When AIMD meets ICN: a bandwidth sharing perspective

Chadi Barakat

Joint work with Damien Saucez (Inria)
Ilaria Cianci (Politecnico di Bari)
Luigi Alfredo Grieco (Politecnico di Bari)

Information Centric Networking (ICN)

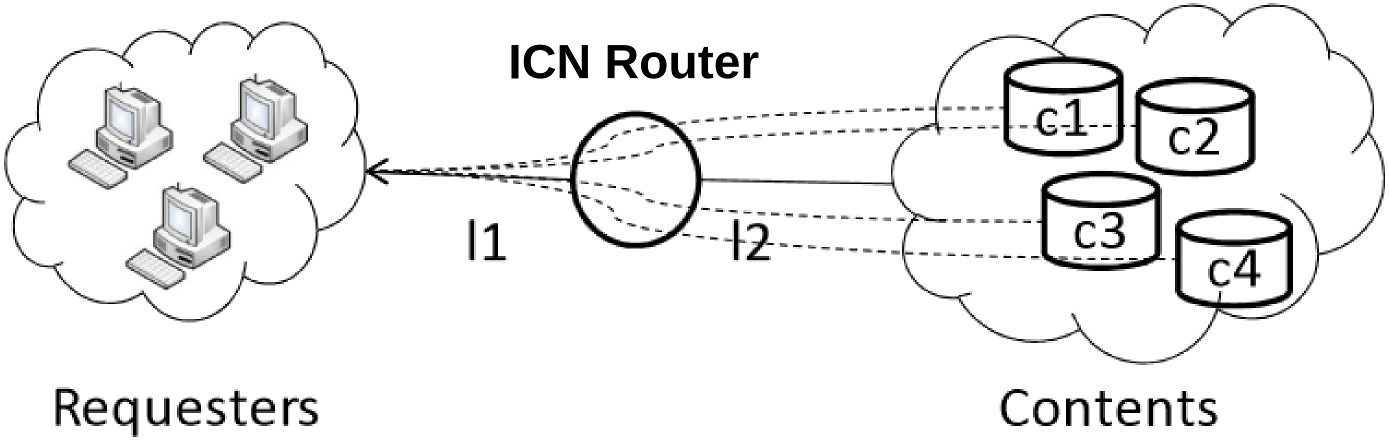
Interplay between in-network caching and congestion control

- What is ICN?
  - A new content-centric networking paradigm
  - Routing by content names
  - Caching inside the network
- In-network caching and congestion control
  - Popular contents get closer to edge (smaller network delay)
  - For AIMD-like (or TCP-like) congestion control

    smaller delay = faster rate increase
  - How bandwidth is shared?
  - Who will win and who will lose when in-network caching is enabled?
- Compared to today, ICN will strongly correlate network delay with popularity

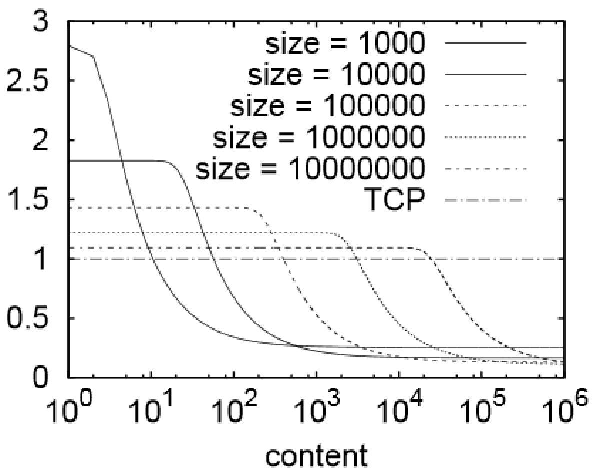
The case of long-lived flows and one bottleneck router

- Flow = Content download
- Contents of different popularity (e.g. Zipf)
- N downloads in parallel
- Bandwidth bottlenecked at one router
- A cache of finite size
- Requesters implement AIMD congestion control (TCP-SACK like)

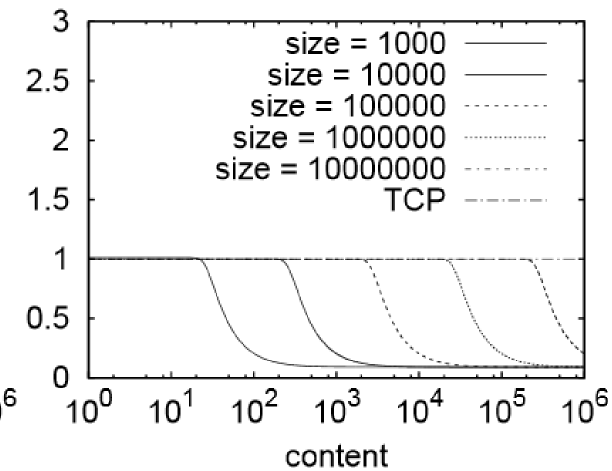


The case of long-lived flows and one bottleneck router

- From TCP modeling history, *Download Rate ∞ 1/RTT*
- RTT = Mean Round-Trip Time
  - No Caching: *RTT = End-To-End Delay*
  - With Caching: *RTT = Hit × Delay-to-Cache + Miss × End-To-End-Delay*
  - Hit and Miss rates, and thus RTT, depend on content popularity
- Gain for content c = $\dfrac{\textit{Harmonic Mean of RTT}}{\textit{RTTc}}$



Zipf(1.1)



Zipf(2)

The case of finite-size contents

From instantaneous to long-run bandwidth sharing

- Instantaneous bias against unpopular contents because of their longer RTT
  - Can be seen as waiting for popular contents to finish
  - Is there enough free time left to recover from this bias?
- Long-run bandwidth sharing
  - Contents of finite size
  - Contents of different popularity
  - Stochastic request process with some constant rate (load < 1)
  - Mean download time vs. popularity ?
  - Expansion (contraction) factor for content c:

$$\frac{\textit{Mean Download Time with Caching}}{\textit{Mean Download Time without Caching}}$$

ICN as a Discriminatory Processor Sharing Queue

- What is a DPS ?
  - Work conserving system
  - Parallel processing of contents
  - Resources split between active contents proportionally to their weights

$$\text{Share of content c} = \frac{\textit{Weight of Content c}}{\textit{Sum of Weights of Active Downloads}},$$

- Implicit equations exist for mean download time per weight (class)

G. Fayolle, et al. Sharing a processor among many job classes. Journal of the ACM, 1980.

Assumptions: Poisson arrivals of requests

Exponentially distributed service times (i.e. content sizes)

- ***For AIMD and ICN, weight of content c = 1/RTTc***

The simple case of two extreme classes

- Two classes of contents: Highly and poorly popular

  Popular contents cached very close to edge

- **Analytical result:**
  - Popular contents see same download time as without caching
  - Non popular contents see a download time inflated by $1/(1-\rho)$

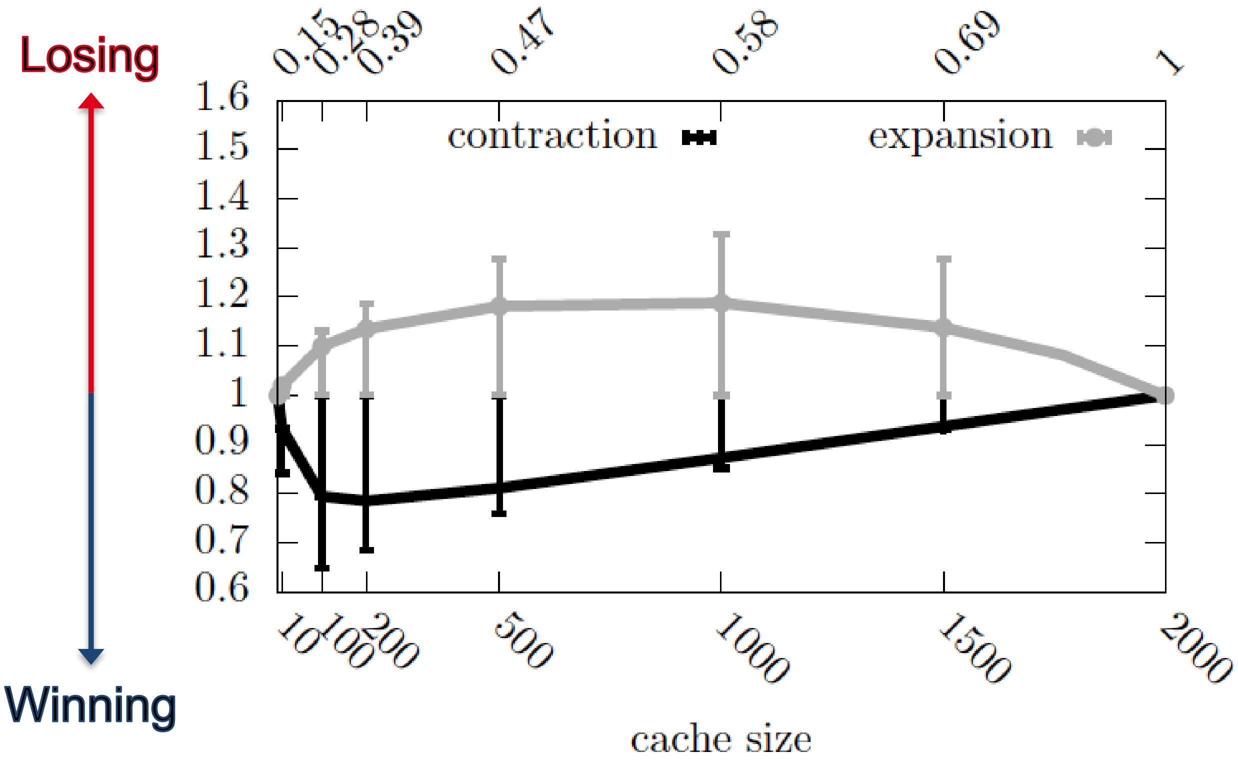
    ρ = load on bottleneck link

  - The larger the load, the larger the bias

The case of M classes

Expansion factor vs. Cache size

- A load of 90% and a catalogue of 2000 contents

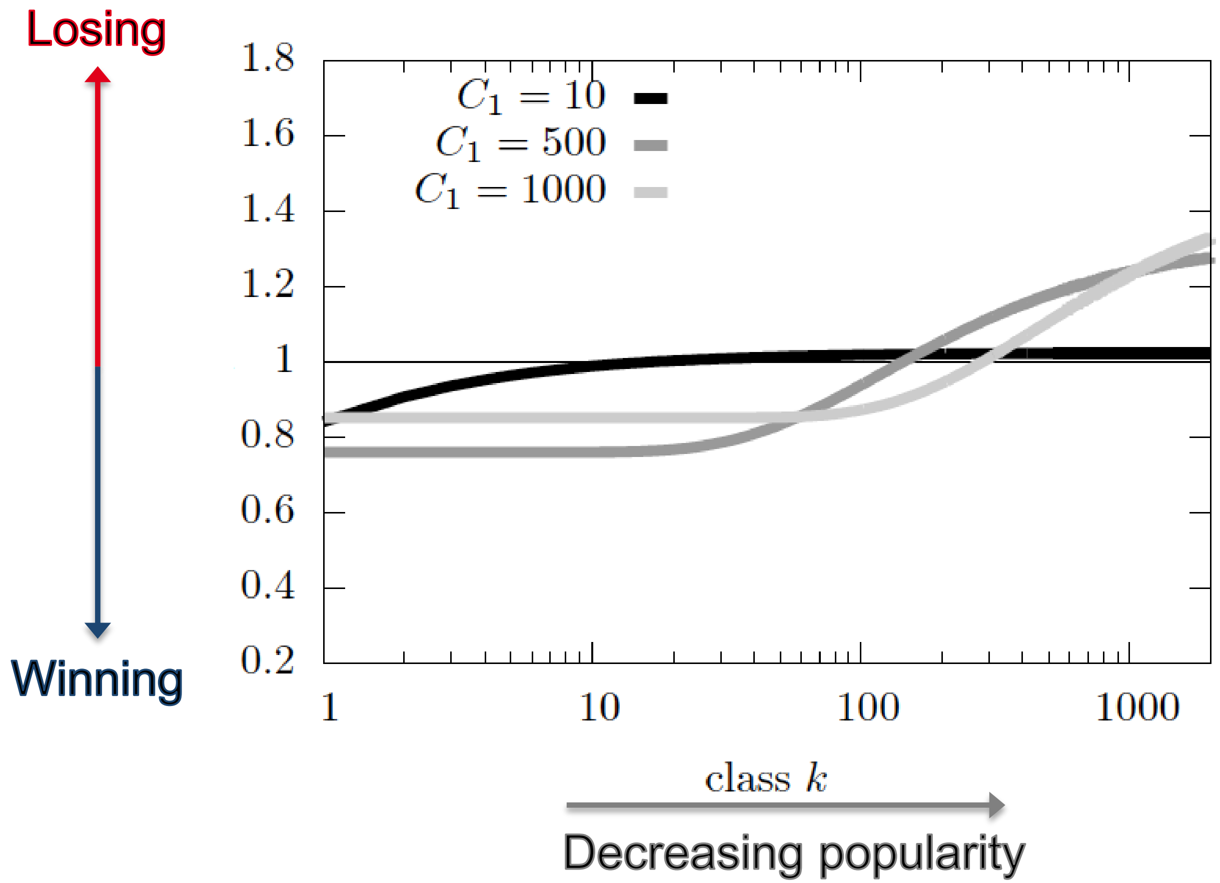


Zipf(0.8)

The case of M classes

Expansion factor vs. Popularity

- A load of 90% and a catalogue of 2000 contents

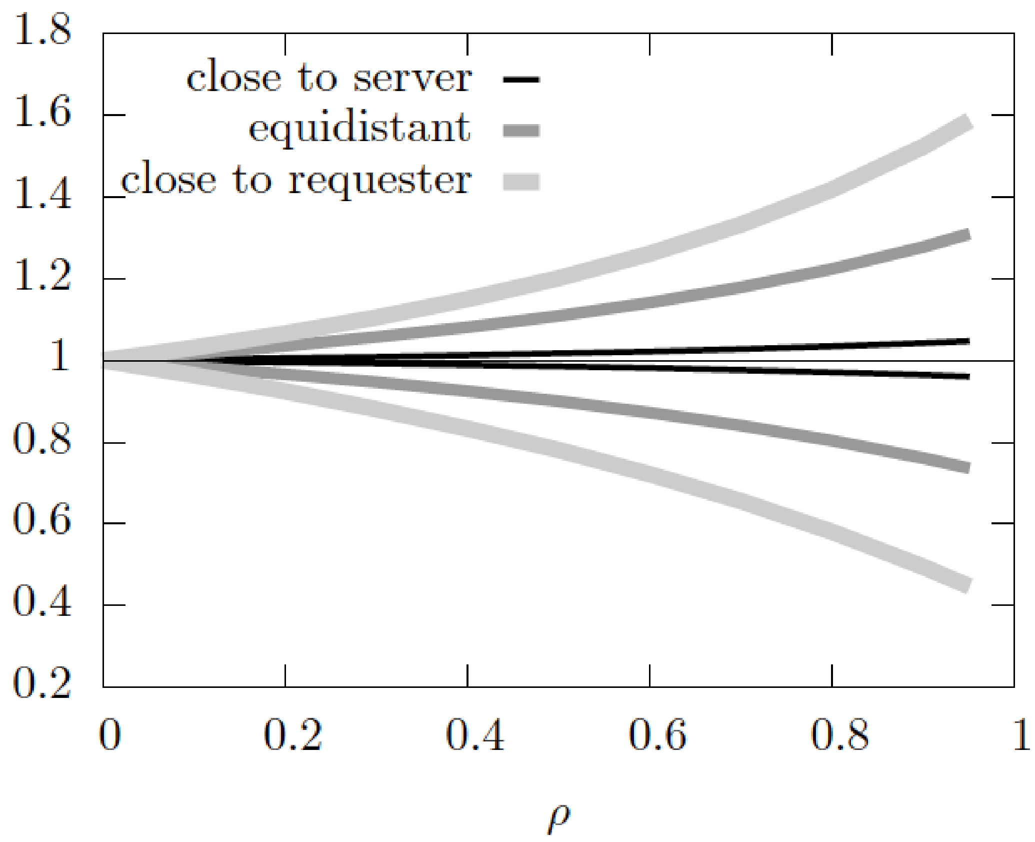


The case of M classes

Expansion factor vs. Load

- A catalogue of 2000 contents and a cache of 500 contents

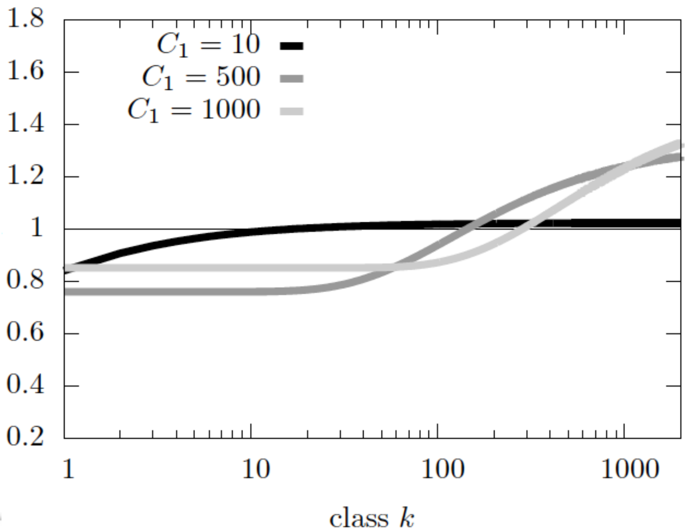
Losing
Worst case

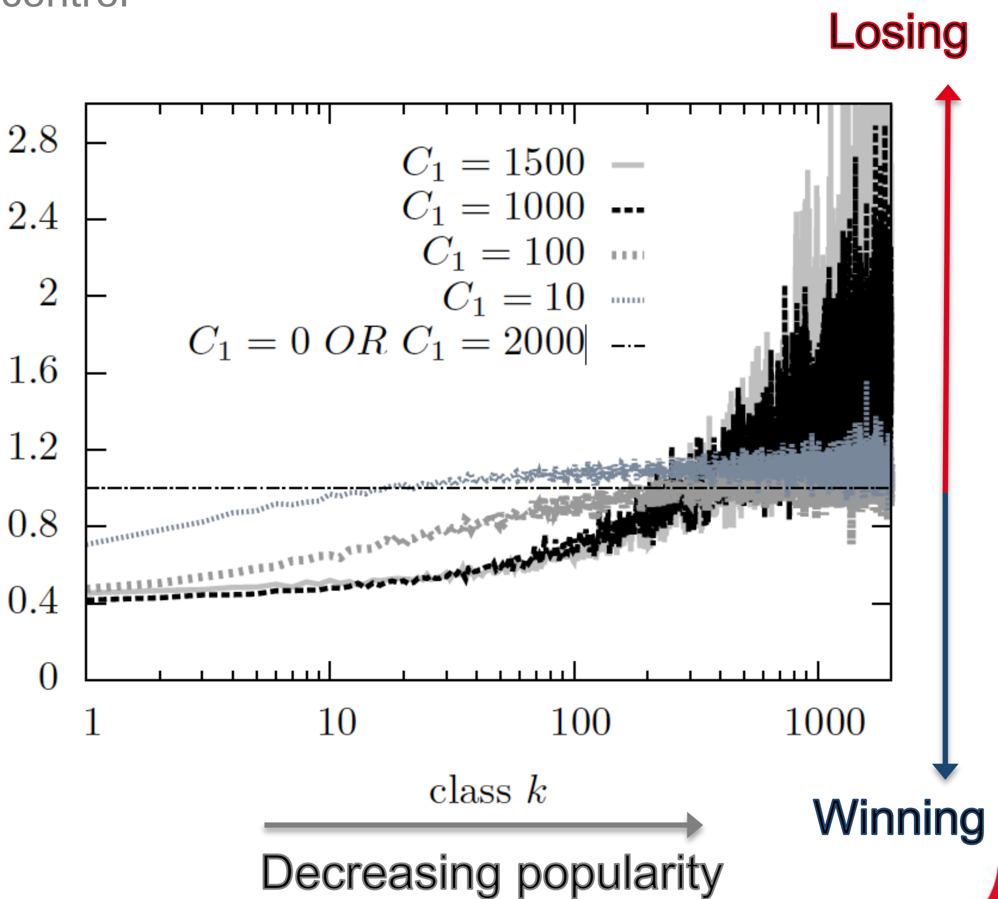


Winning
Best case

Experimental validation

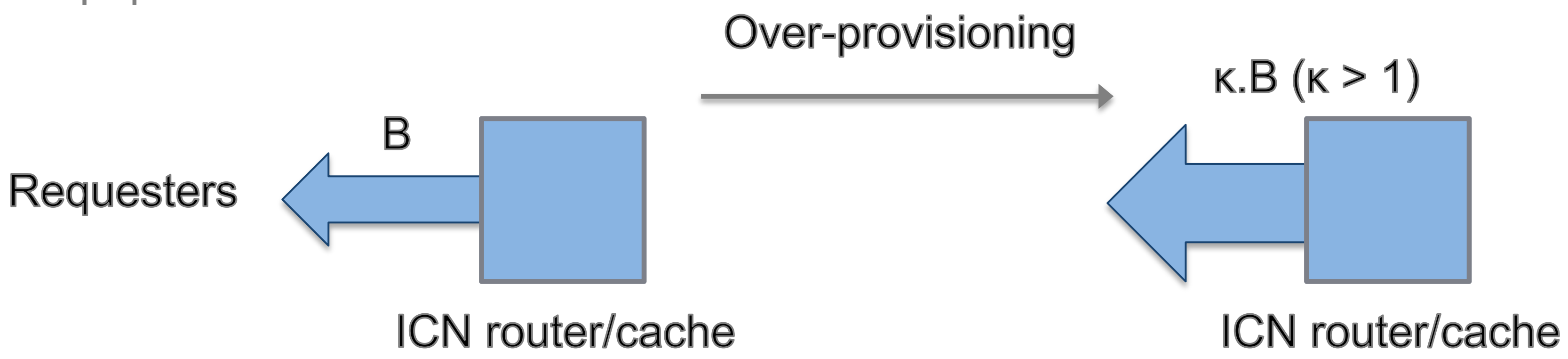
- CCN-Joker (by Poli Bari):
  - A java-based emulator of CCN
  - Implementation of AIMD congestion control
  - Congestion inferred by Timeouts
  - Selective ACKs and retransmissions
- Dummynet for net emulation
- LRU as cache policy





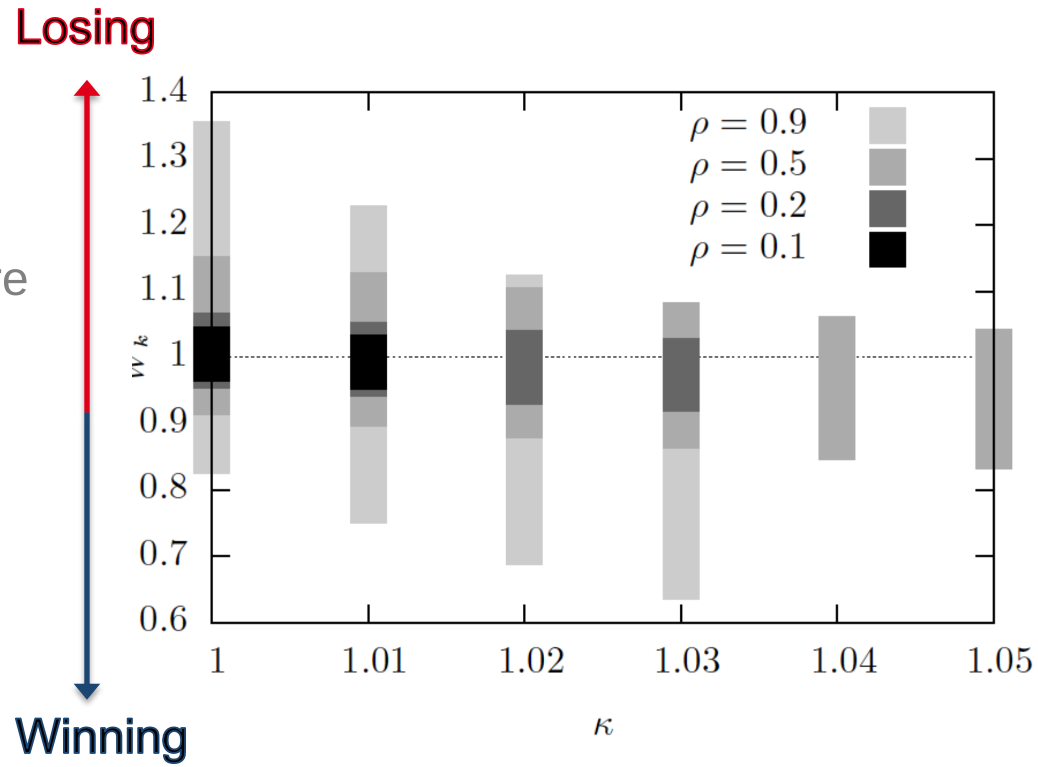
Possible solutions for removing the bias

- In-network flow isolation (scalability issues)
- Making AIMD delay independent
  - Download rate increase independent of RTT
  - Congestion window increase proportional to RTT
  - Problem modeled as DPS with same weights for all contents
- Over-provisioning the bottleneck link
  - How much extra bandwidth is needed to compensate for the greediness of popular contents?



Over-provisioning the bottleneck link

- With few percents more bandwidth, unpopular contents see same performance as in the case of “no in-network caching”
- A catalogue of 2000 and a cache of 500
- *Bandwidth provisioning rule:*
  - Make sure the network before the cache is always faster by at least few percents of the bottleneck link in the “no caching” case



Conclusion

When AIMD meets CCN

- Serious *instantaneous* unfairness issue against non popular contents
- Tempered in the long run
  - Getting 50% longer download time for unpopular contents is very likely
- Two possible solutions identified:
  - Rethinking AIMD to be delay independent
  - Over-provisioning the access network
- Analysis on average, what about variance of performance?
  - Experiments show more important loss for some contents

Thank you

chadi.barakat@inria.fr
planete.inria.fr/chadi/

INVENTEURS DU MONDE NUMÉRIQUE