



THE ULTIMATE MINI GUIDE TO

ABR LOGIC

WHAT IS ADAPTIVE BITRATE?

A bitrate indicates the **speed at which video content is delivered**. It represents the amount of bits per second that are transmitted digitally across a network, and the speed of the bitrate can also determine the cost of the content delivery. In a video context, **higher bitrate usually indicates a higher quality stream**. A video bitrate indicates the average bitrate consumed by players during the playback session.

The most common metrics in use for video bitrates, according to the measurement unit they use, are the Decimal metric **Mbit/s = 1.000.000 bits/second** and the binary metric **Mibit/s = 1.048.567 bits/second**. These average bitrates help users understand how video players manage quality loading.



WHAT IS BITRATE?

A bitrate can essentially tell the video player how many bits of the video content or file can be processed per second during the video playback.

WHAT IS ADAPTIVE BITRATE?

This is where the concept of Adaptive Bitrates (ABR) comes in.

ABR logic in the player ensures the best possible version, resolution and bitrate of the video content are delivered to the video player.

Always transmitting video content at the fastest speed and the highest quality doesn't make much sense in some playback situations. For example, a viewer could go from watching content at home, where a strong connection and high bandwidth are available, to switching to their mobile device, where a flaky cellular network is used.

Using ABR logic allows the player to deliver video content efficiently and at the right quality for each individual user based on the present circumstances



ABR ALGORITHM

The video streaming landscape is in a constant state of flux, with a continual advancement in devices, resolutions and network conditions. When designing an ABR algorithm there are three key things to keep in mind:

- **Minimising rebuffering:** having a buffer that does not affect continuous playback
- **Maximising efficiency:** streaming the highest usable bitrate aka quality possible
- **Encourage stability:** limit track switches for a more seamless experience

```
{ "timestamp": "2017-06-03T18:42:18.018", "deltaStartMillis": "0", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "36", "sizeChars": "5022", "webURL": "/app/page/analyze", "webParams": "null", "sessionID": "14402n620jm9trnd3s3n7wg0k", "level": "INFO", "requestID": "8249868e-afd8-46ac-9745-839146a20f09"}, {"timestamp": "2017-06-03T18:43:33.030", "deltaStartMillis": "0", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "7", "sizeChars": "10190", "webURL": "/app/page/report", "webParams": "null", "sessionID": "14402n620jm9trnd3s3n7wg0k", "level": "INFO", "requestID": "789d89cb-bfa8-4e7d-8047-498454af885d"}, {"timestamp": "2017-06-03T18:46:21.000", "deltaStartMillis": "0", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "10", "sizeChars": "10190", "webURL": "/app/rest/json/file", "webParams": "file=chartdata_new.json", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "23", "sizeChars": "5022", "webURL": "/app/page/analyze", "webParams": "null", "sessionID": "14402n620jm9trnd3s3n7wg0k", "level": "INFO", "requestID": "8249868e-afd8-46ac-9745-839146a20f09"}, {"timestamp": "2017-06-03T18:48:18.018", "deltaStartMillis": "0", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "36", "sizeChars": "5022", "webURL": "/app/page/analyze", "webParams": "file=chartdata_new.json", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "7", "sizeChars": "10190", "webURL": "/app/page/report", "webParams": "null", "sessionID": "14402n620jm9trnd3s3n7wg0k", "level": "INFO", "requestID": "789d89cb-bfa8-4e7d-8047-498454af885d"}, {"timestamp": "2017-06-03T18:46:21.000", "deltaStartMillis": "0", "class": "com.orgmanager.handlers.RequestHandler", "method": "handle", "message": "Duration Log", "durationMillis": "7", "sizeChars": "10190", "webURL": "/app/page/report", "webParams": "null", "sessionID": "14402n620jm9trnd3s3n7wg0k", "level": "INFO", "requestID": "789d89cb-bfa8-4e7d-8047-498454af885d"} }
```

BANDWIDTH ESTIMATION

Available bandwidth estimation is a crucial part of ABR logic, but also **difficult to get entirely accurate**.

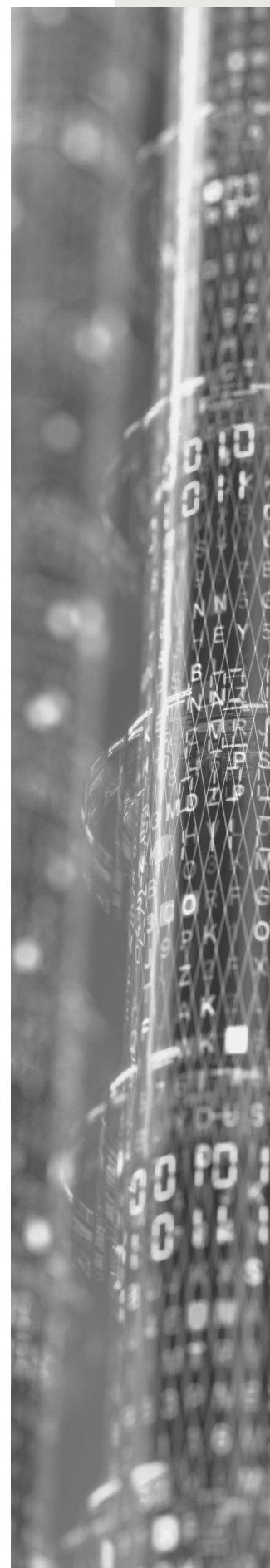
Bandwidth itself is the amount of data that is transmitted over a network in a period of time. The **screen size** or **resolution**, the **start-up time** and the **dropped frame rate** should all be taken into consideration.

A well constructed ABR algorithm will take the screen and player size into account, ensuring that the bitrate is capped at a certain amount.

Some players strive for a **faster join time**. To achieve this the player will start downloading the video content at the **lowest bitrate first** into the buffer and then **aggressively get faster to reduce buffer**. Join times can be **crucial** for the end viewer and **quality of experience** (QoE), as buffering is an annoyance that can cause viewers to **abandon the stream all together** and **exit before video start** (EBVS).

It's also important to take quality of experience and the dropped frame rate into account. At times, the device hosting the stream doesn't have the resources, causing frames of the content to be dropped.

If too many frames are dropped, the player will drop to a lower available bitrate/quality in the stream, and this will clearly reduce the quality of experience for the viewer.



ABR LOGIC BASED ON HISTORIC BANDWIDTH

The other option would be for the network to take the **network history** into account. In this method, the player would take the **history of the current network bandwidth** in consideration to then **estimate** the bitrate.

The player will aim to display the **highest quality** video content based on the **historic bandwidth data** and knowledge of the network conditions.

It is important to note that **network history may be limited to the current playout session** as privacy policies (like GDPR) may rule out the option to leverage the historic bandwidth across sessions.



ADAPTIVE BITRATE STREAMING (ABS)

Adaptive Bitrate Streaming, sometimes referred to as **ABS**, is a method of streaming videos over HTTP, which ensures that viewers will experience an optimal online video playback on all devices and at different bandwidths. ABS **detects the available connection speed in real time** and **adjusts the video stream continuously** to deliver the best possible picture.

First, the source video has to be encoded at **different bitrates**. Those different files of video content can then be viewed with their associated bitrates, which means that **devices can select the speed that is appropriate** to their needs.

During the entire playback, the player will **'adapt'**. When a viewer starts watching a stream, the video player **requests fragments from the lowest bitrate**. If the player finds the download speed is greater than the bitrate of the downloaded segment, the next higher bitrate will be requested.

ABR & ABS WITH THEO

THEOplayer has **three pre-set strategies** that the player will automatically use when it comes to **ABR**:

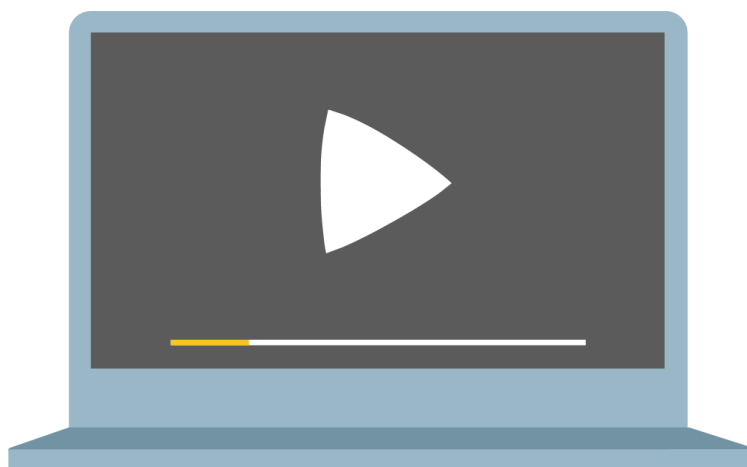
- **Performance**: The player will optimize ABR behavior to focus on the **performance of the player**. This strategy initiates playback with the lowest quality suitable for the device which means **faster start-up time**.
- **Quality**: The player will optimize ABR behavior to focus on displaying the **best visual quality to the end-user**. This strategy initiates playback with the **highest bit rate suitable for the device**.
- **Bandwidth**: This is the **default strategy**. The player will optimize ABR behavior to focus on displaying the **most optimal quality based on historic bandwidth data and knowledge of the network conditions**. When no historic data is available, the player will use the performance strategy.

[Check THEOplayer's ABR demo here.](#)

WHY ABR IS IMPORTANT FOR YOUR QOE AND QOS

With faster internet connections and an expectation for things to be instantaneous, **viewers patience for start up time, buffering and pixels or fuzzy pictures** has all but vanished. Quality of Experience in playback is expected to be **seamless**. And your **QoE is a direct result of your Quality of Service (QoS)**.

Having a player that has advanced ABR and ABS support is **crucial** to guarantee your viewers have the best playback experience possible.



THEO TECHNOLOGIES AND THEOPLAYER

THEOplayer is at the forefront of the video player industry. Our Universal Player with HTML5 capability provides organizations with a single player across all major devices, platforms and browsers. The Universal Player is pre-integrated with components including **streaming, analytics, DRM, and advertising solutions.**

We cover all industry segments and regions, and we are committed to being not just a vendor, but a **solution life cycle partner.** We understand that you want to create a unique experience for your customers, and our THEO experts are available from the inception of the solution through its evolution with innovations over time. At THEO we have the expertise to power some of the biggest **publishers, broadcasters, telcos, and digital natives** across the world, and we are ready to help **you.**



INTERESTED IN LEARNING MORE ABOUT ABR LOGIC?

Get in contact with one of our THEO experts.



Pieter-Jan Speelmans



Willem De Saegher

Ask an expert

THEO Technologies NV
www.theoplayer.com
contact@theoplayer.com