

# Common OTT problems and how to fix them

Jaron Viëtor

 MistServer

2019-09-14

# Who? / What?

Hi! I'm Jaron. I'm the CTO of DDVTech, the team behind MistServer.

# Who? / What?

Hi! I'm Jaron. I'm the CTO of DDVTech, the team behind MistServer.  
Of course, MistServer fixes all common OTT (internet-based streaming) problems!  
But you might want to know how you'd do so without us.

# Who? / What?

Hi! I'm Jaron. I'm the CTO of DDVTech, the team behind MistServer.  
Of course, MistServer fixes all common OTT (internet-based streaming) problems!  
But you might want to know how you'd do so without us.  
Presenting: OTT De-Mist-ified!

# Who? / What?

Hi! I'm Jaron. I'm the CTO of DDVTech, the team behind MistServer.  
Of course, MistServer fixes all common OTT (internet-based streaming) problems!  
But you might want to know how you'd do so without us.  
Presenting: OTT De-Mist-ified!

- A little bit of OTT history
- Protocol selection
- Codec selection
- Segmenting
- Scaling

# OTT history

At first, internet streaming was mostly audio-only. People used modems over phone lines, and their bandwidth tended to be about enough to carry on a conversation or so. (Gee, I wonder why...)

# OTT history

At first, internet streaming was mostly audio-only. People used modems over phone lines, and their bandwidth tended to be about enough to carry on a conversation or so. (Gee, I wonder why...)

Then came what we now call VoD: pre-recorded video. Generally downloads to disk that then played out-of-browser.

# OTT history

At first, internet streaming was mostly audio-only. People used modems over phone lines, and their bandwidth tended to be about enough to carry on a conversation or so. (Gee, I wonder why...)

Then came what we now call VoD: pre-recorded video. Generally downloads to disk that then played out-of-browser.

IP cameras: usually RTSP-based (most still are today)



# OTT history

At first, internet streaming was mostly audio-only. People used modems over phone lines, and their bandwidth tended to be about enough to carry on a conversation or so. (Gee, I wonder why...)

Then came what we now call VoD: pre-recorded video. Generally downloads to disk that then played out-of-browser.

IP cameras: usually RTSP-based (most still are today)

Flash came along and like magic we suddenly had vector animation and progressive streaming!

# OTT history

At first, internet streaming was mostly audio-only. People used modems over phone lines, and their bandwidth tended to be about enough to carry on a conversation or so. (Gee, I wonder why...)

Then came what we now call VoD: pre-recorded video. Generally downloads to disk that then played out-of-browser.

IP cameras: usually RTSP-based (most still are today)

Flash came along and like magic we suddenly had vector animation and progressive streaming!

HTML5 arrived with proper media support, and Flash went away again just as quickly as it appeared

# OTT history

At first, internet streaming was mostly audio-only. People used modems over phone lines, and their bandwidth tended to be about enough to carry on a conversation or so. (Gee, I wonder why...)

Then came what we now call VoD: pre-recorded video. Generally downloads to disk that then played out-of-browser.

IP cameras: usually RTSP-based (most still are today)

Flash came along and like magic we suddenly had vector animation and progressive streaming!

HTML5 arrived with proper media support, and Flash went away again just as quickly as it appeared

So... what should we do today?

# Protocol selection

Mostly about device compatibility.

# Protocol selection

Mostly about device compatibility.

Basically boils down to:

- iOS: HLS or WebRTC for live, MP4 for VoD
- Most other consumer devices: MP4, WebM, WebRTC, HLS/DASH/CMAF
- Special cases: RTSP, RTMP, TS... and/or some of the above.

# Protocol selection

Mostly about device compatibility.

Basically boils down to:

- iOS: HLS or WebRTC for live, MP4 for VoD
- Most other consumer devices: MP4, WebM, WebRTC, HLS/DASH/CMAF
- Special cases: RTSP, RTMP, TS... and/or some of the above.

HLS for live and MP4 for VoD works for most, but is not always ideal

# Codec selection

Also about device compatibility (hardware accelerated decoding support), but a constantly moving target.

# Codec selection

Also about device compatibility (hardware accelerated decoding support), but a constantly moving target.

H264 is most widely supported today, and when targeting consumer devices is an absolute must... for now.



# Codec selection

Also about device compatibility (hardware accelerated decoding support), but a constantly moving target.

H264 is most widely supported today, and when targeting consumer devices is an absolute must... for now.

HEVC is great for newer devices, improved quality per byte of data. Patent pool nightmare.

# Codec selection

Also about device compatibility (hardware accelerated decoding support), but a constantly moving target.

H264 is most widely supported today, and when targeting consumer devices is an absolute must... for now.

HEVC is great for newer devices, improved quality per byte of data. Patent pool nightmare.

AV1 is the (near) future. I hope..!

# Segmenting

Relevant for HLS/DASH/CMAF: those work with small segments of media data that together form the stream.

# Segmenting

Relevant for HLS/DASH/CMAF: those work with small segments of media data that together form the stream.

Affects start-up time, latency, stability.

Start-up time: smaller segments load faster and decrease start-up time

Latency: smaller segments mean the client-side buffer is smaller, and thus latency is lower  
but...

Stability: longer segments are more stable and have less overhead

# Segmenting

Relevant for HLS/DASH/CMAF: those work with small segments of media data that together form the stream.

Affects start-up time, latency, stability.

Start-up time: smaller segments load faster and decrease start-up time

Latency: smaller segments mean the client-side buffer is smaller, and thus latency is lower  
but...

Stability: longer segments are more stable and have less overhead

Key is knowing your market and where your priorities in the above aspects are

# Scaling

Scaling can mean delivering at scale (many viewers and/or many broadcasts), or scaling up/down as demand changes (flexibility).

# Scaling

Scaling can mean delivering at scale (many viewers and/or many broadcasts), or scaling up/down as demand changes (flexibility).

Can be solved by (partially) using cloud capacity, CDNs, peer to peer, adding/removing servers as-needed, doing load balancing, and/or buying more bandwidth... or a combination of any of these.

# Scaling

Scaling can mean delivering at scale (many viewers and/or many broadcasts), or scaling up/down as demand changes (flexibility).

Can be solved by (partially) using cloud capacity, CDNs, peer to peer, adding/removing servers as-needed, doing load balancing, and/or buying more bandwidth... or a combination of any of these.

Good statistics/metrics on your system usage are key to solving this properly!



## Let us help! / Questions?

MistServer provides ready-made solutions for all the problems discussed, so you don't have to reinvent the wheel.

If you are facing some other problem, we can probably help as well!

# Let us help! / Questions?

MistServer provides ready-made solutions for all the problems discussed, so you don't have to reinvent the wheel.

If you are facing some other problem, we can probably help as well!

Feel free to ask questions now,  
or drop by our booth (H14C10, right over ← there!)  
or shoot us an e-mail ([info@mistserver.org](mailto:info@mistserver.org))!

## Let us help! / Questions?

MistServer provides ready-made solutions for all the problems discussed, so you don't have to reinvent the wheel.

If you are facing some other problem, we can probably help as well!

Feel free to ask questions now,  
or drop by our booth (H14C10, right over ← there!)  
or shoot us an e-mail ([info@mistserver.org](mailto:info@mistserver.org))!

Back to you, Ian!