

COMP445

LAB 2

Alawadi Mustafa (40217764)

2/18/2025

PART 2

TASK 1

1.

```
<MPD mediaPresentationDuration="PT634.566S" minBufferTime="PT2.00S" profiles="urn:hbbtv:dash:profile:isoff-live;2012,urn:mpeg:dash:profile:isoff-live;2011" type="static"
 xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd">
 <BaseURL>./</BaseURL>
 <Period>
 <AdaptationSet mimeType="video/mp4" contentType="video" subsegmentAlignment="true" subsegmentStartsWithSAP="1" par="16:9">
 <SegmentTemplate duration="120" timescale="30" media="$RepresentationID$.$Number$.m4v" startNumber="1" initialization="$RepresentationID$/$RepresentationID$_0.m4v"/>
 <Representation id="bbb_30fps_1024x576_2500k" codecs="avc1.64001" bandwidth="3134488" width="1024" height="576" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_1280x720_4000k" codecs="avc1.64001" bandwidth="4952892" width="1280" height="720" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_1920x1080_8000k" codecs="avc1.64002" bandwidth="9914554" width="1920" height="1080" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_320x188_200k" codecs="avc1.64000" bandwidth="254326" width="320" height="188" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_320x188_400k" codecs="avc1.64000" bandwidth="507246" width="320" height="188" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_480x270_800k" codecs="avc1.64001" bandwidth="1254758" width="480" height="270" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_640x360_1600k" codecs="avc1.64001" bandwidth="1013310" width="640" height="360" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_640x360_800k" codecs="avc1.64001" bandwidth="1883700" width="640" height="360" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_768x432_1500k" codecs="avc1.64001" bandwidth="1883700" width="768" height="432" frameRate="30" sar="1:1" scanType="progressive"/>
 <Representation id="bbb_30fps_3840x2160_12000k" codecs="avc1.640033" bandwidth="14931538" width="3840" height="2160" frameRate="30" sar="1:1" scanType="progressive"/>
 </AdaptationSet>
 <AdaptationSet mimeType="audio/mp4" contentType="audio" subsegmentAlignment="true" subsegmentStartsWithSAP="1">
 <Accessibility schemeIdUri="urn:tvai:metadata:cs:AudioPurposeCs:2007" value="6"/>
 <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
 <SegmentTemplate duration="192512" timescale="48000" media="$RepresentationID$.$Number$.m4a" startNumber="1" initialization="$RepresentationID$/$RepresentationID$_0.m4a"/>
 <Representation id="bbb_a64" codecs="mp4a.40.5" bandwidth="67071" audioSamplingRate="48000">
 <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
 </Representation>
 </AdaptationSet>
 </Period>
 </MPD>
```

2.

DASH Reference Client 4.7.0

Stream: https://dash.akamai.net/akama/bbb_30fps/bbb_30fps.mpd

Show Options Stop Load Copy Settings URL

Updated Export settings

Our export settings feature creates shorter URLs now. Click on "Copy Settings URL" on the top right and paste the URL in the address bar of your browser. The current settings are compared to the default settings and the difference is stored using query parameters.

Additional samples can be found in the Sample Section.

Video Audio

Buffer Length: 10.417
 Bitrate Downloading: 1013 kbps
 Index Downloading: 4 / 10
 Index playing: 4 / 10
 Dropped Frames: 163
 Latency (minavg|max): 0.00 | 0.26 | 0.54
 Download (minavg|max): 0.01 | 1.08 | 2.17
 Ratio (minavg|max): 1.84 | 3.70 | 666.67

Frame 4035: PTS= 00:02:14.500

640x360 / 800 kbps / 30 fps

20:14 10:34

Clear Disable

Video Buffer Level Video Bitrate (kbps) Video Download Time (sec)

Video Buffer Level: 0 to 15 seconds
Video Bitrate (kbps): 0 to 1250
Video Download Time (sec): 0 to 2 seconds

Network Tab (m4v)

Name	Status	Type	Initiator	Size	Time
bbb_30fps_640x360_800k_0.m...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_640x360_800k_1.m...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_3840x160_1200k_0...	200	xhr	XHRLoader.js[102]	3.9 MB	20.00 s
bbb_30fps_3840x160_1200k_0...	200	xhr	XHRLoader.js[102]	1.1 MB	5.68 s
bbb_30fps_640x360_1000k_0...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_640x360_1000k_0...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_1280x720_400k_0...	200	xhr	XHRLoader.js[102]	17 kB	165 ms
bbb_30fps_1280x720_400k_0...	200	xhr	XHRLoader.js[102]	2.0 MB	12.81 s
bbb_30fps_640x360_1000k_0...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_640x360_1000k_0...	200	xhr	XHRLoader.js[102]	511 kB	5.33 s
bbb_30fps_320x160_200k_0...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_320x160_200k_0...	200	xhr	XHRLoader.js[102]	0 ms	
bbb_30fps_480x270_200k_0...	200	xhr	XHRLoader.js[102]	0 ms	
bbb_30fps_480x270_200k_0...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_640x360_200k_0...	200	xhr	XHRLoader.js[102]	0 ms	
bbb_30fps_640x360_200k_0...	200	xhr	XHRLoader.js[102]	424 kB	2.54 s
bbb_30fps_640x360_200k_0...	200	xhr	XHRLoader.js[102]	404 kB	2.56 s
bbb_30fps_640x360_200k_10...	200	xhr	XHRLoader.js[102]	401 kB	2.43 s
bbb_30fps_640x360_200k_11...	200	xhr	XHRLoader.js[102]	368 kB	2.29 s
bbb_30fps_640x360_200k_12...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_640x360_200k_13...	200	xhr	XHRLoader.js[102]	392 kB	2.39 s
bbb_30fps_640x360_200k_14...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_640x360_200k_15...	200	xhr	XHRLoader.js[102]	4 ms	
bbb_30fps_640x360_200k_16...	200	xhr	XHRLoader.js[102]	4 ms	
bbb_30fps_640x360_200k_17...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_18...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_19...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_20...	200	xhr	XHRLoader.js[102]	4 ms	
bbb_30fps_640x360_200k_21...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_22...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_23...	200	xhr	XHRLoader.js[102]	3 ms	
bbb_30fps_640x360_200k_24...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_25...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_26...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_640x360_200k_27...	200	xhr	XHRLoader.js[102]	3 ms	
...
46 / 65 requests	10.7 MB	10.7 MB transferred	29.1 MB	23.4 MB resources	

DASH Reference Client 4.7.0

Stream: https://dash.akamai.net/akama/bbb_30fps/bbb_30fps.mpd

Show Options Stop Load Copy Settings URL

Updated Export settings

Our export settings feature creates shorter URLs now. Click on "Copy Settings URL" on the top right and paste the URL in the address bar of your browser. The current settings are compared to the default settings and the difference is stored using query parameters.

Additional samples can be found in the Sample Section.

Video Audio

Buffer Length: 0
 Bitrate Downloading: 254 kbps
 Index Downloading: 1 / 10
 Index playing: 1 / 10
 Dropped Frames: 0
 Latency (minavg|max): 0.00 | 0.00 | 0.00
 Download (minavg|max): 0.00 | 0.01 | 0.01
 Ratio (minavg|max): 571.43 | 727.27 | 2000.00

Frame 1319: PTS= 00:00:43.967

320x180 / 200 kbps / 30 fps

00:46 10:34

Clear Disable

Video Buffer Level Video Bitrate (kbps) Video Download Time (sec)

Video Buffer Level: 0 to 15 seconds
Video Bitrate (kbps): 0 to 300
Video Download Time (sec): 0 to 2 seconds

Network Tab (m4v)

Name	Status	Type	Initiator	Size	Time
bbb_30fps_1024x576_250k_0...	200	xhr	XHRLoader.js[102]	1.7 kB	3.40 s
bbb_30fps_1024x576_250k_1...	200	xhr	XHRLoader.js[102]	65.5 kB	2.00 s
bbb_30fps_320x180_200k_0...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_320x180_200k_2...	200	xhr	XHRLoader.js[102]	64.0 kB	2.00 s
bbb_30fps_1024x576_250k_...	200	xhr	XHRLoader.js[102]	54.6 kB	2.00 s
bbb_30fps_320x180_200k_2...	200	xhr	XHRLoader.js[102]	53.2 kB	2.00 s
bbb_30fps_1024x576_250k_...	200	xhr	XHRLoader.js[102]	53.4 kB	2.00 s
bbb_30fps_320x180_200k_...	200	xhr	XHRLoader.js[102]	78.8 kB	2.00 s
bbb_30fps_320x180_200k_3...	200	xhr	XHRLoader.js[102]	4.4 kB	7.65 s
bbb_30fps_320x180_200k_4...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_320x180_200k_5...	200	xhr	XHRLoader.js[102]	1 ms	
bbb_30fps_320x180_200k_6...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_1024x576_250k_...	200	xhr	XHRLoader.js[102]	84.5 kB	2.00 s
bbb_30fps_320x180_200k_7...	200	xhr	XHRLoader.js[102]	2 ms	
bbb_30fps_320x180_200k_8...	200	xhr	XHRLoader.js[102]	3 ms	
bbb_30fps_320x180_200k_9...	200	xhr	XHRLoader.js[102]	3 ms	
bbb_30fps_320x180_200k_10...	200	xhr	XHRLoader.js[102]	3 ms	
bbb_30fps_320x180_200k_11...	200	xhr	XHRLoader.js[102]	2 ms	
...
18 / 33 requests	407 kB	407 kB transferred	1.9 MB	2.3 MB resources	

DASH Reference Client 4.7.0

Stream: https://dash.akamai.net/akama/bbb_30fps/bbb_30fps.mpd

Show Options Stop Load Copy Settings URL

Updated Export settings

Our export settings feature creates shorter URLs now. Click on "Copy Settings URL" on the top right and paste the URL in the address bar of your browser. The current settings are compared to the default settings and the difference is stored using query parameters.

Additional samples can be found in the Sample Section.

Video Audio

Buffer Length: 10.43
Bitrate Downloading: 4953 kbps
Index Downloading: 8 / 10
Index playing: 7 / 10
Dropped Frames: 0
Latency (min/avg/max): 0.17 | 0.40 | 0.53
Download (min/avg/max): 1.10 | 1.28 | 1.40
Ratio (min/avg/max): 2.85 | 3.13 | 3.65

Frame 4296: PTS= 00:02:23.2000

1024x576 / 2500 kbps / 30 fps

10:34 2:23

Video Buffer Level Video Bitrate (kbps) Video Download Time (sec)

01:30 01:35 01:40 01:45 01:50 01:55 02:00 02:05 02:10 02:15 02:20

0 2500 5000 7500 10000 12500

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

0 2500 5000 7500 10000 12500

20:00ms 40:00ms 60:00ms 80:00ms 100:00ms 120:00ms 140:00ms 160:00ms 180:00ms

Name Status Type Initiator Size Time

b6b_30fps_1280x720_4000k_0... 200 xhr XHRLoader.js[102] (disk ca... 2 ms

b6b_30fps_1280x720_4000k_1... 200 xhr XHRLoader.js[102] (disk ca... 1.24 s

b6b_30fps_1280x720_4000k_2... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_3... 200 xhr XHRLoader.js[102] (disk ca... 0 ms

b6b_30fps_1280x720_4000k_4... 200 xhr XHRLoader.js[102] (disk ca... 0 ms

b6b_30fps_1280x720_4000k_5... 200 xhr XHRLoader.js[102] (disk ca... 2.9 MB

b6b_30fps_1280x720_4000k_6... 200 xhr XHRLoader.js[102] (disk ca... 1.6 MB

b6b_30fps_1280x720_4000k_7... 200 xhr XHRLoader.js[102] (disk ca... 2.2 MB

b6b_30fps_1280x720_4000k_8... 200 xhr XHRLoader.js[102] (disk ca... 2.1 MB

b6b_30fps_1280x720_4000k_9... 200 xhr XHRLoader.js[102] (disk ca... 2.0 MB

b6b_30fps_1280x720_4000k_10... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_11... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_12... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_13... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_14... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_15... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_16... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_17... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_18... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_19... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_20... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_21... 200 xhr XHRLoader.js[102] (disk ca... 7 ms

b6b_30fps_1280x720_4000k_22... 200 xhr XHRLoader.js[102] (disk ca... 8 ms

b6b_30fps_1280x720_4000k_23... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_24... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_25... 200 xhr XHRLoader.js[102] (disk ca... 5 ms

b6b_30fps_1280x720_4000k_26... 200 xhr XHRLoader.js[102] (disk ca... 6 ms

b6b_30fps_1280x720_4000k_27... 200 xhr XHRLoader.js[102] (disk ca... 7 ms

b6b_30fps_1280x720_4000k_28... 200 xhr XHRLoader.js[102] (disk ca... 522 ms

b6b_30fps_1280x720_4000k_29... 200 xhr XHRLoader.js[102] (disk ca... 1.7 KB

b6b_30fps_1280x720_4000k_30... 200 xhr XHRLoader.js[102] (disk ca... 0.08 s

b6b_30fps_1280x720_4000k_31... 200 xhr XHRLoader.js[102] (disk ca... 3.9 MB

b6b_30fps_1280x720_4000k_32... 200 xhr XHRLoader.js[102] (disk ca... 3.45 ms

b6b_30fps_1280x720_4000k_33... 200 xhr XHRLoader.js[102] (disk ca... 1.2 MB

b6b_30fps_1280x720_4000k_34... 200 xhr XHRLoader.js[102] (disk ca... 1.75 ms

b6b_30fps_1280x720_4000k_35... 200 xhr XHRLoader.js[102] (disk ca... 2.41 ms

b6b_30fps_1280x720_4000k_36... 200 xhr XHRLoader.js[102] (disk ca... 2.1 MB

b6b_30fps_1280x720_4000k_37... 200 xhr XHRLoader.js[102] (disk ca... 2.83 ms

48 / 89 requests 35.0 MB / 35.0 MB transferred 78.8 MB / 80.1 MB resources

reference.dashif.org/dash.js/v4.7.0/samples/dash-if-reference-player/index.html

DASH Reference Client 4.7.0

Stream: https://dash.akamai.net/akama/bbb_30fps/bbb_30fps.mpd

Show Options Stop Load Copy Settings URL

Updated Export settings

Our export settings feature creates shorter URLs now. Click on "Copy Settings URL" on the top right and paste the URL in the address bar of your browser. The current settings are compared to the default settings and the difference is stored using query parameters.

Additional samples can be found in the Sample Section.

Video Audio

Buffer Length: 63.377
Bitrate Downloading: 14932 kbps
Index Downloading: 10 / 10
Index playing: 10 / 10
Dropped Frames: 0
Latency (min/avg/max): 0.07 | 0.17 | 0.42
Download (min/avg/max): 0.10 | 0.18 | 0.25
Ratio (min/avg/max): 16.06 | 22.41 | 38.83

Frame 5658: PTS= 00:03:08.6000

3840x2160 / 12000 kbps / 30 fps

10:34 03:08

Video Buffer Level Video Bitrate (kbps) Video Download Time (sec)

02:20 02:25 02:30 02:35 02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00

0 5000 10000 15000 20000

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

0 5000 10000 15000 20000

60,000ms 80,000ms 100,000ms 120,000ms 140,000ms 160,000ms 180,000ms 200,000ms

Name Status Type Initiator Size Time

b6b_30fps_1280x720_4000k_0... 200 xhr XHRLoader.js[102] (disk ca... 1 ms

b6b_30fps_1280x720_4000k_1... 200 xhr XHRLoader.js[102] (disk ca... 4 ms

b6b_30fps_1280x720_4000k_2... 200 xhr XHRLoader.js[102] (disk ca... 1 ms

b6b_30fps_1280x720_4000k_3... 200 xhr XHRLoader.js[102] (disk ca... 7 ms

b6b_30fps_1280x720_4000k_4... 200 xhr XHRLoader.js[102] (disk ca... 14 ms

b6b_30fps_1280x720_4000k_5... 200 xhr XHRLoader.js[102] (disk ca... 6.0 MB

b6b_30fps_1280x720_4000k_6... 200 xhr XHRLoader.js[102] (disk ca... 383 ms

b6b_30fps_1280x720_4000k_7... 200 xhr XHRLoader.js[102] (disk ca... 9 ms

b6b_30fps_1280x720_4000k_8... 200 xhr XHRLoader.js[102] (disk ca... 6.4 MB

b6b_30fps_1280x720_4000k_9... 200 xhr XHRLoader.js[102] (disk ca... 467 ms

b6b_30fps_1280x720_4000k_10... 200 xhr XHRLoader.js[102] (disk ca... 6.0 MB

b6b_30fps_1280x720_4000k_11... 200 xhr XHRLoader.js[102] (disk ca... 6.1 MB

b6b_30fps_1280x720_4000k_12... 200 xhr XHRLoader.js[102] (disk ca... 252 ms

b6b_30fps_1280x720_4000k_13... 200 xhr XHRLoader.js[102] (disk ca... 10 ms

b6b_30fps_1280x720_4000k_14... 200 xhr XHRLoader.js[102] (disk ca... 16 ms

b6b_30fps_1280x720_4000k_15... 200 xhr XHRLoader.js[102] (disk ca... 407 ms

b6b_30fps_1280x720_4000k_16... 200 xhr XHRLoader.js[102] (disk ca... 6.5 MB

b6b_30fps_1280x720_4000k_17... 200 xhr XHRLoader.js[102] (disk ca... 220 ms

b6b_30fps_1280x720_4000k_18... 200 xhr XHRLoader.js[102] (disk ca... 6.4 MB

b6b_30fps_1280x720_4000k_19... 200 xhr XHRLoader.js[102] (disk ca... 15 ms

b6b_30fps_1280x720_4000k_20... 200 xhr XHRLoader.js[102] (disk ca... 5.7 MB

b6b_30fps_1280x720_4000k_21... 200 xhr XHRLoader.js[102] (disk ca... 7.3 ms

b6b_30fps_1280x720_4000k_22... 200 xhr XHRLoader.js[102] (disk ca... 6.3 MB

b6b_30fps_1280x720_4000k_23... 200 xhr XHRLoader.js[102] (disk ca... 15 ms

b6b_30fps_1280x720_4000k_24... 200 xhr XHRLoader.js[102] (disk ca... 5.7 MB

b6b_30fps_1280x720_4000k_25... 200 xhr XHRLoader.js[102] (disk ca... 6.7 MB

b6b_30fps_1280x720_4000k_26... 200 xhr XHRLoader.js[102] (disk ca... 156 ms

b6b_30fps_1280x720_4000k_27... 200 xhr XHRLoader.js[102] (disk ca... 5.5 MB

b6b_30fps_1280x720_4000k_28... 200 xhr XHRLoader.js[102] (disk ca... 6.0 MB

b6b_30fps_1280x720_4000k_29... 200 xhr XHRLoader.js[102] (disk ca... 230 ms

b6b_30fps_1280x720_4000k_30... 200 xhr XHRLoader.js[102] (disk ca... 6.5 MB

b6b_30fps_1280x720_4000k_31... 200 xhr XHRLoader.js[102] (disk ca... 196 ms

b6b_30fps_1280x720_4000k_32... 200 xhr XHRLoader.js[102] (disk ca... 5.4 MB

b6b_30fps_1280x720_4000k_33... 200 xhr XHRLoader.js[102] (disk ca... 168 ms

b6b_30fps_1280x720_4000k_34... 200 xhr XHRLoader.js[102] (disk ca... 6.1 MB

b6b_30fps_1280x720_4000k_35... 200 xhr XHRLoader.js[102] (disk ca... 233 ms

TASK 2

The screenshot shows the Chrome DevTools Console tab with the following content:

```
> setInterval() => {
  const dashMetrics = player.getDashMetrics();
  const dashAdapter = player.getDashAdapter();
  const videoType = "video";

  if (dashMetrics && dashAdapter) {
    const bufferLevel = dashMetrics.getCurrentBufferLevel(videoType, true);
    const throughput = player.getAverageThroughput(videoType);
    const httpRequests = dashMetrics.getHttpRequests(videoType);

    let latency = 0;
    if (httpRequests.length > 0) {
      const lastRequest = httpRequests[httpRequests.length - 1];
      if (lastRequest.trequest && lastRequest.tresponse) {
        latency = (lastRequest.tresponse.getTime() - lastRequest.trequest.getTime()) / 1000;
      }
    }

    console.log(`Buffer Level: ${bufferLevel ? bufferLevel.toFixed(2) : 'N/A'} seconds`);
    console.log(`Measured Throughput: ${throughput ? throughput.toFixed(2) : 'N/A'} kbps`);
    console.log(`Latency: ${latency ? latency.toFixed(2) : 'N/A'} seconds`);
  }
}, 8000);
< 606
Buffer Level: 62.22 seconds
Measured Throughput: 5237852.00 kbps
Latency: 0.01 seconds
Buffer Level: 60.84 seconds
Measured Throughput: 5340051.25 kbps
Latency: 0.01 seconds
Buffer Level: 60.87 seconds
Measured Throughput: 5578914.50 kbps
Latency: 0.01 seconds
Buffer Level: 60.91 seconds
Measured Throughput: 5422311.20 kbps
Latency: 0.01 seconds
Buffer Level: 60.94 seconds
Measured Throughput: 5337263.67 kbps
Latency: 0.01 seconds
Buffer Level: 60.97 seconds
Measured Throughput: 5177401.75 kbps
Latency: 0.00 seconds
Buffer Level: 61.00 seconds
Measured Throughput: 4986432.60 kbps
Latency: 0.01 seconds
Buffer Level: 61.03 seconds
Measured Throughput: 5023460.20 kbps
Latency: 0.01 seconds
Buffer Level: 61.06 seconds
Measured Throughput: 5319011.75 kbps
Latency: 0.01 seconds
>
```

The log output shows periodic measurements of buffer level, measured throughput, and latency over time. The log entries are timestamped with file and line numbers (VM2529:19, VM2529:20, etc.). The buffer level fluctuates between 60.84 and 62.22 seconds. Measured throughput values range from 4986432.60 to 5578914.50 kbps. Latency values are mostly 0.01 seconds, except for one entry at 0.00 seconds.

```
> player.updateSettings({
  streaming: {
    abr: {
      useDefaultABRRules: false, // Disable default ABR rules
      ABRStrategy: "abrThroughput" // Change to "abrBOLA" or "abrDynamic" for other tests
    }
  }
});
<- undefined
Buffer Level: 63.65 seconds
Measured Throughput: 4989966.80 kbps
Latency: 0.01 seconds
Buffer Level: 63.01 seconds
Measured Throughput: 4989966.80 kbps
Latency: 0.01 seconds
Buffer Level: 63.04 seconds
Measured Throughput: 5168850.00 kbps
Latency: 0.01 seconds
Buffer Level: 63.07 seconds
Measured Throughput: 5725514.25 kbps
Latency: 0.01 seconds
Buffer Level: 62.89 seconds
Measured Throughput: 5309128.40 kbps
Latency: 0.01 seconds
Buffer Level: 62.98 seconds
Measured Throughput: 5428442.25 kbps
Latency: 0.01 seconds
Buffer Level: 62.98 seconds
Measured Throughput: 6413509.00 kbps
Latency: 0.01 seconds
Buffer Level: 62.93 seconds
Measured Throughput: 5983095.20 kbps
Latency: 0.01 seconds
Buffer Level: 62.97 seconds
Measured Throughput: 5313335.80 kbps
Latency: 0.01 seconds
Buffer Level: 62.80 seconds
Measured Throughput: 4737496.50 kbps
Latency: 0.01 seconds
>
```

```
> player.updateSettings({
  streaming: {
    abr: {
      useDefaultABRRules: false, // Disable default ABR rules
      ABRStrategy: "abrBOLA" // Use BOLA (Buffer Occupancy-based ABR)
    }
  }
});
< undefined
Buffer Level: 60.88 seconds
Measured Throughput: 7359199.40 kbps
VM2529:19
VM2529:20
Latency: 0.00 seconds
VM2529:21
Buffer Level: 60.74 seconds
VM2529:19
Measured Throughput: 7141532.17 kbps
VM2529:20
Latency: 0.01 seconds
VM2529:21
Buffer Level: 60.94 seconds
VM2529:19
Measured Throughput: 6532785.00 kbps
VM2529:20
Latency: 0.01 seconds
VM2529:21
Buffer Level: 62.23 seconds
VM2529:19
Measured Throughput: 406543.00 kbps
VM2529:20
Latency: 0.11 seconds
VM2529:21
Buffer Level: 62.26 seconds
VM2529:19
Measured Throughput: 406543.00 kbps
VM2529:20
Latency: 0.01 seconds
VM2529:21
Buffer Level: 62.29 seconds
VM2529:19
Measured Throughput: 467692.00 kbps
VM2529:20
Latency: 0.07 seconds
VM2529:21
Buffer Level: 62.20 seconds
VM2529:19
Measured Throughput: 467692.00 kbps
VM2529:20
Latency: 0.01 seconds
VM2529:21
Buffer Level: 62.09 seconds
VM2529:19
Measured Throughput: 429278.67 kbps
VM2529:20
Latency: 0.01 seconds
VM2529:21
Buffer Level: 62.15 seconds
VM2529:19
Measured Throughput: 429278.67 kbps
VM2529:20
Latency: 0.01 seconds
VM2529:21
Buffer Level: 62.18 seconds
VM2529:19
```

```

> player.updateSettings({
  streaming: {
    abr: {
      useDefaultABRRules: false, // Disable default ABR rules
      ABRStrategy: "abrDynamic" // Use Dynamic (Hybrid) ABR
    }
  }
});
< undefined
  Buffer Level: 60.73 seconds
  Measured Throughput: 486300.73 kbps
  Latency: 0.01 seconds
  Buffer Level: 60.76 seconds
  Measured Throughput: 486300.73 kbps
  Latency: 0.01 seconds
  Buffer Level: 60.52 seconds
  Measured Throughput: 517555.88 kbps
  Latency: 0.07 seconds
  Buffer Level: 60.55 seconds
  Measured Throughput: 502056.94 kbps
  Latency: 0.15 seconds
  Buffer Level: 60.59 seconds
  Measured Throughput: 479258.68 kbps
  Latency: 0.07 seconds
  Buffer Level: 60.55 seconds
  Measured Throughput: 467400.75 kbps
  Latency: 0.83 seconds
  Buffer Level: 60.65 seconds
  Measured Throughput: 475029.88 kbps
  Latency: 0.04 seconds
  Buffer Level: 60.68 seconds
  Measured Throughput: 444118.36 kbps
  Latency: 0.06 seconds
  Buffer Level: 60.71 seconds
  Measured Throughput: 478140.50 kbps
  Latency: 0.05 seconds
  Buffer Level: 60.74 seconds
  Measured Throughput: 480212.20 kbps
  Latency: 0.07 seconds
  Buffer Level: 60.51 seconds
  Measured Throughput: 476392.00 kbps
  Latency: 0.35 seconds

```

ABR algo	Avg Buffer	Avg Throughput (Kbps)	Avg Latency
Throughput	Around 62.80	Around 4,73,7496.50	0.01
BOLA	Around 62.20	Around 477,502.20	0.01-0.12
Dynamic	Around 60.55	Around 478,140.50	0.04-0.83

1.

The purpose of this assignment was to compare three Adaptive Bitrate (ABR) algorithms, Throughput-based, BOLA, and Dynamic, by periodically collecting data on buffer levels, throughput, and latency every 8 seconds during video streaming. These metrics were analyzed to assess the performance of each ABR algorithm under varying network conditions. The experiment involved capturing key metrics during the streaming of video content using each of the ABR algorithms. Results from these algorithms were compared to understand their behavior in terms of video quality, buffer stability, and responsiveness to network changes.

2.

The motivation for this work is to compare three common ABR algorithms, Throughput-based, BOLA, and Dynamic, in order to understand their performance and trade-offs in real-world streaming scenarios. As viewers, we always prioritize high video quality for an optimal experience. However, we often encounter issues such as buffering, low video quality, and inconsistent playback. The goal of this study is to explore how each ABR algorithm balances these concerns, focusing on Throughput-based's aim to maximize video quality through high throughput, while BOLA and Dynamic focus on minimizing buffer instability for smoother playback. By analyzing these algorithms, we can better understand why we sometimes experience poor video quality and find potential solutions for improving streaming experiences.

3.

The Throughput-based ABR maintained the highest buffer levels, prioritizing larger buffers to prevent rebuffering. However, this approach occasionally led to delays due to larger buffers filling and draining. The BOLA ABR showed a more balanced approach with relatively stable buffer levels but at the expense of throughput. It demonstrated more consistent latency and fewer fluctuations. The Dynamic ABR exhibited the most unpredictable buffer behavior, with occasional spikes that made it less stable in terms of latency. This instability is due to its adaptive nature, switching between strategies based on varying network conditions. Throughput-based ABR had the highest throughput (due to its focus on utilizing maximum bitrate) but the highest buffer levels, causing occasional delays. BOLA was more conservative in its bitrate selection, leading to lower throughput but ensuring stability in buffer levels, which minimized delays. Dynamic ABR had the most erratic buffer levels, with occasional spikes, and focused on maintaining buffer stability, but this came with reduced throughput. To support this analysis, we have included screenshots that visually represent the performance of each ABR algorithm, showing the buffer levels, throughput, and latency over time for better clarity.

4.

In conclusion, the study highlighted the inherent trade-offs between the three ABR algorithms. The Throughput-based algorithm is excellent at delivering high-quality video, but it can experience buffer instability and occasional delays. On the other hand, both BOLA and Dynamic prioritize buffer stability, which results in lower throughput and potentially reduced video quality in certain network conditions. For future improvements, ABR algorithms could be designed to better balance these factors, adjusting more effectively to fluctuating network conditions, ensuring smoother playback without compromising video quality.

5.

Akram, S., & Khan, A. (2020). *Adaptive Bitrate Video Streaming for Wireless Nodes: A Survey*. ResearchGate. Retrieved from
https://www.researchgate.net/publication/343415080_Adaptive_Bitrate_Video_Streaming_for_Wireless_nodes_A_Survey

DASH Industry Forum. (2024). *DASH.js v4.7.0 - DASH-IF Reference Player*. Retrieved from
<https://reference.dashif.org/dash.js/v4.7.0/samples/dash-if-reference-player/index.html>