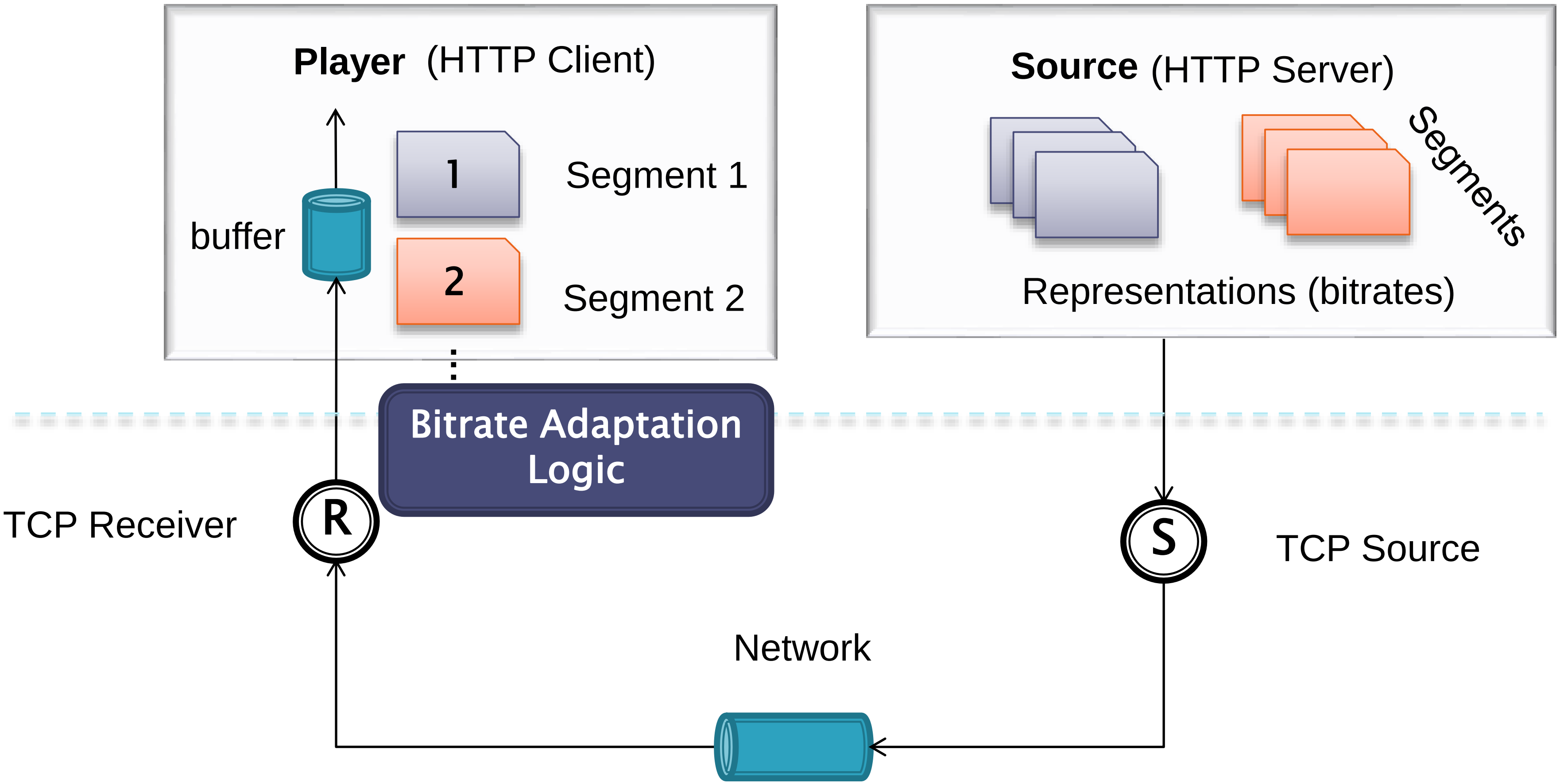
Cache-Friendly Streaming Bitrate Adaptation by Congestion Feedback in ICN

Dinh Nguyen[1], Jiong Jin[2], and Atsushi Tagami[1]
[1]KDDI R&D Laboratories Japan, [2]Swinburne Univ. of Tech. Australia

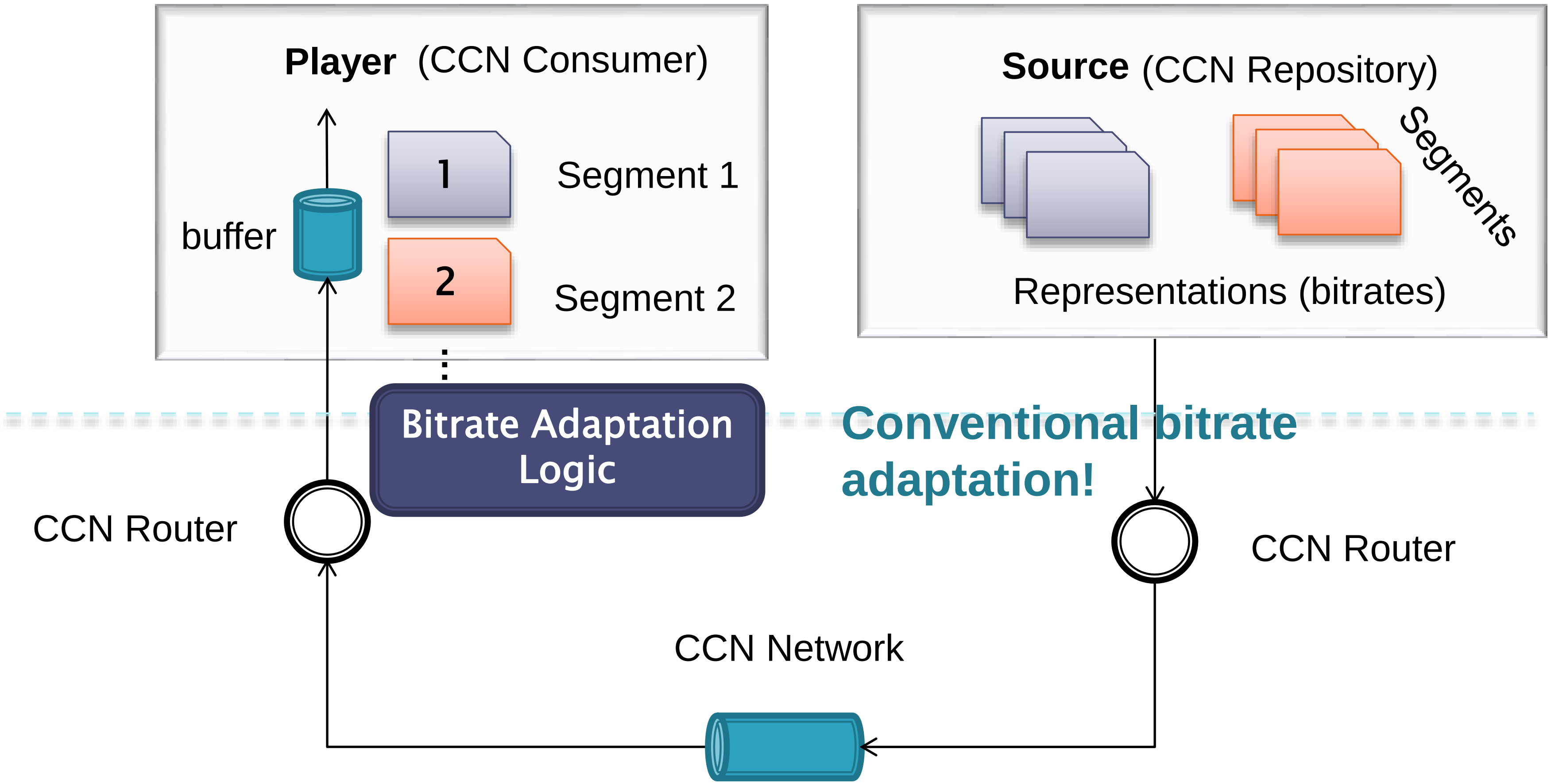
Content

1. Motivation
2. Fair and stable bitrate adaptation
3. Impact on caching
4. Evaluation
5. Conclusion and future work

Dynamic Adaptive Streaming over HTTP



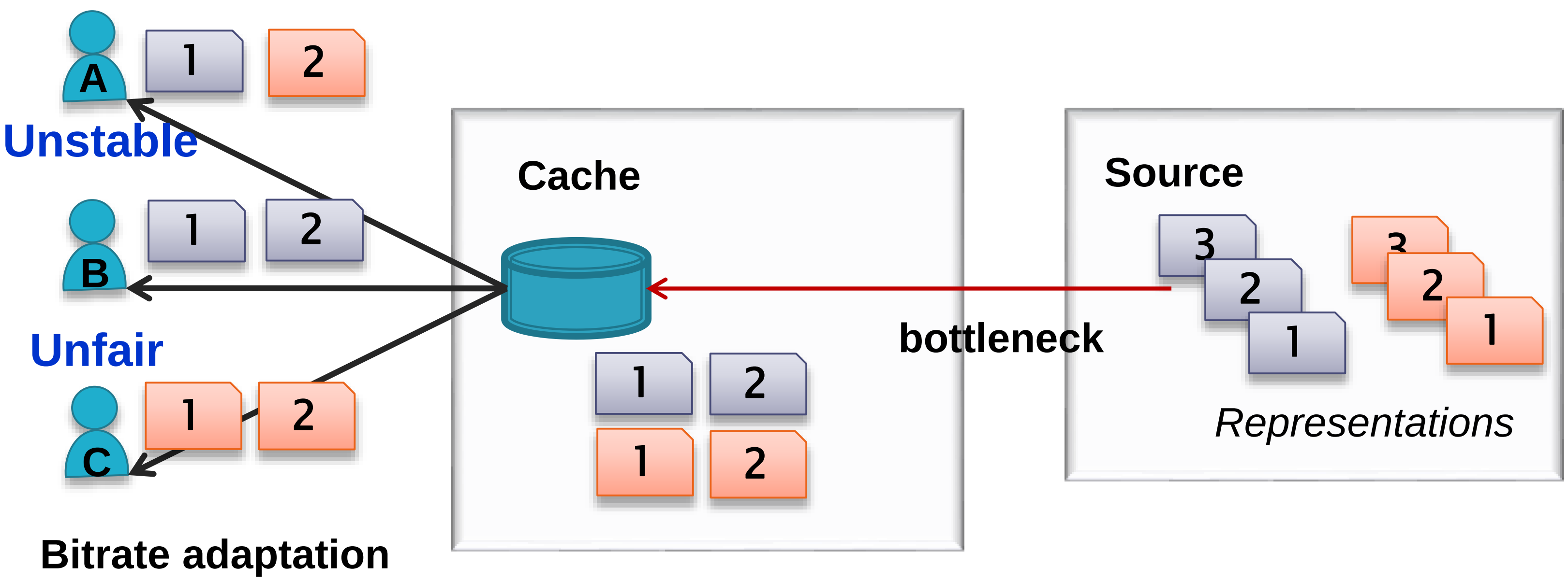
DASH over CCN



Liu, Y.; Geurts, J.; Point, J.-C.; Lederer, S.; Rainer, B.; Muller, C.; Timmerer, C. & Hellwagner, H.
Dynamic adaptive streaming over CCN: A caching and overhead analysis, *ICC* 2013, 3629-3633

The problem

- **Bitrate adaptation results in low cache hit**
  - Vanilla ICN caches admit every video segment
  - Non-repetitive requests from users due to unfair and unstable bitrate adaptation
  - **Multiple representations of the same segment**



Motivation

- Avoid the *side-effect* of bitrate adaptation to cache
  - High cache hit rate even with adaptive streaming
- Bitrate adaptation should be *friendly* to in-network caches
  - <u>Repetitive requests:</u> same bottleneck same bitrate
  - <u>Less fluctuation:</u> quickly settle at fair, stable bitrate
- To devise a streaming bitrate adaptation with
  - Fairness
  - Stability

Conventional bitrate adaptation

1. Estimate bandwidth in the last download

Estimate bandwidth

$$\tilde{x} = \frac{\text{segment\_size}}{\text{download\_time}}$$

inaccurate bandwidth share

2. Smooth the estimated bandwidth using e.g. EWMA

Smooth

$$\text{EWMA}(\tilde{x})$$

3. Choose nearest bitrate as the target video bitrate

Quantize

$$\hat{x} \leq \text{EWMA}(\tilde{x})$$

4. Request next segment of bitrate $\hat{x}$ after

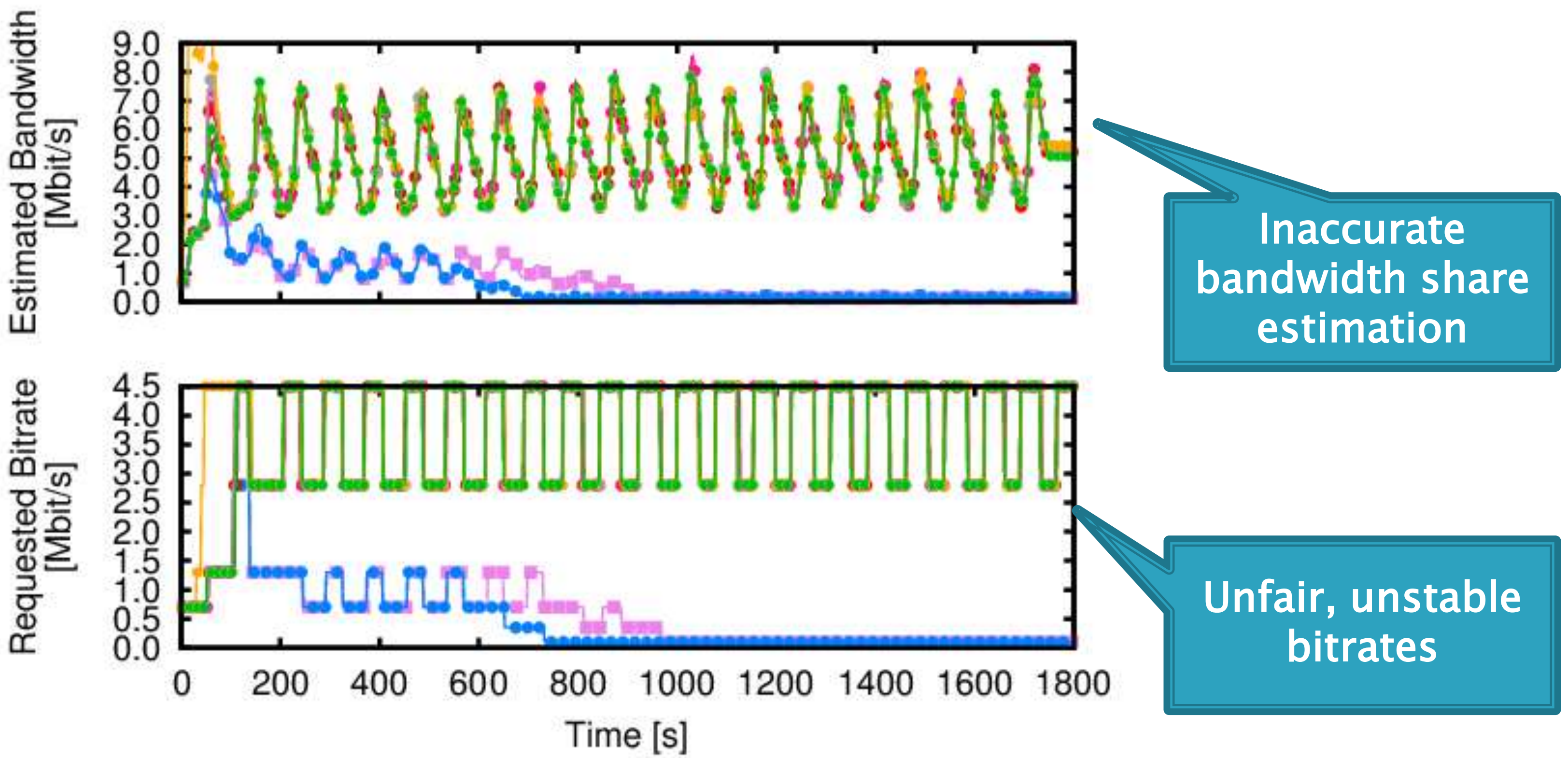
Schedule next request

$$\Delta t = \begin{cases} 0 \text{ if } B < B_{\max} \\ \tau \text{ if } B \geq B_{\max} \end{cases}$$

/* $B/B_{\max}$: current/max playback buffer */

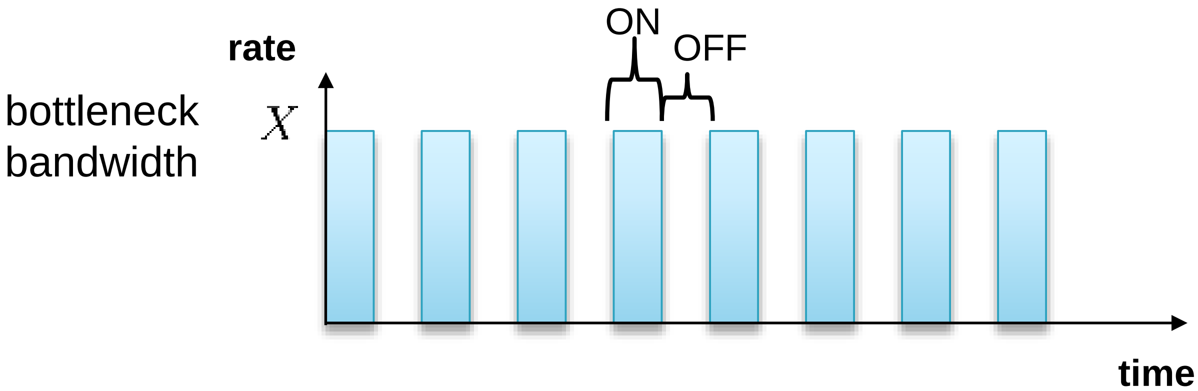
/* τ: segment length in second */

A simple test



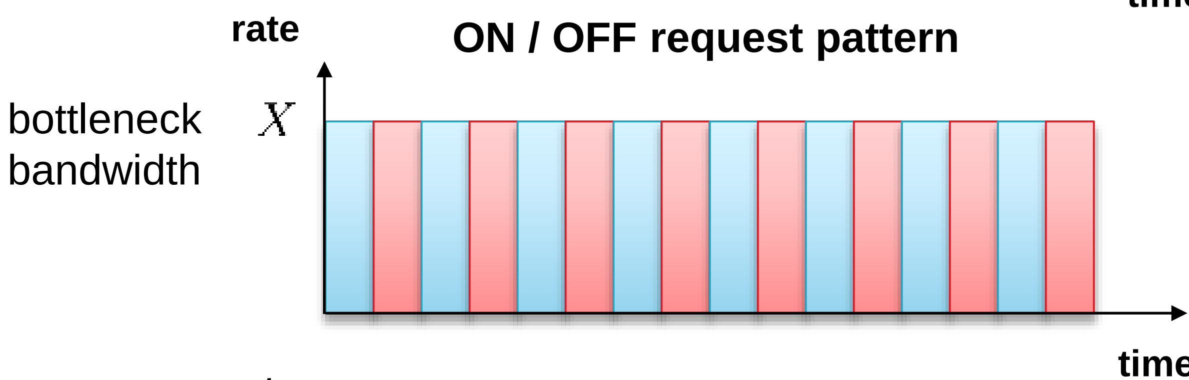
- 8 conventional adaptive streams compete over 24Mbit/s bottleneck

Inaccurate bandwidth estimation



ON / OFF request pattern

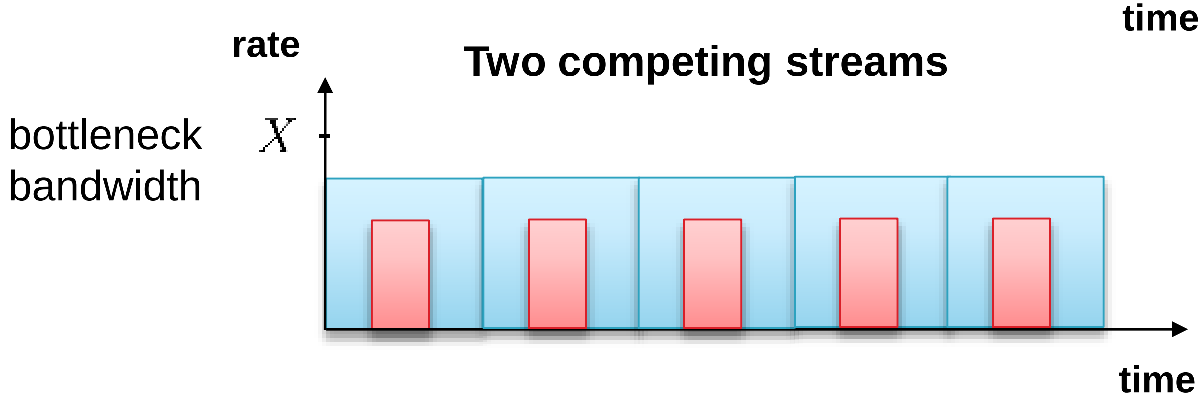
$\tilde{x} = X$



Two competing streams

$\tilde{x}_1 = \tilde{x}_2 = X$!!

Unstable



$\tilde{x}_1 > \tilde{x}_2$

Unfair

Utility fairness resource allocation

- Utility fairness optimization [Wang06]

$$\max_{x \geq 0} \quad \sum_{n \in N} \int_{m_n}^{x_n} \frac{1}{U_n(y)} dy$$

$$s.t. \quad \sum_{n:i,j \in L(n)} x_n \leq c_{i,j} \ \forall i,j$$

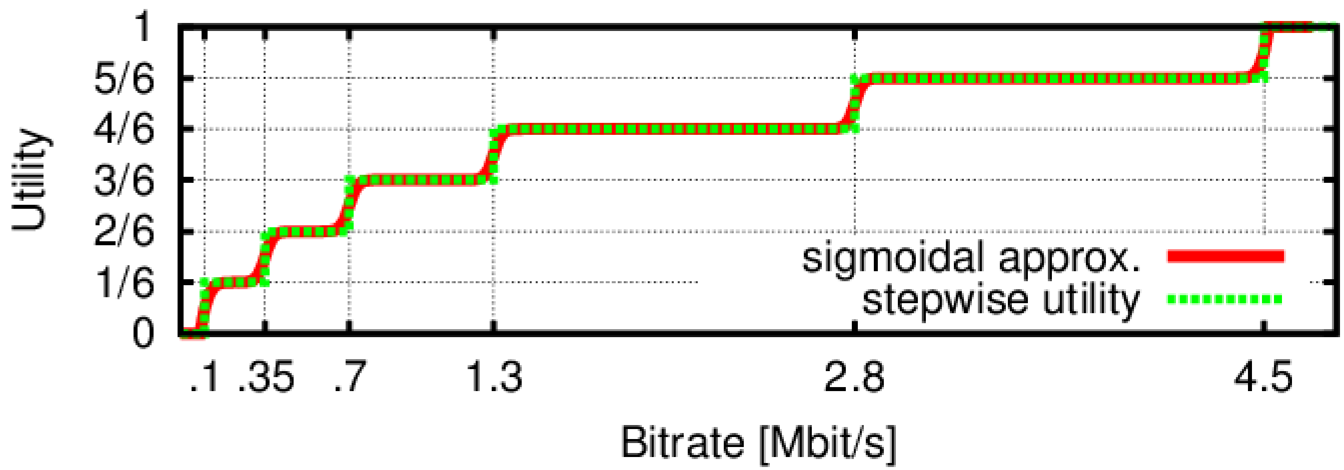
- Utility-fair bandwidth share

$$x_n(t+1) = U_n^{-1}\left(\underbrace{\frac{1}{q_n(t)}}_{\text{congestion feedback}}\right)$$

congestion feedback

Wang, W.-H.; Palaniswami, M. & Low, S., Application-Oriented Flow Control: Fundamentals, Algorithms and Fairness, *IEEE/ACM Trans. Netw.,* 2006*, 14*, 1282-1291

Utility functions



- Step-wise utility functions
- Logistic approximation at each step for continuity
- Could be different for users requesting the same content

Utility–fair bitrate adaptation

1a. Measure and smooth congestion price feedback

$$q_n = \mathrm{EWMA}[q(D)]$$

`/* ` $q_n/q(D)$`:  smooth/current congestion price */`

1b. Compute utility-fair bandwidth share

$$\tilde{x}_n = U_n^{-1}\left(\frac{1}{q_n}\right)$$

utility-fair bandwidth estimation

2. Smooth the estimated bandwidth

$$\mathrm{EWMA}(\tilde{x})$$

3. Choose nearest bitrate as the target video bitrate

$$\hat{x} \le \mathrm{EWMA}(\tilde{x})$$

4. Request next segment of bitrate $\hat{x}$ after

$$\Delta t = \begin{cases} 0 \text{ if } B < B_{\max} \\ \tau \text{ if } B \ge B_{\max} \end{cases}$$

`/* ` $B/B_{\max}$`:  current/max playback buffer */`
`/* ` τ`:  segment length in second */`

KDDI R&D LABS

Impact on caching (ext. to Che's)

- Fewer # representations in cache R
- Larger effective cache size

$$\bar{\boldsymbol{C}} = \frac{C}{S\left(\sum_{k=1}^{\boldsymbol{R}} b_k\right)}$$

- Longer characteristic time $T_{\bar{C}}$

$$\sum_{i=1}^{M}(1 - e^{-p_i \boldsymbol{T}_{\bar{\boldsymbol{C}}}}) = \bar{\boldsymbol{C}}$$

- Higher cache hit rate

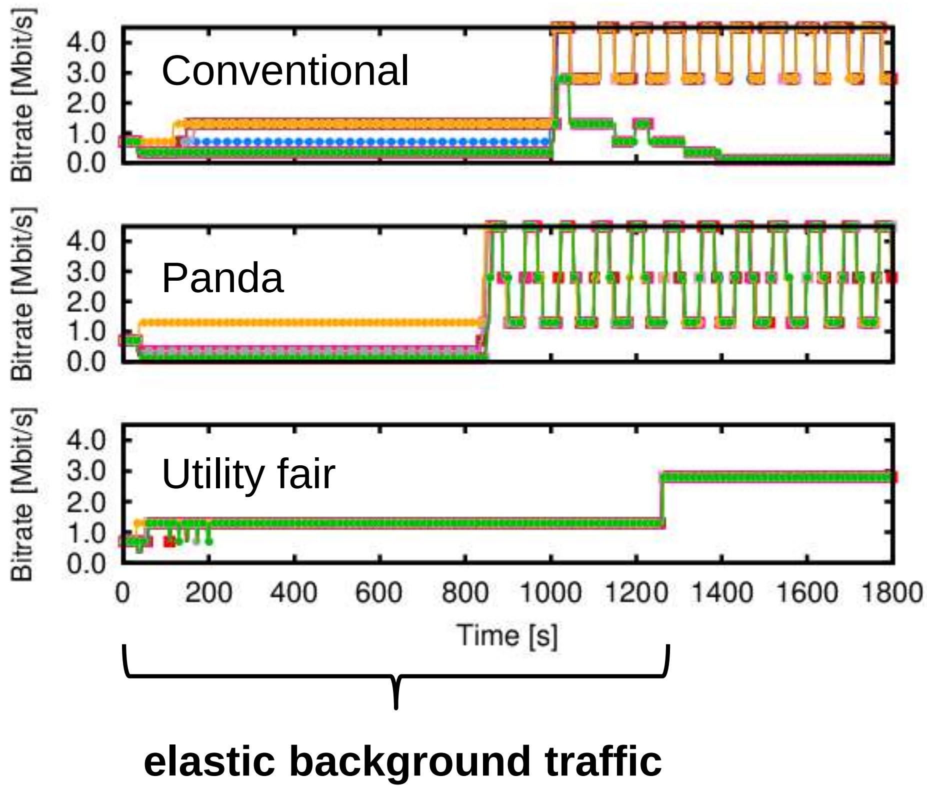
$$P_{\text{hit}} = \sum_{i=1}^{M} p_i(1 - e^{-p_i \boldsymbol{T}_{\bar{\boldsymbol{C}}}})$$

Simulation

- Chunk-level CCN simulation
- Congestion signal: queuing delay
- LRU caches
- Chain topology with bottleneck
- Playback buffer: 30 seg.
- Evaluate
  - Conventional adaptation
  - PANDA [Li2014]: AIMD bandwidth probe
  - Proposed utility-fair adaptation

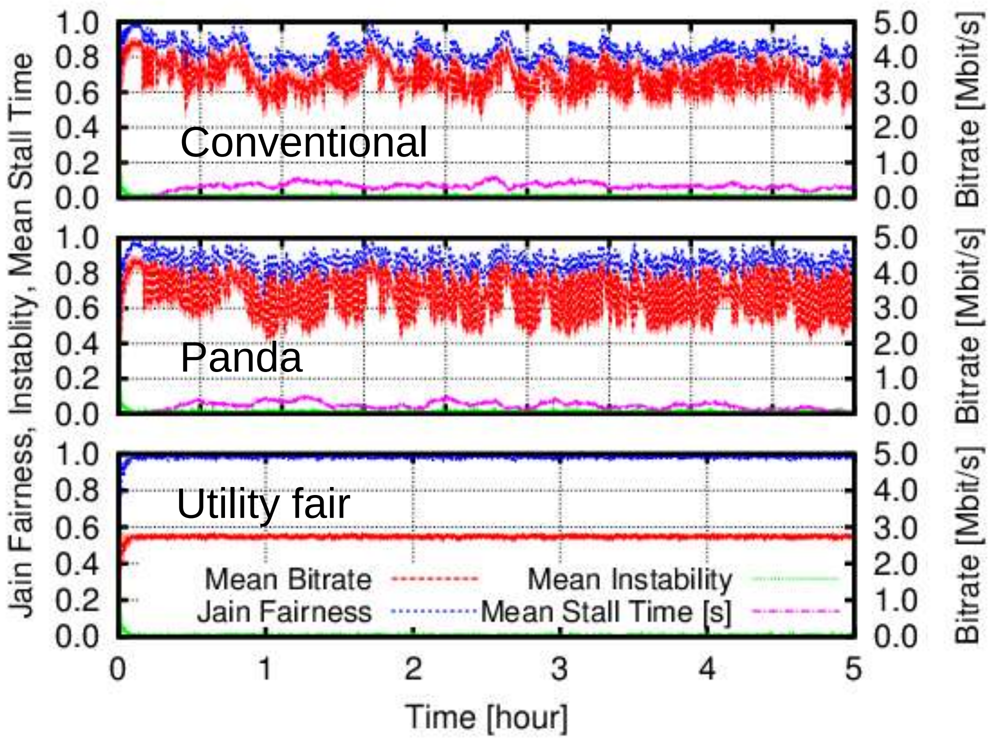
Li, Z.; Zhu, X.; Gahm, J.; Pan, R.; Hu, H.; Begen, A. & Oran, D.
Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale
Selected Areas in Communications, IEEE Journal on, 2014, *32*, 719-733

Elastic background traffic



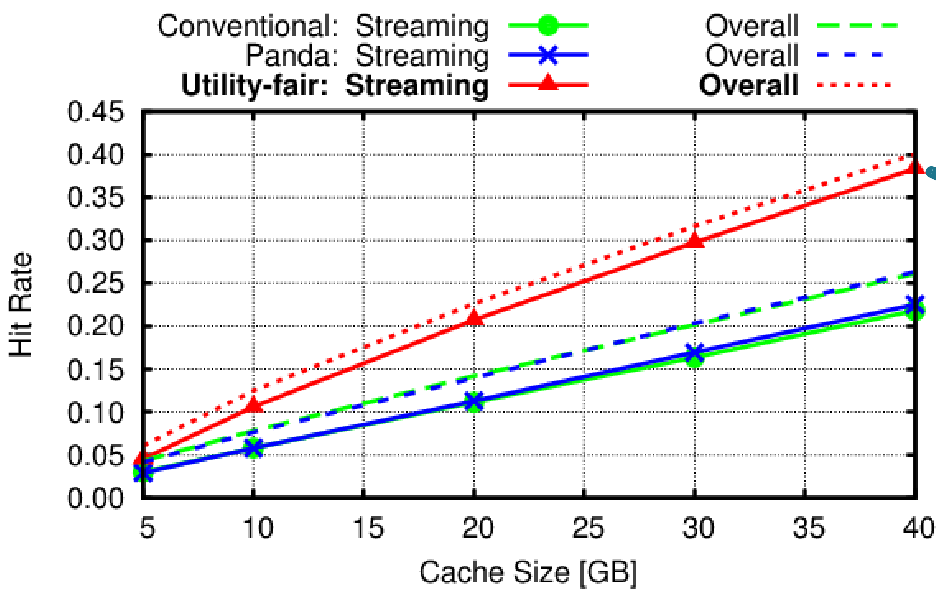
- 24 Mbit/s bottleneck
- 8 streaming sessions + 1 download

Stability and fairness

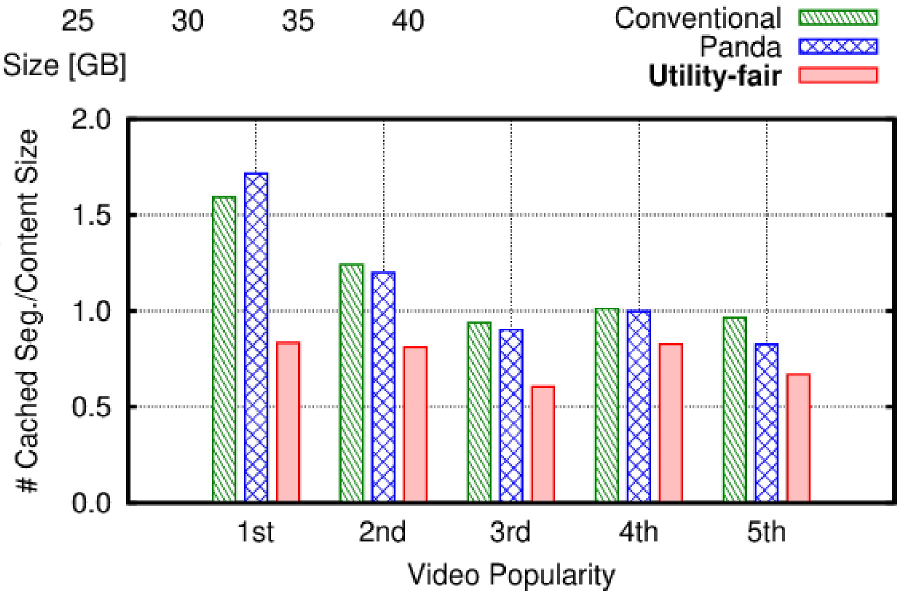


- 1Gbit/s bottleneck, chain topology
- 60% streaming + 40% downloading
- 2000 content objects (Zipf.8), Poisson arrival 0.8 req/s
- 20GB LRU cache @ all routers

Cache-friendliness



Much higher hit ratio



Fewer redundant representations

Conclusion

- Promising result on bitrate adaptation by congestion feedback in CCN
  - Accurate bandwidth estimation: using utility fairness framework
  - Cache-friendliness: fairness and stability in bitrate adaptation increase cache hit
- Future work
  - Testbed evaluation
  - Caching for video streaming
  - Exploiting better congestion feedback signal available in CCN/NDN

Thank you!