Optimising DASH over AQM-enabled Gateways using intra-chunk parallel retrieval (chunklets)

Jonathan Kua, Grenville Armitage
Centre for Advanced Internet Architectures
Swinburne University of Technology
Melbourne, Australia

Outline

1. Introduction
2. Related work
3. Use scenario
4. Experiment settings
5. Performance metrics
6. Results & analysis
7. Conclusion

Introduction

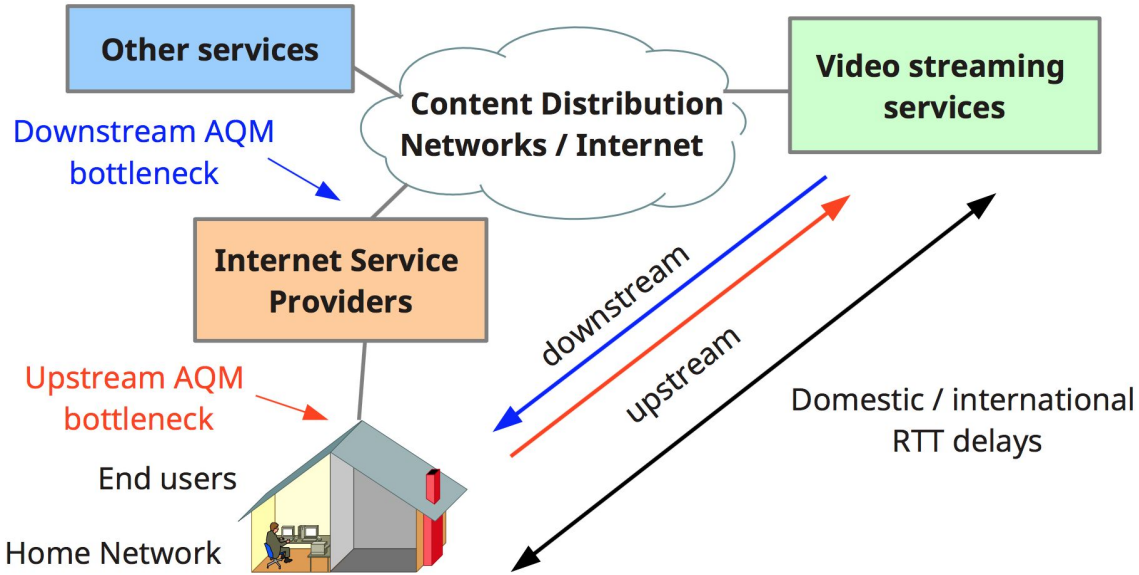
- Dynamic Adaptive Streaming over HTTP (DASH) is a recent standard for live and on-demand video streaming services
- Active Queue Management (AQM) schemes are being progressively deployed either at the ISP-end or home gateway to counter bufferbloat
- DASH adapt the video quality to match the network capacity by requesting multi-rate video by using a single TCP connection
- They propose a system using intra-chunk parallel connections (chunklets) to accelerate the DASH content transfer when bottlenecks implement AQMs
- Chunklet: different parts of a video chunk when DASH using multiple concurrent TCP connections in parallel

Related work

- DASH
  - In DASH streaming systems, video is encoded into multiple versions at different bitrates (quality)
  - Then each encoded video is segmented into small video segments (chunks), each containing a few seconds of video
- AQMs
  - It aims to keep queuing delays low, such as PIE and FQ-CoDel
  - PIE: it operates on single queue and keep queuing delays low by dropping packets when queuing delays exceed a target delay 15ms
  - FQ-CoDel: it isolates traffic flows into sub-queues (by default is 1024) then serves each sub-queue with a Deficit Round Robin (DRR) scheduler (by dropping packets every periodical time when queuing delay exceed 5ms)

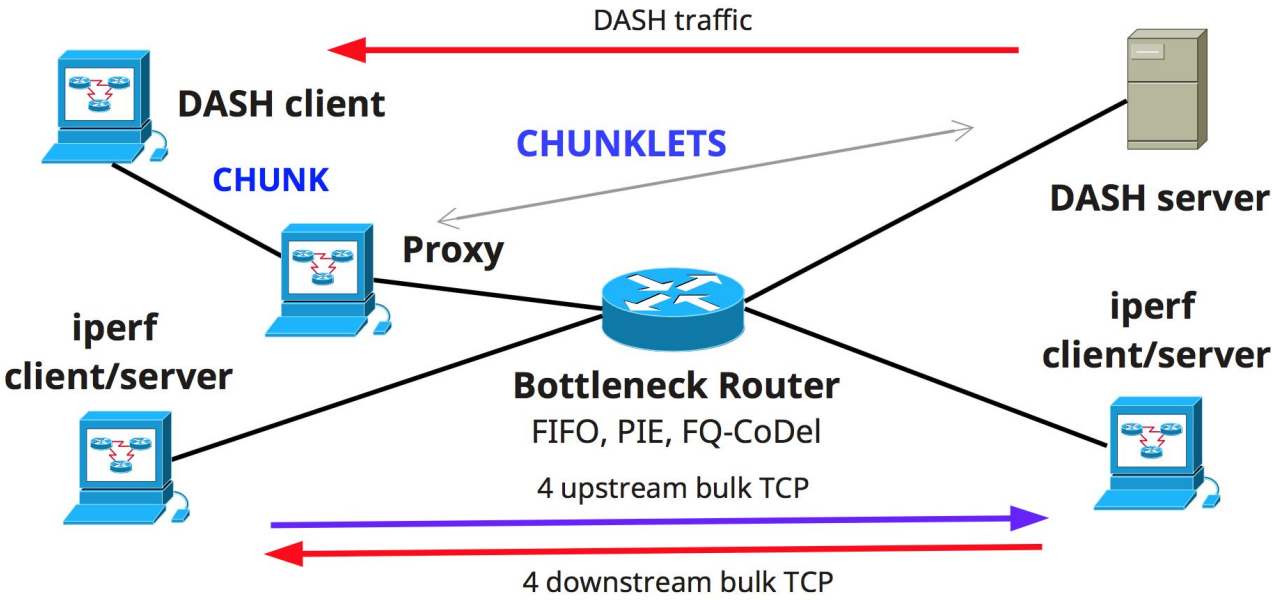
Use scenario

- A home network that is connected to local or international video streaming services and other Internet-based services
- AQMs replace traditional FIFO home gateways in modern



Experiment

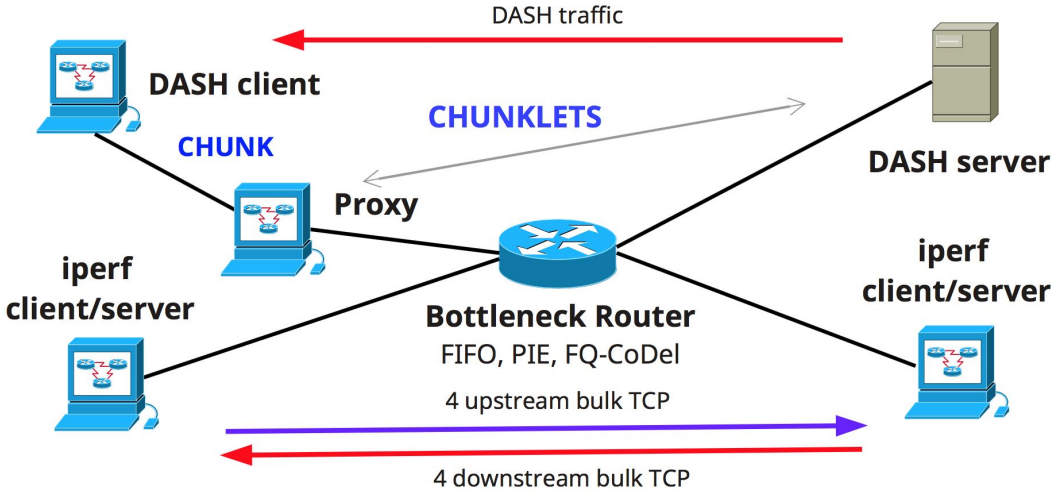
- Experiment testbed built based on TEACUP [1] software, a system for automated TCP testbed experiments



[1] http://caia.swin.edu.au/reports/150529A/

Experiment (cont.)

- Bottleneck Router: a FreeBSD's dummynet/ipfw to provide FIFO, PIE and FQ-CoDel queue management schemes
- DASH server: a lighttpd version 1.4.35 web server
- DASH client: dash.js version 2.4.17
- Proxy server: 64-bit FreeBSD



Performance metrics

TABLE I
REPRESENTATION RATES AVAILABLE IN 2-SEC DATASET

Resolution	Encoding Level	Representation Rates
320x240	1 - 3	46, 89, 131kbps
480x360	4 - 8	178, 222, 263, 334, 396kbps
854x480	9 - 10	533, 595kbps
1280x720	11 - 14	0.8, 1.0, 1.2, 1.5Mbps
1920x1080	15 - 20	2.1, 2.5, 3.1, 3.5, 3.8, 4.2Mbps

- Representation Rate (RR): video sequences encoded at different bitrates
- Smoothed Achieved Rate (SA): the average of the achieved rate from the last three chunks
- Instability index: calculate the weighted sum of the number of RR

$$\frac{\sum_{d=0}^{k-1} |b_{x,t-d} - b_{x,t-d-1}|.w(d)}{\sum_{d=1}^{k} b_{x,t-d}.w(d)}$$

  - video chunks k = 10
  - w(d) = k − d
  - b(x,t) is the encoding level retrieved at time t

Result & analysis

- It shows SA increases for all FIFO, PIE and FQ-CoDel as N increase
- N < 6, PIE has wider spread in both SA and RR
- N > 6, FQ-CoDel has wider spread

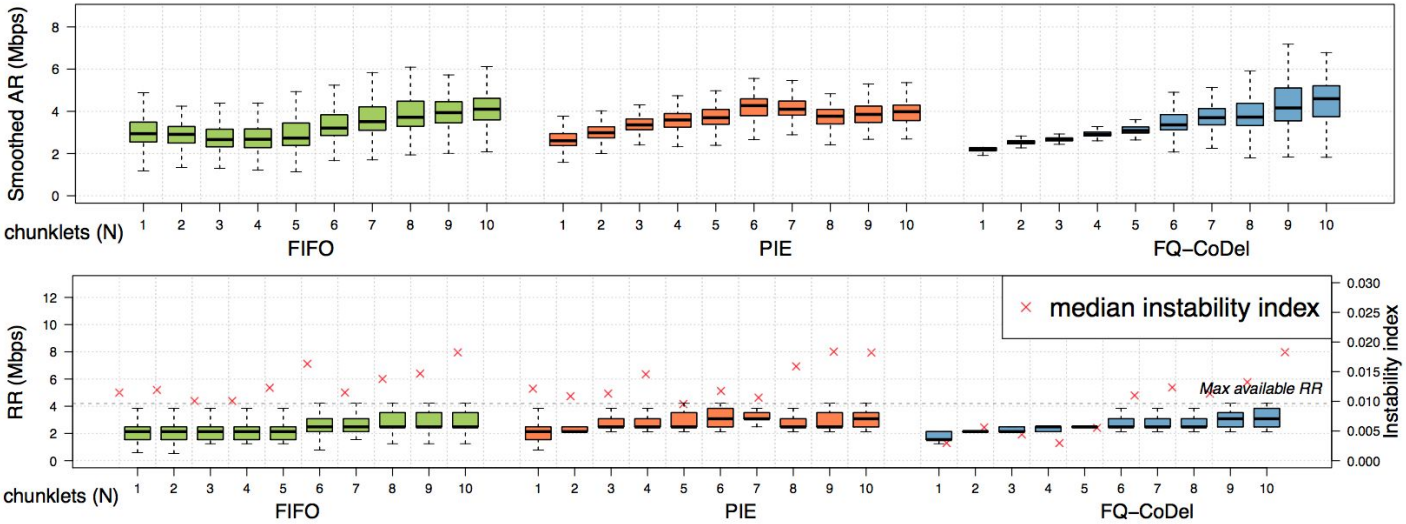


Fig. 6. Smoothed AR, RR and instability index *vs N* for FIFO, PIE and FQ-CoDel @ {12/1Mbps, 20ms RTT} path, 4 downstream elastic flows

Result & analysis (cont.)

- FIFO performs the worst
- PIE & FQ-CoDel allow low-rate ACK streams to traverse the upstream bottleneck and adjust DASH sufficiently

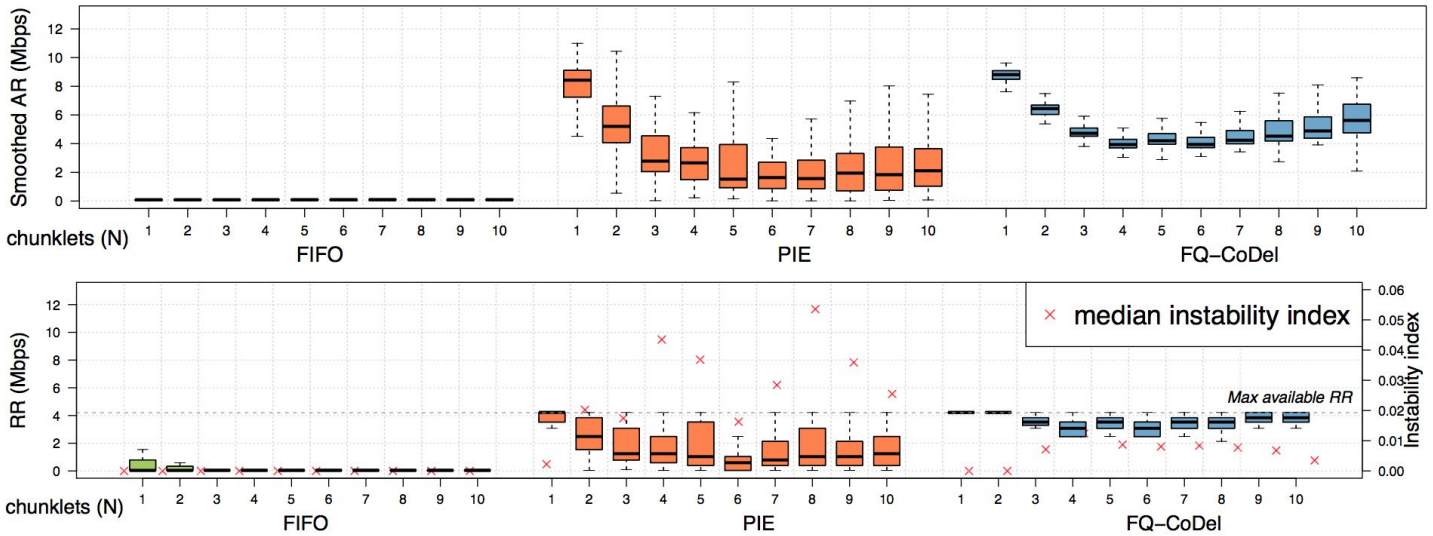


Fig. 9. Smoothed AR, RR and instability index *vs N* for FIFO, PIE and FQ-CoDel @ {12/1Mbps, 20ms RTT} path, 4 upstream bulk TCP flows

Conclusion

- Optimise DASH performance over AQM schemes by using parallel TCP connection and DASH chunklets
- With downstream or upstream TCP flows, it can provide better performance over AQMs than regular DASH over FIFO

- Experiment? or not? (not sure)
- Instability index or SA cannot be used to determine the QoE of DASH video content
- Is chunklet efficient? Due to the overhead of chunklet, N HTTP headers are required to transmit N chunklets

Q & A