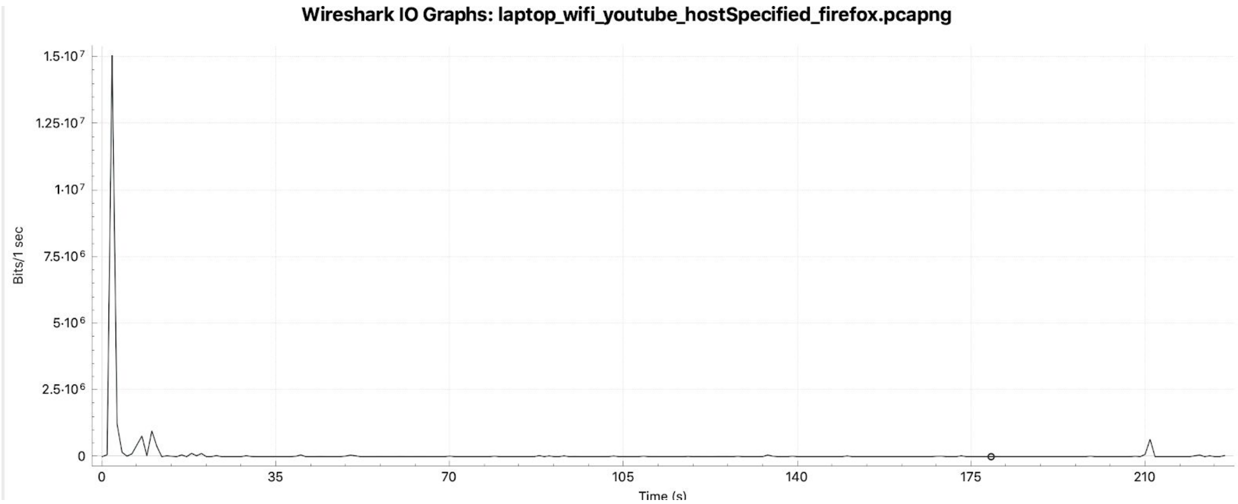
Part B Explanation:

A. A laptop that connects to a WiFi network.

a. Youtube:



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then tapers out later. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

No.	Time	Source	Destination	Protocol	Length	Sequence number	Acknowledgment n	Source Port	Info
57	2.150437	172.217.18.14	192.168.5.10	TLSv1.2	1474	861412189	2041842150	443	Application Data
58	2.150456	172.217.18.14	192.168.5.10	TLSv1.2	1474	861413597	2041842150	443	Application Data
59	2.150556	172.217.18.14	192.168.5.10	TLSv1.2	1474	861415005	2041842150	443	Application Data
60	2.150742	172.217.18.14	192.168.5.10	TLSv1.2	1474	861416413	2041842150	443	Application Data
61	2.150744	172.217.18.14	192.168.5.10	TLSv1.2	1474	861417821	2041842150	443	Application Data
64	2.151012	172.217.18.14	192.168.5.10	TLSv1.2	1474	861419229	2041842150	443	Application Data [TCP segment of a reassembled
65	2.151015	172.217.18.14	192.168.5.10	TLSv1.2	1474	861420637	2041842150	443	Application Data [TCP segment of a reassembled
68	2.151175	172.217.18.14	192.168.5.10	TLSv1.2	1474	861422045	2041842150	443	Application Data [TCP segment of a reassembled
69	2.151319	172.217.18.14	192.168.5.10	TLSv1.2	1474	861423453	2041842150	443	Application Data [TCP segment of a reassembled
71	2.151476	172.217.18.14	192.168.5.10	TLSv1.2	1474	861424861	2041842150	443	Application Data [TCP segment of a reassembled
72	2.151633	172.217.18.14	192.168.5.10	TLSv1.2	1474	861426269	2041842150	443	Application Data [TCP segment of a reassembled
74	2.151796	172.217.18.14	192.168.5.10	TLSv1.2	1183	861427677	2041842150	443	Application Data
76	2.151915	172.217.18.14	192.168.5.10	TLSv1.2	1474	861428794	2041842150	443	Application Data
77	2.152070	172.217.18.14	192.168.5.10	TLSv1.2	1474	861430202	2041842150	443	Application Data
79	2.152229	172.217.18.14	192.168.5.10	TLSv1.2	402	861431610	2041842150	443	Application Data
84	2.153992	172.217.18.14	192.168.5.10	TLSv1.2	744	861431946	2041842150	443	Application Data
85	2.153997	172.217.18.14	192.168.5.10	TLSv1.2	445	861432624	2041842150	443	Application Data

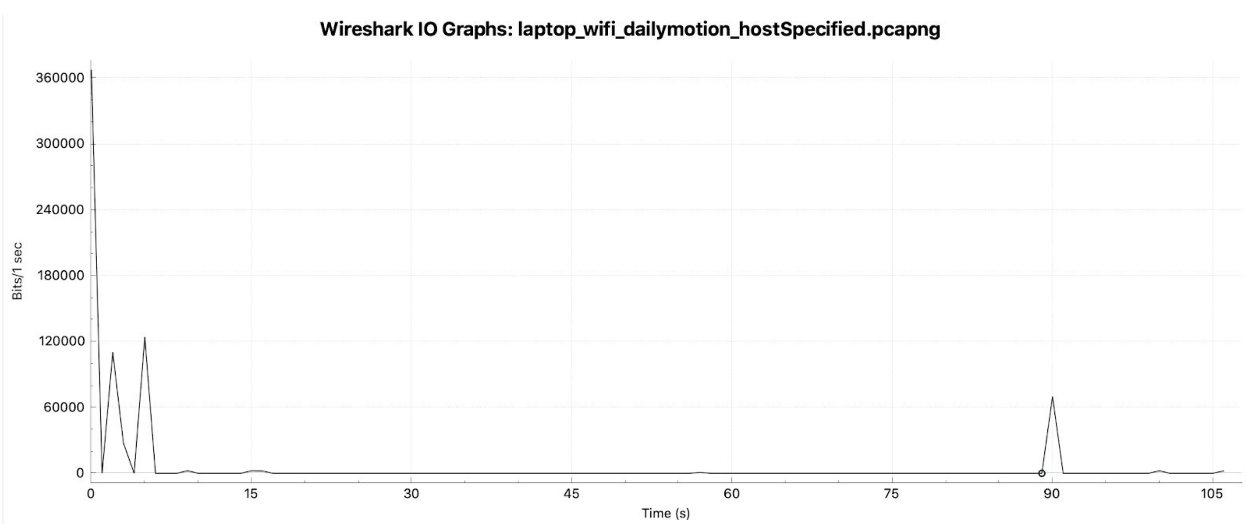
No.	Time	Source	Destination	Protocol	Length	Sequence number	Acknowledgment n	Source Port	Info
2980	50.253791	172.217.21.238	192.168.5.10	TCP	66	314926091	2380191954	443	443 → 60868 [ACK] Seq=314926091 Ack=2380191954
2981	50.254406	172.217.21.238	192.168.5.10	TCP	66	314926091	2380192279	443	443 → 60868 [ACK] Seq=314926091 Ack=2380192279
2982	50.254461	192.168.5.10	172.217.21.238	TLSv1.3	120	2380192279	314926091	60868	Application Data
2983	50.311377	172.217.21.238	192.168.5.10	TCP	66	314926091	2380192333	443	443 → 60868 [ACK] Seq=314926091 Ack=2380192333
2984	51.029889	172.217.21.238	192.168.5.10	TLSv1.3	1474	314926091	2380192333	443	Application Data
2985	51.132138	192.168.5.10	172.217.21.238	TCP	66	2380192333	314927499	60868	60868 → 443 [ACK] Seq=2380192333 Ack=314927499
2986	51.193853	172.217.21.238	192.168.5.10	TCP	549	314927499	2380192333	443	443 → 60868 [PSH, ACK] Seq=314927499 Ack=23801
2987	51.193859	172.217.21.238	192.168.5.10	TLSv1.3	144	314927982	2380192333	443	Application Data
2988	51.193860	172.217.21.238	192.168.5.10	TLSv1.3	93	314928060	2380192333	443	Application Data
2989	51.193861	172.217.21.238	192.168.5.10	TCP	93	314928060	2380192333	443	[TCP Retransmission] 443 → 60868 [PSH, ACK] Se
2990	51.193969	192.168.5.10	172.217.21.238	TCP	66	2380192333	314927982	60868	60868 → 443 [ACK] Seq=2380192333 Ack=314927982
2991	51.193970	192.168.5.10	172.217.21.238	TCP	66	2380192333	314928060	60868	60868 → 443 [ACK] Seq=2380192333 Ack=314928060
2992	51.193971	192.168.5.10	172.217.21.238	TCP	66	2380192333	314928087	60868	60868 → 443 [ACK] Seq=2380192333 Ack=314928087
2993	51.194051	192.168.5.10	172.217.21.238	TCP	78	2380192333	314928087	60868	[TCP Dup ACK 2992#1] 60868 → 443 [ACK] Seq=238
2994	51.391945	172.217.21.238	192.168.5.10	TLSv1.3	1474	314926091	2380192333	443	[TCP Spurious Retransmission] , Application Da
2995	51.392067	192.168.5.10	172.217.21.238	TCP	78	2380192333	314928087	60868	[TCP Window Update] 60868 → 443 [ACK] Seq=2380
2996	55.934594	192.168.5.10	172.217.22.14	TCP	54	2055439379	1290610808	60770	[TCP Keep-Alive] 60770 → 443 [ACK] Seq=2055439
2997	56.018294	172.217.22.14	192.168.5.10	TCP	66	1290610808	2055439380	443	[TCP Keep-Alive ACK] 443 → 60770 [ACK] Seq=129
2998	56.235920	192.168.5.10	172.217.21.206	TCP	54	2207875279	838374073	60787	[TCP Keep-Alive] 60787 → 443 [ACK] Seq=2207875
2999	56.755008	172.217.21.206	192.168.5.10	TCP	66	838374073	2207875280	443	[TCP Keep-Alive ACK] 443 → 60787 [ACK] Seq=838
3000	57.761036	192.168.5.10	172.217.18.14	TCP	54	69442447	2270395980	60710	[TCP Keep-Alive] 60710 → 443 [ACK] Seq=6944244
3001	58.088508	172.217.18.14	192.168.5.10	TCP	66	2270395980	69442448	443	[TCP Keep-Alive ACK] 443 → 60710 [ACK] Seq=227
3002	60.116910	172.217.21.238	192.168.5.10	TLSv1.2	161	747601112	1063215930	443	Application Data
3003	60.117007	192.168.5.10	172.217.21.238	TCP	66	1063215930	747601207	59953	59953 → 443 [ACK] Seq=1063215930 Ack=747601207
3004	64.723977	192.168.5.10	172.217.18.14	TCP	54	803949172	927992891	60718	[TCP Keep-Alive] 60718 → 443 [ACK] Seq=8039491
3005	64.749065	172.217.18.14	192.168.5.10	TCP	66	927992891	803949173	443	[TCP Keep-Alive ACK] 443 → 60718 [ACK] Seq=927
3006	70.246791	192.168.5.10	172.217.18.14	TCP	1474	3534561412	2198748234	60722	60722 → 443 [ACK] Seq=3534561412 Ack=219874823
3007	70.246792	192.168.5.10	172.217.18.14	TLSv1.2	563	3534562820	2198748234	60722	Application Data
3008	70.272285	172.217.18.14	192.168.5.10	TCP	66	2198748234	3534563317	443	443 → 60722 [ACK] Seq=2198748234 Ack=353456331
3009	70.288951	172.217.18.14	192.168.5.10	TLSv1.2	583	2198748234	3534563317	443	Application Data
3010	70.289013	192.168.5.10	172.217.18.14	TCP	66	3534563317	2198748751	60722	60722 → 443 [ACK] Seq=3534563317 Ack=219874875

Compared to the first image, where the majority of TLS packets had almost a steady length of 1474 bytes, the latter image has TLS packets with a severely lowerer number of bytes, 144, 93, 161, and 583, when only looking at data sent from the server to the client. We can assume that link quality has gone down due to the need for TCP keep alive packets to be sent. Therefore, we can say that when the link quality is bad, then the variable bit rate goes down, which eventually makes the streaming bit rate go down.

Over Time:

The trace starts off initially with a strong link quality and later becomes weak (indicated by error packets). Therefore the streaming bitrate is larger in the beginning, and later becomes smaller.

b. Daily Motion



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then has smaller peaks, indicating a smaller bit rate. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

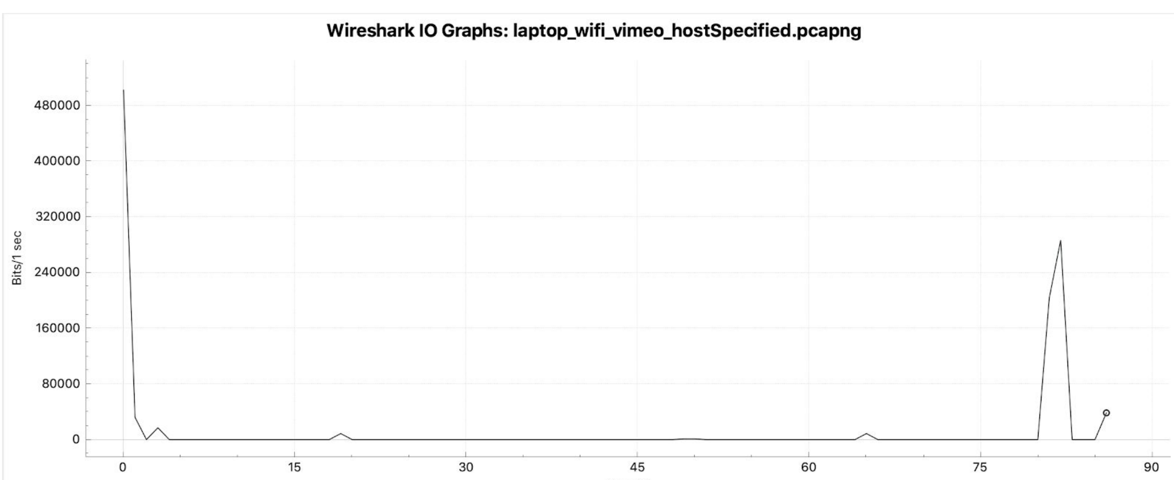
No.	Time	Source	Destination	Protocol	Length	Sequence number	Acknowledgment num	Source Port
96	5.238035	192.168.5.10	195.8.215.137	TCP	1474	3999924471	4213870821	64085
97	5.238035	192.168.5.10	195.8.215.137	TLSv1.2	318	3999925879	4213870821	64085
98	5.238224	192.168.5.10	195.8.215.137	TLSv1.2	97	3678596861	835424487	64087
99	5.264697	195.8.215.137	192.168.5.10	TCP	74	591944116	504729040	443
100	5.264759	192.168.5.10	195.8.215.137	TCP	66	504729040	591944117	64177
101	5.267641	195.8.215.137	192.168.5.10	TCP	54	835424487	0	443
102	5.268362	195.8.215.137	192.168.5.10	TCP	54	835424487	0	443
103	5.271944	195.8.215.137	192.168.5.10	TCP	66	4213870821	3999924471	443
104	5.272608	195.8.215.137	192.168.5.10	TCP	66	4213870821	3999926131	443
105	5.273116	195.8.215.137	192.168.5.10	TCP	54	835424487	0	443
106	5.273943	195.8.215.137	192.168.5.10	TLSv1.2	463	4213870821	3999926131	443
107	5.274015	192.168.5.10	195.8.215.137	TCP	66	3999926131	4213871218	64085
108	5.281070	192.168.5.10	195.8.215.137	TLSv1.2	583	504729040	591944117	64177
109	5.293393	192.168.5.10	195.8.215.137	TCP	1474	3999926131	4213871218	64085
110	5.293395	192.168.5.10	195.8.215.137	TLSv1.2	1123	3999927539	4213871218	64085
111	5.314292	195.8.215.137	192.168.5.10	TCP	66	591944117	504729557	443
112	5.320298	195.8.215.137	192.168.5.10	TLSv1.2	1474	591944117	504729557	443
113	5.320304	195.8.215.137	192.168.5.10	TCP	1474	591945525	504729557	443
114	5.320412	195.8.215.137	192.168.5.10	TLSv1.2	658	591946933	504729557	443

In this image, packet 101 and 102 represent packets with rst flags set, indicating that there is poor link quality. From there, we see that the TLS packet directly below, sent from the server to the client with application data has only 463 bytes, compared to a "good" connection which would have 1474 bytes. We can infer from this, that as the link quality is weak, then the streaming bit rate goes down.

Over Time:

In this example, we see that the streaming bitrate starts off strong (large) for a brief period of time, but is later lowered to accommodate for the weak link quality. This is concurred with the fact that we observe more packets indicating some sort of transmission failure later.

c. Vimeo



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then has smaller peaks, indicating a smaller bit rate, then again another spike, indicating a larger bit rate. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

No.	Time	Source	Destination	Protocol	Length	Sequence numl	Acknowledgment	Source F	Info
95	1.934554	151.101.0.217	192.168.5.10	TLSv1.2	994	985603778	1382833211	443	Application Data, Application Data
96	1.934779	192.168.5.10	151.101.0.217	TCP	66	1382833211	985604706	62615	62615 → 443 [ACK] Seq=1382833211 Ack=9
97	1.977626	151.101.0.217	192.168.5.10	TLSv1.2	1109	892131748	294410456	443	Application Data, Application Data
98	1.977703	192.168.5.10	151.101.0.217	TCP	66	294410456	892132791	62619	62619 → 443 [ACK] Seq=294410456 Ack=89
99	3.145197	192.168.5.10	151.101.0.217	TLSv1.2	866	294410456	892132791	62619	Application Data
100	3.181491	151.101.0.217	192.168.5.10	TCP	66	892132791	294411256	443	443 → 62619 [ACK] Seq=892132791 Ack=29
101	3.305432	151.101.0.217	192.168.5.10	TLSv1.2	922	892132791	294411256	443	Application Data
102	3.305507	192.168.5.10	151.101.0.217	TCP	66	294411256	892133647	62619	62619 → 443 [ACK] Seq=294411256 Ack=89
103	3.306184	151.101.0.217	192.168.5.10	TLSv1.2	138	892133647	294411256	443	Application Data
104	3.306246	192.168.5.10	151.101.0.217	TCP	66	294411256	892133719	62619	62619 → 443 [ACK] Seq=294411256 Ack=89
105	19.012197	192.168.5.10	151.101.0.217	TCP	54	620867061	2207419003	62269	62269 → 443 [ACK] Seq=620867061 Ack=22
106	19.039056	151.101.0.217	192.168.5.10	TCP	66	2207419003	620867062	443	[TCP ACKed unseen segment] 443 → 62269
107	19.048438	192.168.5.10	151.101.128.217	TCP	54	3332550993	4105326702	62497	62497 → 443 [ACK] Seq=3332550993 Ack=4
108	19.105913	192.168.5.10	151.101.128.217	TCP	54	1377076132	74642393	62490	62490 → 443 [ACK] Seq=1377076132 Ack=7
109	19.115320	192.168.5.10	151.101.128.217	TCP	54	1132418173	811457637	62495	62495 → 443 [ACK] Seq=1132418173 Ack=8
110	19.115320	192.168.5.10	151.101.128.217	TCP	54	355668791	3399827742	62486	62486 → 443 [ACK] Seq=355668791 Ack=33
111	19.116586	192.168.5.10	151.101.0.217	TCP	54	3186830046	585152961	62180	62180 → 443 [ACK] Seq=3186830046 Ack=5
112	19.130608	151.101.128.217	192.168.5.10	TCP	66	4105326702	3332550994	443	[TCP ACKed unseen segment] 443 → 62497
113	19.149608	192.168.5.10	151.101.128.217	TCP	54	494193703	2153757569	62487	62487 → 443 [ACK] Seq=494193703 Ack=21
114	19.213025	151.101.128.217	192.168.5.10	TCP	66	74642393	1377076133	443	[TCP ACKed unseen segment] 443 → 62490
115	19.234263	151.101.128.217	192.168.5.10	TCP	66	3399827742	355668792	443	[TCP ACKed unseen segment] 443 → 62486
116	19.236860	151.101.128.217	192.168.5.10	TCP	66	811457637	1132418174	443	[TCP ACKed unseen segment] 443 → 62495
117	19.241881	151.101.0.217	192.168.5.10	TCP	66	585152961	3186830047	443	[TCP ACKed unseen segment] 443 → 62180
118	19.246261	151.101.128.217	192.168.5.10	TCP	66	2153757569	494193704	443	[TCP ACKed unseen segment] 443 → 62487
119	19.360970	192.168.5.10	151.101.128.217	TCP	54	3962034344	2487881658	62494	62494 → 443 [ACK] Seq=3962034344 Ack=2
120	19.360971	192.168.5.10	151.101.0.217	TCP	54	916750574	3478037265	62179	62179 → 443 [ACK] Seq=916750574 Ack=34
121	19.476274	151.101.128.217	192.168.5.10	TCP	66	2487881658	3962034345	443	[TCP ACKed unseen segment] 443 → 62494
122	19.483144	151.101.0.217	192.168.5.10	TCP	66	3478037265	916750575	443	[TCP ACKed unseen segment] 443 → 62179
123	49.178563	192.168.5.10	151.101.0.217	TCP	54	1382833210	985604706	62615	[TCP Keep-Alive] 62615 → 443 [ACK] Seq
124	49.200119	151.101.0.217	192.168.5.10	TCP	66	985604706	1382833211	443	[TCP Keep-Alive ACK] 443 → 62615 [ACK] Seq=9

As indicated by the TCP keep alive flags, the link quality is weak (if it had been strong, it would have been known that the connection was present). During the small period before these packets get ultimately sent, we see several TLSv1.2 packets get sent from the server (port 443) to the client (port 62619). A typical "good" connection will have these packets send 1474 bytes of data. However, none of these packets amounts to that much, all of them have different sizes (1109 bytes, 922 bytes, 138 bytes), and the number of bytes transmitted (from server to client) goes down as we get closer to the error packets. Therefore, it can be assumed that a worse link quality will cause the streaming bitrate to go down.

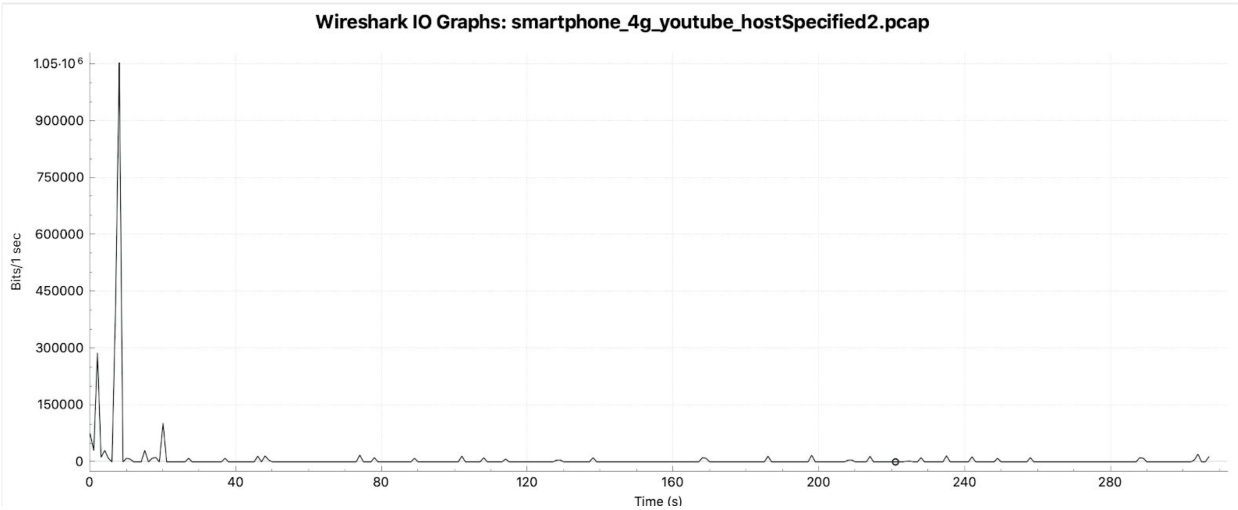
Over Time:

The trace starts off initially with a strong link quality and later becomes weak and once again strong. This is concurred with the fact that we observe more packets indicating some sort of transmission failure at lower streaming bit rate areas.

Therefore, the streaming bitrate is larger in the beginning, and somewhere in the middle becomes smaller, and then increases once again at the end.

B. <u>A smartphone that connects to a 4G cellular network.</u>

a. Youtube



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then tapers out later. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

No.	Time	Source	Destination	Protocol	Length	Sequence numl	Acknowledgment	Source F	Info
324	8.094006	172.217.19.110	10.153.113.219	TLSv1.2	137	4019278084	639770746	443	Application Data
325	8.094008	172.217.19.110	10.153.113.219	TLSv1.2	1454	4019278165	639770746	443	Application Data
326	8.094010	172.217.19.110	10.153.113.219	TLSv1.2	1216	4019279563	639770746	443	Application Data
327	8.094012	172.217.19.110	10.153.113.219	TLSv1.2	1454	4019280723	639770746	443	Application Data
328	8.094015	172.217.19.110	10.153.113.219	TLSv1.2	1454	4019282121	639770746	443	Application Data
329	8.094017	172.217.19.110	10.153.113.219	TLSv1.2	1454	4019283519	639770746	443	Application Data
330	8.094019	172.217.19.110	10.153.113.219	TLSv1.2	1454	4019284917	639770746	443	Application Data

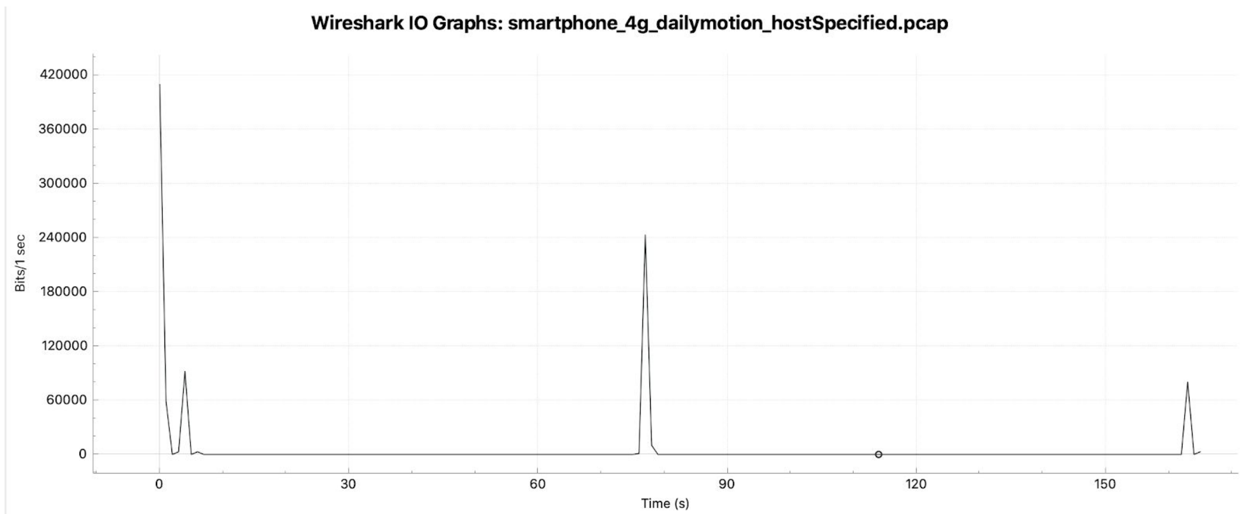
	Time	Source	Destination	Protocol	Length	Sequence numl	Acknowledgment	Source F	Info
682	214.242137	10.153.113.219	172.217.19.110	TCP	56	639795557	4019374479	57556	57556 → 443 [ACK] Seq=639795557 Ack=4019374
683	214.242468	10.153.113.219	172.217.19.110	TLSv1.2	95	639795557	4019374479	57556	Application Data
684	214.279507	172.217.19.110	10.153.113.219	TCP	56	4019374479	639795596	443	443 → 57556 [ACK] Seq=4019374479 Ack=639795
685	224.935496	10.153.113.219	172.217.19.110	TLSv1.2	95	2985115860	4005128727	57544	Application Data
686	224.937110	10.153.113.219	172.217.19.110	TLSv1.2	80	2985115899	4005128727	57544	Application Data
687	224.937708	10.153.113.219	172.217.19.110	TCP	56	2985115923	4005128727	57544	57544 → 443 [FIN, ACK] Seq=2985115923 Ack=4
688	225.134438	172.217.19.110	10.153.113.219	TCP	56	4005128727	2985115860	443	[TCP Dup ACK 211#1] 443 → 57544 [ACK] Seq=4
689	225.140434	172.217.19.110	10.153.113.219	TCP	56	4005128727	2985115899	443	443 → 57544 [ACK] Seq=4005128727 Ack=298511
690	225.140437	172.217.19.110	10.153.113.219	TCP	56	4005128727	2985115924	443	443 → 57544 [ACK] Seq=4005128727 Ack=298511
691	225.140440	172.217.19.110	10.153.113.219	TCP	56	4005128727	2985115924	443	443 → 57544 [FIN, ACK] Seq=4005128727 Ack=2
692	225.140859	10.153.113.219	172.217.19.110	TCP	56	2985115924	4005128728	57544	57544 → 443 [ACK] Seq=2985115924 Ack=400512
693	228.021575	10.153.113.219	172.217.19.110	TLSv1.3	766	1731785046	2138988142	57578	Application Data
694	228.068619	172.217.19.110	10.153.113.219	TCP	56	2138988142	1731785756	443	443 → 57578 [ACK] Seq=2138988142 Ack=173178
695	228.097622	172.217.19.110	10.153.113.219	TLSv1.3	125	2138988142	1731785756	443	Application Data
696	228.098072	10.153.113.219	172.217.19.110	TCP	56	1731785756	2138988211	57578	57578 → 443 [ACK] Seq=1731785756 Ack=213898
697	228.103692	172.217.19.110	10.153.113.219	TLSv1.3	87	2138988211	1731785756	443	Application Data
698	228.103695	172.217.19.110	10.153.113.219	TLSv1.3	95	2138988242	1731785756	443	Application Data
699	228.104017	10.153.113.219	172.217.19.110	TCP	56	1731785756	2138988281	57578	57578 → 443 [ACK] Seq=1731785756 Ack=213898
700	228.104242	10.153.113.219	172.217.19.110	TLSv1.3	95	1731785756	2138988281	57578	Application Data

In the latter image, we can see that a tcp duplicate ack has been sent. Doing such can indicate poor link quality. We see that the TLS packets following the tcp duplicate ack have a notably smaller number of bytes transferred (as compared to the first image which showed 1454 bytes transferred). Due to the great difference, it can be assumed that the video streaming bit rate went down when this link quality went down.

Over Time:

The streaming bitrate started off relatively low, then spiked up, then went low again when the link quality went down. This is concurred with the fact that we observe more packets indicating some sort of transmission failure at lower streaming bitrate areas.

b. Daily Motion



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then lowers, then increases, and lowers once again. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

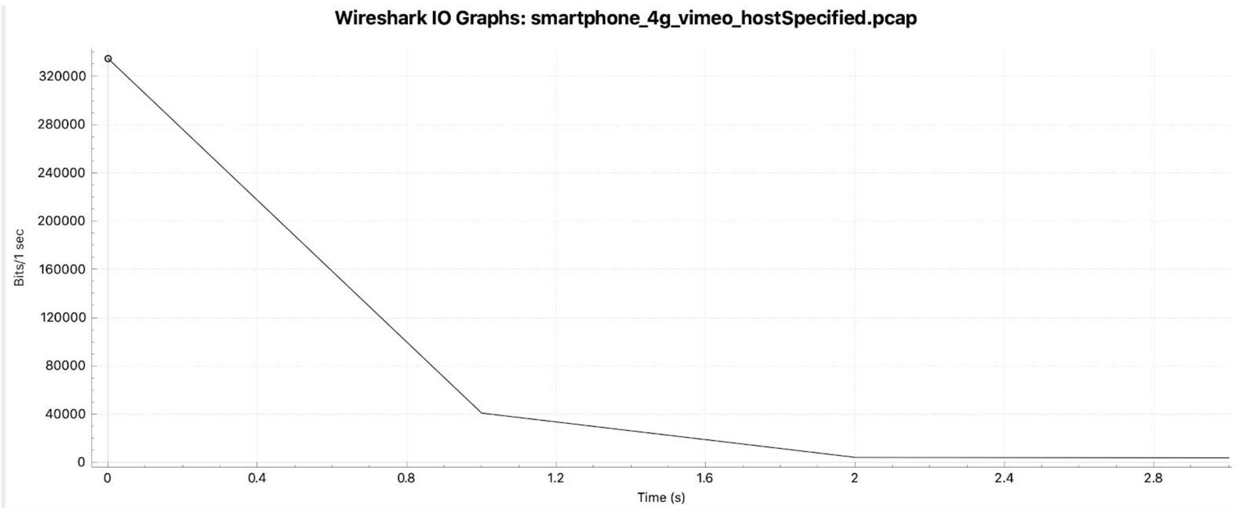
Link quality:

There were few TLS packets located close in time to a possible link quality drop from the client (port 443) to the server. The few that were, showed that the number of bytes had dropped from the better quality scenario. In this trace, there were several tcp connections that closed and reopened when the quality was “bad,” making the pattern harder to see.

Over Time:

The streaming quality seemingly went up and down at different periods of time, most likely to account for the varying link quality. This is concurred with the fact that we observe more packets indicating some sort of transmission failure during periods when streaming bitrate went down.

c. Vimeo



For this part, I used wireshark in order to create an IO graph for a particular stream. I altered the y axis to project bits/ second. As you can see from this graph, the bit rate varies drastically, meaning that Vimeo on a smart phone connected to 4g has variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

No.	Time	Source	Destination	Protocol	Length	Sequence numl	Acknowledgment	Source F	Info
45	0.714205	151.101.64.217	10.153.113.219	TLSv1.2	1454	3231382811	2307189531	443	Application Data [TCP segment of a rea
46	0.714207	151.101.64.217	10.153.113.219	TLSv1.2	1454	3231384209	2307189531	443	Application Data [TCP segment of a rea
47	0.714209	151.101.64.217	10.153.113.219	TLSv1.2	1454	3231385607	2307189531	443	Application Data [TCP segment of a rea
48	0.714212	151.101.64.217	10.153.113.219	TLSv1.2	1454	3231387005	2307189531	443	Application Data [TCP segment of a rea
49	0.714214	151.101.64.217	10.153.113.219	TLSv1.2	1454	3231388403	2307189531	443	Application Data [TCP segment of a rea
50	0.714216	151.101.64.217	10.153.113.219	TLSv1.2	1454	3231389801	2307189531	443	Application Data [TCP segment of a rea
51	0.714218	151.101.64.217	10.153.113.219	TLSv1.2	387	3231391199	2307189531	443	Application Data

No.	Time	Source	Destination	Protocol	Length	Sequence numl	Acknowledgment	Source F	Info
61	0.846419	10.153.113.219	151.101.64.217	TLSv1.2	107	1509940652	2523473164	56364	Change Cipher Spec, Encrypted Handshak
62	0.847643	10.153.113.219	151.101.64.217	TLSv1.2	862	1509940703	2523473164	56364	Application Data
63	0.890821	151.101.64.217	10.153.113.219	TCP	56	2523473164	1509940703	443	443 → 56364 [ACK] Seq=2523473164 Ack=1
64	0.890822	151.101.64.217	10.153.113.219	TCP	56	2523473164	1509941509	443	443 → 56364 [ACK] Seq=2523473164 Ack=1
65	1.027997	151.101.64.217	10.153.113.219	TLSv1.2	984	2523473164	1509941509	443	Application Data, Application Data
66	1.028088	10.153.113.219	151.101.64.217	TCP	56	1509941509	2523474092	56364	56364 → 443 [ACK] Seq=1509941509 Ack=2
67	1.256463	10.153.113.219	151.101.64.217	TLSv1.2	886	2307189531	3231391530	56359	Application Data
68	1.287088	151.101.64.217	10.153.113.219	TCP	56	3231391530	2307190361	443	443 → 56359 [ACK] Seq=3231391530 Ack=2
69	1.419193	151.101.64.217	10.153.113.219	TLSv1.2	1056	3231391530	2307190361	443	Application Data, Application Data
70	1.419447	10.153.113.219	151.101.64.217	TCP	56	2307190361	3231392530	56359	56359 → 443 [ACK] Seq=2307190361 Ack=3
71	1.710364	10.153.113.219	151.101.64.217	TLSv1.2	865	2307190361	3231392530	56359	Application Data
72	1.769133	151.101.64.217	10.153.113.219	TCP	56	3231392530	2307191170	443	443 → 56359 [ACK] Seq=3231392530 Ack=2
73	1.952052	151.101.64.217	10.153.113.219	TLSv1.2	1056	3231392530	2307191170	443	Application Data, Application Data
74	1.952680	10.153.113.219	151.101.64.217	TCP	56	2307191170	3231393530	56359	56359 → 443 [ACK] Seq=2307191170 Ack=3
75	2.881841	10.153.113.219	151.101.64.217	TLSv1.2	87	1509941509	2523474092	56364	Encrypted Alert
76	2.882573	10.153.113.219	151.101.64.217	TCP	56	1509941540	2523474092	56364	56364 → 443 [FIN, ACK] Seq=1509941540
77	2.919225	151.101.64.217	10.153.113.219	TCP	56	2523474092	1509941509	443	[TCP Dup ACK 64#1] 443 → 56364 [ACK] S
78	2.927095	151.101.64.217	10.153.113.219	TCP	56	2523474092	1509941541	443	443 → 56364 [ACK] Seq=2523474092 Ack=1
79	2.935189	151.101.64.217	10.153.113.219	TLSv1.2	87	2523474092	1509941541	443	Encrypted Alert
80	2.935435	10.153.113.219	151.101.64.217	TCP	44	1509941541	0	56364	56364 → 443 [RST] Seq=1509941541 Win=0
81	2.941235	151.101.64.217	10.153.113.219	TCP	56	2523474123	1509941541	443	443 → 56364 [FIN, ACK] Seq=2523474123
82	2.941364	151.101.64.217	10.153.113.219	TCP	56	2523474123	1509941541	443	[TCP Out-Of-Order] 443 → 56364 [FIN, A
83	2.941496	10.153.113.219	151.101.64.217	TCP	44	1509941541	0	56364	56364 → 443 [RST] Seq=1509941541 Win=0

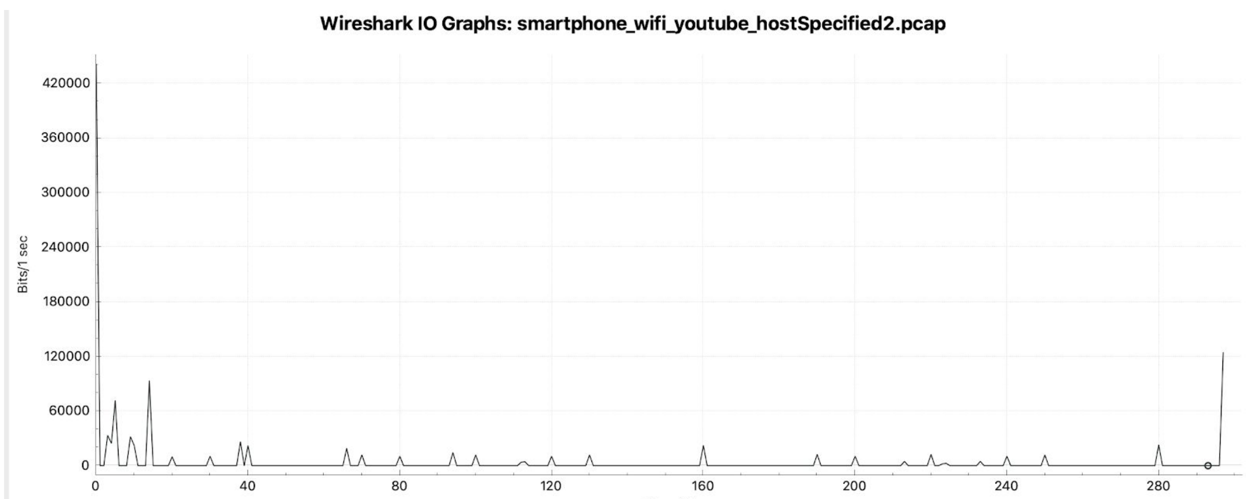
From the later image you can see several TLSv1.2 packets getting sent from the server (port 443) to the client. Compared to the first image which had lengths of 1454 bytes, the lengths of these packets are significantly lesser (1056 bytes, 984 bytes, and 862 bytes). It can be assumed that this drop in average number of bytes (and therefore bit rate) is caused by the link quality going down, as indicated by the subsequent error packets.

Over Time:

It is observed that the streaming bitrate is greater in the beginning, but then lowers toward the end. This is most likely due to link quality worsening. This is concurred with the fact that we observe more packets indicating some sort of transmission failure.

C. <u>A smartphone that connects to a WiFi network.</u>

a. Youtube



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then we have smaller, varying peaks of bit rates. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

No.	Time	Source	Destination	Protocol	Length	Sequence num	Acknowledgment	Source P	Info
427	280.464049	192.168.5.11	172.217.19.110	TCP	66	1457857507	1902513365	60678	60678 → 443 [ACK] Seq=1457857507 Ack=1
428	280.464782	172.217.19.110	192.168.5.11	TLSv1.2	105	1902513365	1457857507	443	Application Data
429	280.465066	192.168.5.11	172.217.19.110	TCP	66	1457857507	1902513404	60678	60678 → 443 [ACK] Seq=1457857507 Ack=1
430	280.465284	192.168.5.11	172.217.19.110	TLSv1.2	105	1457857507	1902513404	60678	Application Data
431	280.573378	172.217.19.110	192.168.5.11	TCP	66	2312474719	165275091	443	443 → 60667 [ACK] Seq=2312474719 Ack=1
432	280.573852	172.217.19.110	192.168.5.11	TCP	66	1902513404	1457857546	443	443 → 60678 [ACK] Seq=1902513404 Ack=1
433	297.647393	192.168.5.11	172.217.19.110	TLSv1.2	631	165275091	2312474719	60667	Application Data
434	297.676201	172.217.19.110	192.168.5.11	TCP	66	2312474719	165275656	443	443 → 60667 [ACK] Seq=2312474719 Ack=1
435	297.907408	172.217.19.110	192.168.5.11	TLSv1.2	152	2312474719	165275656	443	Application Data
436	297.907935	192.168.5.11	172.217.19.110	TCP	66	165275656	2312474805	60667	60667 → 443 [ACK] Seq=165275656 Ack=23
437	297.912869	172.217.19.110	192.168.5.11	TLSv1.2	1474	2312474805	165275656	443	Application Data
438	297.912873	172.217.19.110	192.168.5.11	TLSv1.2	1474	2312476213	165275656	443	Application Data
439	297.912877	172.217.19.110	192.168.5.11	TLSv1.2	1474	2312477621	165275656	443	Application Data

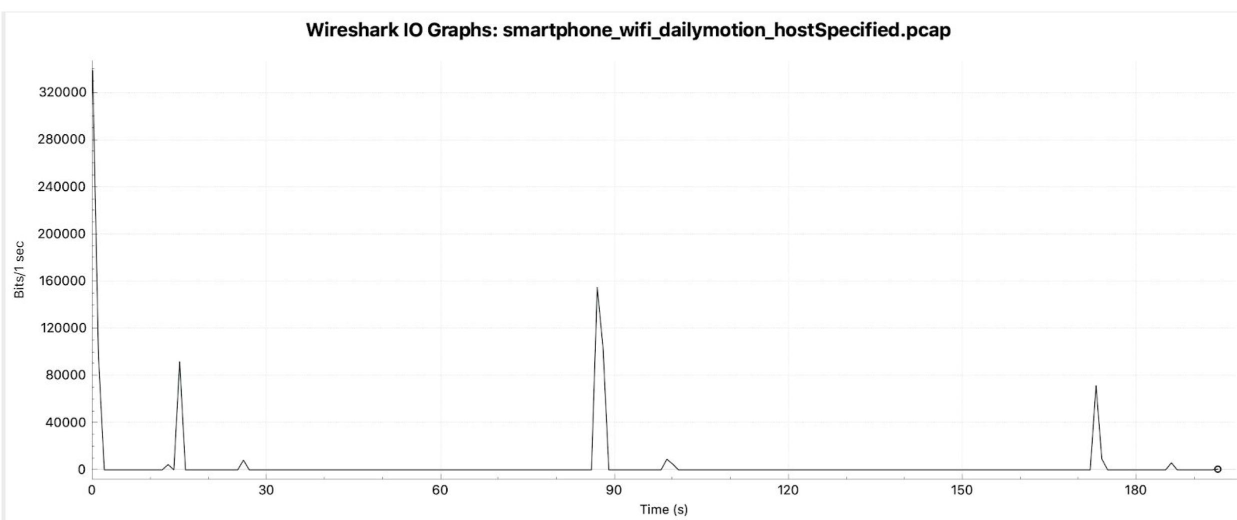
No.	Time	Source	Destination	Protocol	Length	Sequence num	Acknowledgment	Source P	Info
82	3.024732	192.168.5.11	216.58.214.206	TLSv1.2	357	3814424351	2534757065	60656	Application Data
83	3.025675	192.168.5.11	216.58.214.206	TLSv1.2	1041	3814424642	2534757065	60656	Application Data
84	3.235549	192.168.5.11	216.58.214.206	TCP	1332	3814424351	2534757065	60656	[TCP Retransmission] 60656 → 443 [PSH,
85	3.475739	216.58.214.206	192.168.5.11	TCP	66	2534757065	3814424642	443	443 → 60656 [ACK] Seq=2534757065 Ack=3
86	3.475742	216.58.214.206	192.168.5.11	TCP	66	2534757065	3814425617	443	443 → 60656 [ACK] Seq=2534757065 Ack=3
87	3.475744	216.58.214.206	192.168.5.11	TLSv1.2	135	2534757065	3814425617	443	Application Data
88	3.475746	216.58.214.206	192.168.5.11	TLSv1.2	128	2534757134	3814425617	443	Application Data
89	3.475748	216.58.214.206	192.168.5.11	TLSv1.2	97	2534757196	3814425617	443	Application Data
90	3.475750	216.58.214.206	192.168.5.11	TLSv1.2	105	2534757227	3814425617	443	Application Data
91	3.475752	216.58.214.206	192.168.5.11	TCP	105	2534757227	3814425617	443	[TCP Retransmission] 443 → 60656 [PSH,
92	3.475754	216.58.214.206	192.168.5.11	TCP	78	2534757266	3814425617	443	[TCP Dup ACK 86#1] 443 → 60656 [ACK] S
93	3.475756	216.58.214.206	192.168.5.11	TCP	267	2534757065	3814425617	443	[TCP Retransmission] 443 → 60656 [PSH,
94	3.476385	192.168.5.11	216.58.214.206	TCP	66	3814425617	2534757266	60656	60656 → 443 [ACK] Seq=3814425617 Ack=2
95	3.476387	192.168.5.11	216.58.214.206	TCP	78	3814425617	2534757266	60656	[TCP Dup ACK 94#1] 60656 → 443 [ACK] S
96	3.476389	192.168.5.11	216.58.214.206	TCP	78	3814425617	2534757266	60656	[TCP Dup ACK 94#2] 60656 → 443 [ACK] S
97	3.477400	192.168.5.11	216.58.214.206	TLSv1.2	105	3814425617	2534757266	60656	Application Data

From the first image, we were beginning to see TLS packets send 1474 bytes of data. In the latter image, when we can only presume that the link quality has gone down due to the need for retransmissions, we see that the packets transfer 128, 97, and 105 bytes of data from client to server. From this, we can infer that as the link quality goes down, so does the streaming rate.

Over Time:

We observe that the streaming bitrate is high in the beginning, but gets quickly lowered. It can be presumed that this was simply an adjustment made due to link quality. This is concurred with the fact that we observe more packets indicating some sort of transmission failure later.

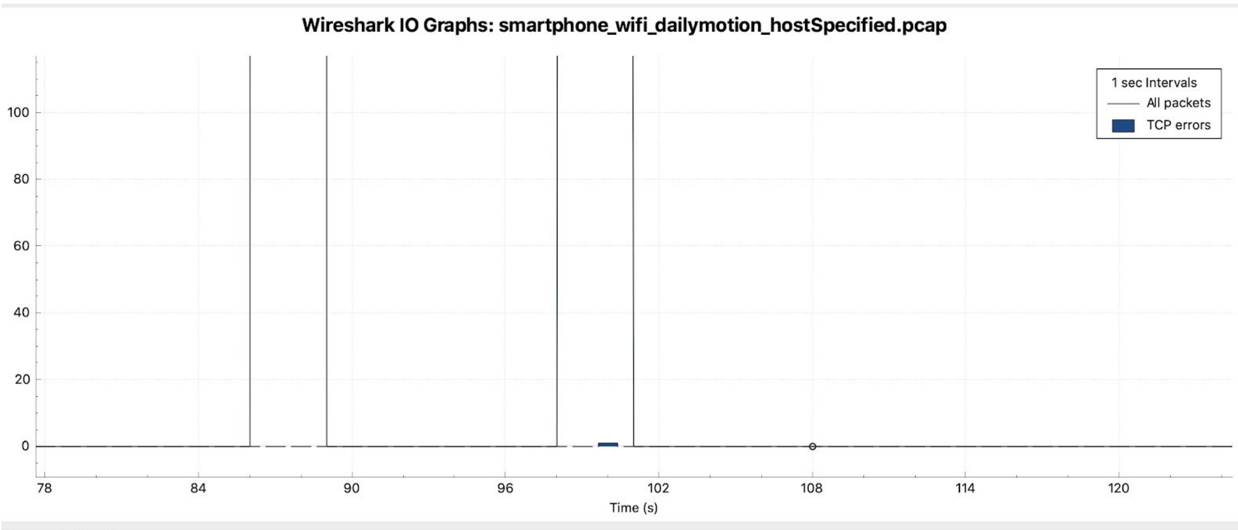
b. Daily Motion



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then we have smaller, varying peaks of bit rates. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:

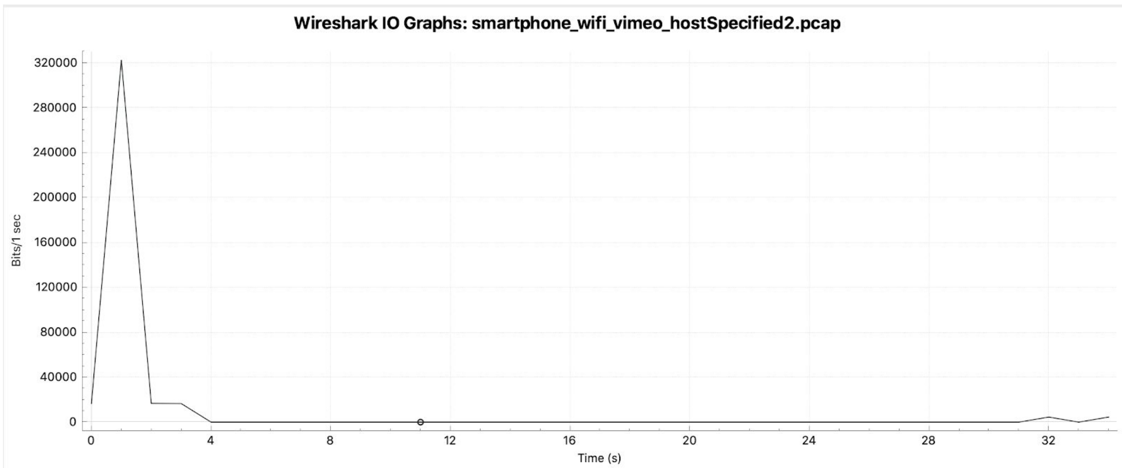


We observe that as we view more packets indicating transmission failure, the streaming bitrate goes down. We can see this by comparing the bit rate just below the 90 marker, which has no tcp errors, and the bit rate just above the 90 marker which has tcp errors (bit rate shown by first image). This means that there is a direct correlation between link quality and streaming bitrate, as link quality goes down, streaming bit rate goes down.

Over Time:

It is observed that the streaming bitrate is greater in the beginning, then lowers, then rises, then lowers. These dips correspond to weaker link quality.

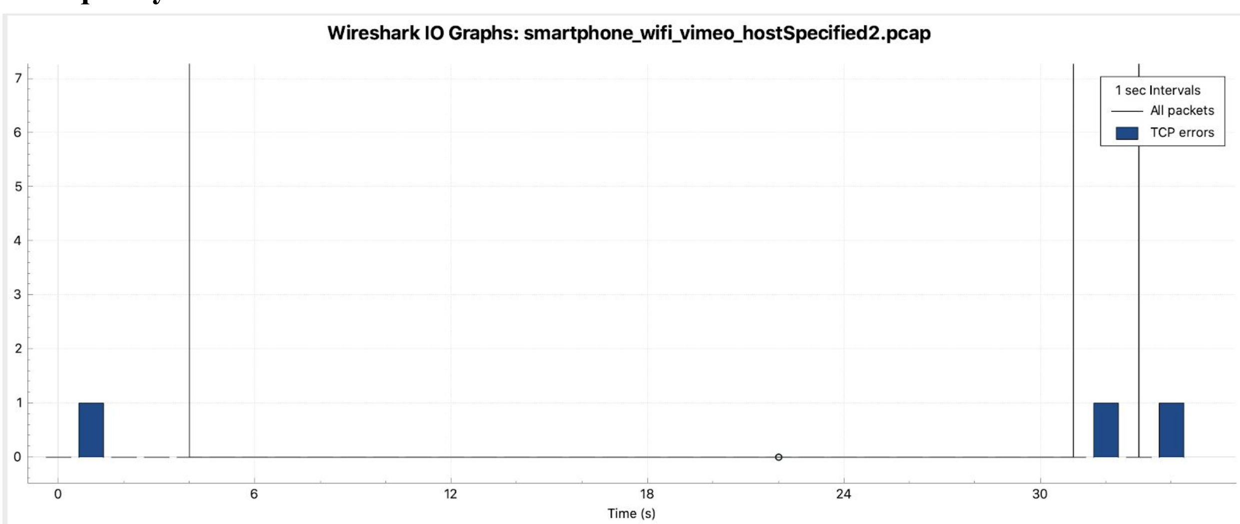
c. Vimeo



Using wireshark's ability to produce IO graphs, we are able to get this graph of number of bits transmitted per one second, ie. the bit rate. From this, we are able to see that the bit rate spikes in the beginning, then tapers out later. Therefore, it is using a variable bit rate.

i. Constant Bit Rate/Variable Bit Rate: **Variable Bit Rate**

Link quality:



We observe that as we view more packets indicating transmission failure, the streaming bitrate goes down, meaning that there is a direct correlation between link quality and streaming bitrate. As link quality goes down, streaming bit rate goes down.

We also note that compared to other traces, the first peak in bit rate is less than, most likely due to poor link quality, as indicated by the initial error packet.

Over Time:

It is observed that the streaming bitrate is greater in the beginning, but then lowers toward the end. This is most likely due to link quality worsening. This is concurred with the fact that we observe more packets indicating some sort of transmission failure.