Cache-friendly Streaming Bitrate Adaptation over NDN using Explicit Congestion Feedback

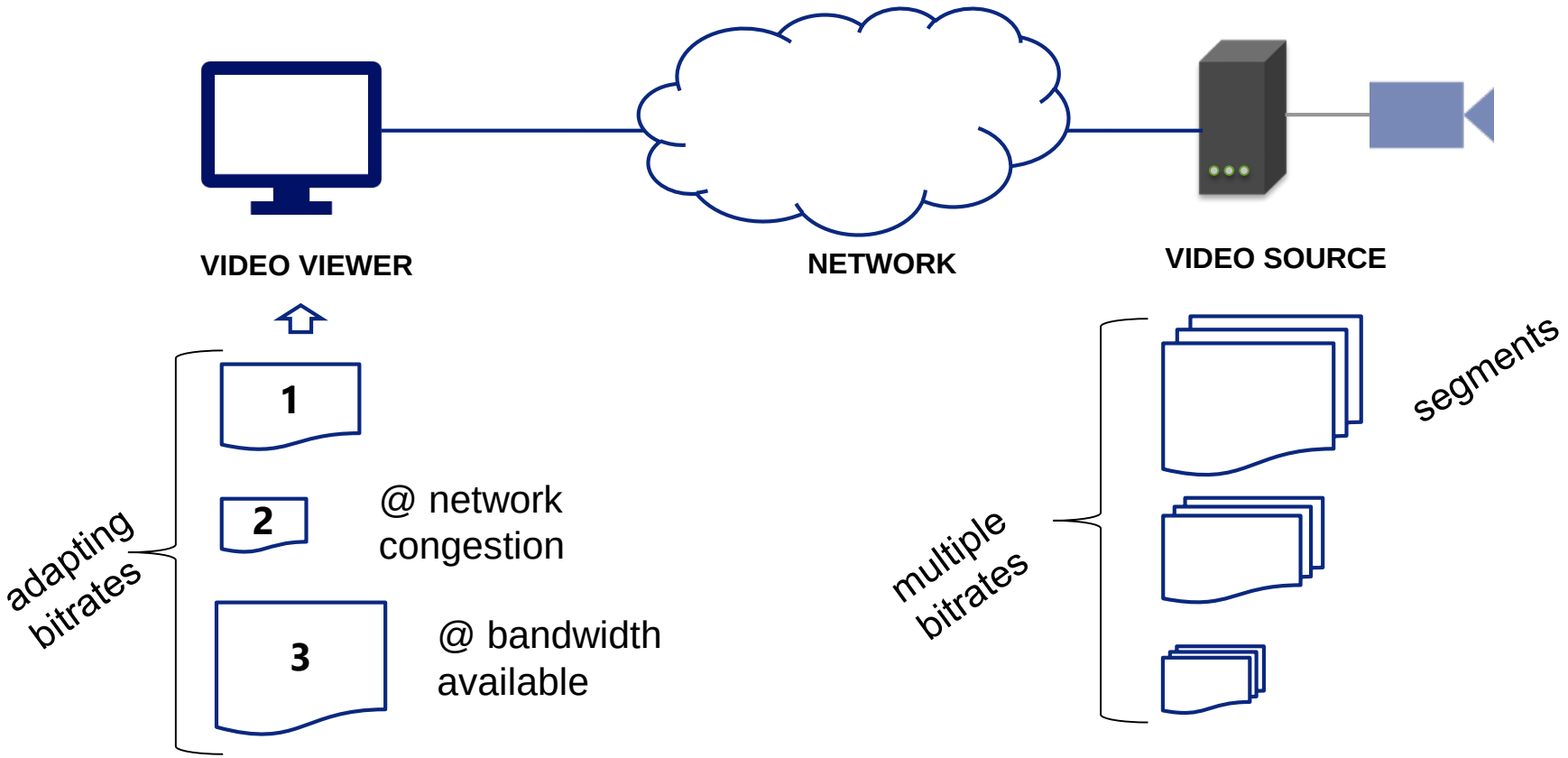
Dinh Nguyen, 植田一暁, 田上敦士
KDDI総合研究所

KDDI Research

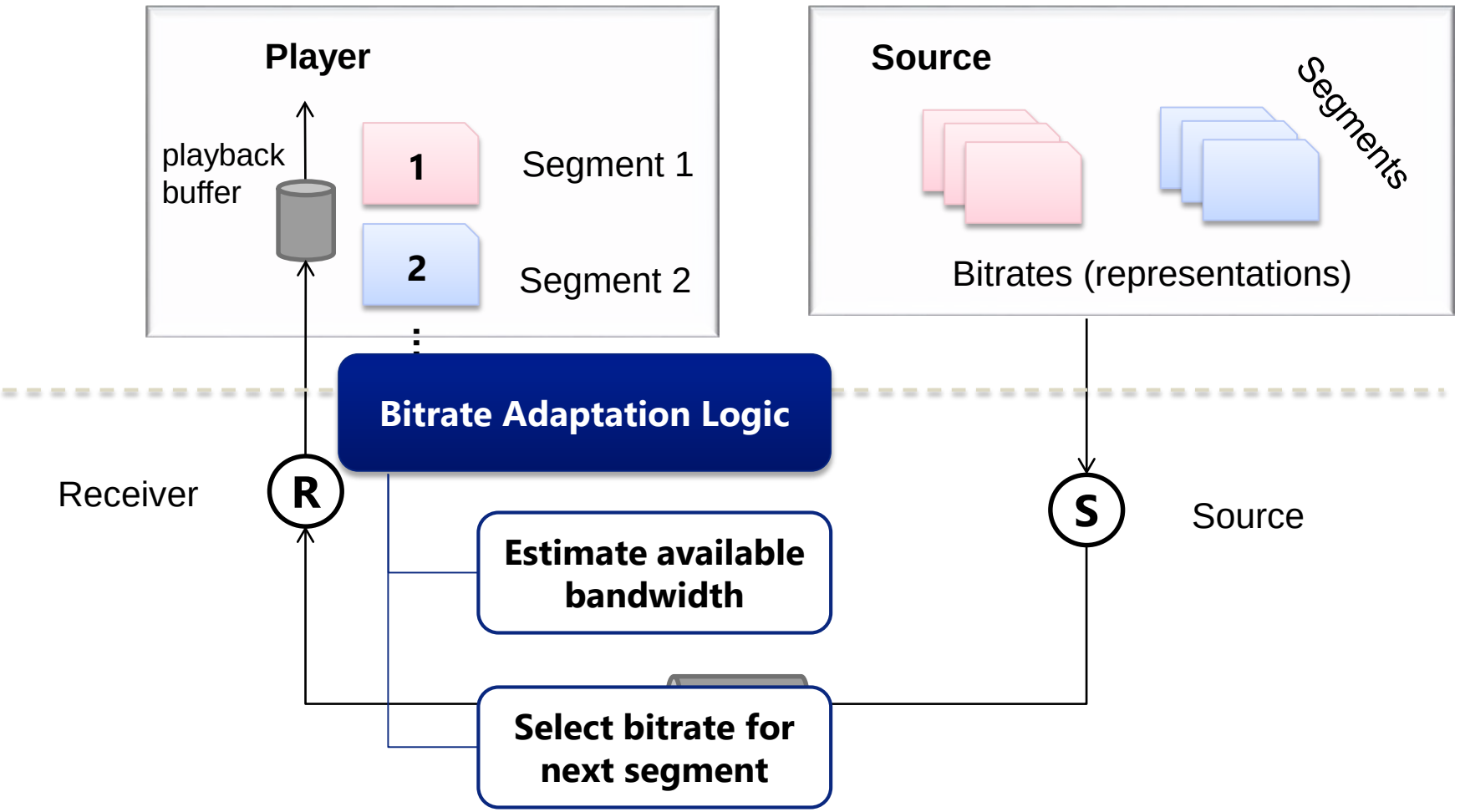
Content

1. Challenges of dynamic adaptive streaming
2. Cache-friendly bitrate adaptation in NDN
3. Performance evaluation
4. Conclusion and future work

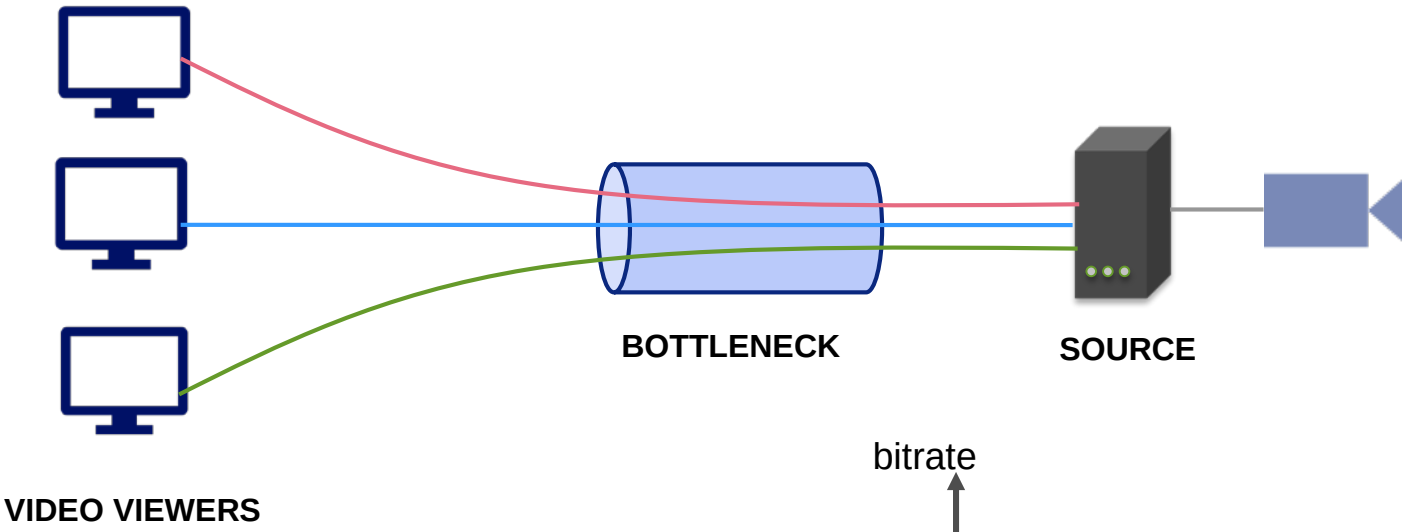
Dynamic Adaptive Streaming (DAS)

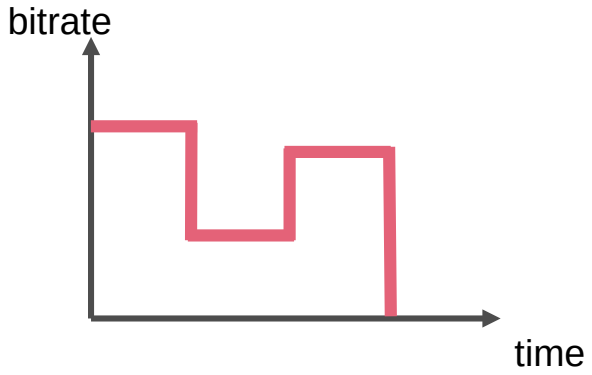


Bitrate adaptation in DAS

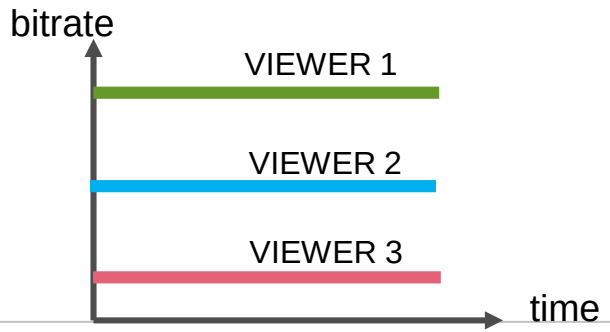


Problems with current bitrate adaptation



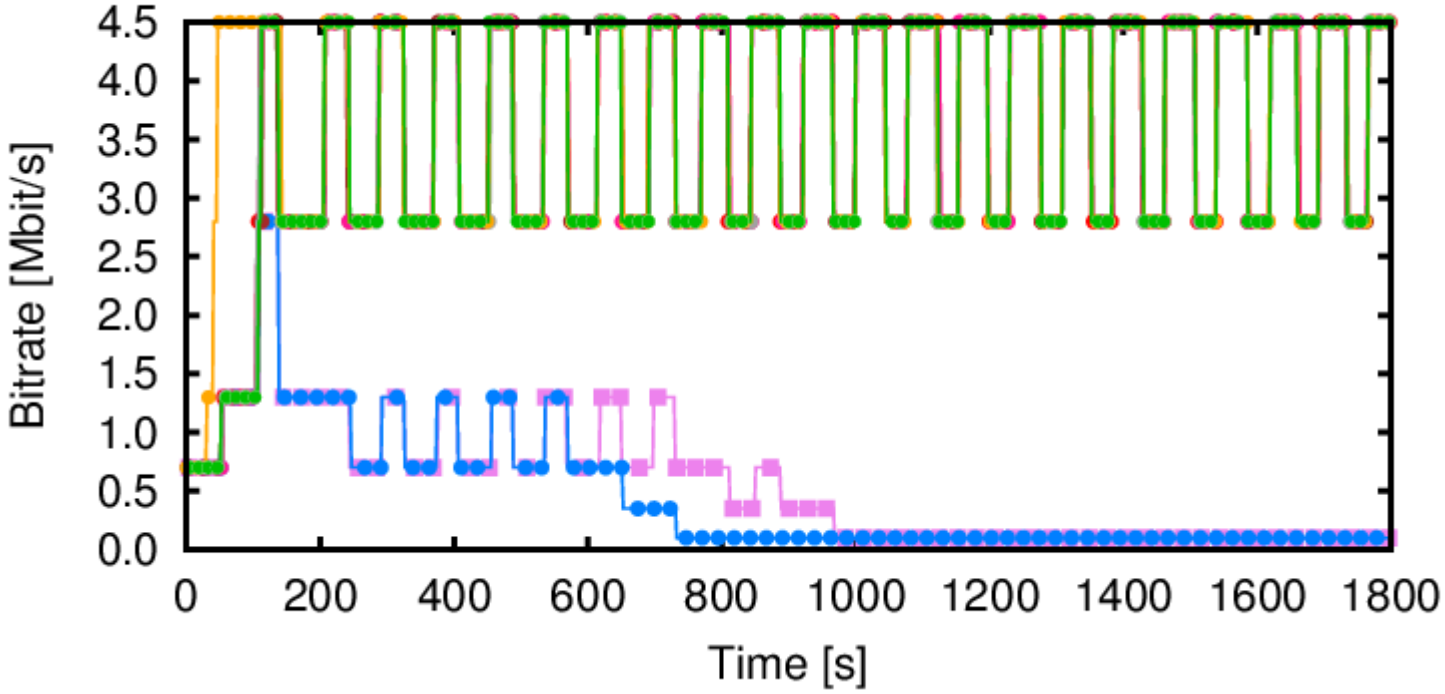


- Unstable video bitrate
  - Fluctuating streaming bitrate



- Unfair bitrate allocation
  - Unfair bandwidth share among multiple users over the same bottleneck

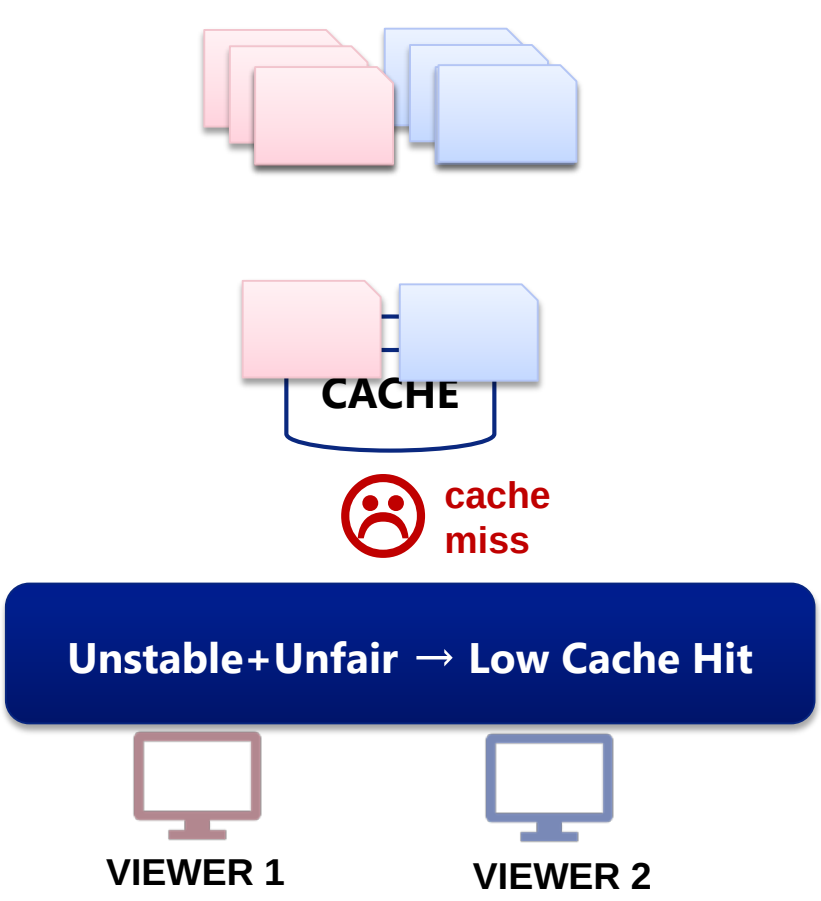
How unstable and unfair conventional bitrate adaptation?



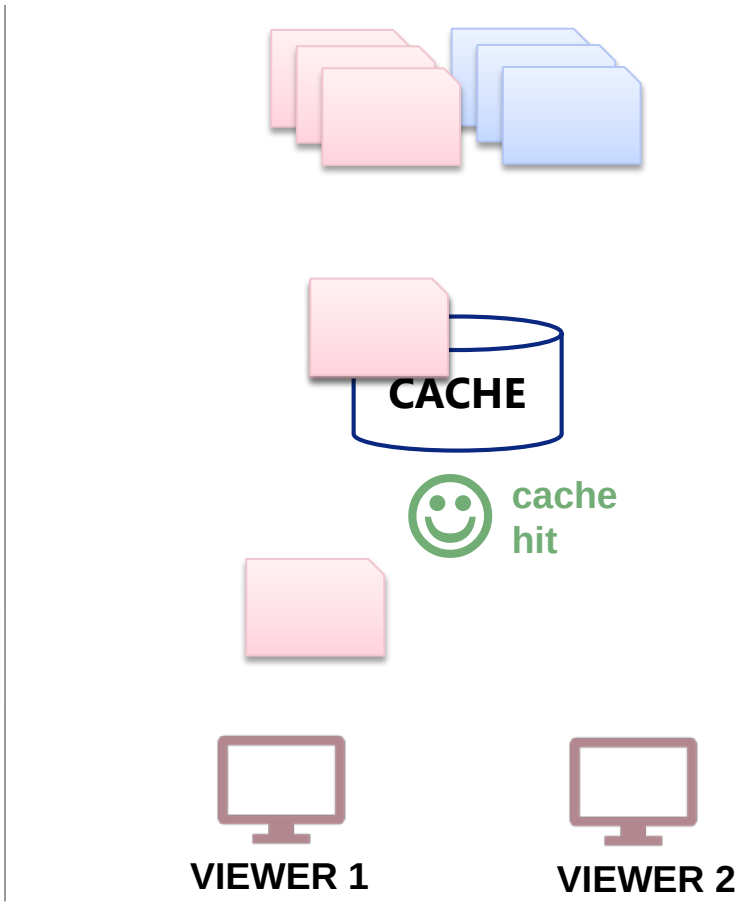
■ 8 conventional adaptive streams over 24Mbit/s bottleneck

Problems with bitrate adaptation over NDN

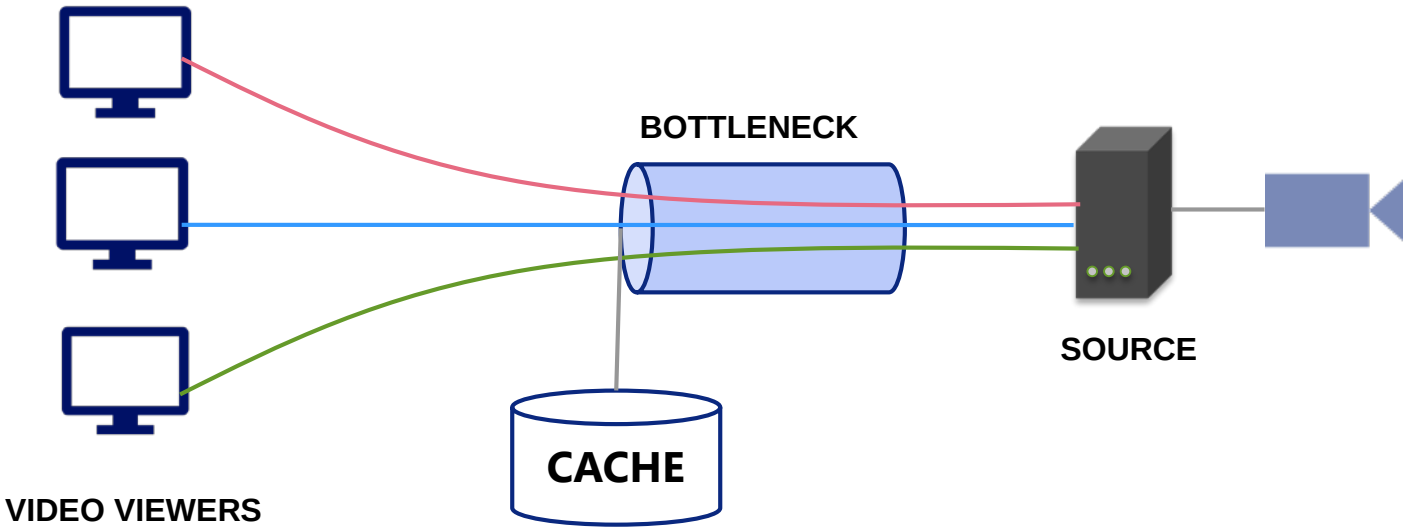
■ Unstable + Unfair Adaptation (Existing)



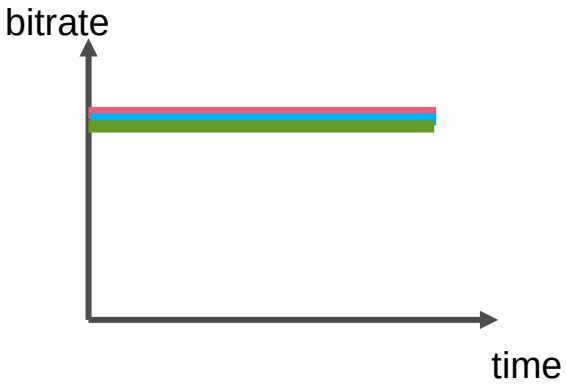
■ Stable + Fair Adaptation (Desired)



Motivation



- High QoE streaming to users
  - High bitrate
  - Stable and fair bitrate
- Cache-friendly bitrate adaptation
  - High cache hit rate even with adaptive streaming



Conventional bitrate adaptation

1. Estimate bandwidth in the last download

Estimate fair bandwidth share

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

Mean download throughput

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

Smooth

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

3. Choose nearest bitrate as the target video bitrate

Quantize

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

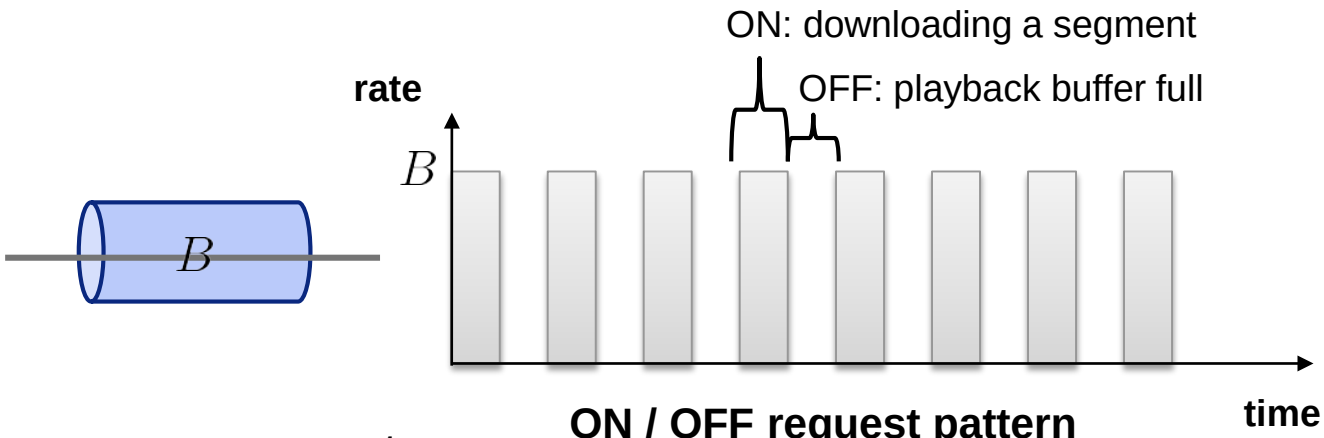
4. Request next segment of bitrate $\hat{x}$ after

Schedule next request

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

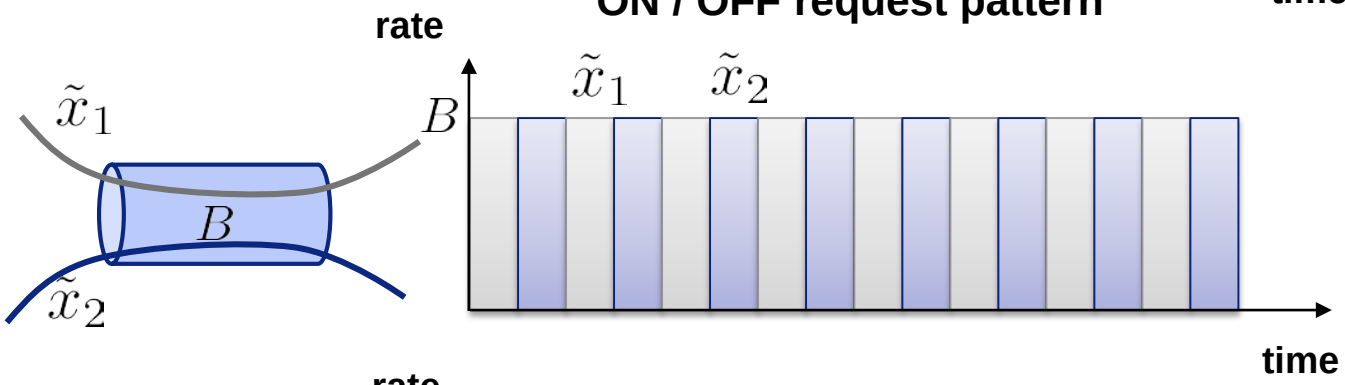
```
/* B/B_max:  current/max playback buffer                    */
/* τ:  segment length in second                             */
```

Mean download throughput is unstable and unfair



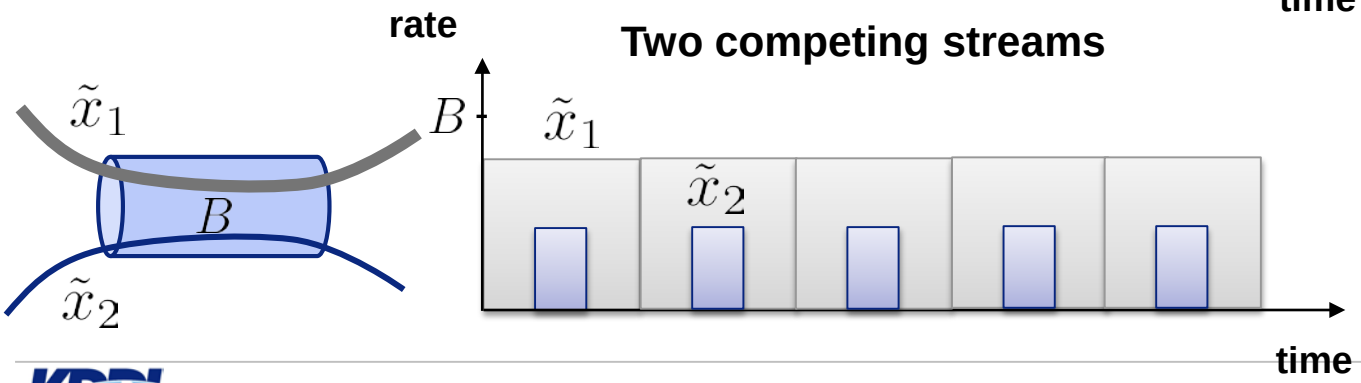
MEAN THROUGHPUT

$$\tilde{x} = B$$



$$\tilde{x}_1 = \tilde{x}_2 = B \;!!$$

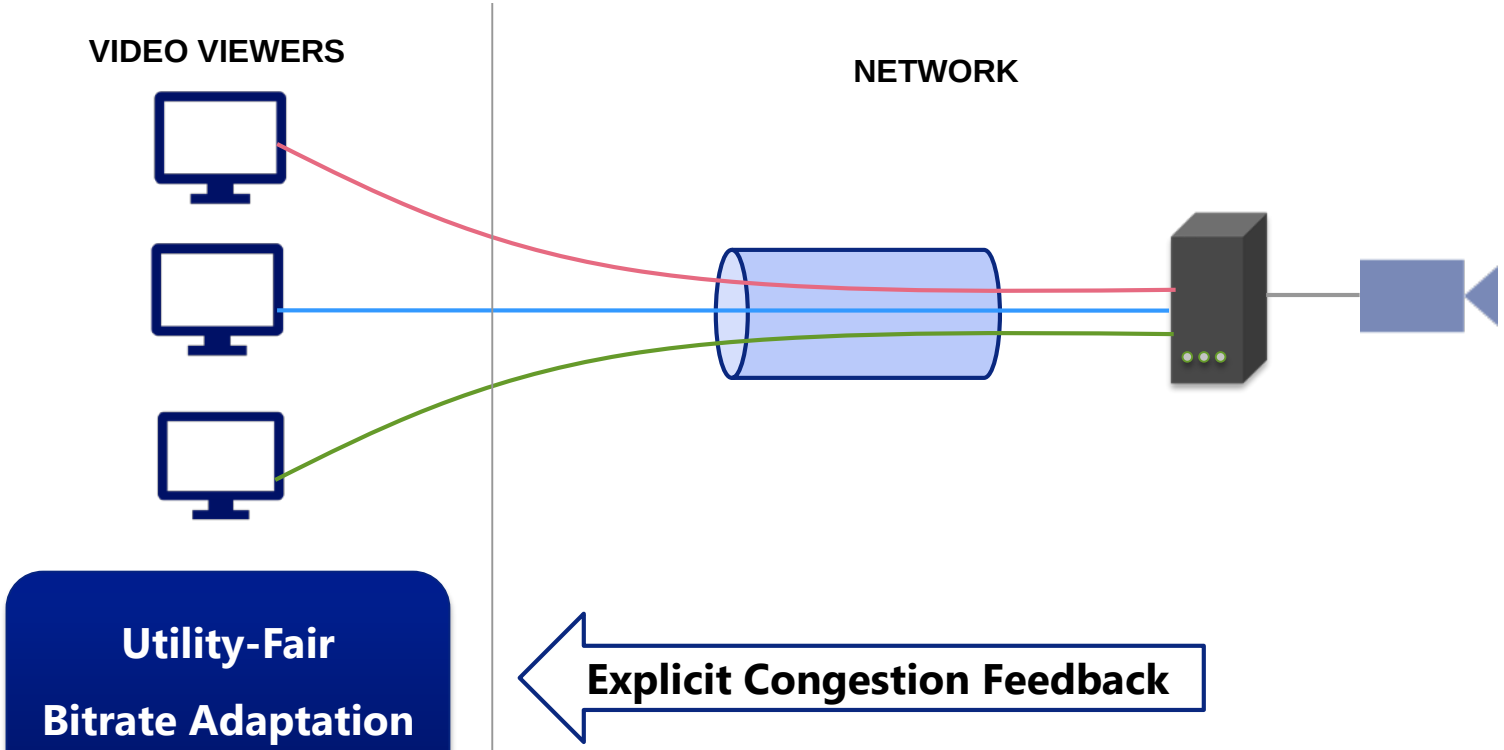
Unstable



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

Unfair

Our proposed solution



Utility-fair framework

- ❑ Fair and stable bitrate

Congestion price (e.g. queuing delay)

- ❑ accumulated over delivery path
- ❑ fed back in Data packets

Utility fairness framework

■Utility fairness optimization [Wang06]

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

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

bandwidth allocation

utility function

link capacity

■Utility-fair bandwidth share

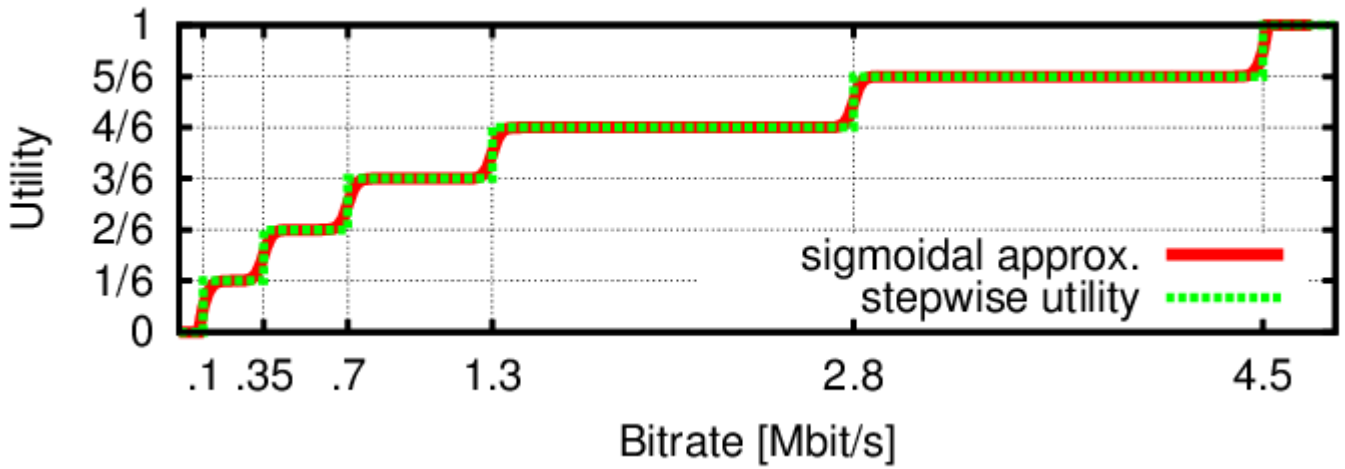
$$x_n(t+1) = U_n^{-1}\left(\underbrace{\frac{1}{q_n(t)}}\right)$$

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 function

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



Utility function of a streaming video with 6 bitrates

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

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

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

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

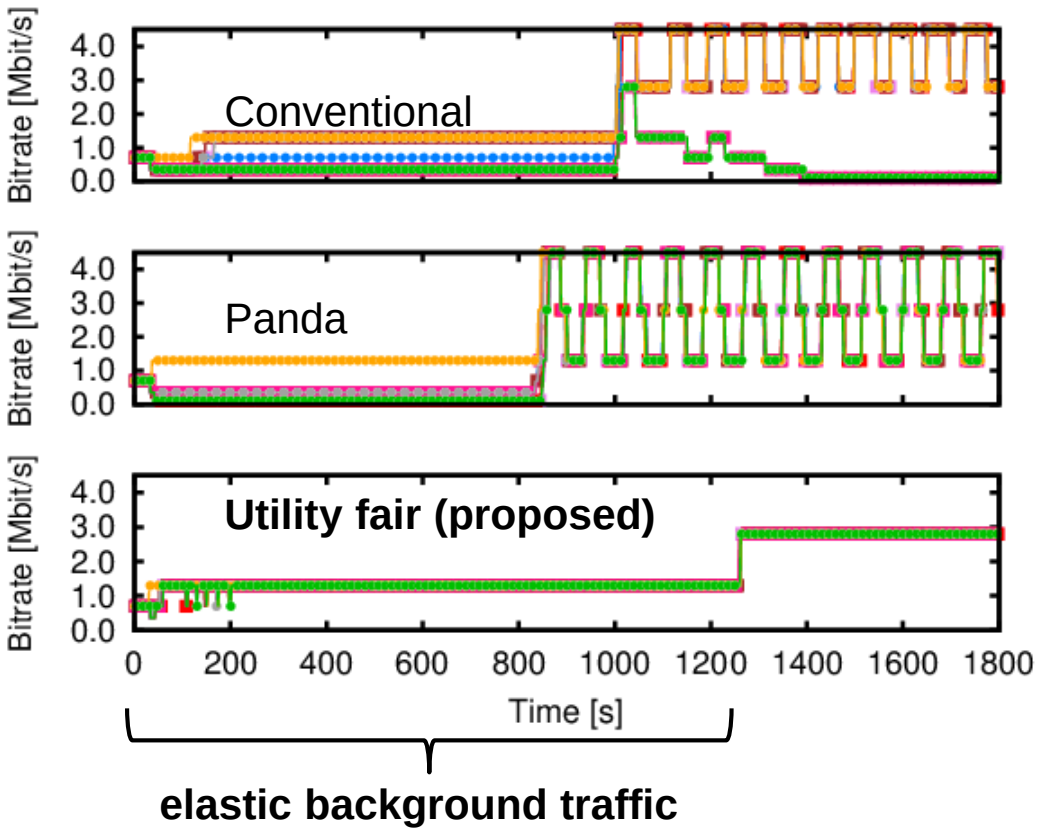
/* $B/B_{\max}$: current/max playback buffer */
/* τ: segment length in second */

Simulation evaluation

- Chunk-level CCN/NDN simulation
- Congestion signal: queuing delay
- LCE+LRU caches
- Chain topology of 4 routers with bottleneck
- Playback buffer: 30 seg.
- Evaluate
  - Conventional bitrate adaptation
  - PANDA [Li2014]: AIMD bandwidth probe
  - Proposed utility-fair adaptation

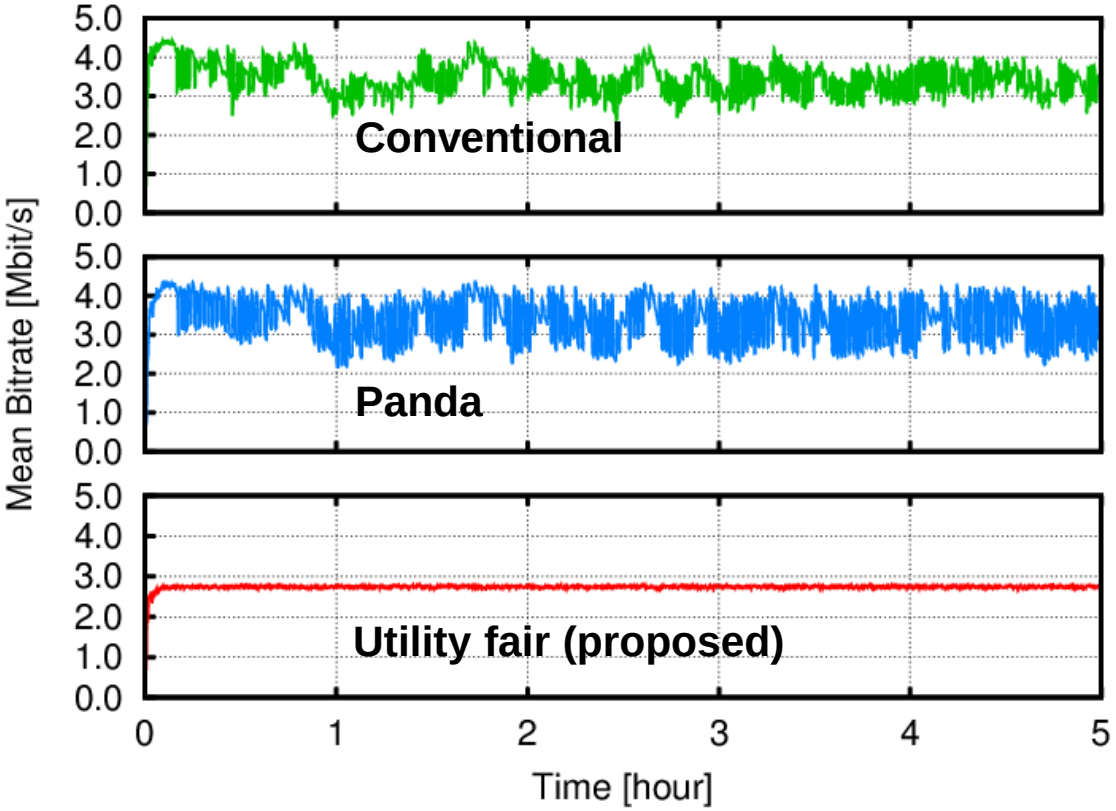
Li et al. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale *JSAC* 2014

Elastic background traffic



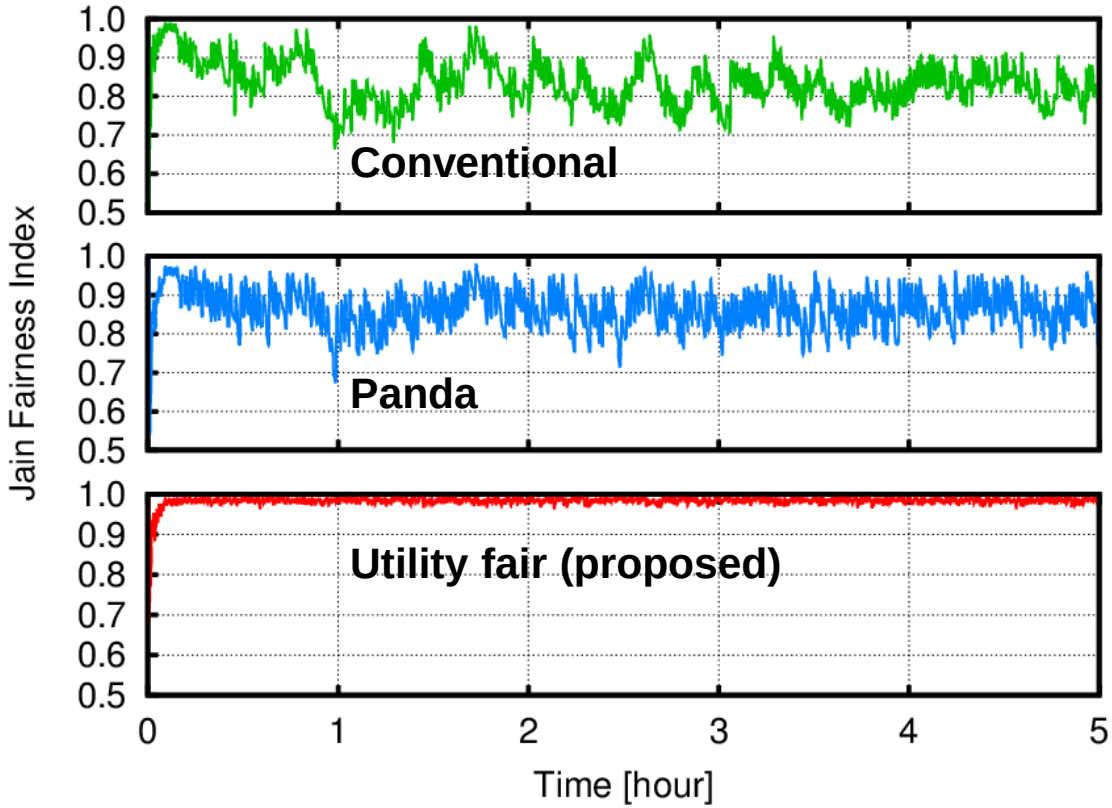
- 24 Mbit/s bottleneck
- 8 streaming sessions + 1 download

Stability of bitrate adaptation



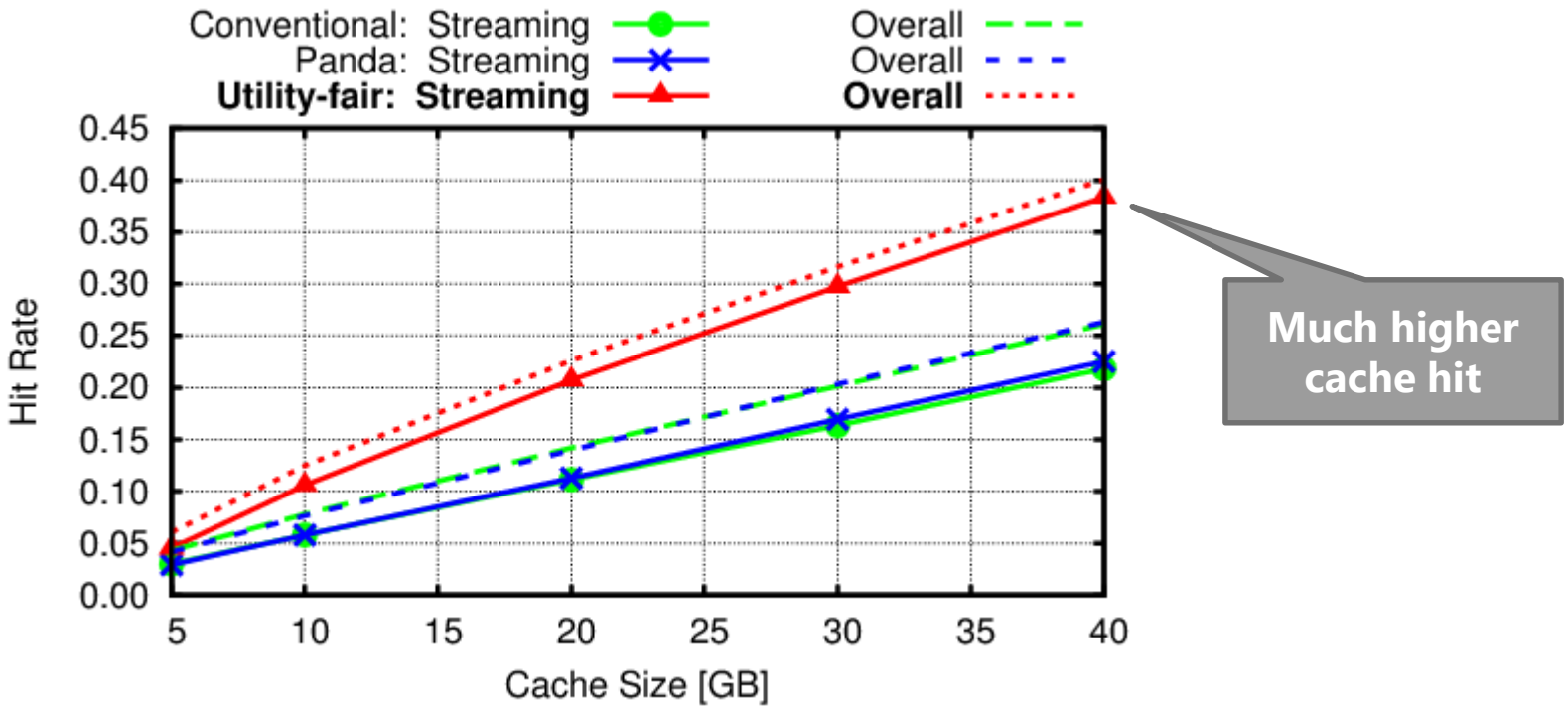
- 1Gbit/s bottleneck, chain topology
- 60% streaming (10min)+ 40% downloading (50MB)
- 2000 content objects (Zipf.8), Poisson arrival 0.8 req/s
- 20GB LRU cache @ all routers

Fairness of bitrate adaptation



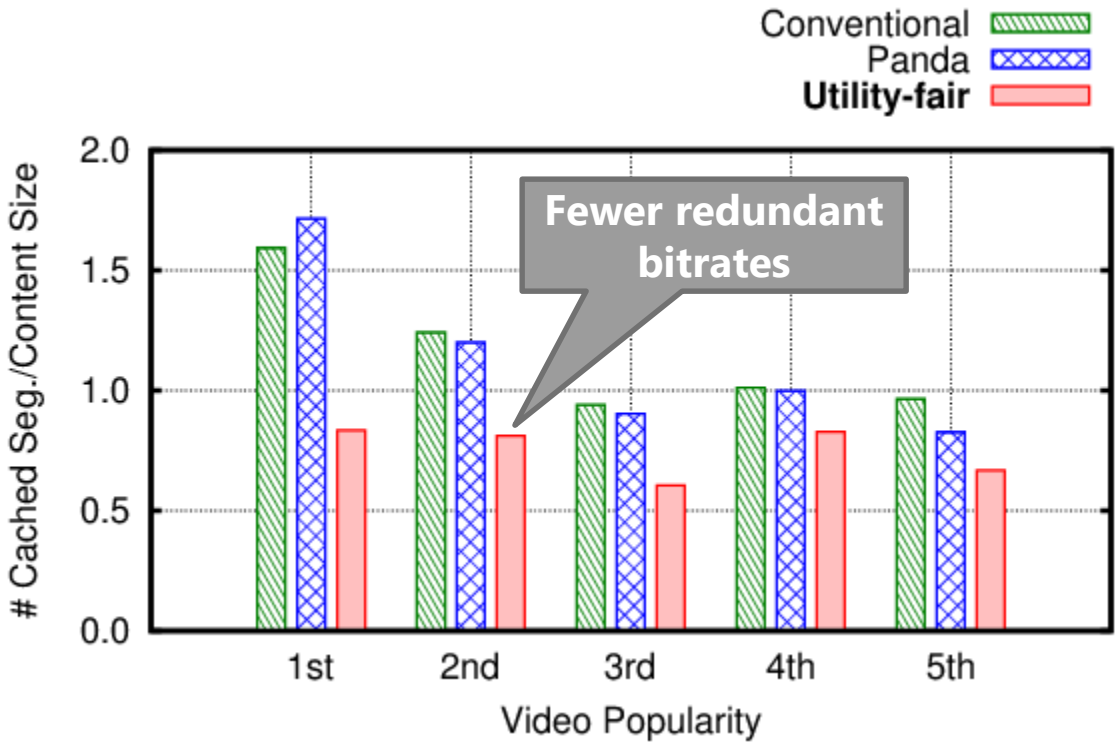
- 1Gbit/s bottleneck, chain topology
- 60% streaming (10min)+ 40% downloading (50MB)
- 2000 content objects (Zipf.8), Poisson arrival 0.8 req/s
- 20GB LRU cache @ all routers

Cache hit rate



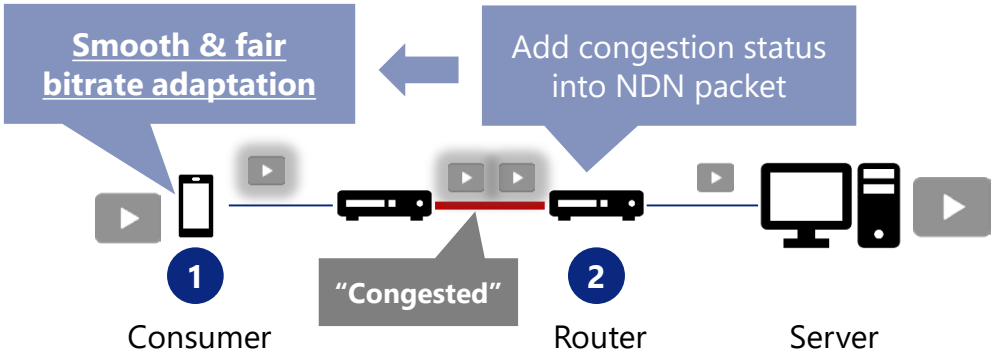
- 1Gbit/s bottleneck, chain topology
- 60% streaming (10min)+ 40% downloading (50MB)
- 2000 content objects (Zipf.8), Poisson arrival 0.8 req/s
- 20GB LRU cache @ all routers

Number of bitrates in cache



- 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

NDN-based implementation



1. DAS over NDN + congestion feedback

NDN-enabled VLC player

1. NDN connection module for VLC
   - Based on Consumer/Producer API
2. Bitrate adaptation logic
   - Use feedback value for bitrate selection

2. Congestion feedback enabled NFD

NFD adds congestion status into the header of data packet

- Queuing delay is used for congestion status
- Use libnl3 library to get *txqueue* length
- Store that value to NDNLPv2 header
  - Outside of the "signature envelop"

Conclusion

- Streaming bitrate adaptation by congestion feedback in CCN/NDN
  - Fair and stable bitrate adaptation: using utility fairness framework and explicit congestion feedback
  - Cache-friendliness: fairness and stability in bitrate adaptation increase cache hit
- Future work
  - Evaluation in global Testbed: CUTEi
  - Caching for video streaming


KDDI
KDDI Research