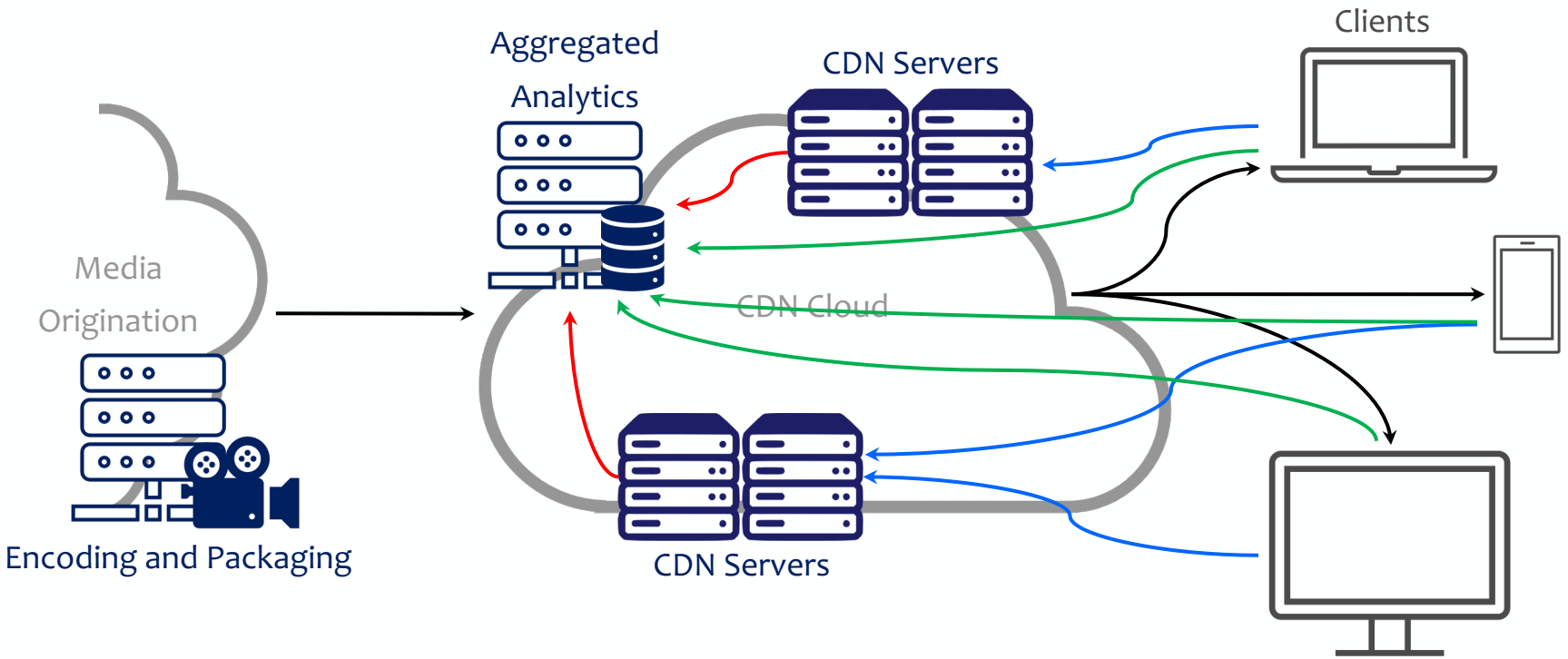
# Use of CMCD in HTTP Adaptive Streaming: Initial Findings

*(accepted for ACM NOSSDAV 2021)*

DASH-IF Special Session – Apr. 9th, 2021

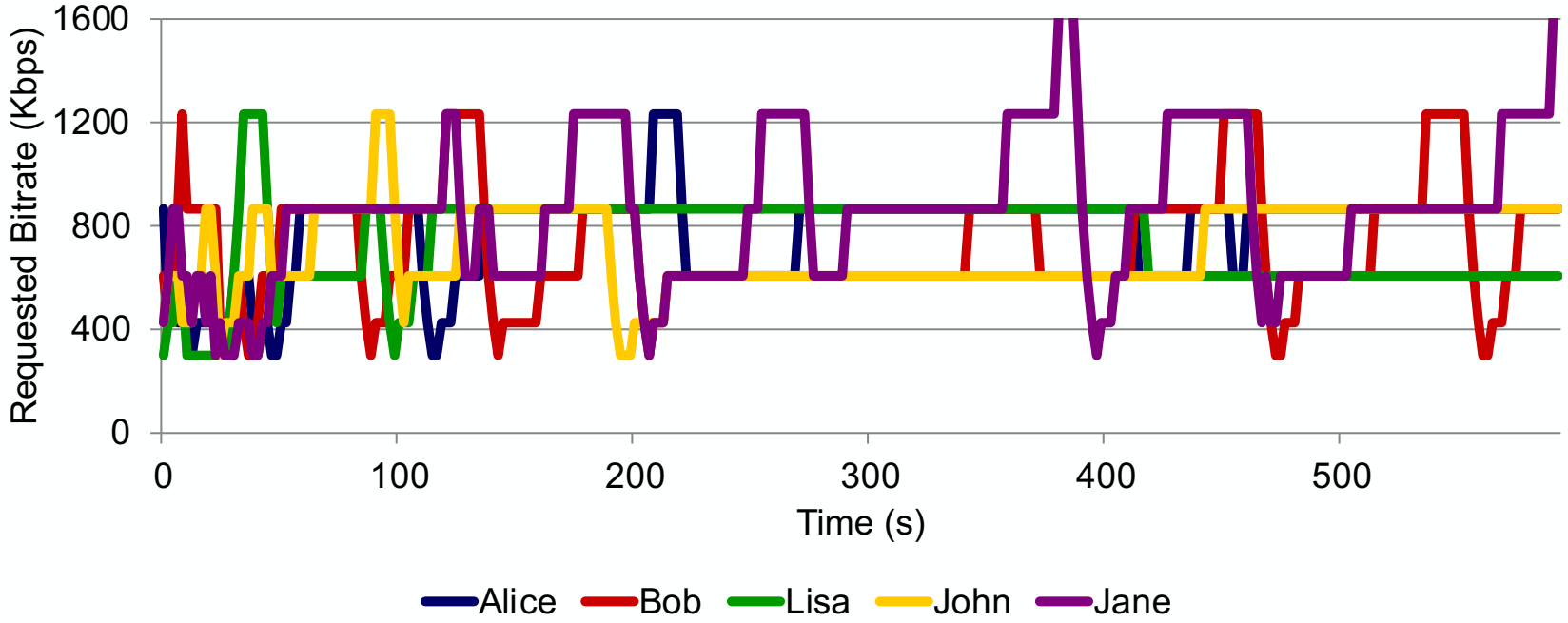
**Abdelhak Bentaleb, May Lim, Mehmet N. Akcay, Ali C. Begen and Roger Zimmermann**

# CMCD: The Baba Yaga of Streaming



# Bad Things Happens When Streaming Clients Compete

## 10 (Identical) Streaming Clients Sharing a 10 Mbps Link




Reading: "What happens when HTTP adaptive streaming players compete for bandwidth?," ACM NOSSDAV 2012

# Clients Compete on Home and ISP Networks

- Observations
  - Sustained starvation, consistent unfairness, frequent bitrate switches can occur
  - Things usually get worse with diverse clients
  - Rebufferings are bad
  - Infeasible to fix it purely on the client side
- Proposition
  - Server-side throttling can help
- Plan of action
  - Clients
    - Perform ABR as usual
    - Use CMCD to send real-time *info*
  - Server
    - Run a lightweight throttling scheme

Information exchange is useful when it is relevant, actionable and up-to-date

But what information is relevant and actionable?

Reading: "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," ACM NOSSDAV 2013

# Key Contributions

Investigate the feasibility of the newly rectified CTA-5004

Demonstrate benefits in a multi-client shared-network scenario

Implement a PoC conforming to the spec

Publish the code to allow others to perform further investigations

# Let's Use Query Arguments

| Parameter | Key | Type | Unit |
|---|---|---|---|
| … | | | |
| **Buffer length** | bl | integer | ms |
| … | | | |
| **Buffer starvation** | bs | boolean | - |
| … | | | |
| **Max buffer** | bmx | integer | ms |
| **Min buffer** | bmn | integer | ms |

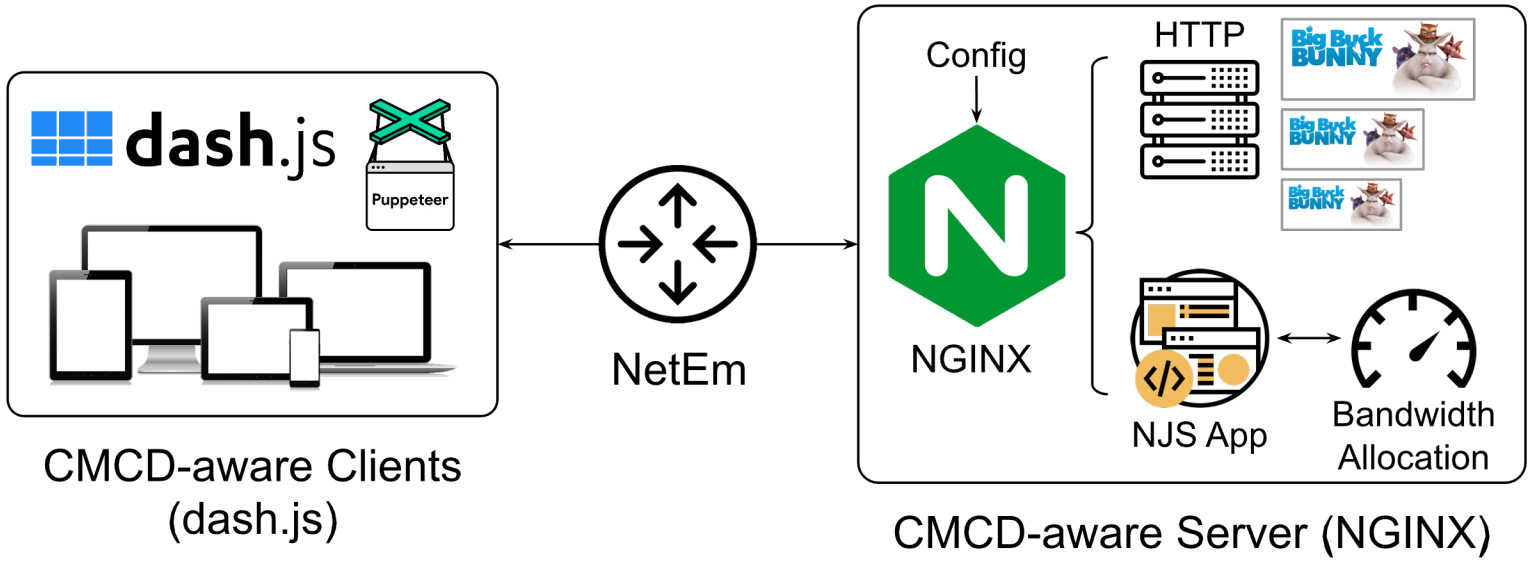
- Max buffer
  - Client cannot take more than this
- Min buffer
  - Below this, rebuffering is imminent
- Example query string

  **?CMCD=bl=4500,bmx=30000,bmn=500**

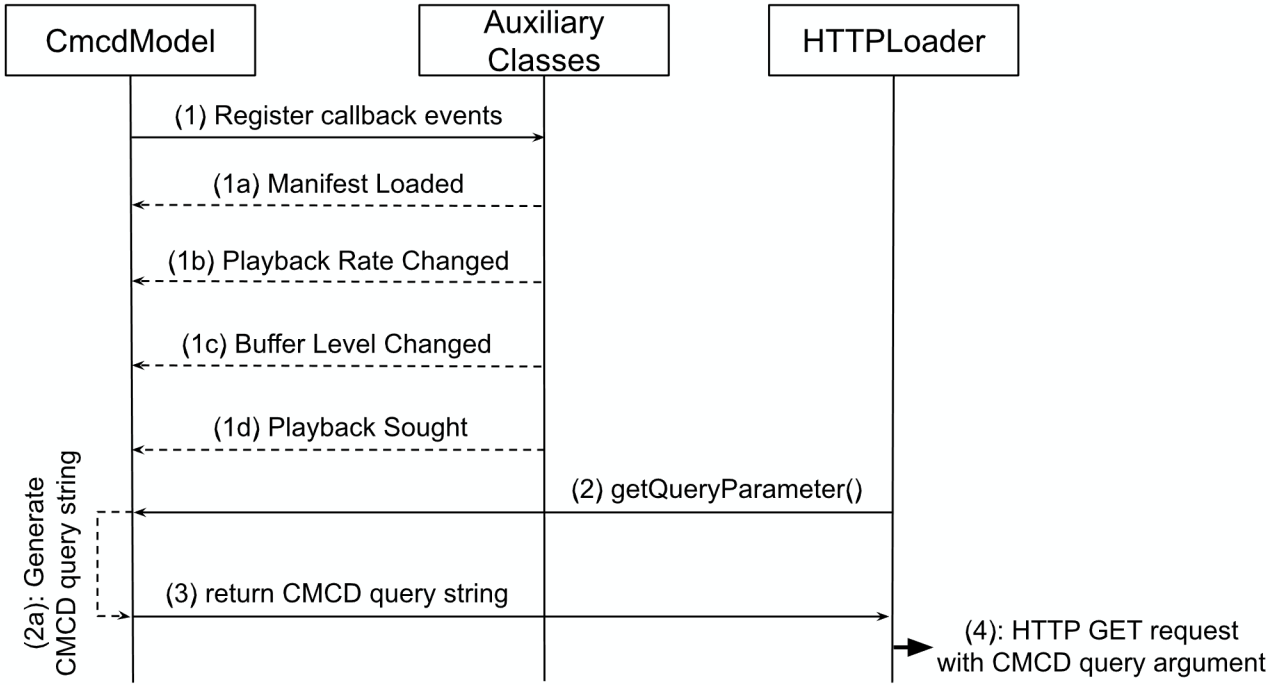
  which is encoded as

  **?CMCD=bl%3D4500%2Cbmx%3D30000%2Cbmn%3D500**

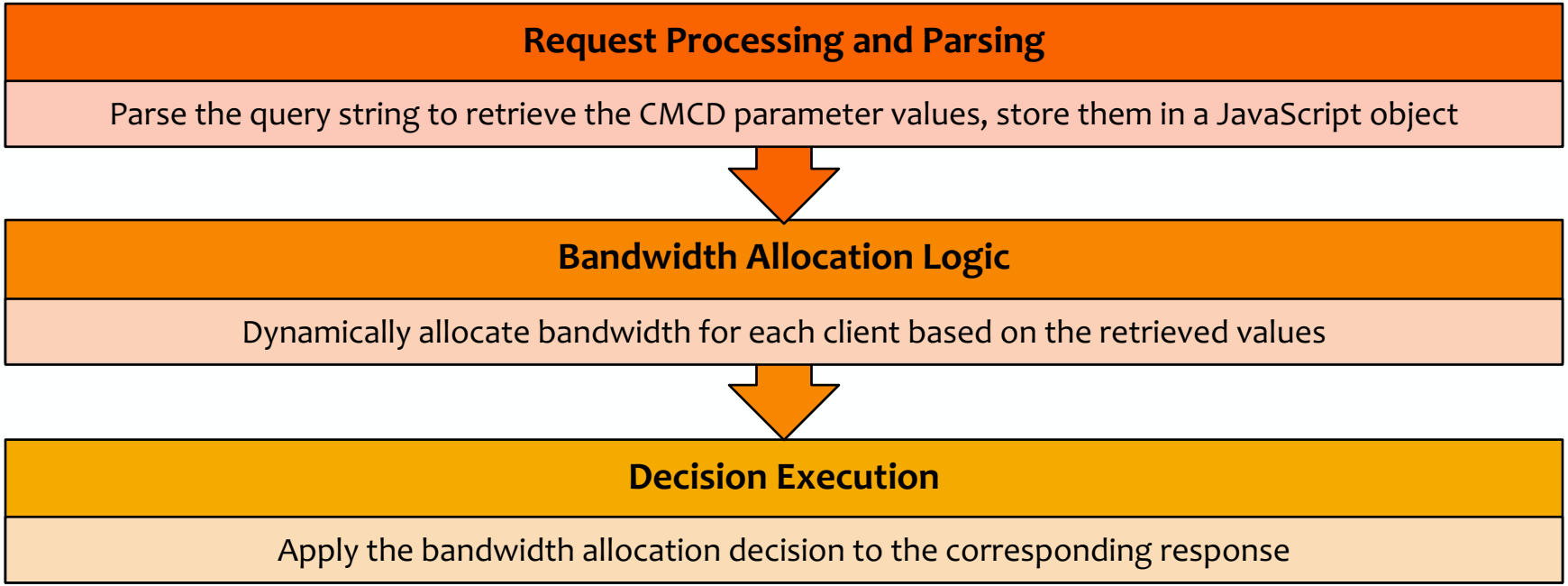
# CMCD-Aware System



# CMCD-Aware Client (dash.js)



# CMCD-Aware Server (NJS Application)



```
function bufferAwareBandwidthAllocation(req) {
    var cmcd_params = processQueryArgs(req);
    var r = 0;
    var C = getAvailibleTotalCapacity();
    if (!('bl' in cmcd_params) || !('bmn' in
         cmcd_params) || !('bmx' in cmcd_params) || !(
         'ot' in cmcd_params)) {
        return 0;  /* Disable bandwidth allocation */
    }
    if (cmcd_params['ot'] != 'v' && cmcd_params['ot']
          != 'av') { /* not video object */
        return 0;  /* Disable bandwidth allocation */
    }
    /* Buffer-to-rate mapping */
    var Cmin = C x (1 - α);
    var Cmax = C x α;
    var Bmin = cmcd_params['bmn'];
    var Bmax = cmcd_params['bmx'];
    var Bufferlength = cmcd_params['bl'];
    /* Case S1 */
    if (Bufferlength < Bmin || ('bs' in cmcd_params)){
        r = Cmax;
    }
    /* Case S2 */
    else if (Bufferlength > Bmax) {
        r = Cmin;
    }
    /* Case S3 */
    else {
        var Brange =  Bmax - Bmin;
        var Crange = Cmax - Cmin;
        r = ((1 - ((Bufferlength - Bmin) / Brange)) *
             Crange) + Cmin;
    }
    return r;
}
```

# Buffer-Aware Bandwidth Allocation

- S1 (buffer underflow)

  $r = C_{max}$

- S2 (buffer overflow)

  $r = C_{min}$

- S3 (buffer safe)

  $r = C_{min} + [(1 - ((bl - B_{min})/B_{range})) \times C_{range}]$ where

  $B_{range} = B_{max} - B_{min}$

  $C_{range} = C_{max} - C_{min}$

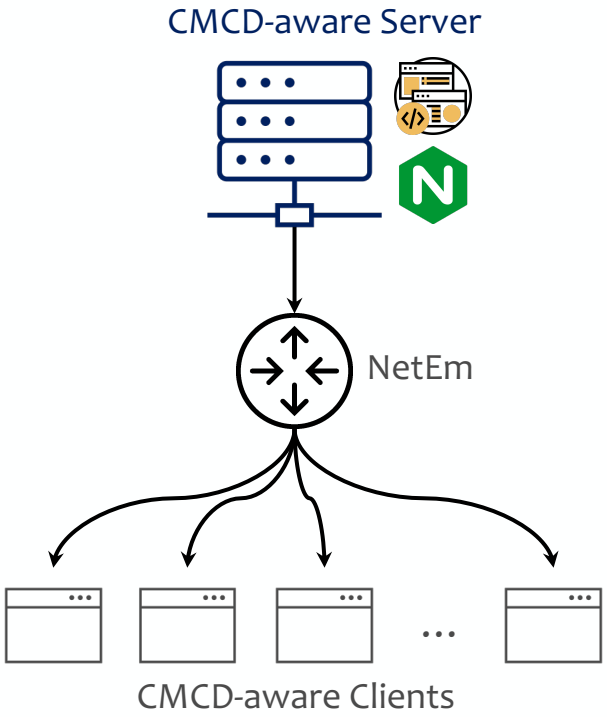
  $C_{max} = \alpha \times C$

  $C_{min} = (1 - \alpha) \times C$

  In this study, $\alpha = 0.9$

# Test Scenarios and Network Emulation

## Downstream Traffic is Throttled by tc-NetEm




Source: DASH-IF, bandwidth changes every 30 s

| Profile Name | Values (Mbps) |
|---|---|
| CascadeX5 | 50, 20, 10, 5, 10, 40 |
| SpikeX5 | 50, 10 |
| CascaseX10 | 100, 40, 20, 10, 20, 40 |
| SpikeX10 | 100, 20 |
| CascadeX20 | 200, 80, 40, 20, 40, 80 |
| SpikeX20 | 200, 40 |
| CascaseX30 | 300, 120, 60, 30, 60, 120 |
| SpikeX30 | 300, 60 |

# Performance Evaluation

## Setup and Configs

- Physical machine
  - Ubuntu 18.04.5 LTS with dual 20-core Intel E5-2630 v4 @ 2.20GHz processors, 192 GB memory
- Server and clients
  - NGNIX (v1.18) + NJS application
  - dash.js (v3.1.3) clients on Chrome browser (v88) with headless mode, default ABR scheme (Dynamic)
- BBB: 10 minutes long, segments of four seconds (from Akamai)
  - 180p@0.4 Mbps, 360p@0.8 Mbps, 432p@1.5 Mbps, 576p@2.5 Mbps, 720p@4.0 Mbps
- Mixed content: Four minutes long, segments of four seconds, 144p, 240p, 360p, 480p, 720p and 1080p
  - v1 (sports): 0.5, 0.7, 1.0, 1.5, 3.5, 5.0 Mbps
  - v2 (movie): 0.7, 1.0, 1.5, 2.0, 3.0, 6.0 Mbps
  - v3 (animation): 0.4, 0.7, 1.0, 1.5, 2.5, 4.0 Mbps
  - v4 (gaming): 0.2, 0.5, 0.7, 1.0, 3.5, 4.5 Mbps
- Low-latency content: BBB, chunks of one frame (30 fps) and segments of one second
  - 360p@0.2 Mbps, 480p@0.6 Mbps, 720p@1.0 Mbps

# Results and Analysis

## Metrics

| Metric | Definition |
|---|---|
| **Avg. BR** | Average bitrate across all clients (Mbps) |
| **Min. BR** | Average bitrate for the client that consumed the lowest average bitrate (Mbps) |
| **Avg. RD** | Average total rebuffering duration across all clients (s) |
| **Max. RD** | Total rebuffering duration for the client that suffered from the longest rebuffering duration (s) |
| **Avg. RC** | Average rebuffering count across all clients |
| **Avg. SC** | Average bitrate switching count across all clients |
| **Avg. LL** | Average live latency across all clients (s) |
| **Max. LL** | Average live latency for the client that experienced the longest latency (s) |

# Access Link: BBB

## 10 Concurrent Sessions, $B_{min}$= 4 s, $B_{max}$= 8 s

**CascadeX10**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 3.13 | 3.33 |
| **Min. BR** | 2.9 | 3.12 |
| **Avg. RD** | 5.36 | 20.84 |
| **Max. RD** | 10.72 | 38.84 |
| **Avg. RC** | 4.72 | 11.04 |
| **Avg. SC** | [illegible] | 36.7 |

**SpikeX10**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 2.61 | 3.2 |
| **Min. BR** | 2.3 | 2.68 |
| **Avg. RD** | 12.43 | 71.9 |
| **Max. RD** | 18.49 | 83.54 |
| **Avg. RC** | 8.68 | 25.48 |
| **Avg. SC** | [illegible] | 33.9 |

**Observation:** Results for $B_{min}$= 12 s, $B_{max}$= 24 s are similar but % improvement is smaller (since larger buffer already provides more robustness)

# Aggregation Link: BBB

## 20 Concurrent Sessions, $B_{min}$= 4 s, $B_{max}$= 8 s

**CascadeX20**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 3.20 | 3.39 |
| **Min. BR** | 2.88 | 3.03 |
| **Avg. RD** | 14.42 | 24.58 |
| **Max. RD** | 27.57 | 44.43 |
| **Avg. RC** | 9.24 | 11.05 |
| **Avg. SC** | [illegible] | 34.84 |

**SpikeX20**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 2.78 | 3.16 |
| **Min. BR** | 2.22 | 2.55 |
| **Avg. RD** | 32.14 | 61.87 |
| **Max. RD** | 51.13 | 78.13 |
| **Avg. RC** | 14.41 | 22.36 |
| **Avg. SC** | [illegible] | 30.36 |

**Observation:** The significant improvements in the rebuffering statistics have an adverse impact on Avg. BR

# Aggregation Link: BBB

## 30 Concurrent Sessions, $B_{min}$= 4 s, $B_{max}$= 8 s

**CascadeX30**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 3.32 | 3.34 |
| **Min. BR** | 2.93 | 2.97 |
| **Avg. RD** | 31.57 | 37.00 |
| **Max. RD** | 59.86 | 71.70 |
| **Avg. RC** | 14.17 | 15.13 |
| **Avg. SC** | 35.86 | 35.54 |

**SpikeX30**

| | | |
|---|---|---|
| **Avg. BR** | 3.07 | 3.11 |
| **Min. BR** | 2.33 | 2.41 |
| **Avg. RD** | 47.99 | 48.39 |
| **Max. RD** | 70.66 | 70.21 |
| **Avg. RC** | 19.03 | 19.26 |
| **Avg. SC** | 46.73 | |

**Observation:** Benefits start diminishing when the number of clients increases due to the simplicity of our approach, more sophisticated methods may be warranted

# Aggregation Link: Mixed Content

## 20 Concurrent Sessions, $B_{min}$= 4 s, $B_{max}$= 8 s

**CascadeX20**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 4.19 | 4.58 |
| **Min. BR** | 0.97 | 0.99 |
| **Avg. RD** | 2.01 | 13.5 |
| **Max. RD** | 8.69 | 36.99 |
| **Avg. RC** | 1.10 | 4.15 |
| **Avg. SC** | | 8.10 |

**SpikeX20**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 4.05 | 4.55 |
| **Min. BR** | | 0.98 |
| **Avg. RD** | 4.15 | 12.60 |
| **Max. RD** | 8.69 | 19.47 |
| **Avg. RC** | 2.50 | 4.60 |
| **Avg. SC** | | 6.15 |

**Observation:** Similar trend as the BBB but the rebuffering is less with mixed content (statistical muxing)

# Access Link: Low Latency

## Five Concurrent Sessions, $B_{min}$= 1 s, $B_{max}$= 3 s

**CascadeX5**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 0.44 | 0.21 |
| **Min. BR** | 0.37 | 0.21 |
| **Avg. RD** | 3.56 | 4.45 |
| **Max. RD** | 3.87 | [illegible] |
| **Avg. RC** | 10.50 | 16.50 |
| **Avg. SC** | 21.75 | 18.00 |
| **Avg. LL** | 2.34 | 2.75 |
| **Max. LL** | 2.91 | 3.38 |

**SpikeX5**

| | CMCD | No CMCD |
|---|---|---|
| **Avg. BR** | 0.21 | 0.20 |
| **Min. BR** | 0.20 | 0.20 |
| **Avg. RD** | 3.62 | 4.91 |
| **Max. RD** | [illegible] | 5.44 |
| **Avg. RC** | 13.25 | 16.75 |
| **Avg. SC** | 5.75 | 3.25 |
| **Avg. LL** | 2.28 | 2.31 |
| **Max. LL** | 2.30 | 3.35 |

# Thank you

- Many thanks to Daniel S. for the help and support
- CMCD-aware system
  - Download the code and test it
  - Open issues and report bugs
- Reach out to any of us for questions
  - bentaleb@comp.nus.edu.sg
  - maylim@comp.nus.edu.sg
  - necmettin.akcay@ozu.edu.tr
  - ali.begen@ozyegin.edu.tr
  - rogerz@comp.nus.edu.sg