Bandwidth and memory sharing in CCN: results from CONNECT

Jim Roberts, INRIA

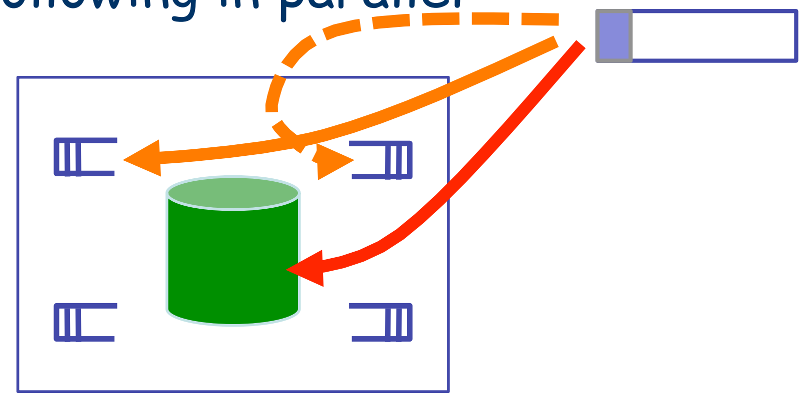
COMET-ENVISION Workshop
Slough, 10-11 November 2011

CONNECT

- a French national project (Jan 2011 – Dec 2012)
- Alcatel, Orange, INRIA, Univ Paris VI, Telecom ParisTech
- objective: consider content-centric networking, starting from the PARC design, adding missing pieces within our area of competence (traffic control, cache management,...)
- 5 work packages
  - traffic control and resource sharing
  - naming, routing and forwarding
  - caching strategies and bandwidth/memory tradeoffs
  - use cases and security
  - evaluation, experimentation
- this talk relates work from 1st and 3rd work packages

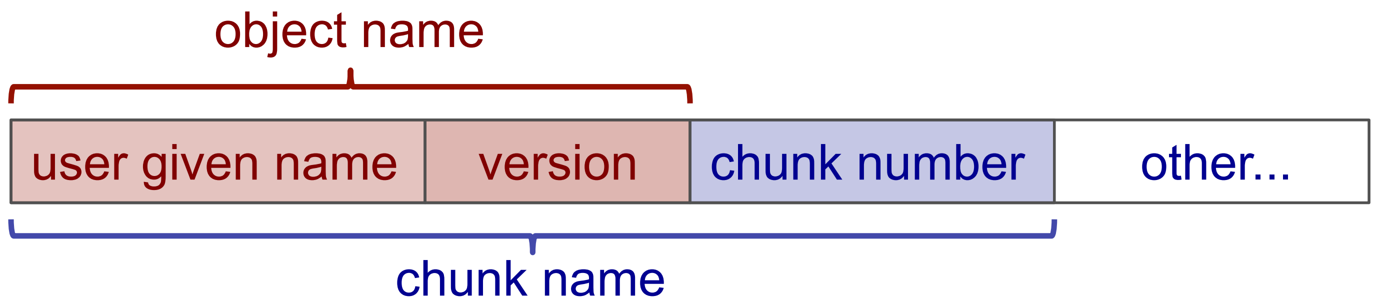
CCN traffic control

- traffic control by network mechanisms and forwarding strategies
  - to ensure low latency for real time applications
  - to control bandwidth sharing between elastic downloads
  - to enable a viable business model for the network provider
- a need to separate buffer and cache
  - a huge cache of $O(10^{12})$ bytes to significantly reduce traffic volume
  - a small buffer of $O(10^{6})$ bytes on each face for responsive traffic management
- on arrival of a Data packet do the following in parallel
  - cache, if appropriate
  - place in buffer on relevant faces
  - discard, if necessary



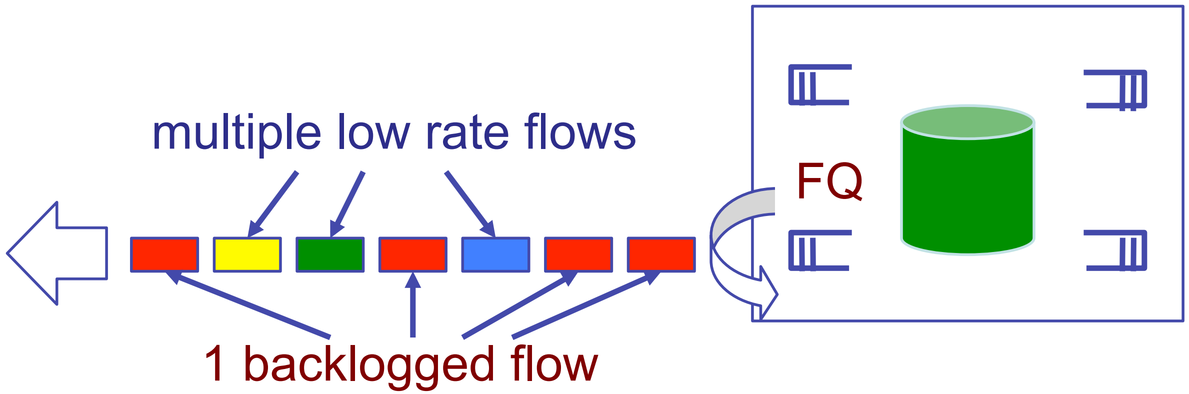
Our choice: flow-aware CCN

- identify flows by object name...
  - included in chunk name and parse-able
- ... on-the-fly, locally, e.g., at a given face



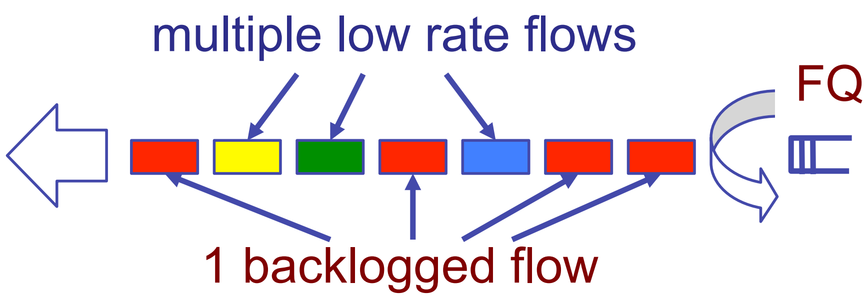
Our choice: flow-aware CCN

- identify flows by object name...
  - included in chunk name and parse-able
- ... on-the-fly, locally, e.g., at a given face
- at each face apply per-flow fair queuing
  - to ensure low latency for real time applications
  - to control bandwidth sharing between elastic downloads



Our choice: flow-aware CCN

- identify flows by object name...
  - included in chunk name and parse-able
- ... on-the-fly, locally, e.g., at a given face
- at each face apply per-flow fair queuing
  - to ensure low latency for real time applications
  - to control bandwidth sharing between elastic downloads
- a provably scalable mechanism: O(100) active flows at load < 90%
  - under a realistic model of dynamic traffic
  - "active flows" have 1 or more packets in buffer
  - load = flow arrival rate × mean size / link rate

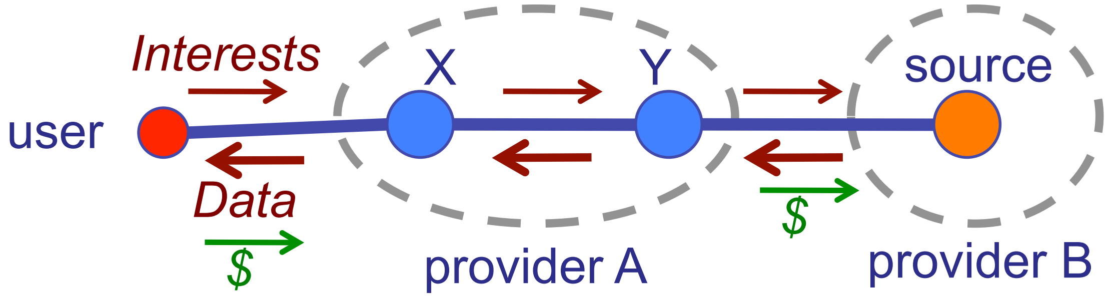


Our choice: flow-aware CCN

- identify flows by object name...
  - included in chunk name and parse-able
- ... on-the-fly, locally, e.g., at a given face
- at each face apply per-flow fair queuing
  - to ensure low latency for real time applications
  - to control bandwidth sharing between elastic downloads
- a provably scalable mechanism: O(100) active flows at load < 90%
  - under a realistic model of dynamic traffic
  - "active flows" have 1 or more packets in buffer
  - load = flow arrival rate × mean size / link rate
- traffic engineering and overload control required to ensure load < 90%

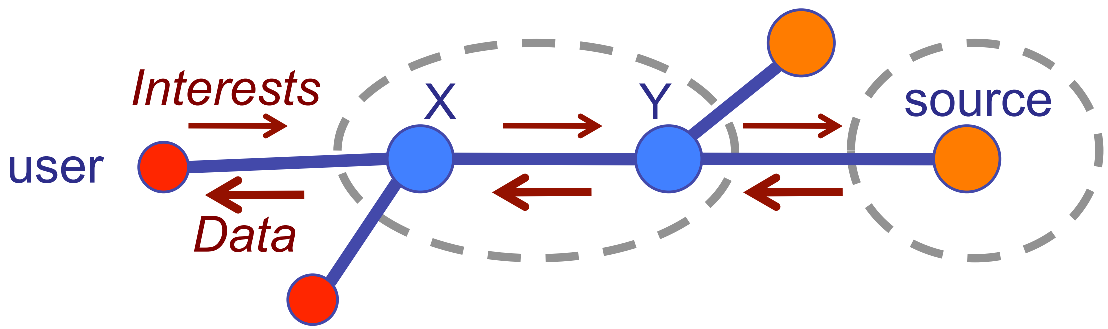
Paying for transport

- a proposed direction of charging: Interests "buy" Data
  - user pays provider A, A pays provider B,..., for *delivered* Data
  - not excluding flat rates, peering...
- brings return on investment and incentive to invest
  - in transmission capacity (to be able to sell Data)
  - in cache memory to avoid paying repeatedly for popular content
- no charge for Interests but an incentive to avoid buying Data that can't be delivered due to congestion...
- ... by discarding excess Interests
  - using FQ scheduler status to determine excess



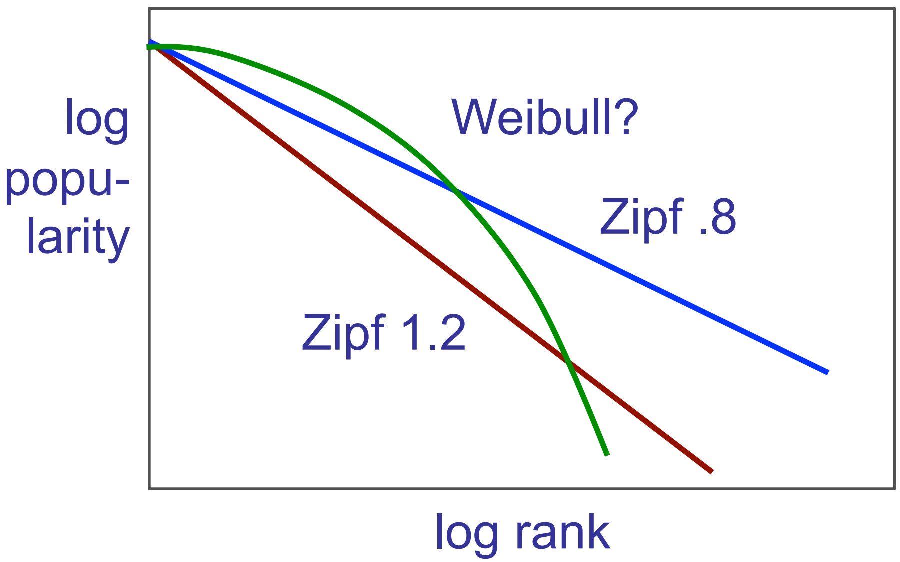
Forwarding strategies

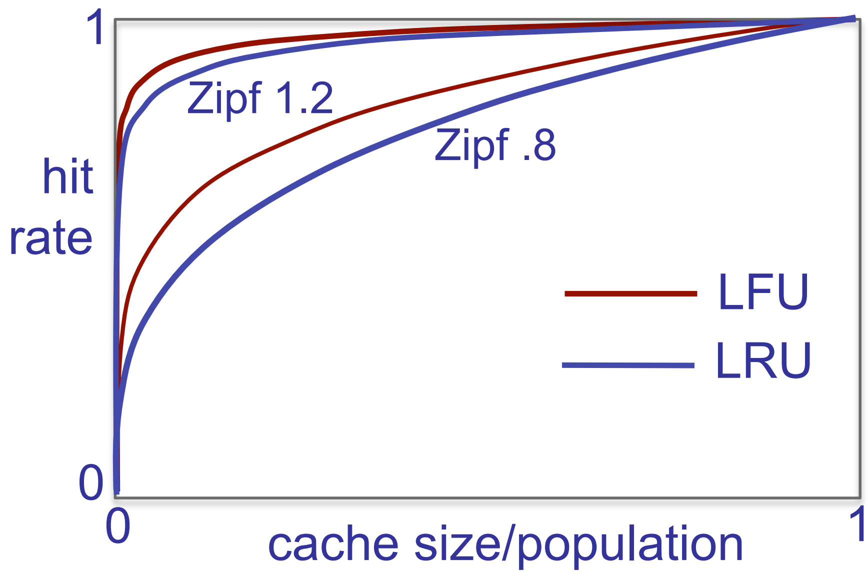
- network performance is broadly independent of user strategies in emitting Interests
  - greedy strategies are OK (e.g., using source coding)
  - AIMD avoids unnecessary end-system complexity
- multicast and multipath forwarding work OK with fair queuing
  - *provided* multicast streams are in cache
  - *provided* multipath intelligently avoids long paths
- enhance CCN with explicit congestion notification: discard *payload* if necessary but return the *header*
  - limits PIT size in routers and end-systems



Cache performance: re-visiting the literature

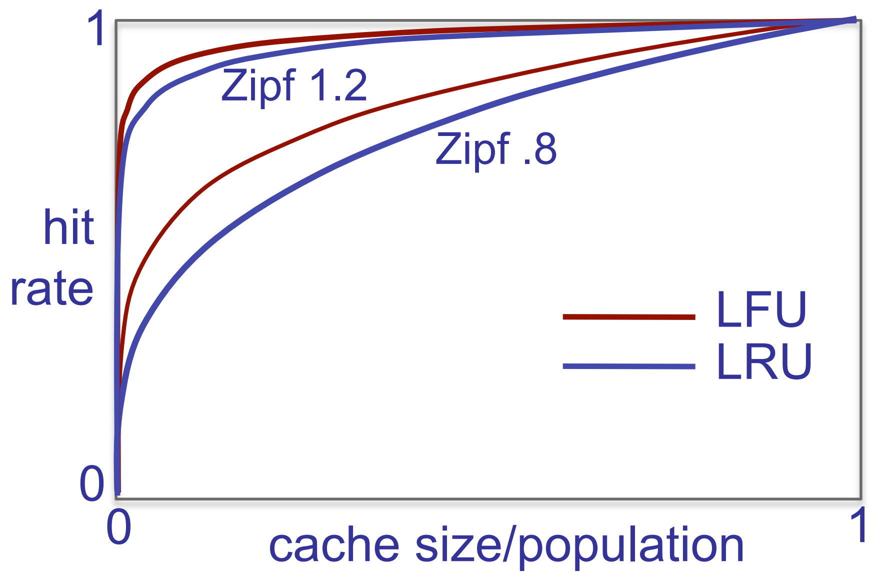
- popularity distributions: Zipf ($\sim 1/i^{\alpha}$), $\alpha<1$ or $\alpha>1$, other laws
- replacement policies: LFU, LRU, LRU with filters, random,...
- hit rate estimates: Flajolet, Jelenkovic, Gelenbe, Che,...





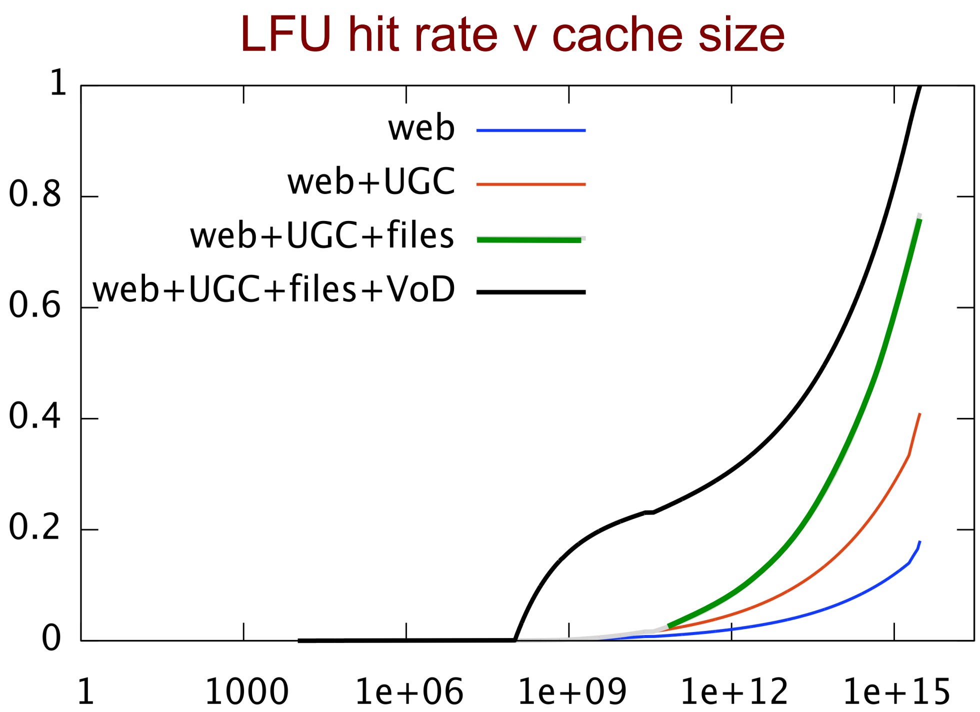
Rules of thumb...

- populations (approx)
  - web 10^{11} x 10 KB
  - UGC 10^{8} x 10 MB
  - file sharing 10^{5} x 10 GB
  - VoD 10^{4} x 100 MB
- very large cache needed for web, UGC, file sharing
  - popularity ~ Zipf .8
  - population ~ 1 PB
  - cache ~ 10-100 TB
- small cache enough for VoD
  - popularity ~ Zipf 1.2 (?)
  - population ~ 1 TB
  - cache ~ <1 TB



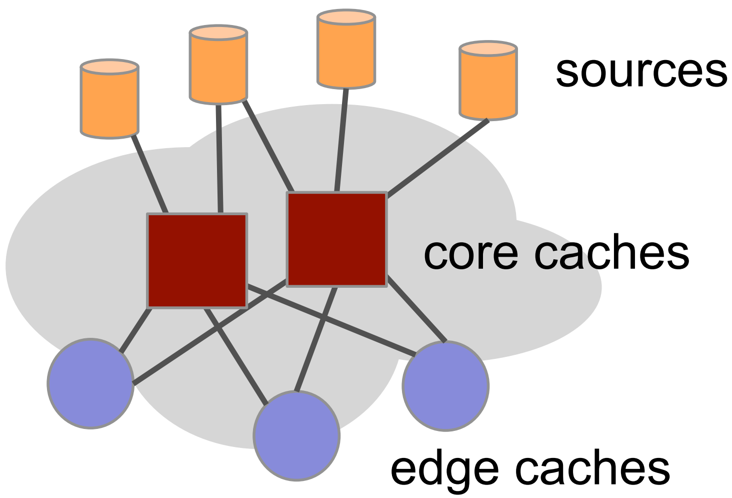
Cache sharing

- cache partitions for service differentiation
  - careful static partitions for optimal bandwidth savings...
  - ... but dynamic partitions are OK and ensure maximal cache utilization
  - cf. ICC 2011 paper by Carofiglio et al.
- fully shared cache, web, file sharing, UGC, VoD
  - cache mainly used by VoD unless very large



Networks of caches

- a cache hierarchy
  - all routers have cache (as proposed in CCN)?
  - or small caches at edge and large data centres in the core?
- cache coordination
  - LRU everywhere brings too much duplication
  - LRU at lower level, MRU at higher level is better
  - need for optimized placements?
- analytical models
  - evolution of popularity distributions
  - impact of correlation



Work in progress

- multipath routing
  - simulations show impact of topology, popularity, cache policies
  - first results: limited impact of topology, simple randomized policies efficient, strongest impact from population size and popularity distribution
  - open source simulator
- multicast using digital fountains (not CCN)
  - periodic interest packets, source coding, congestion control using packet loss rate indications
  - performance depends on popularity distribution
- transport
  - design of receiver-based CCN transport protocols
  - Interest flow shaping to alleviate congestion

Publications

- G. Carofiglio, M. Gallo, L. Muscariello, D.Perino Modeling data transfer in content-centric networking
  - Proc. of 23rd International Teletraffic Congress, ITC23 San Francisco, CA, USA, 2011.
- G. Carofiglio, M. Gallo, L. Muscariello, Bandwidth and storage sharing performance in information-centric networking
  - SIGCOMM workshop on information-centric networking, Toronto, 2011.
- D. Perino and M. Varvello, A reality check for content-centric networking,
  - SIGCOMM workshop on information-centric networking, Toronto, 2011.
- G. Carofiglio, V. Gehlen, D. Perino, Experimental evaluation of storage management in Content-Centric Networking,
  - IEEE ICC 2011, Kyoto, Japan.
- M. Diallo, S. Fdida, V. Sourlas, P. Flegkas, L. Tassiulas, Leveraging caching for Internet-scale content-based publish/subscribe networks,
  - IEEE ICC 2011, Kyoto, Japan.

Conclusions

- flow-aware networking is a complete traffic control for CCN
- "Interests buy Data" implies a rational direction of charging
  - some requirements: object name in packet headers, fair queuing in face buffers
  - some enhancements: Interest discard, explicit congestion notification
- cache management is the key to efficient content distribution
  - small (TB) caches good for VoD but not for other content types
  - larger caches (PB) in core might mean CDN-like solutions (not CCN using data centres
- ongoing developments in CONNECT
  - forwarding & cache management strategies, experimental evaluations, links with naming and routing, CCN use cases