

High Efficiency Streaming Protocol

The video delivery protocol to reduce latency and bandwidth at scale.

WHITEPAPER



High Efficiency Streaming Protocol

The video delivery protocol to reduce latency and bandwidth at scale.

The media industry is constantly pushing to innovate in order to improve viewer experience in an attempt to attract and tie customers to their service. This is however hampered by technical hurdles like high-latency, mind boggling video start and channel change times and bandwidth constraints. With the High-Efficiency Streaming Protocol (HESP), a massive leap forward is made possible. The new protocol enables streaming services to be delivered at scale with a significantly reduced bandwidth and with a sub-second latency. This already impressive list of improvements is further complemented with instantaneous, near real time interactivity and true synchronized viewing, both of multiple videos on the same device and on multiple devices allowing to push the viewer experience and engagement to the next level.

A Continuous State of Flux

Media used to be simple: you had a contract with a telco or cable MSO, running a TV signal to your house over multicast/broadcast. Intending to innovate and reach retail devices striving to better engage audiences, HTTP Adaptive Streaming (HAS) protocols with adaptive bitrate (ABR) capabilities were leveraged. As a result, the media experience on those devices took a step backwards compared to broadcast delivery in relation to latencies and zapping times, as well as throwing up a number of technical hurdles in regards to scalability.

With a rise of OTT-only and cable cutting services, all streaming services are battling for the attention of their audiences. The strategy behind most approaches remains the same: bringing an impeccable viewer experience and engaging viewers to tie them to the service, preventing them from churning to an army of alternatives.

Current streaming services are, however,

struggling with a number of different industry challenges. One such challenge became very apparent during the 2018 FIFA World Cup. High latencies for streams delivered using HAS protocols resulted in spoiled experiences for viewers. This was due to neighbours watching on broadcast services, and shouting loudly, or push notifications and tweets arriving before the start of an attack was even visible to some viewers.

Another common problem, which clearly shows an area where HAS based services can catch up with broadcast, is that of long zapping and buffering times. Where broadcast systems are often able to pick up the multicast signal in a fraction of a second, and continue playing without a glitch or hiccup. Loading times for online unicast streams are significantly higher. Customers of traditional PayTV service providers are expecting to access those services no longer on Service Provider provided Set-top-boxes but increasingly on

a wide range of retail devices. These retail devices require service providers to adopt HTTP Adaptive Streaming (HAS) protocols, however this results in a significant step back in terms of video delivery user-experience.

Besides these obvious drawbacks on a user experience side, there are a few other important challenges with existing technologies. In an attempt to attract viewers, some streaming services have started to offer higher quality video content, boasting full HD and 4K video streaming at a time where most services are still delivering 720p streams.

However, the cost aspect of such business decisions is not to be underestimated. With 1080p images being 2.25 times bigger, and 4k being 4.5 times bigger compared to 720p, the required bandwidth is soaring to new heights.

However the cost aspect of such business decisions is not to be underestimated. With 1080p images being 2.25 times bigger, and 4k being 4.5 times bigger compared to 720p, the required bandwidth is soaring to new heights. As a direct result, cost of delivery is increasing significantly.

This problem is aggravated as streaming services are growing their audiences and media consumption is significantly increasing. Based on a Cisco study, by 2022, it is expected that online video will attribute to more than 82% of consumer internet traffic. The one advantage HAS protocols can leverage, is the use of standard HTTP-based CDNs to deliver media at scale. The egress cost for media services on these CDNs will however increase as they grow their audience, and consumption increases. Interestingly, not growing the audience is often not an option with a trend emerging where larger services manage to spend bigger budgets on content and technology growth, reducing churn and attracting new audiences from smaller services.

The solutions used today

The origin of most of the challenges outlined in the previous section are the result of adoption of HTTP adaptive streaming protocols. As generally understood, these protocols work by splitting long streams in short segments. These segments are generated within the packager and listed within a manifest file, after which they can be distributed over standard HTTP CDNs. Each segment starts with a keyframe, allowing playback to start immediately when the segment is loaded. This however results in a substantial increase in the bandwidth required to deliver the video.

There are however also some downsides to this approach. In most cases, to load segments, players need to set up connections to the server, load the manifest, start loading the segment, and push these into the playback buffer. As Internet connections are often unstable, we require in the player large buffers to absorb reasonable amounts of bandwidth fluctuations and guarantee smooth video playback. This results in an increased latency and zapping/start-up time.

There are a few solutions for the challenges outlined in the previous section which are being leveraged by recent low-latency proposals:

- ▶ The first approach aims to reduce latencies by shortening streaming segments to about 1 to 2 seconds. However, as segments start with keyframes, this approach significantly increases the size of the segment and accordingly the streaming bandwidth.
- ▶ Another often seen approach to optimise delivery and reducing latency is shifting towards non-HTTP based (and less cacheable protocols) like Real-Time Messaging Protocol (RTMP), Web Real-Time Communications (WebRTC) or WebSockets. As these protocols

often do not rely on HTTP, they can not be delivered using traditional CDNs and as a consequence require a dedicated and costly infrastructure to scale. Quite often these protocols have also other drawbacks compared to standard HAS protocols, such as the inability to dynamically adapt to the network connectivity changes like with the HAS Adaptive Bitrate (ABR) algorithms. Another solution that has seen quite some industry attention is the use of Chunked Transfer Encoding (CTE) over HTTP in conjunction with MPEG's Common Media Application Format (CMAF) based DASH, often referred to as DASH CMAF-CTE. This is however only a partial solution as one has to effectively trade-off the latency and start time with the bandwidth overhead, because the principal mechanism used is the shortening or the lengthening segments and chunks.

- ▶ In order to battle bandwidths, we see more and more experiments within the market to deliver content

using alternative codecs , like H2.65/ HEVC, AV1 and VP9. While looking very promising, there are a few drawbacks on this approach still. There are no next generation codecs which are already available across most used devices. As a result, this approach often requires a secondary encoding to be made, packaged and distributed. Some positive news in this area however is that for some use-cases, a business case can already be made to justify this increased complexity.

In summary, these solutions tend to increase significantly the costs for the service provider due to:

- ▶ increased bandwidth;
- ▶ parallel encoding and caching workflows;
- ▶ non HTTP origins and caches.

The High Efficiency Streaming Protocol

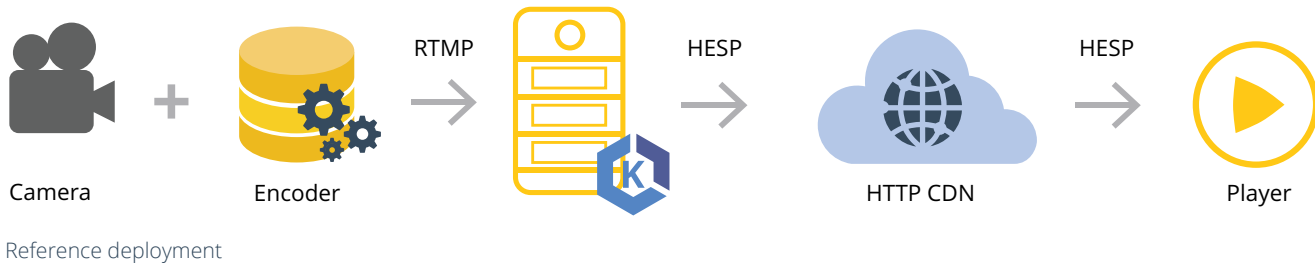
In order not to optimize a single aspect of the problem, and having to make a tradeoff like with current approaches, THEO developed a radically new HTTP Adaptive Streaming protocol. It was designed to improve user experience and engagement by:

- ▶ significantly reducing latency to allow for sub second latency;
- ▶ shortening zapping times to about 100ms to allow for similar to traditional broadcast experiences;
- ▶ bringing down bandwidth costs and optimising viewer bandwidth usage up to 20%;
- ▶ while still being scalable using HTTP CDNs, allowing for virtually endless scaling in a cost efficient manner.

As additional requirements, this protocol was designed to be integratable in existing

video delivery pipelines and workflows, meaning it is compatible with existing encoders and 3rd party CDNs, only requiring changes within the packager and player. To validate these claims, a number of tests were executed to compare end-to-end latency, bandwidth usage and zapping/start-up times with a state-of-the-art CMAF-CTE setup.

The setup for these tests took a camera feed coming from the THEO office in Belgium, being encoded on-site into a 720p@24fps signal (using ffmpeg), and transported to the AWS Ireland hub for a packaging step and served back from an origin instance back to the THEO offices, where the latency is measured. For the CMAF-CTE pipeline, the Github Streamline Project's Low Latency Preview [1] was used, which uses a Go packager and low-latency optimized dash.js player.



Comparing end-to-end latency

In order to compare latency achievements of the HESP protocol, we designed a test scenario where HESP is compared with CMAF-CTE, using the same encoding settings, and a selected set of chunk and segment sizes for the CMAF setup (meaning a changing GOP size).

Both encoded signals are sent over RTMP to a packager on AWS. Both signals are delivered into players running on Chrome version 73. The camera was pointed towards a millisecond accurate clock, for which the latency was compared with the video displayed in the player.

Following test setup/settings were compared:

1. HESP
2. CMAF-CTE with 1s segments and 1 frame chunks
3. CMAF-CTE with 2s segments and 5 frame chunks
4. CMAF-CTE with 6s segments and 5 frame chunks

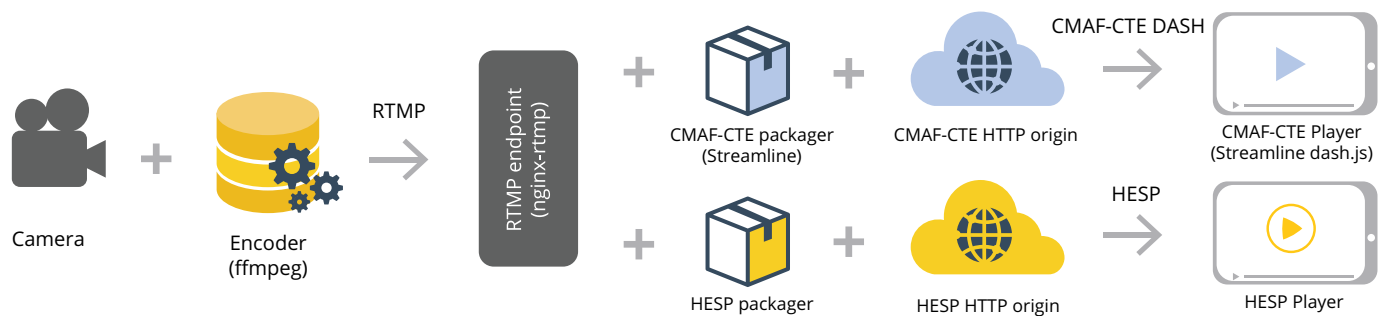
The following results were measured.

	Latency	Bandwidth	Zapping & ABR Switching
HESP	330ms	209MB	94ms
	~ 7 times less	~ up to 20% less	~ up to 20 times faster
CMAF-CTE 1s/1f	2,330ms	253MB +20.80%	1,950ms
CMAF-CTE 2s/5f	2,345ms	249MB +19.21%	1,961ms
CMAF-CTE 6s/5f	2,385ms	247MB +17.98%	2,202ms

As the results of this test indicate, end-to-end latency for the HESP protocol is significantly lower compared to all CMAF-CTE configurations which were tested. The latency measured of 330ms is low enough to enable new interactive formats, where for example viewers can directly interact with the TV broadcaster.

Comparing bandwidth usage with optimal latency

As a second test, we compared the streaming bandwidth between the HESP and CMAF-CTE streams from the previous test setup. Bandwidths were measured when delivered to the client by measuring the total amount of data transferred over a period of 10 minutes for a live channel distributing a few different types of content. All assets were encoded in the same way, the only variance being GOP sizes.



HESP vs CMAF-CTE DASH

The test assets which were looped within the live channels were three videos from the Xiph collection:

1. Big Buck Bunny: The slow moving movie we all love.
2. Elephants Dream: Relatively fast moving movie asset.
3. Meridian: The Netflix test asset designed to test encoders.

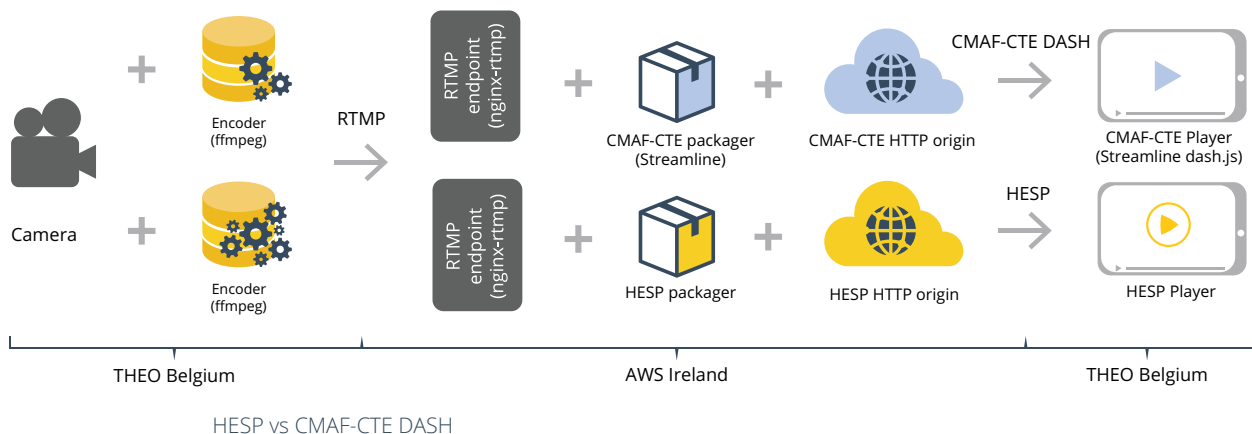
When comparing bandwidth used after 10 minutes, we found the following results. The average latencies were the same as in the previous test.

Video asset	HESP (0.33ms latency)	CMAF-CTE 1/1 (2.33s latency)	CMAF-CTE 2/5 (2.35s latency)	CMAF-CTE 6/5 (2.39s latency)
Big Buck Bunny	313.41MB	344.55MB (+9.94%)	331.87MB (+5.59%)	318.44MB (+1.60%)
Elephants Dream	306.68MB	317.63MB (+3.51%)	310.27MB (+1.17%)	308.53MB (+0.60%)
Meridian	214.27MB	253.14MB (+18.14%)	249.83MB (+16.60%)	247.24MB (+15.39%)

Based on these results, we can see that the HESP protocol, with its optimized container format is able to achieve ultra-low latency without compromising on bandwidth. In fact, HESP achieves a lower latency than CMAF-CTE while at the same time achieving a lower bandwidth utilization.

Comparing bandwidth usage at the same latency

In most use-cases, achieving ultra low latency is not a requirement. For example, most TV services and formats do not require their latency to be lower than the current broadcast service latency which is around 4 to 5 seconds. In order to measure gains achieved by trading latency for bandwidth, we designed a test setup where the encoding parameters were modified to achieve a 2.3s end to end latency for HESP aligned with the lowest latency measured for DASH CMAF-CTE. The remainder of the setup was kept the same as the previous setup.



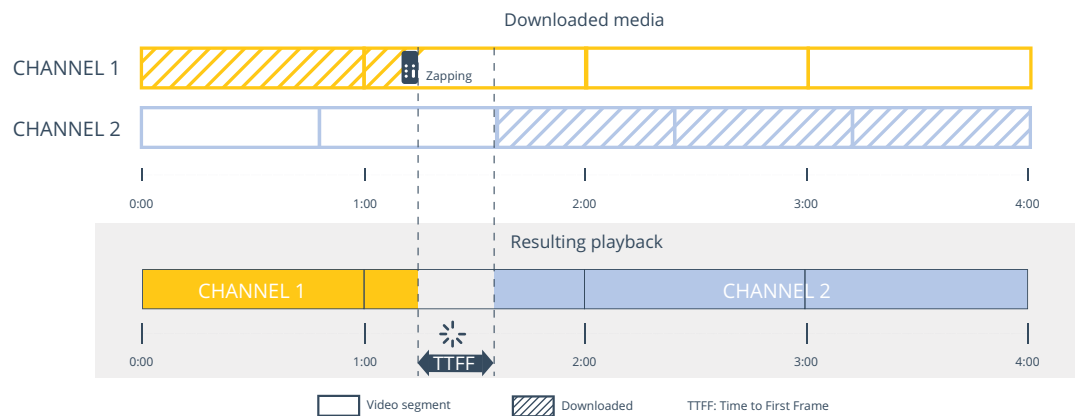
The results measured were:

Video asset	HESP (2.30ms latency)	CMAF-CTE 1/1 (2.33s latency)	CMAF-CTE 2/5 (2.35s latency)	CMAF-CTE 6/5 (2.39s latency)
Big Buck Bunny	297.18MB	344.55MB (+15.94%)	331.87MB (+11.67%)	318.44MB (+7.15%)
Elephants Dream	300.05MB	317.63MB (+5.85%)	310.27MB (+3.41%)	308.53MB (+2.83%)
Meridian	209.56MB	253.14MB (+20.80%)	249.83MB (+19.21%)	247.24MB (+17.98%)

When comparing these results with the results from the previous test, we can see the additional time spent in compressing the video allows to increase the bandwidth savings. The percentage in bandwidth saved, can of course map one-to-one with cost reduction for example the CDN egress. Depending on the content being broadcasted (ideal GOP-length), the bandwidth saving can become quite significant, averaging between 10-15%.

Comparing zapping times

As a next test setup, we compared zapping times between HESP and the different CMAF-CTE configurations used in the earlier tests. The time measured was the time between clicking the “zap” button and the first frame showing up on the screen, referred to as Time to First Frame (TTFF). The setup from the first latency comparison test was used in order to make the measurements.



Time to First Frame: Measuring the zapping time

Scenario	Average TTFF	Minimum TTFF	Maximum TTFF
1) HESP	94ms	79ms	111ms
2) CMAF-CTE 1/1	1950ms	1592ms	2394ms
3) CMAF-CTE 2/5	1961ms	1531ms	2548ms
4) CMAF-CTE 6/5	2202ms	1721ms	2870ms

Based on these results, we can see zapping times for HESP streams are significantly lower compared to the CMAF-CTE setup, reaching values below 100ms on average. Compared even to current digital TV broadcast the zapping with HESP feels instantaneous.

Leveraging CDNs with HESP

As mentioned earlier, the HESP protocol was designed to be interoperable with 3rd party CDNs. The protocol has been designed to require only HTTP/1.1 features and capabilities, similar to CMAF-CTE. The HTTP capabilities required in order to be able to use HESP in combination with a CDN are:

- ▶ Support for HTTP chunked transfer encoding
- ▶ Support for HTTP range requests

Additionally, it is recommended CDN edges also cache and aggregate ongoing chunked requests. This means when two requests need to deliver the same (part) of an asset, they only request it to the origin or underlying CDN tier once.

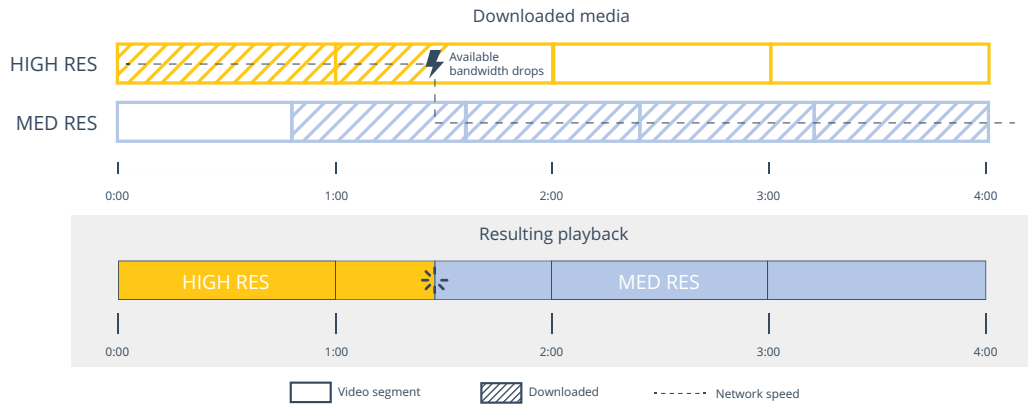
HESP has been tested with 3rd party CDNs. Depending on the CDN, the additional latency introduced by a CDN varied around 50ms, keeping the total end-to-end latency well below 1 second.

Adaptive bitrate switching with HESP

In order to adapt to varying viewer network conditions, the HESP protocol has been designed to support adaptive bitrate switching, and allows immediate switching to alternative renditions. HESP players are capable to measure the used and available bandwidth within the client, by fingerprinting the current network traffic. Based on this information, as well as other viewer environment metadata such as device orientation or resolution. The client on the device is able to switch to a rendition which is better suited for the current environment.

When using the HESP protocol, an ABR switch can be made immediately and does not need to be timed with certain intervals. This is in contrast with HAS protocols as well as CMAF-CTE, where switches to alternative renditions can only be made on the edge of a segment. While this is less of a problem when the choice would be to switch towards a higher quality (and higher bitrate) rendition due to a

bandwidth increase, it's crucial to immediately switch to a lower bitrate rendition in case of a bandwidth reduction in order to avoid player buffer underruns. Especially when targeting low-latency playback we want to keep the playback buffer as small as possible.




Protocol flow – ABR with HESP

As HESP is able to switch renditions immediately, this significantly reduces the risk of stalling and buffering behaviour.

Cross platform delivery

As a cross platform player vendor, THEO Technologies also made certain to design the HESP protocol to be able to reach any platform and device. This includes platforms which are left out by some other streaming protocols such as iOS devices. In order to achieve this, both native and HTML5 based HESP players were developed. Following platforms and devices are supported:

Browsers (across Windows, macOS, Android, iOS and Linux)	Native environments
 Chrome  Firefox  Edge  Safari (including mobile Safari on iOS)	 Android  iOS  androidtv (on Smart-TV and specific STBs)

Summary

As outlined and shown by the test results above, the HESP protocol has a number of interesting properties:

1 | Sub second latency

2 | Bandwidth reduction

3 | An ultra fast zapping time of only 100ms.

4 | Scalability across standard HTTP CDNs.

5 | Instant ABR switching capabilities.

6 | Easy integration in existing streaming architectures.

These properties enable streaming service providers to improve their business case in multiple ways:

1 | Setting up interactive streaming experiences where viewers can interact with the content being streamed and interact near real-time.

2 | Increase user engagement by providing instant video startup when zapping between channels.

3 | Reduce bandwidth costs of existing solutions across platforms without the need to switch to alternative encoders.

4 | Allow to scale a low latency streaming solution over HTTP to reach all popular platforms and devices, including iOS.

5 | Deliver a low latency stream which can dynamically adapt to viewer environments, switching between ABR qualities immediately.

6 | Reduce latency for time sensitive content, avoiding spoiled experiences for their viewers.



Discover what THEO can do for you.

Would you like to become part of the HESP journey?
Contact one of our HESP experts



PIETER-JAN SPEELMANS
Founder & CTO



BART VAN OOSTERHOUT
HESP Program Director



JOHAN VOUNCKX
VP Innovation



www.theoplayer.com

<https://www.theoplayer.com/contact/hesp-consultation>

LEUVEN (HQ)

NEW YORK

SAN FRANCISCO

SINGAPORE

BARCELONA