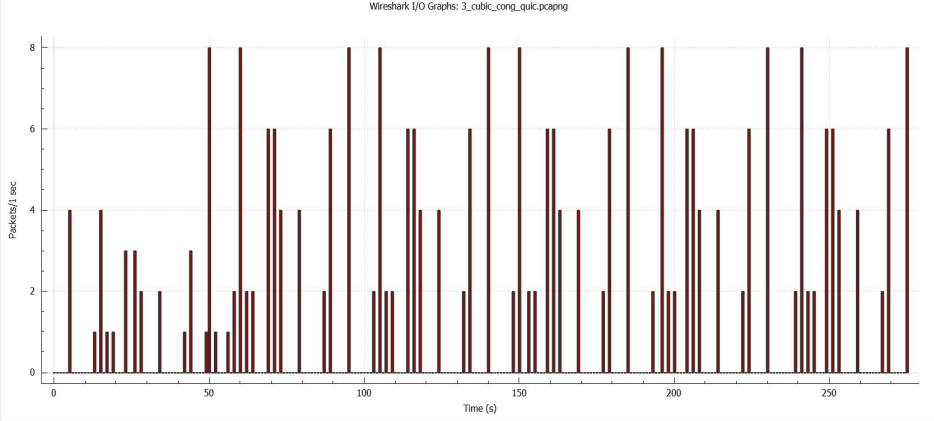
QUIC errors


Wireshark I/O Graphs: 1_cubic_cong_quic.pcapng
Packets/1 sec
20
15
10
5
0
0
50
100
150
200
250
Time (s)


Wireshark I/O Graphs: 2_cubic_cong_quic.pcapng
Packets/1 sec
0
250
500
750
1000
1250
1500
0
50
100
150
200
250
300


Wireshark I/O Graphs: 2_cubic_cong_quic.pcapng
Packets/1 sec
0
2.5
5
7.5
10
12.5
0
50
100
150
200
250
Time (s)


Wireshark I/O Graphs: 3_cubic_cong_quic.pcapng
Packets/1 sec
0
200
400
600
800
0
50
100
150
200
250
Time (s)


Wireshark I/O Graphs: 3_cubic_cong_quic.pcapng
Packets/1 sec
0
2
4
6
8
0
50
100
150
200
250
Time (s)

QUIC comparison vs TCP

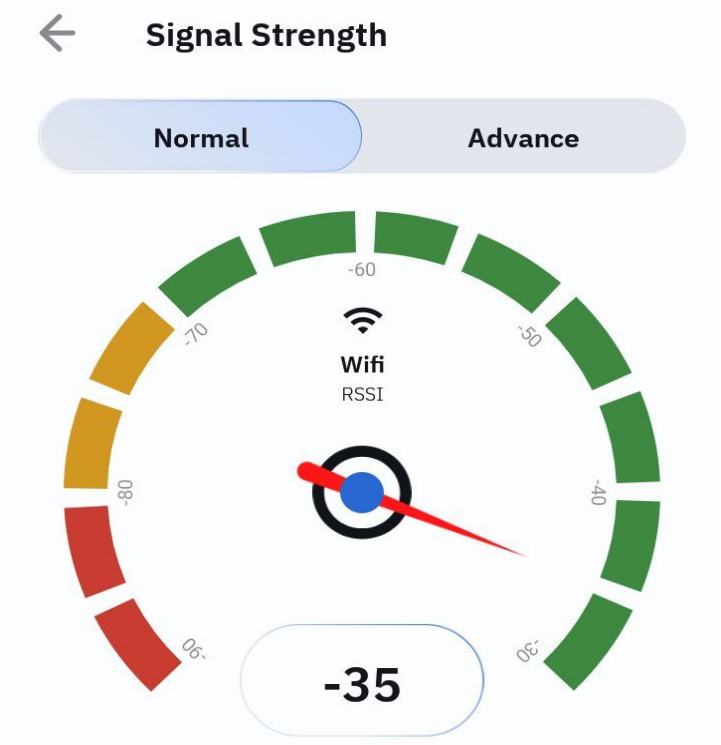
- User page load time was better in comparison to TCP by 40-50secs.
- QUIC was able to perform better in congested environment and able to playback the video without any re-buffering required during the playback.
- Max error packets in quic : 12.5 packets/sec
- Max error packets in TCP: 100 packets/sec

- In congested environment QUIC performed better than all the TCP variants based on the re-buffering time.

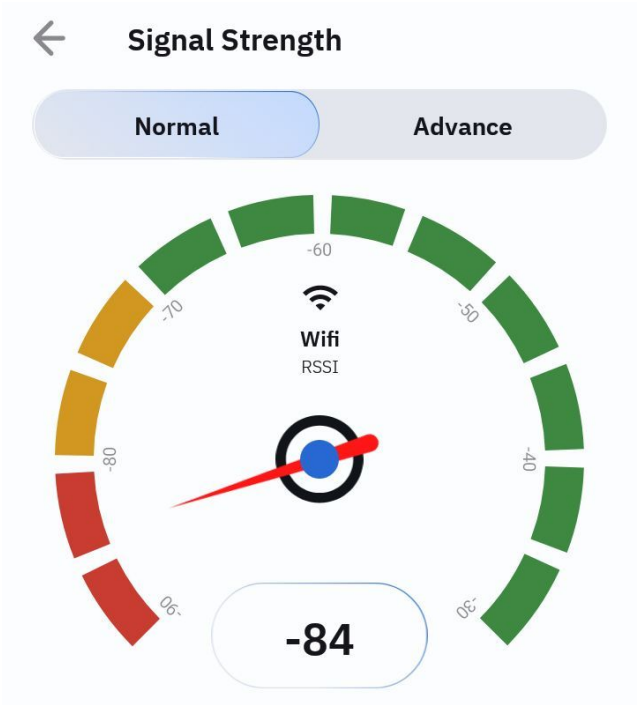
TCP variants performance based on signal strength

Network Conditions

Strong signal (dBm)



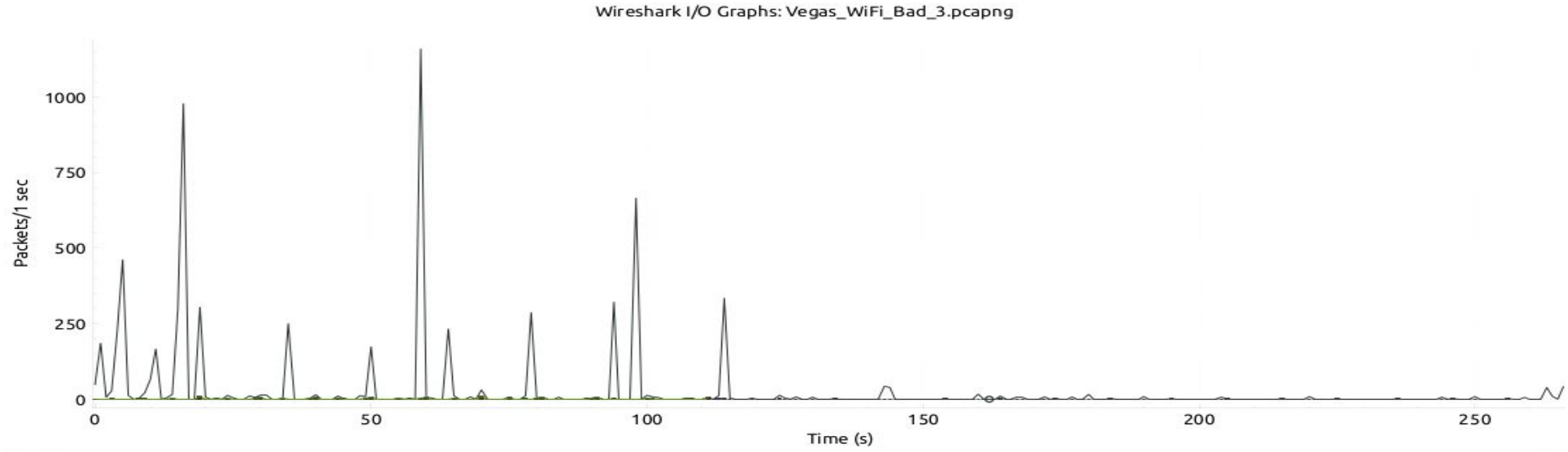
Weak signal(dBm)

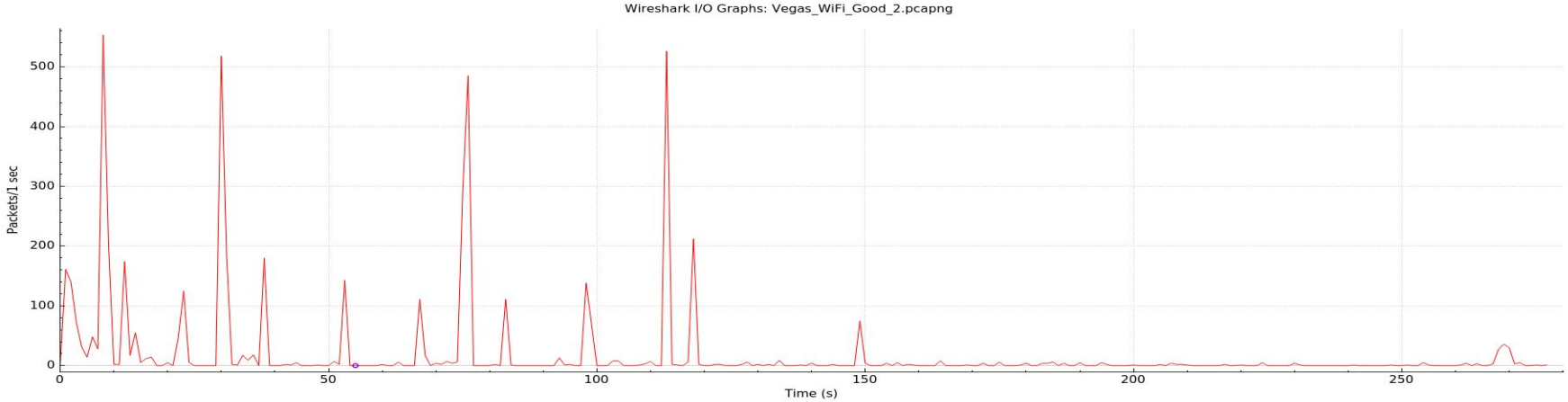


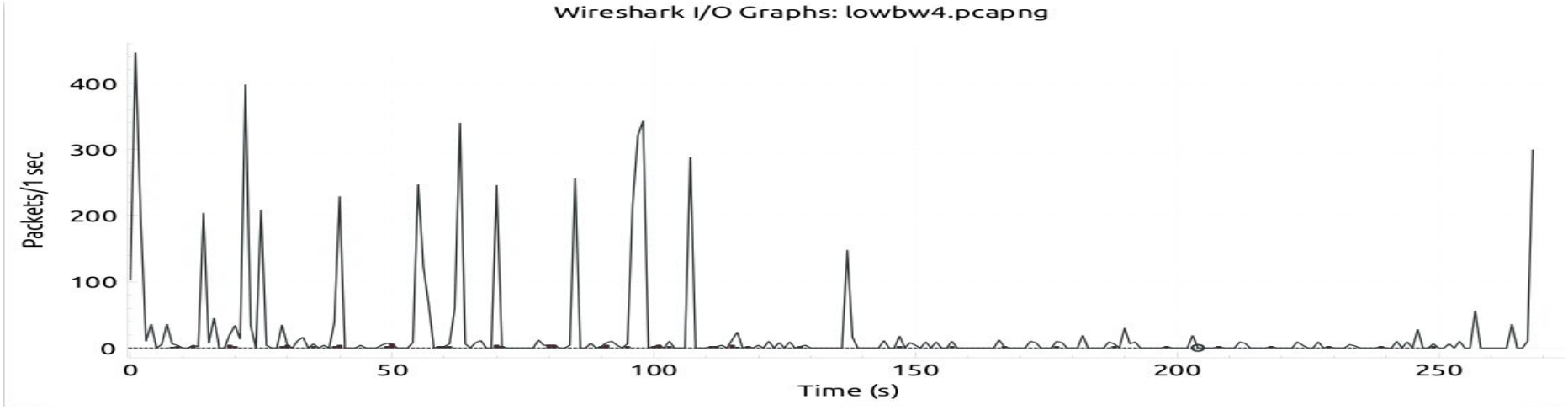
Median metrics

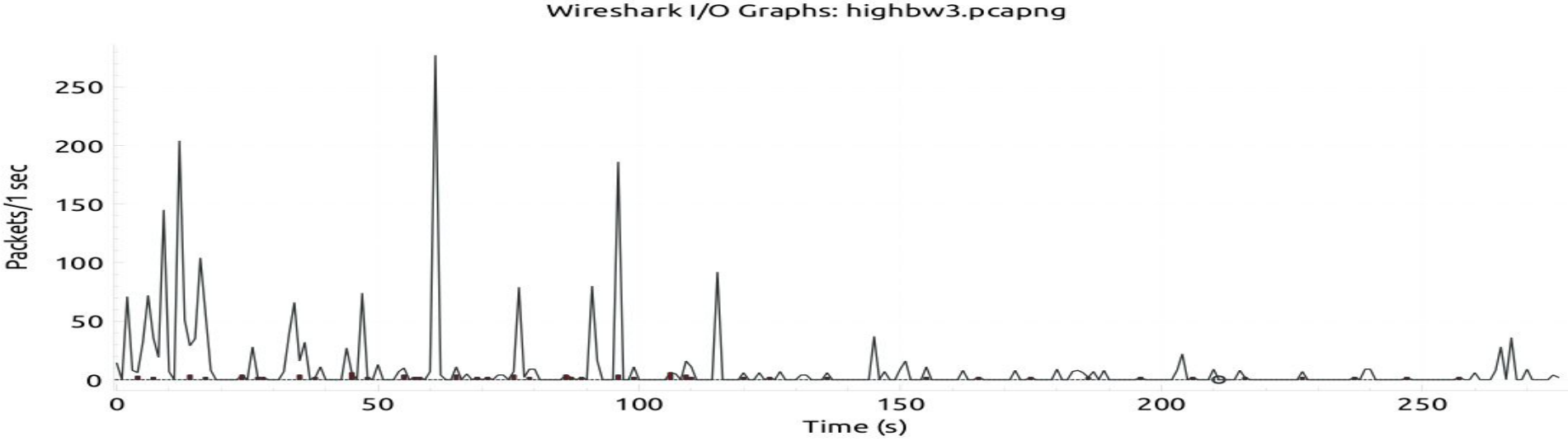
Variant	Throughput (bytes/sec)	Peaks	No of duplicate acks
Weak Signal			
Veno	32k	11	88
Cubic	33k	12	112
Vegas	33k	13	94
Strong Signal			
Veno	32k	12	108
Cubic	33k	13	72
Vegas	35k	15	90

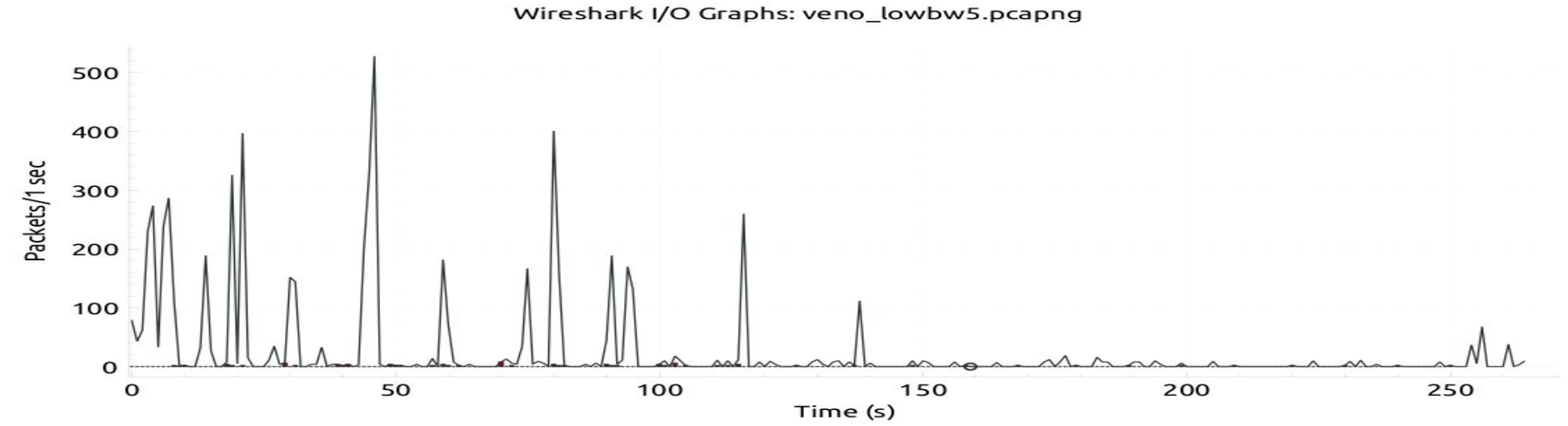
Instantaneous Throughput

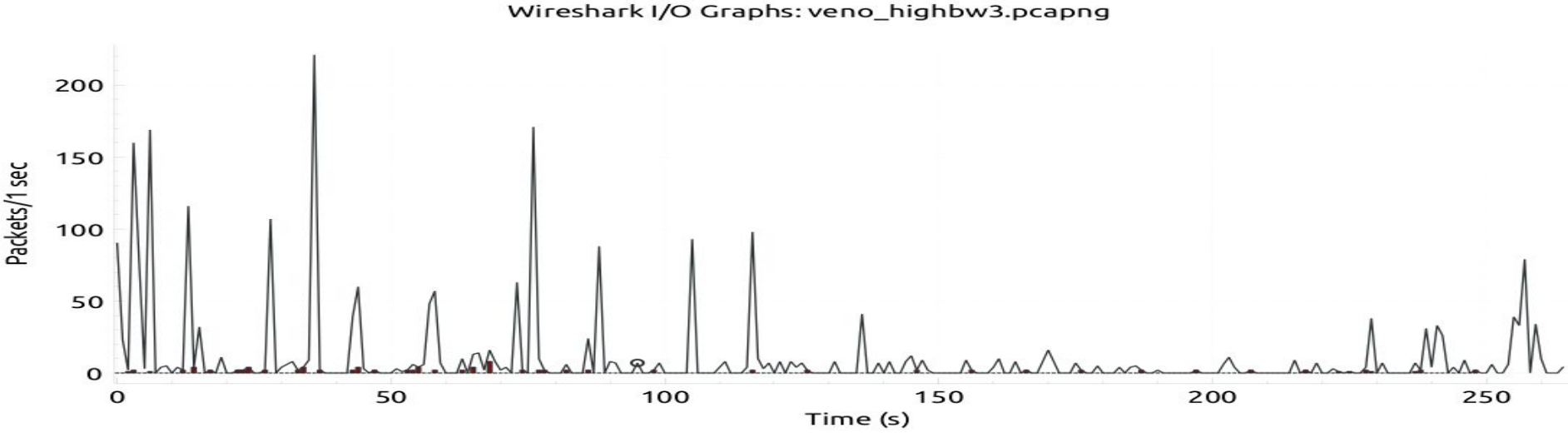

Wireshark I/O Graphs: Vegas_WiFi_Bad_3.pcapng
Packets/1 sec
1000
750
500
250
0
0
50
100
150
200
250
Time (s)


Wireshark I/O Graphs: Vegas_WiFi_Good_2.pcapng
Packets/1 sec
500
400
300
200
100
0
0
50
100
150
200
250
Time (s)

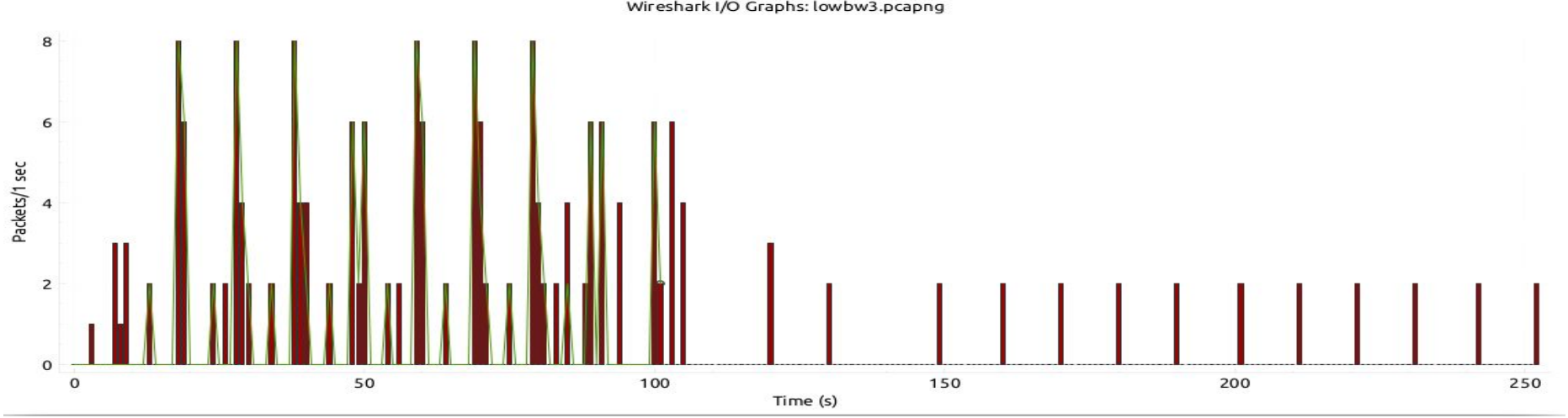

Wireshark I/O Graphs: lowbw4.pcapng
Packets/1 sec
400
300
200
100
0
0
50
100
150
200
250
Time (s)


Wireshark I/O Graphs: highbw3.pcapng
Packets/1 sec
250
200
150
100
50
0
0
50
100
150
200
250
Time (s)

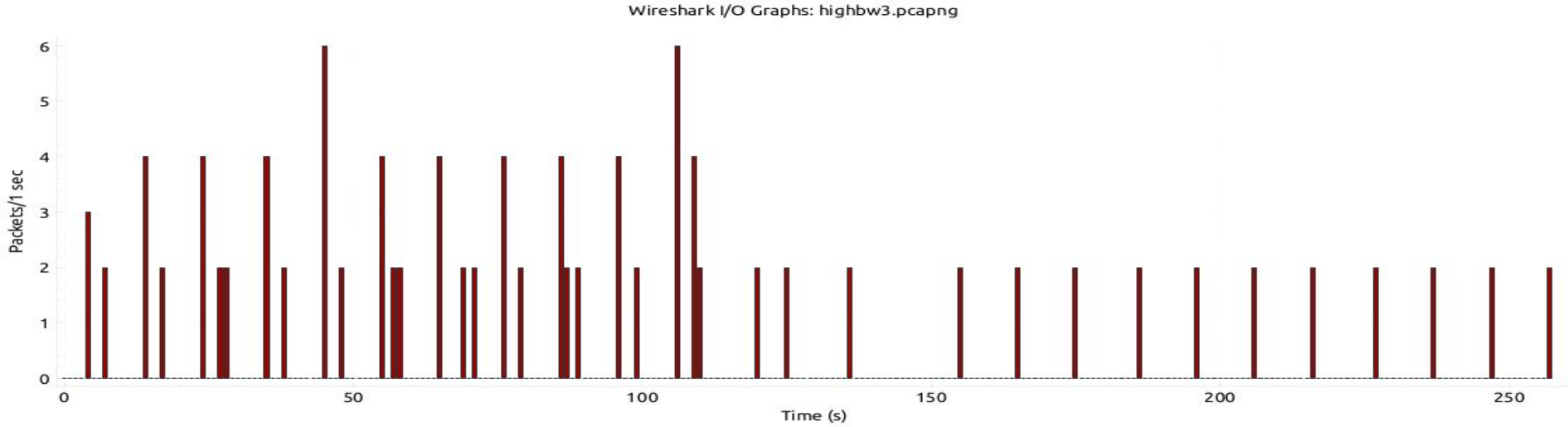

Wireshark I/O Graphs: veno_lowbw5.pcapng
Packets/1 sec
500
400
300
200
100
0
0
50
100
150
200
250
Time (s)

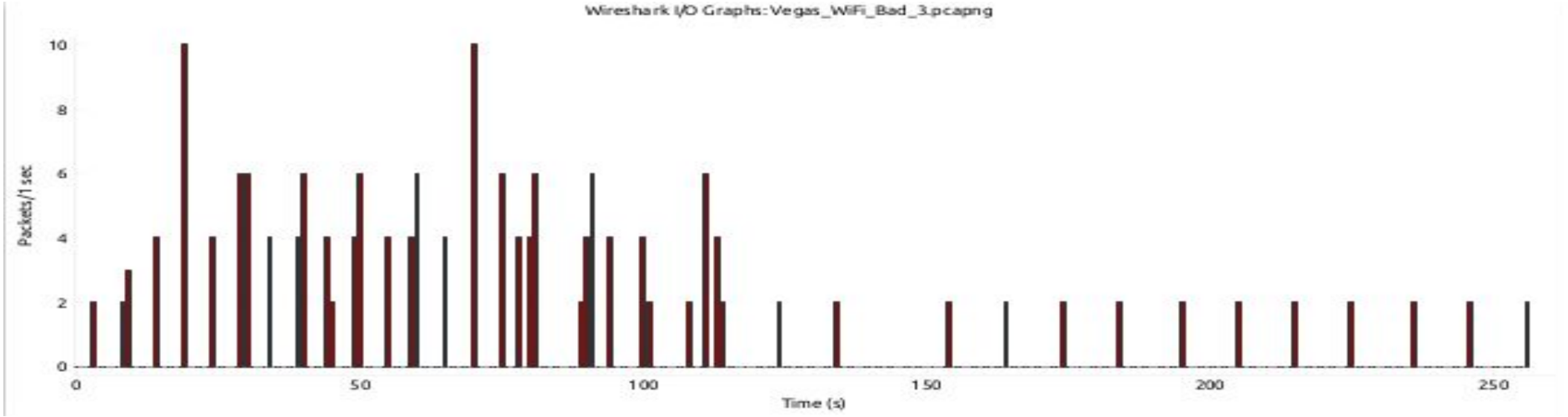

Wireshark I/O Graphs: veno_highbw3.pcapng
Packets/1 sec
200
150
100
50
0
0
50
100
150
200
250
Time (s)

Packet errors

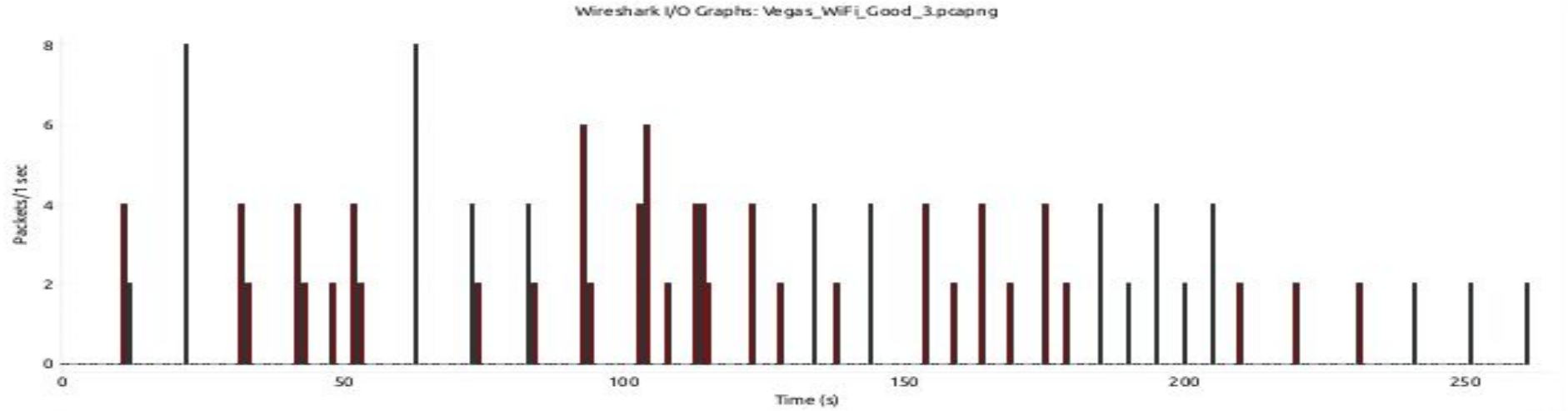

Wireshark I/O Graphs: lowbw3.pcapng
Packets/1 sec
Time (s)
0
2
4
6
8
0
50
100
150
200
250

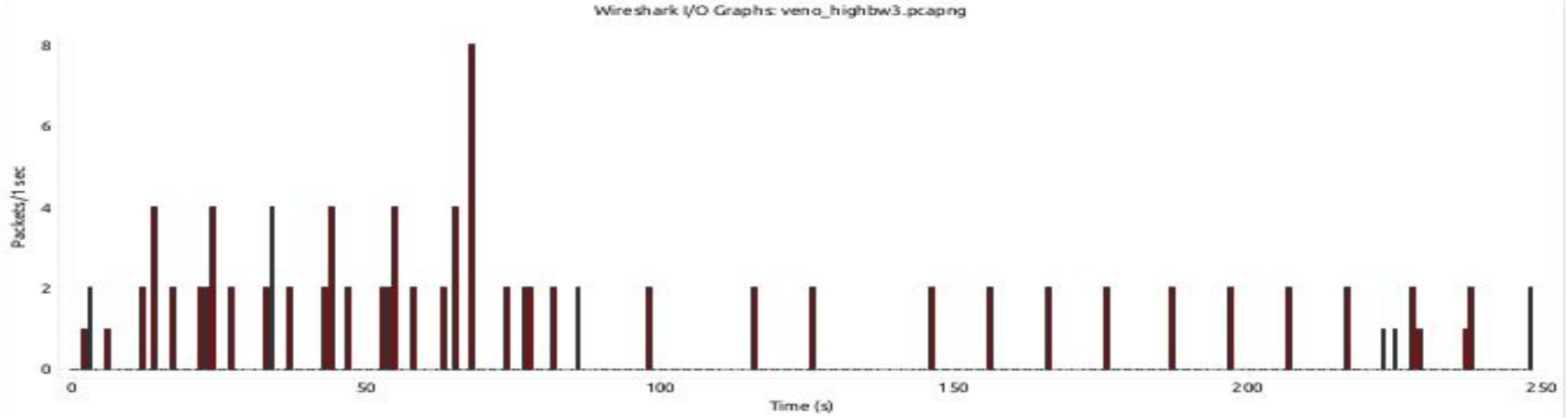
Cubic


Wireshark I/O Graphs: highbw3.pcapng
Packets/1 sec
Time (s)
0
1
2
3
4
5
6
0
50
100
150
200
250

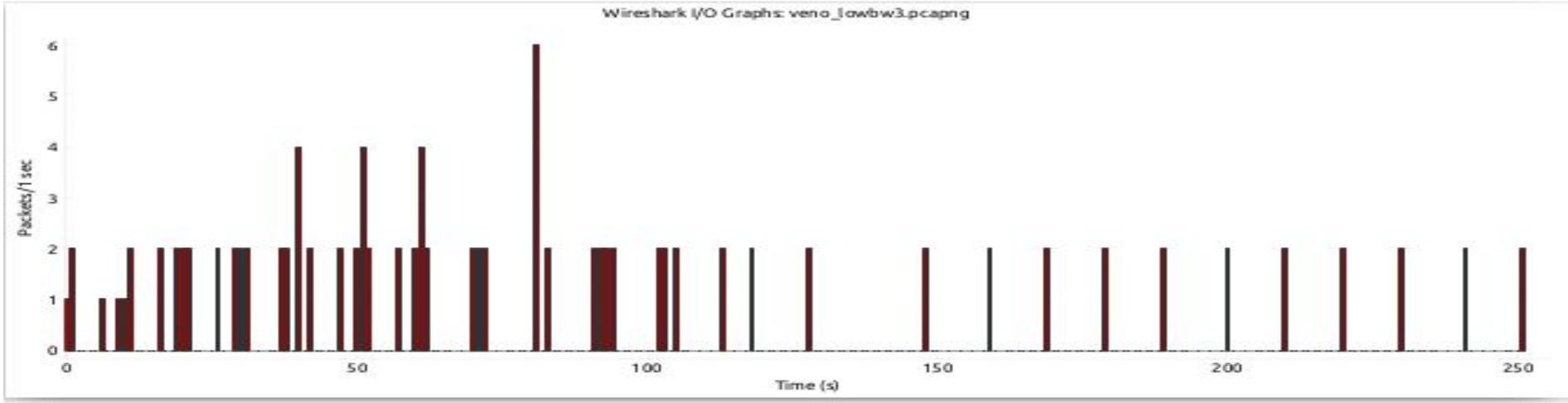

Wireshark I/O Graphs: Vegas_WiFi_Bad_3.pcapng
Packets/1 sec
Time (s)
0
2
4
6
8
10
0
50
100
150
200
250

Vegas


Wireshark I/O Graphs: Vegas_WiFi_Good_3.pcapng
Packets/1 sec
Time (s)
0
2
4
6
8
0
50
100
150
200
250


Wireshark I/O Graphs: veno_highbw3.pcapng
Packets/1 sec
Time (s)
0
2
4
6
8
0
50
100
150
200
250

Veno


Wireshark I/O Graphs: veno_lowbw3.pcapng
Packets/1 sec
Time (s)
0
1
2
3
4
5
6
0
50
100
150
200
250

Thank You