# Inferring Video Streaming Quality of Experience at scale using incremental statistics from CDN Logs

Hugo Girault, Loïck Bonniot, **Christoph Neumann**

February 11th, 2024

mhv/2024

broadpeak

This is streaming at its peak

# 01. Motivation

Content service providers need to monitor Quality of Experience

- startup duration, stalls?
- layer changes, video bitrate?
- …

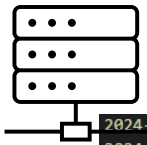

Usage of player plugins or SDKs

- Not always possible: legacy devices/players, cost of integration…

*How to estimate Quality of Experience without player metrics?*
*At scale?*

# 01. How to estimate QoE without player metrics?

Can we exploit CDN cache server logs to infer QoE?



startup duration, stalls,
layer changes, video bitrate
…

```
2024-01-03T14:47:21+01:00|138.14.8.8.GET|/bpk-tv/bbb/default/index.mpd||-|200|14.32.1.103:443|20 EXPIRED|0.046|0.046|3825|-
2024-01-03T14:47:21+01:00|138.14.8.8.GET|/bpk-tv/bbb/default/dash/bbb-audio=128000-7515918226480 .dash|-|-|200|14.32.1.31.443|200|MISS|0.100|0.049|63873|-
2024-01-03T14:47:21+01:00|138.14.8.8.GET|/bpk-tv/bbb/default/dash/bbb-video=1194000-1533860864 5.dash|-|-|200|14.32.1.12:443|200|MISS|0.150|0.048|293366|-
2024-01-03T14:47:23+01:00|138.14.8.8.GET|/bpk-tv/bbb/default/index.mpd||-|200|14.32.1.66:443|206|EXPIRED|0.044|0.044|3810|-
2024-01-03T14:47:23+01:00|138.14.8.8.GET|/bpk-tv/bbb/default/dash/bbb-video=1194000-153386086614075.dash|-|-|200|14.32.1.31.443|200|MISS|0.132|0.050|211400|-
```

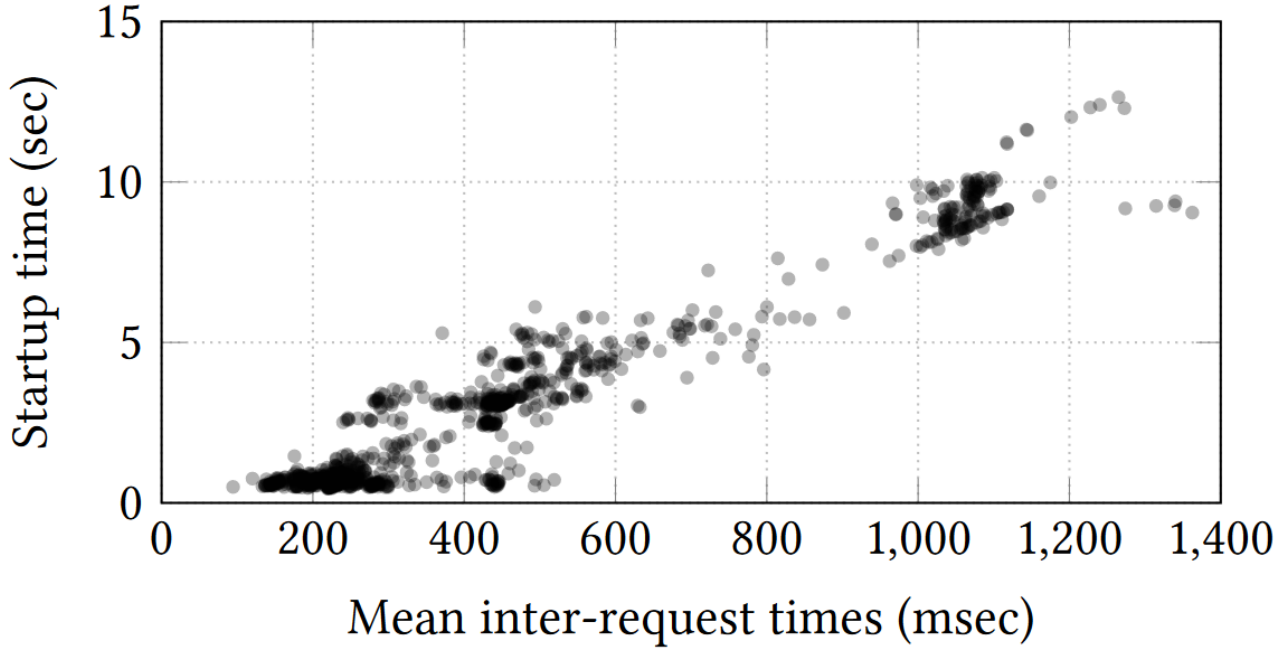
- One line per HTTP request: timings, payload size, response codes, hit/miss
- An individual log line does not provide QoE information

*Reasoning about QoE becomes possible with a session identifier (e.g., CMCD sid)*

➔ *calculate timeseries, patterns, distributions within a streaming session*

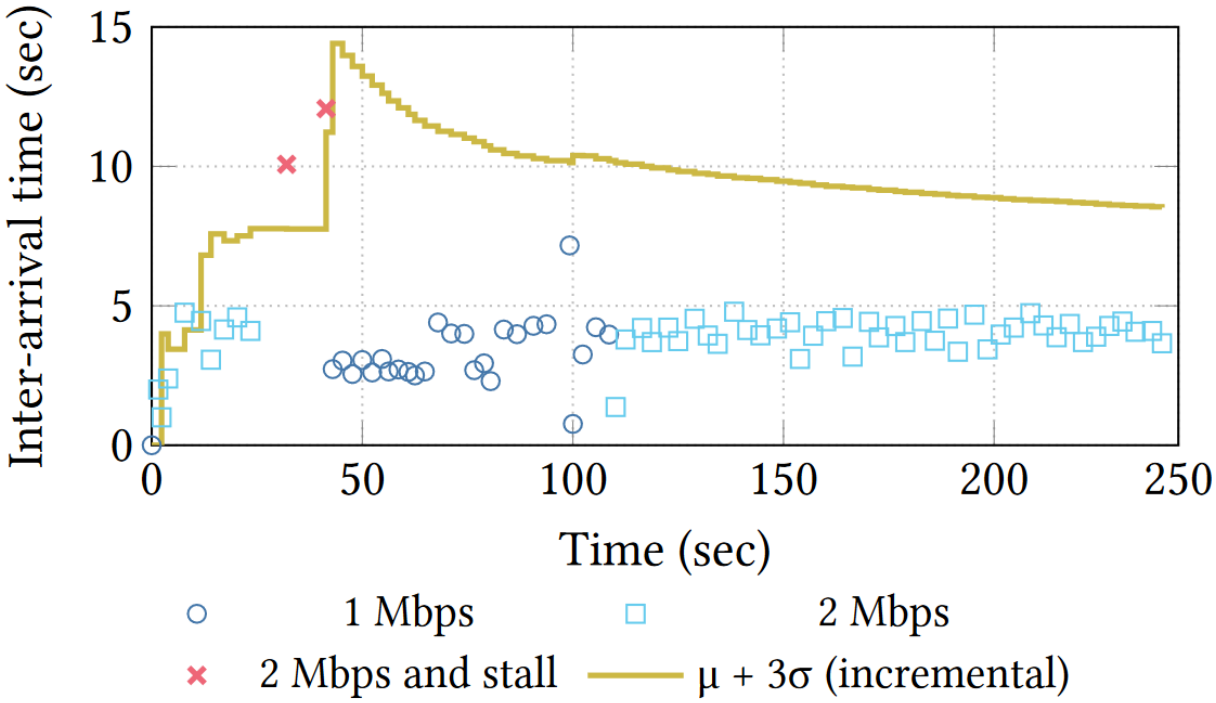
# 02. Can we exploit CDN cache server logs?

Example: inter-arrival times of ten first requests versus startup time for a given video player

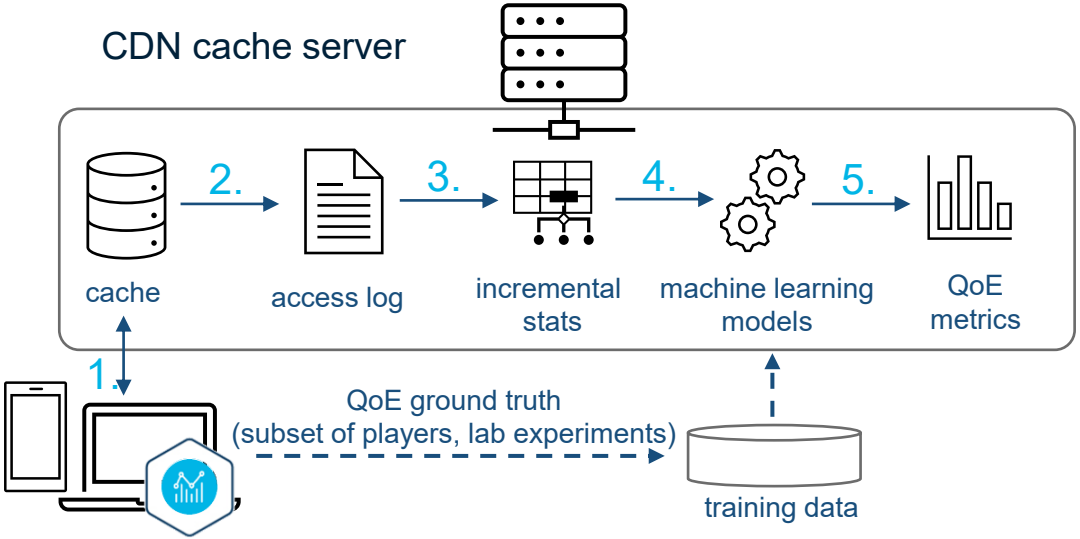


# 02. Can we exploit CDN cache server logs?

Example: request inter-arrival times versus stalls



# 03. Inferring QoE from access logs




1. Request content
2. Write to access log; annotate with session identifiers
3. Calculate *incremental* statistics: means, approximate quantile sketches, counts…
4. At the end of streaming session: feed calculated values to machine learning models
5. Output QoE metrics

# 04. Implementation

Implementation geared towards performance

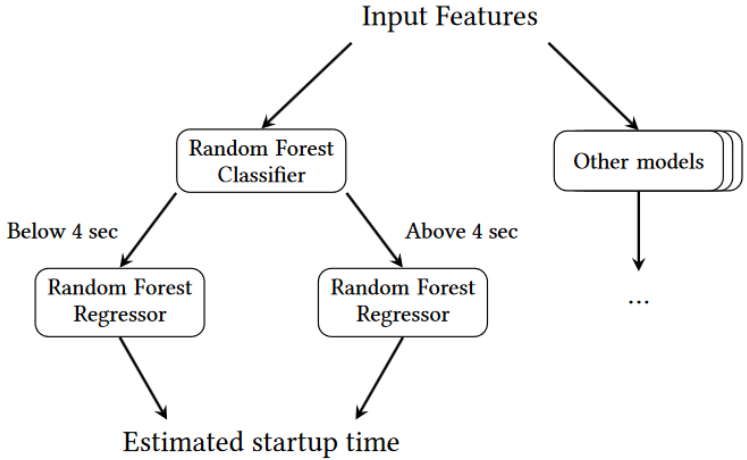
- Relying on Nginx
- Machine learning models run on top of ONNX
- Constant memory accumulator for streaming session statistics

Able to process several 10s of thousand of sessions in parallel

Inferred QoE metrics

- Startup duration
- Average video layer
- Layer switches
- Stalls count and duration

One model per metric



# 04. Evaluation

Training and validation dataset using dedicated testbed

- Large panel of network conditions (production datasets tend to be biased)
- Generated around 10 000 streaming sessions
- DASH.js, HLS.js, Shaka, THEO

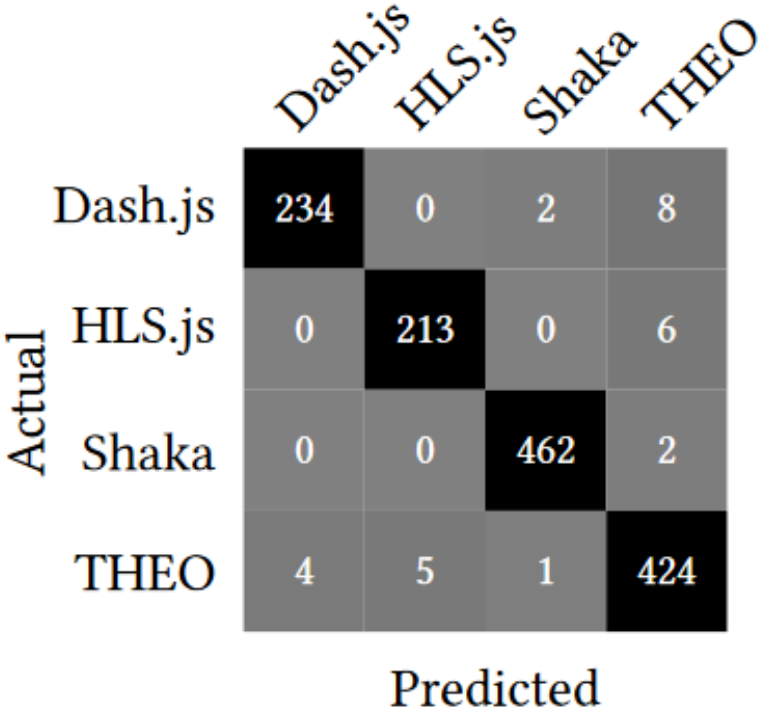
Comparison against Shah et al. [1]

| QoE metric | Baseline MAE | Baseline $R^2$ | Our approach MAE | Our approach $R^2$ |
|---|---|---|---|---|
| Startup time (sec) | N/A | | **0.94** | **0.89** |
| Average layer (kbps) | 337 | 0.83 | **210** | **0.89** |
| Layer switches (count) | 4.5 | -0.02 | **1.7** | **0.90** |
| Stalls (count) | 4.2 | 0.23 | **1.51** | **0.51** |
| Stalls duration (sec) | 25 | -0.05 | **8.3** | **0.72** |

[1] Anant Shah, Juan Bran, Kyriakos Zarifis, and Harkeerat Bedi. 2022. SSQoE: Measuring Video QoE from the Server-side at a Global Multi-tenant CDN. In PAM.

# 04. Evaluation

## Player identification



# 05. Conclusion

Collecting QoE information directly from video player is not always possible

Proposed to infer QoE from cache server logs

- By aggregating statistics for each streaming session
- By relying on a separate machine learning model for each QoE metric

Showed accurate inference of QoE metrics

Implementation choices guided by performance

- Relying on incremental constant memory statistics
- Able to process several 10s of thousand of sessions in parallel

# Thank you!

broadpeak.tv