ICN traffic control and network performance

Jim Roberts (Inria)
james.roberts@inria.fr

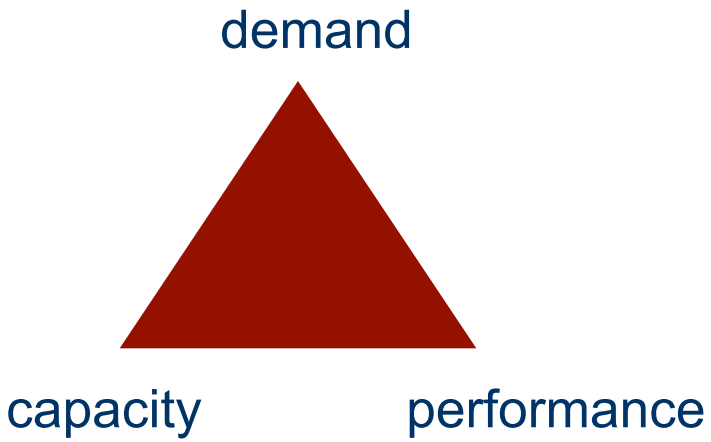
Rescom, May 2013

Traffic control is...

- how to realize performance objectives
- at least cost
- accounting for traffic characteristics (multi-service)
- by traffic engineering...
- and resource sharing mechanisms
    - congestion control
    - queue management
    - overload controls
    - ...
- i.e., traffic control is the basis for quality of service (QoS)

The dual role of QoS

- perform effective traffic control
  - to meet diverse application requirements
  - for delay, jitter, loss, throughput,...
- create a viable business model for the network operator
  - a range of differentiated services
  - to maximize revenue and avoid commoditization
- a source of some confusion!

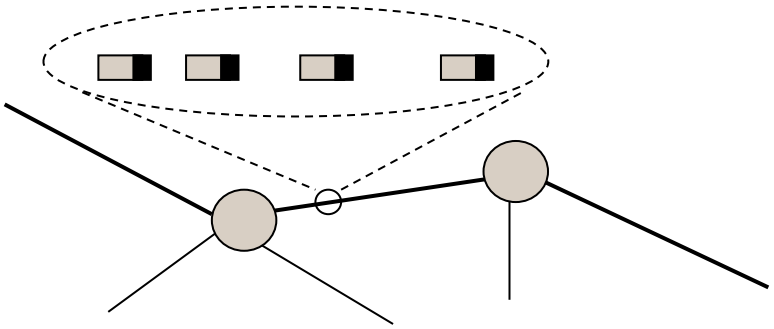


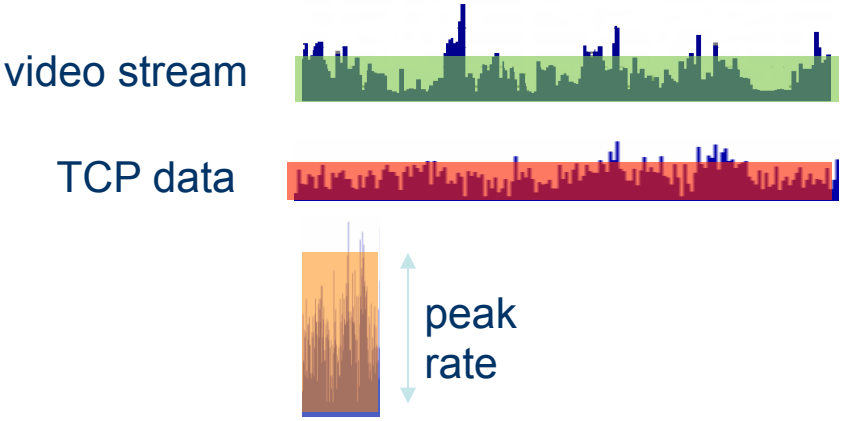
Outline

- Internet traffic control
- ICN traffic control
- Impact of in-network caching

Understanding traffic at flow level

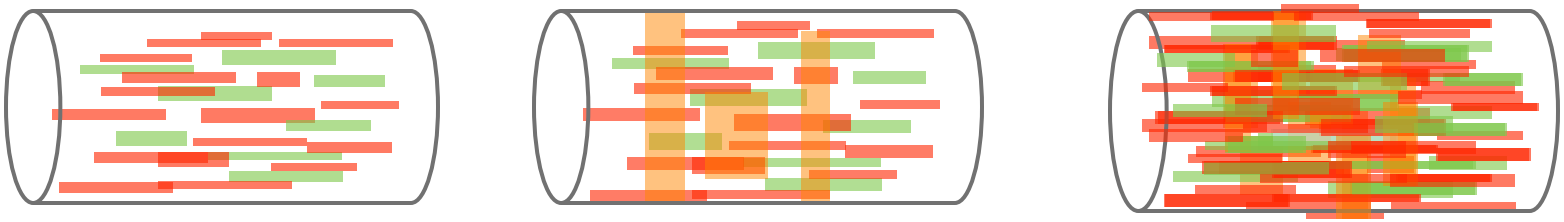
- users experience quality at flow level
  - a flow is an instance of some application (document transfer, voice signal,...)
  - a set of packets with like header fields, local in space and time
- flows of four types
  - conversational, streaming, interactive data, background
  - with different requirements for latency, integrity, throughput
  - and different traffic characteristics – rates, volumes,...





Bandwidth sharing regimes

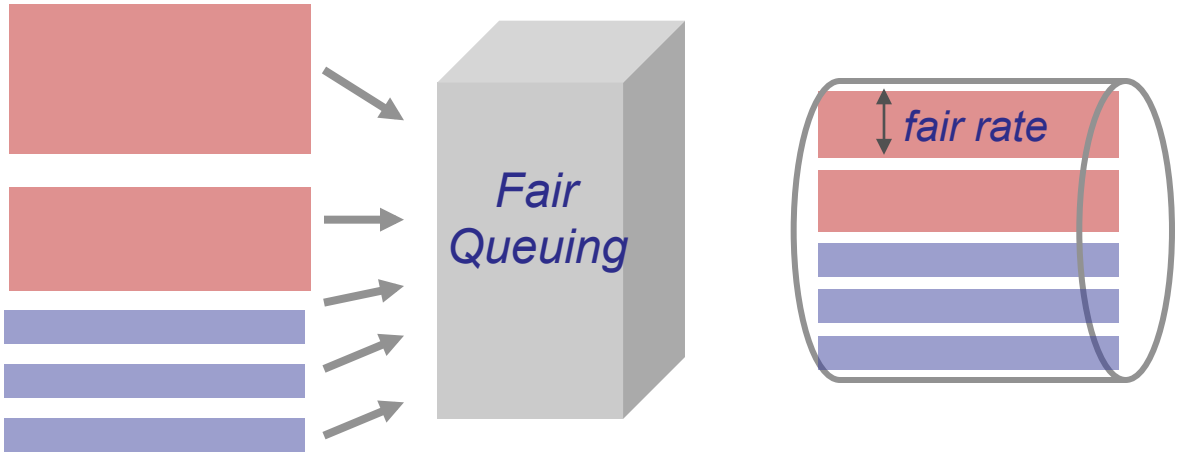
- three regimes



- transparent regime:
  - all flows suffer negligible loss, no throughput degradation
- elastic regime:
  - some high rate flows can saturate residual bandwidth; without control these can degrade quality for all other flows
- overload regime:
  - traffic load is greater than capacity; all flows suffer from congestion unless they are handled with priority

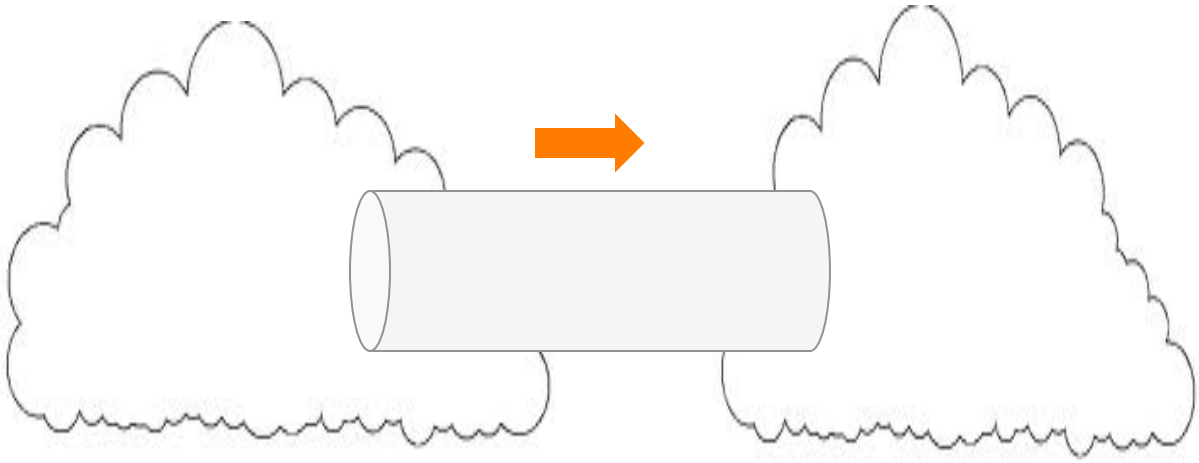
Internet traffic control

- rather than Intserv, DiffServ, MPLS TE,…
  routers should impose per-flow fair sharing and not rely on end-system implemented congestion control
- fair queuing is feasible and scalable…
  … and realizes implicit service differentiation…
  … for network neutral traffic control
- fairness is an expedient, not a socio-economic objective

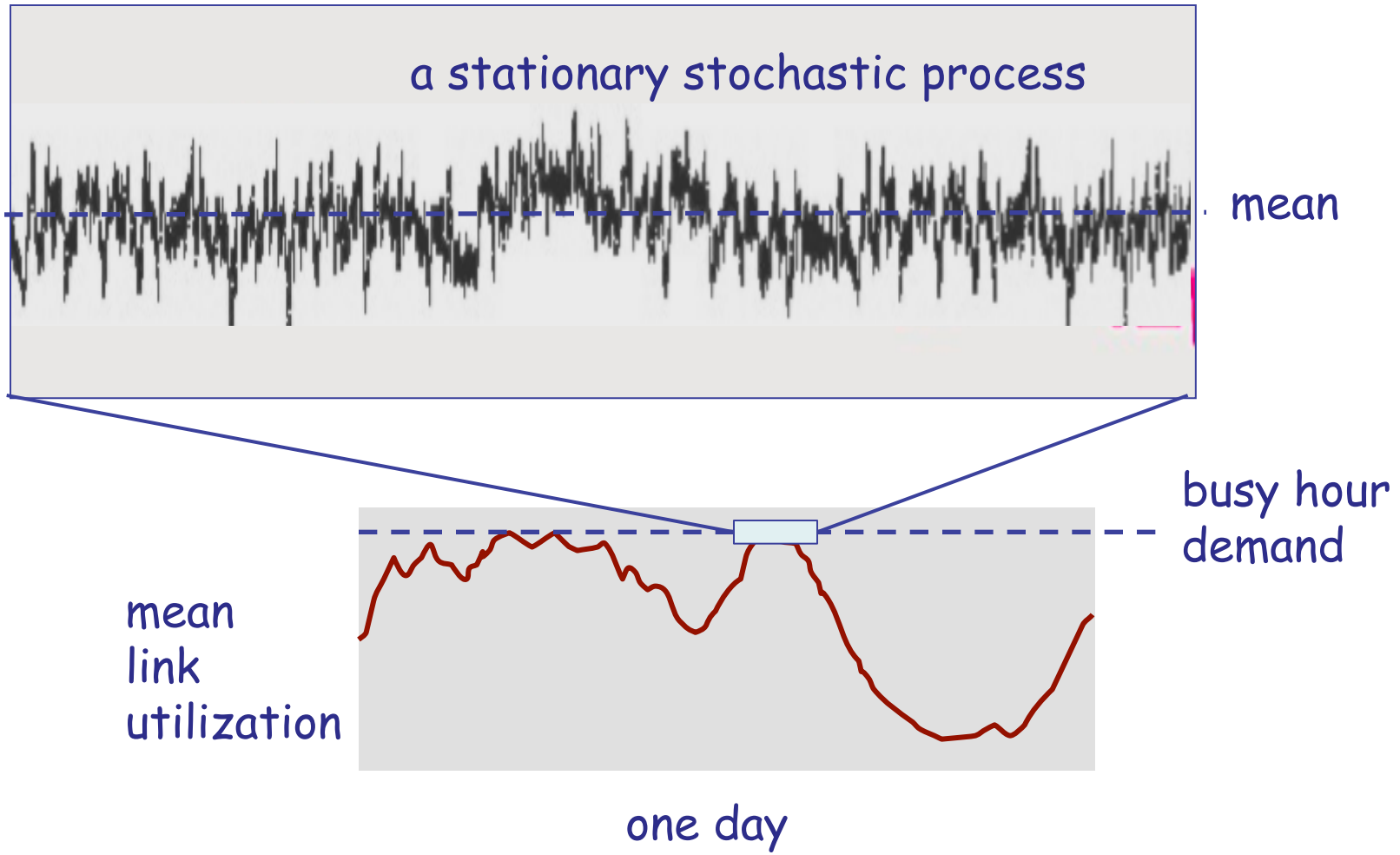


Statistical bandwidth sharing

- ie, statistical multiplexing with elastic traffic
- consider a network link handling flows between users, servers, data centers,...
- define, link load = flow arrival rate x mean flow size / link rate
  = packet arrival rate x mean packet size / link rate
  = link utilization

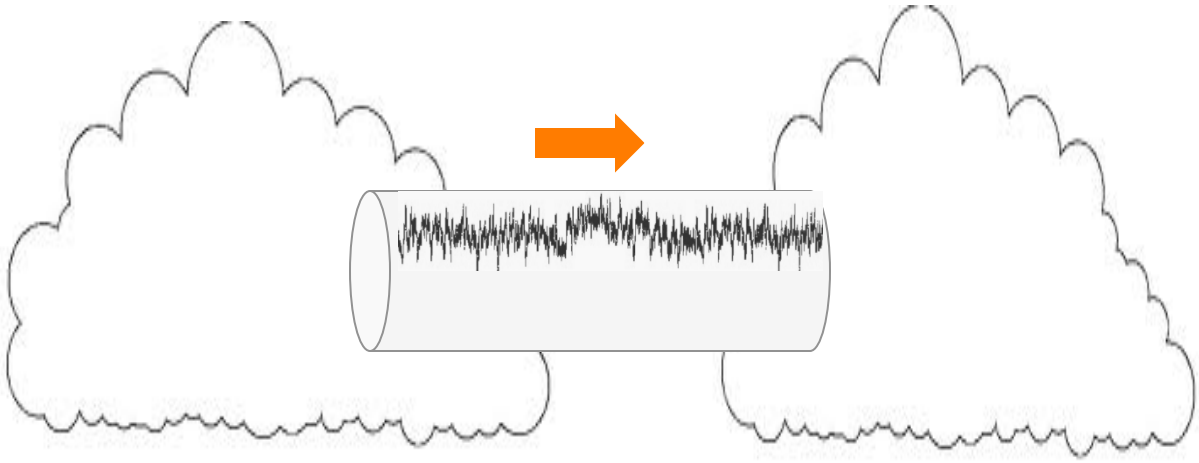


Traffic variations and stationarity



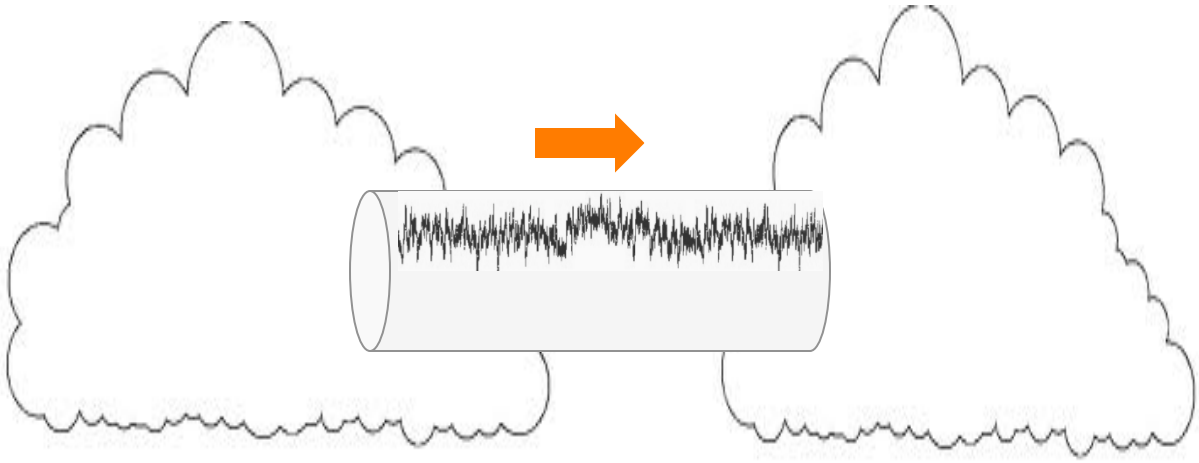
Statistical bandwidth sharing

- ie, statistical multiplexing with elastic traffic
- consider a network link handling flows between users, servers, data centers,...
- define, link load = flow arrival rate x mean flow size / link rate
  = packet arrival rate x mean packet size / link rate
  = link utilization

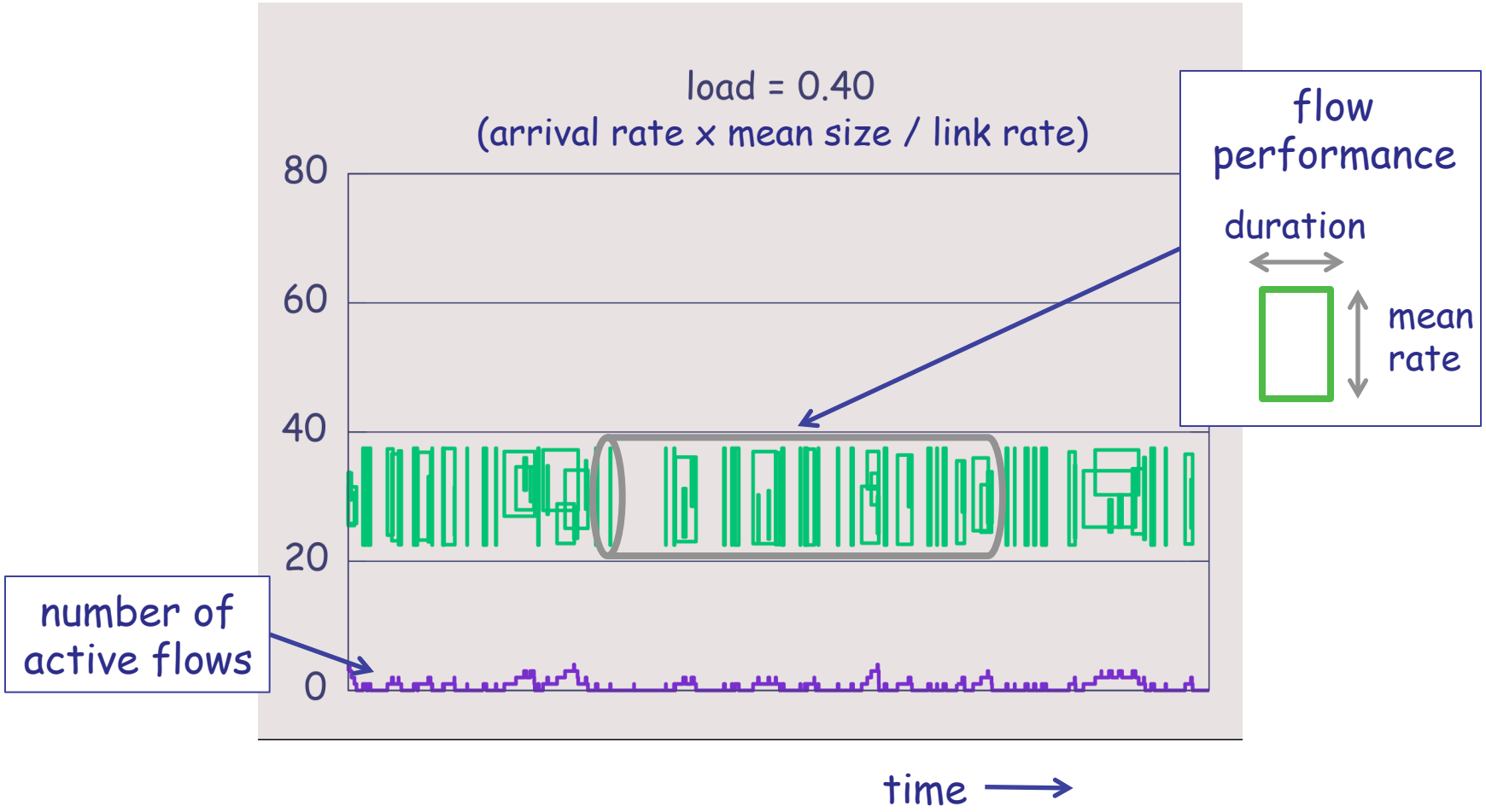


Bandwidth sharing performance

- in the following simulation experiments, assume flows
  - arrive as a Poisson process
  - have exponential size distribution
  - instantaneously share link bandwidth fairly
- results apply more generally thanks to insensitivity

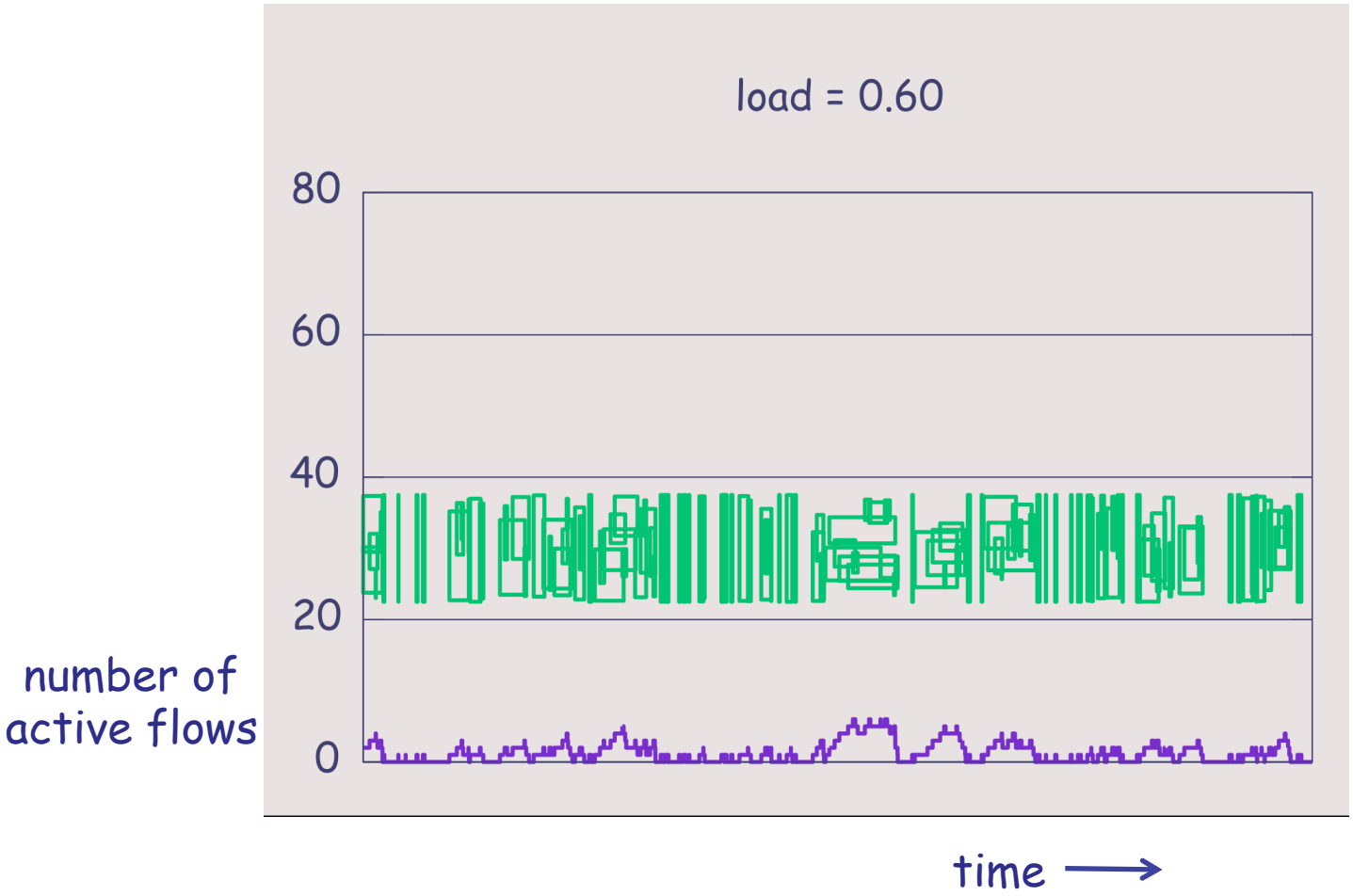


Performance of fair shared link

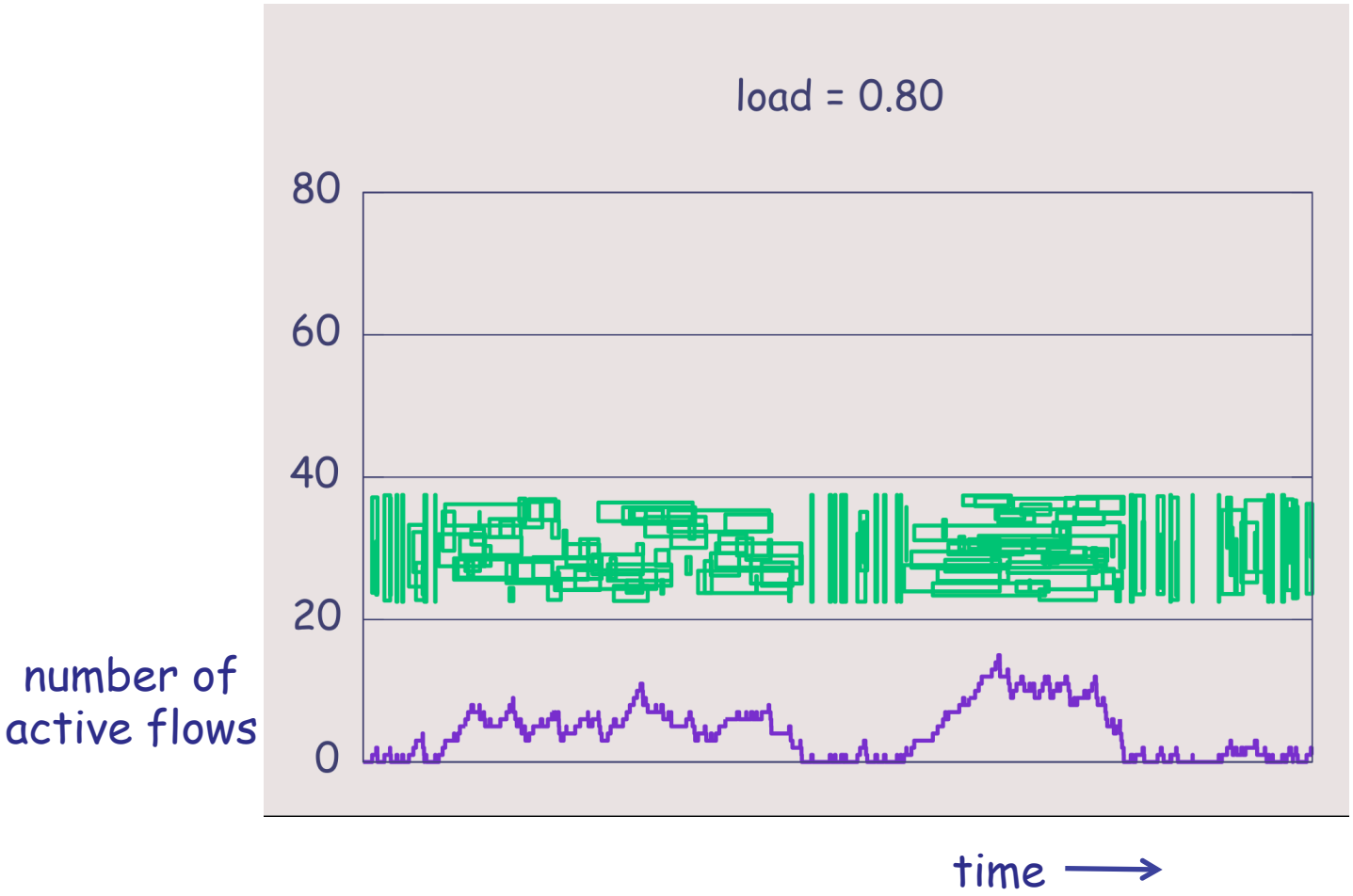


Performance of fair shared link

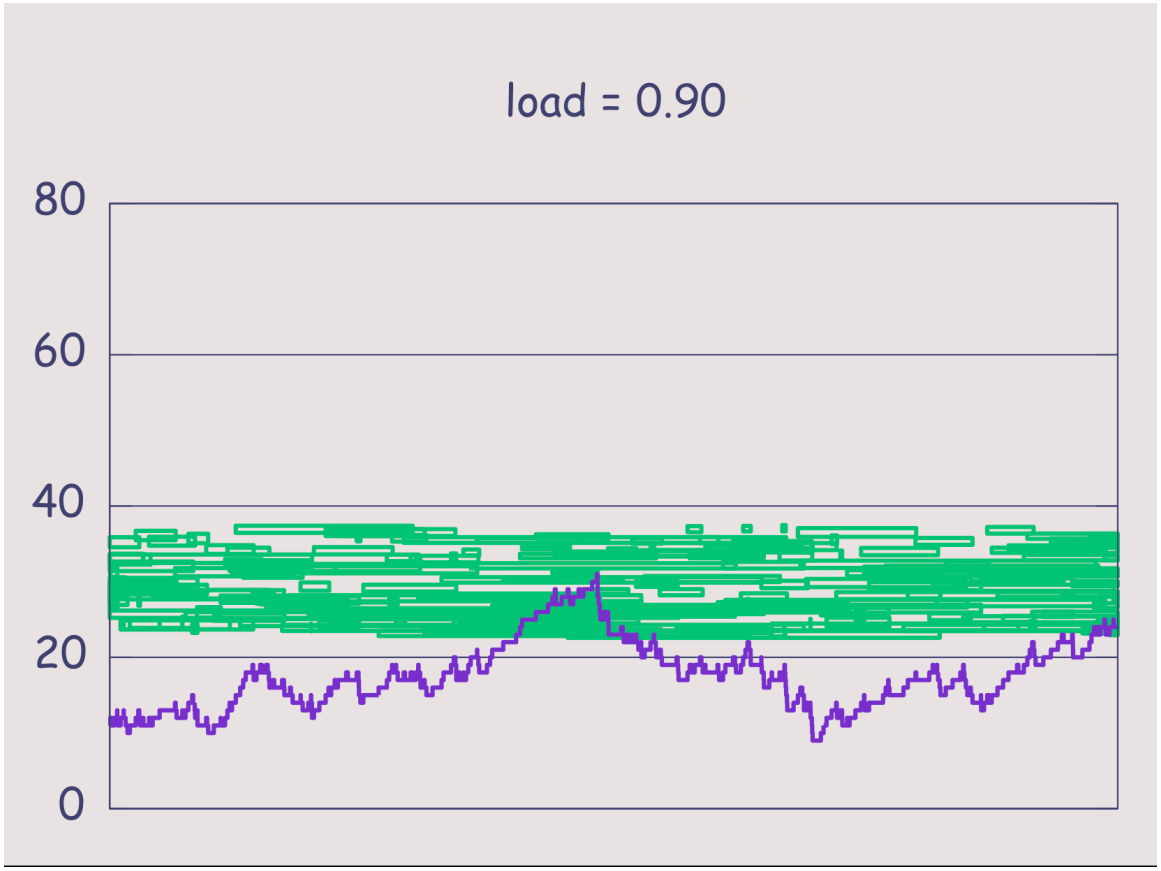
Performance of fair shared link



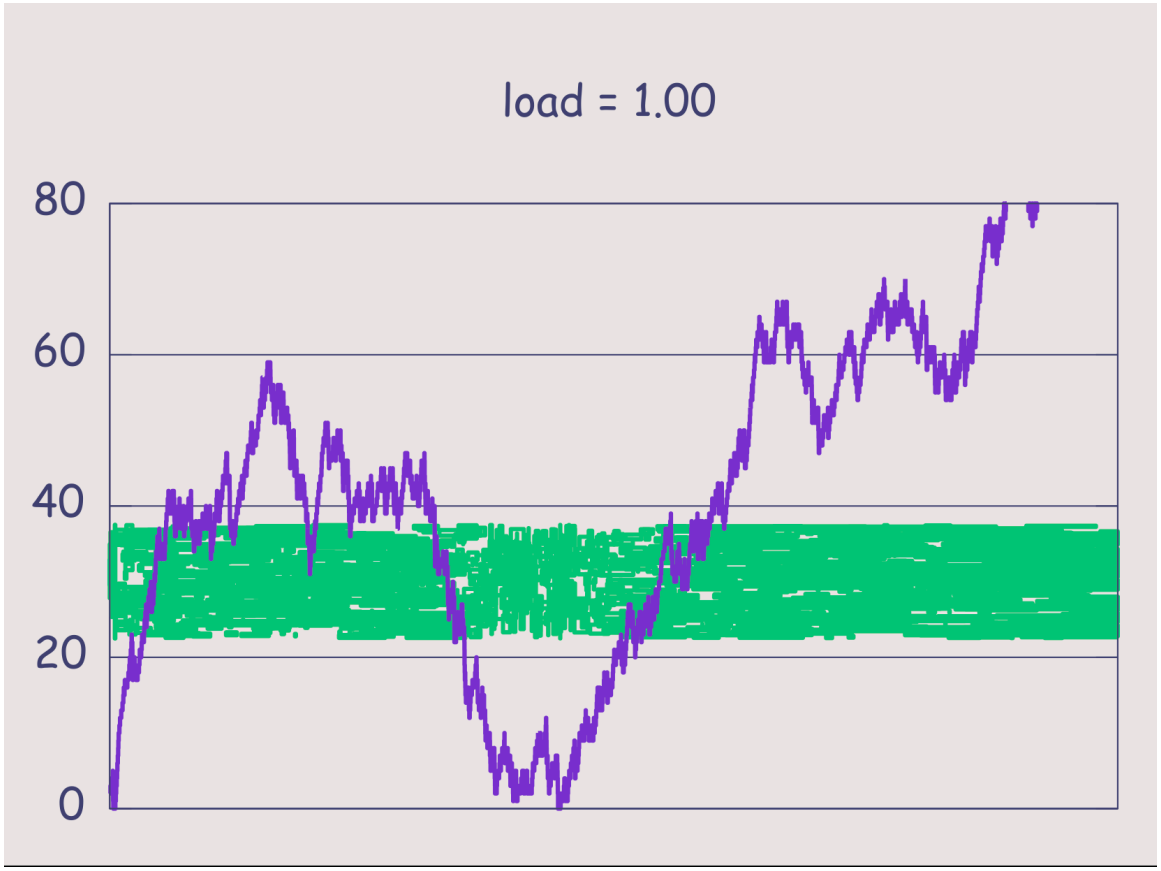
Performance of fair shared link



Performance of fair shared link

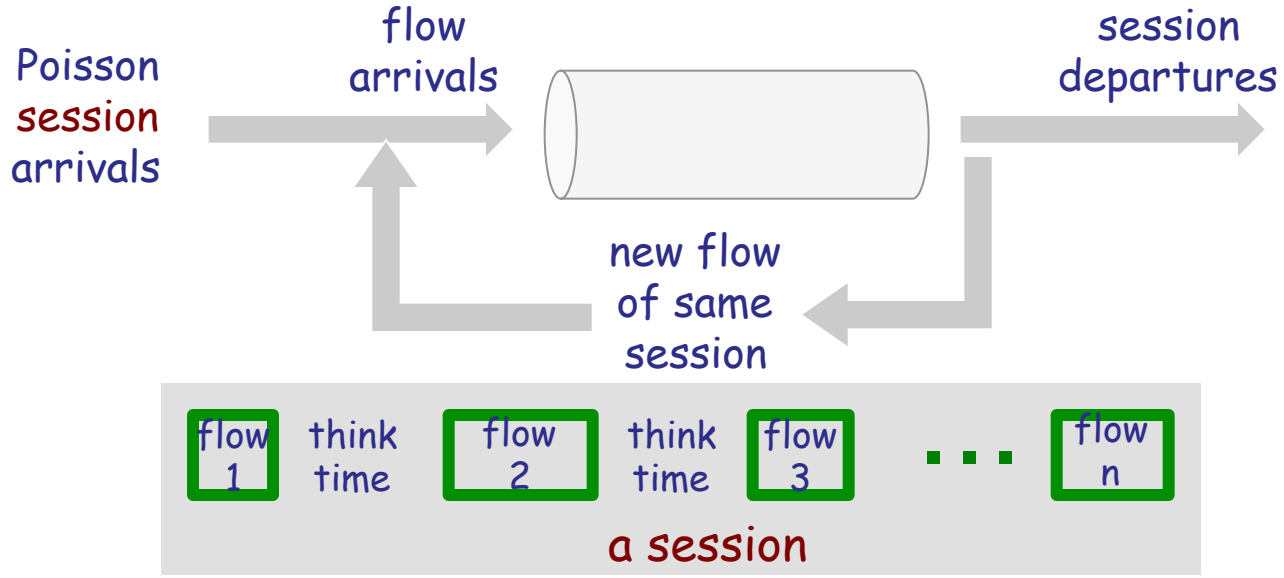


Performance of fair shared link



Observations

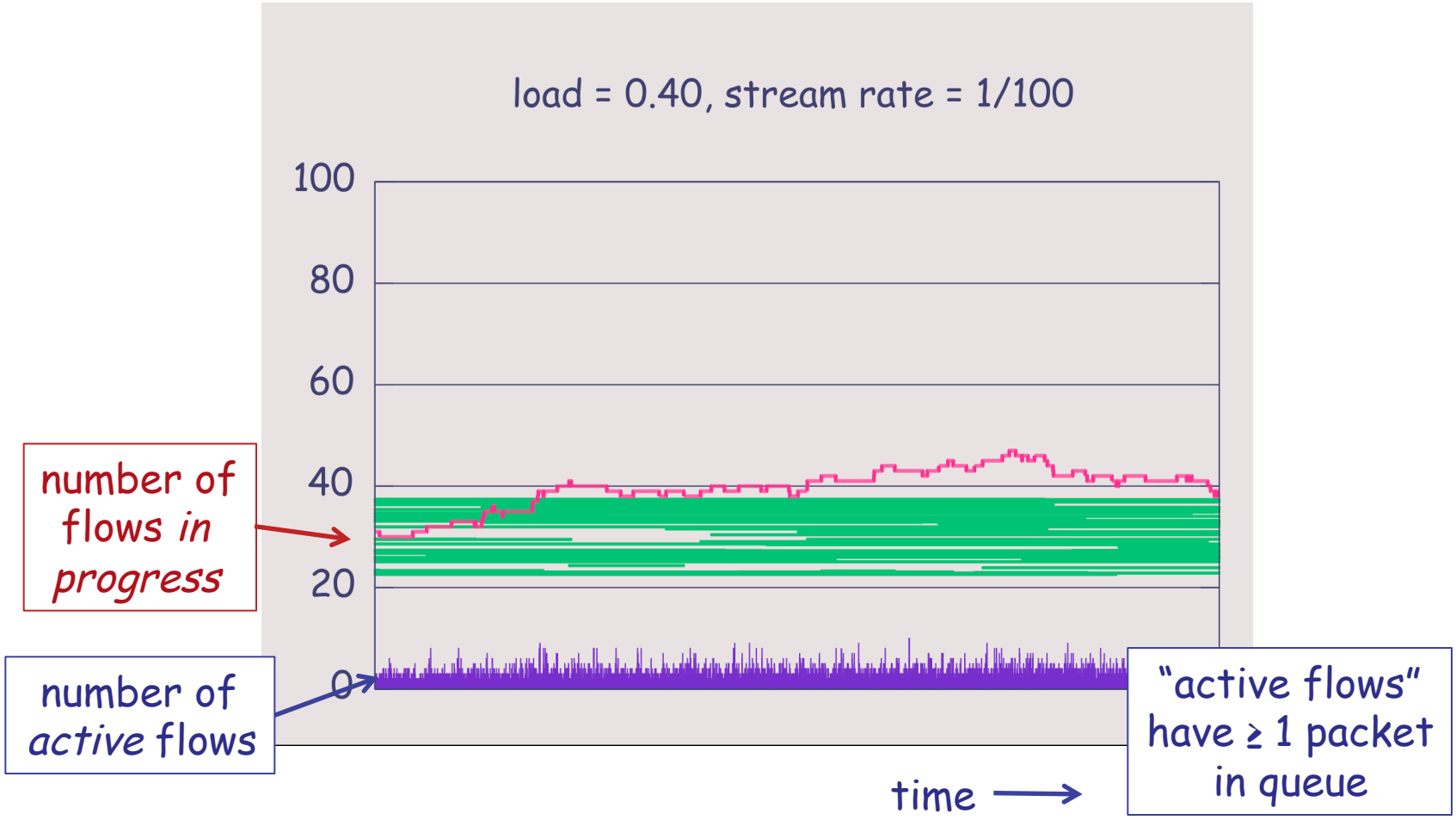
- the number of flows using a fairly shared link is small until load approaches 100% (*for any link capacity*)
- therefore, fair queuing schedulers are feasible and scalable
- our simulations make Markovian assumptions but the results for the number of active flows are true for much more general traffic [Ben-Fredj et al, Sigcomm 2001]



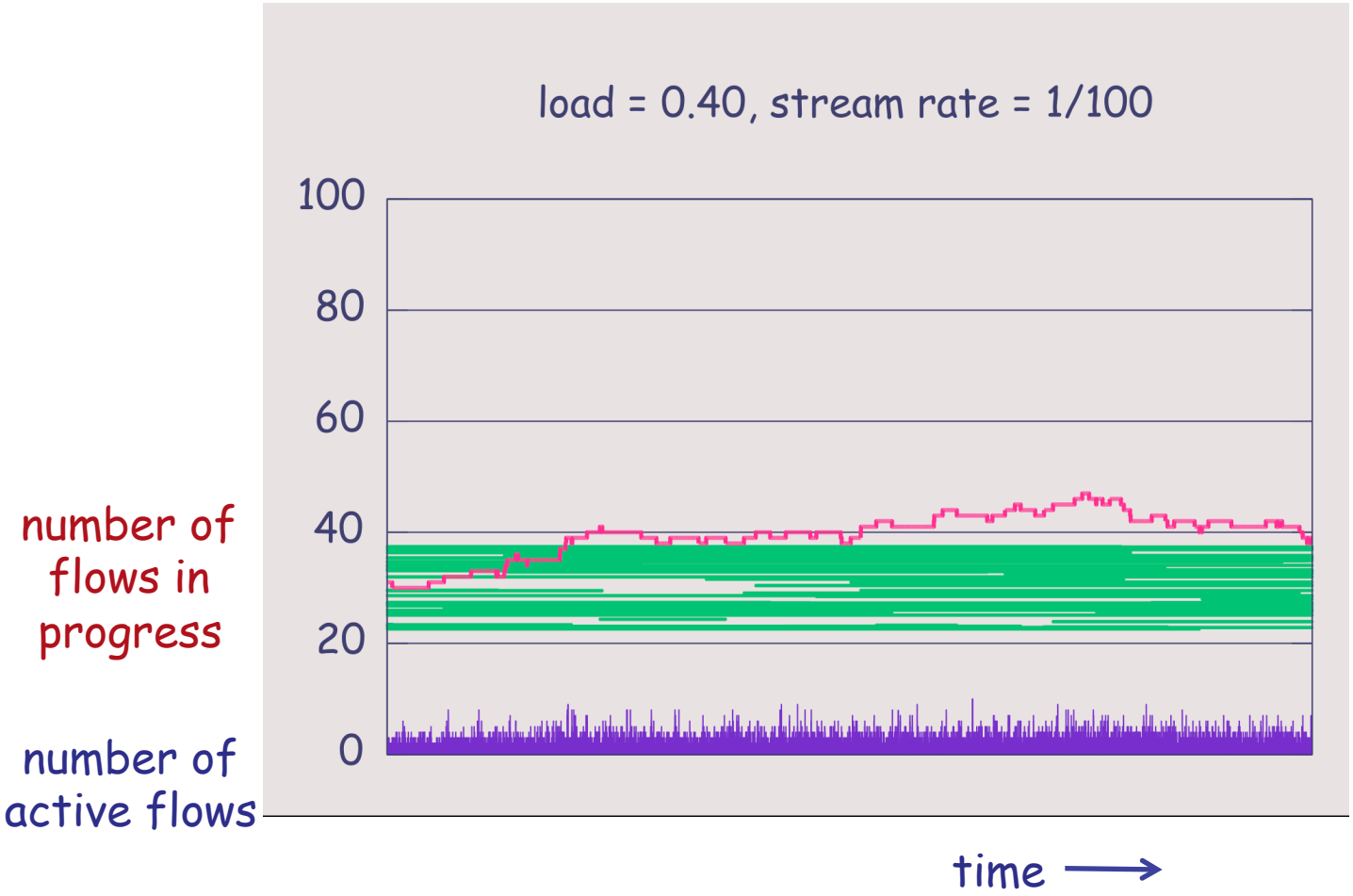
More simulations

- on Internet core links (≥ 10 Gbps), the vast majority of flows cannot use all available capacity; their rate is constrained elsewhere on their path (eg, ≤ 10 Mbps)
- consider a link shared by flows whose maximum rate is only 1% of the link rate
  - conservatively assume these flows emit packets as a Poisson process at rate proportional to the number of flows in progress

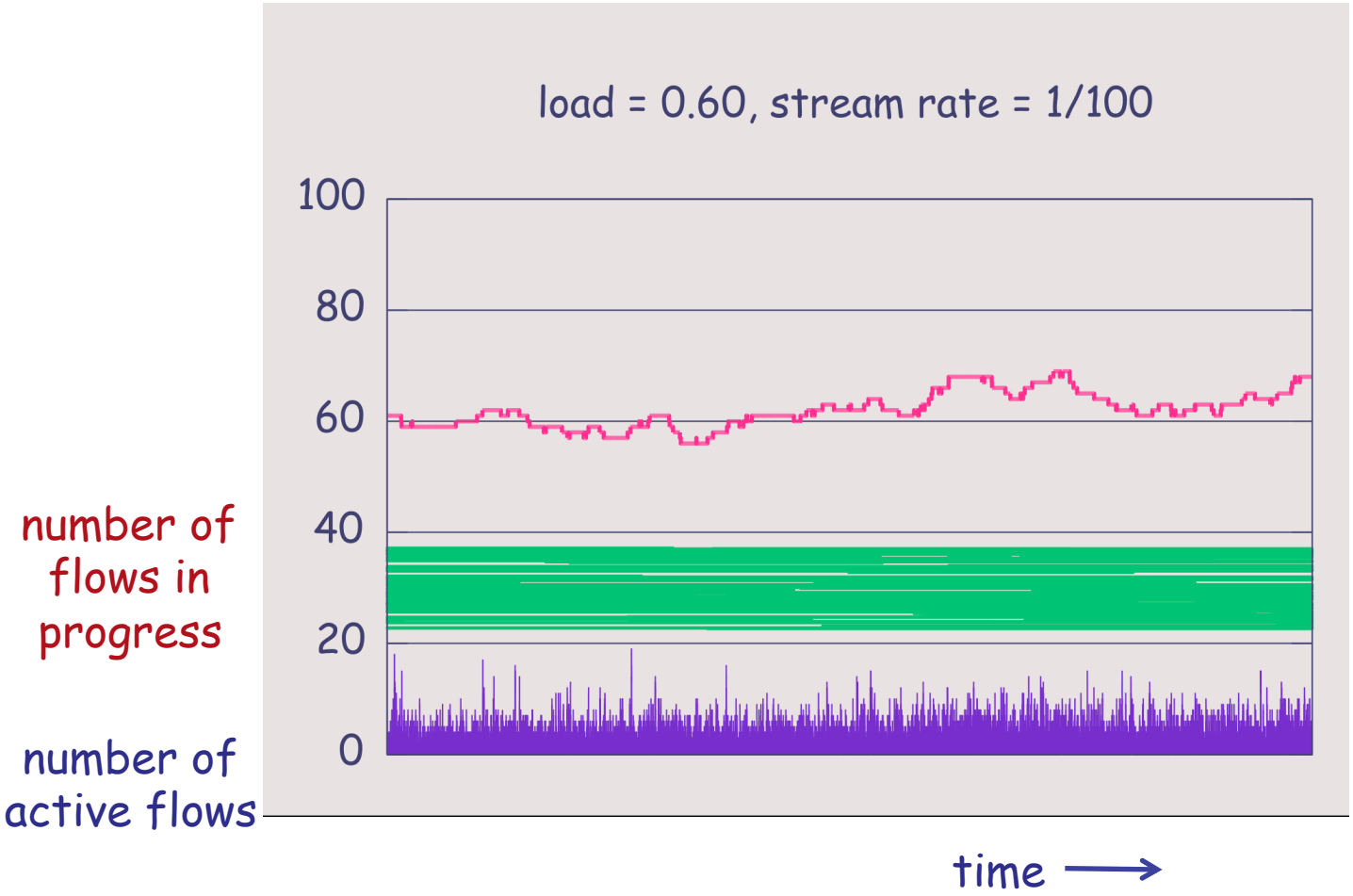
Performance with rate limited flows



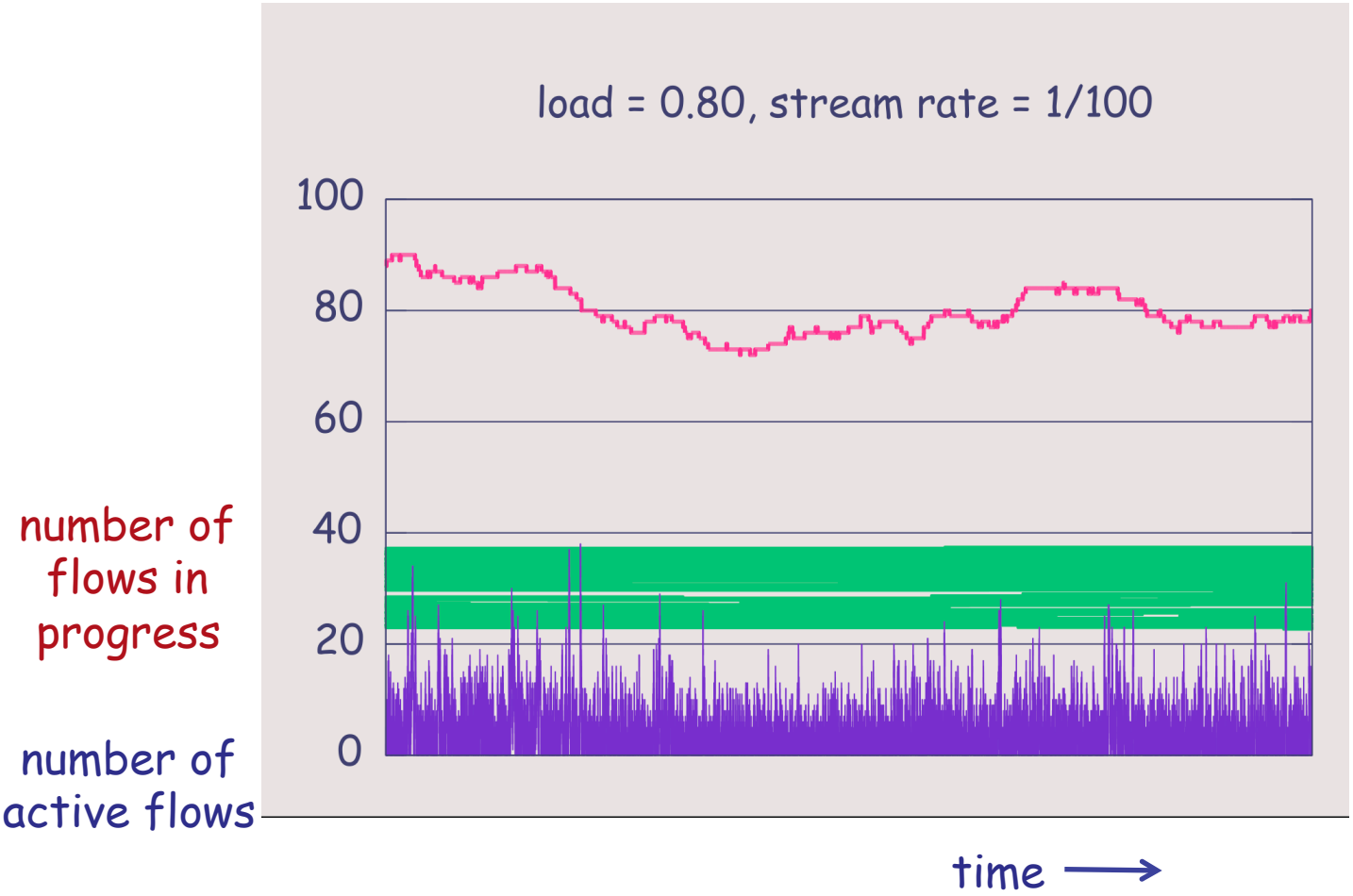
Performance with rate limited flows



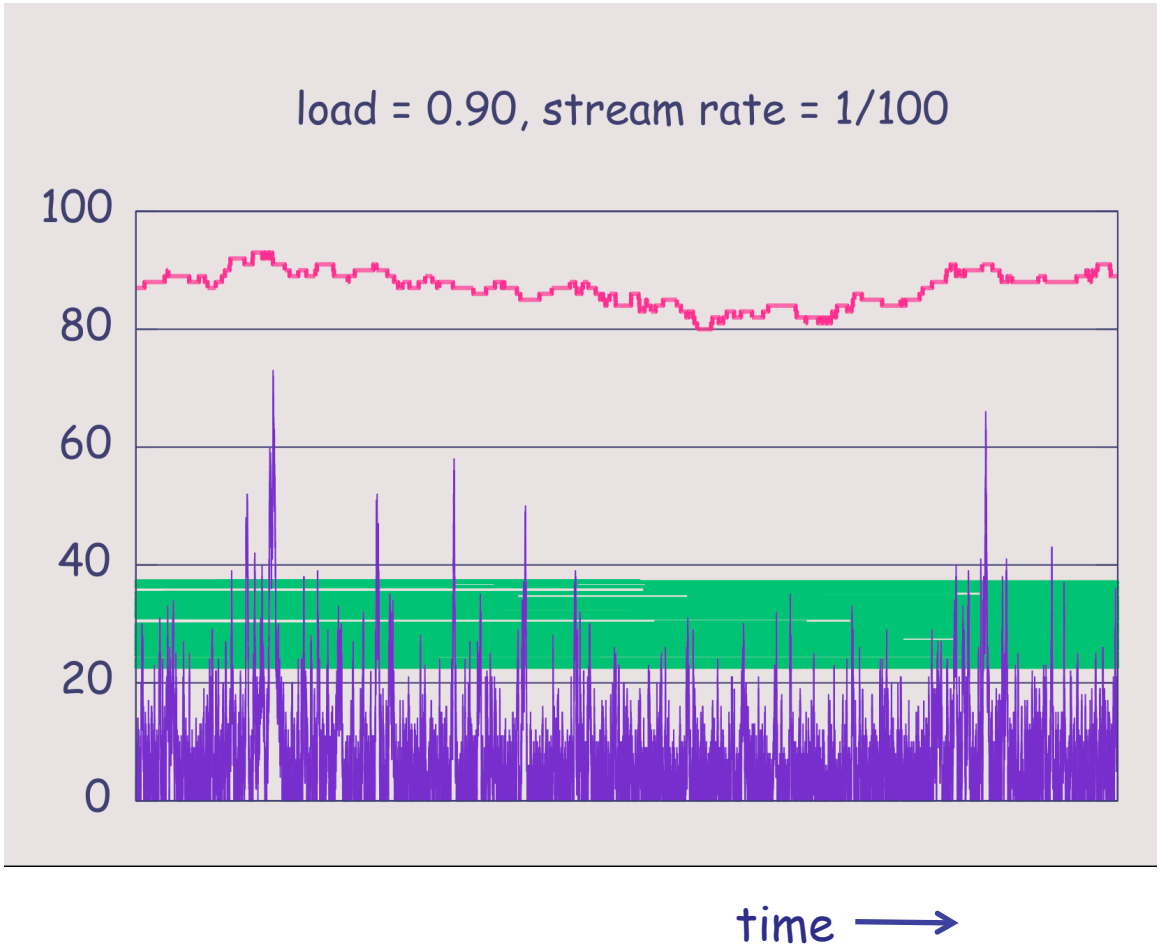
Performance with rate limited flows



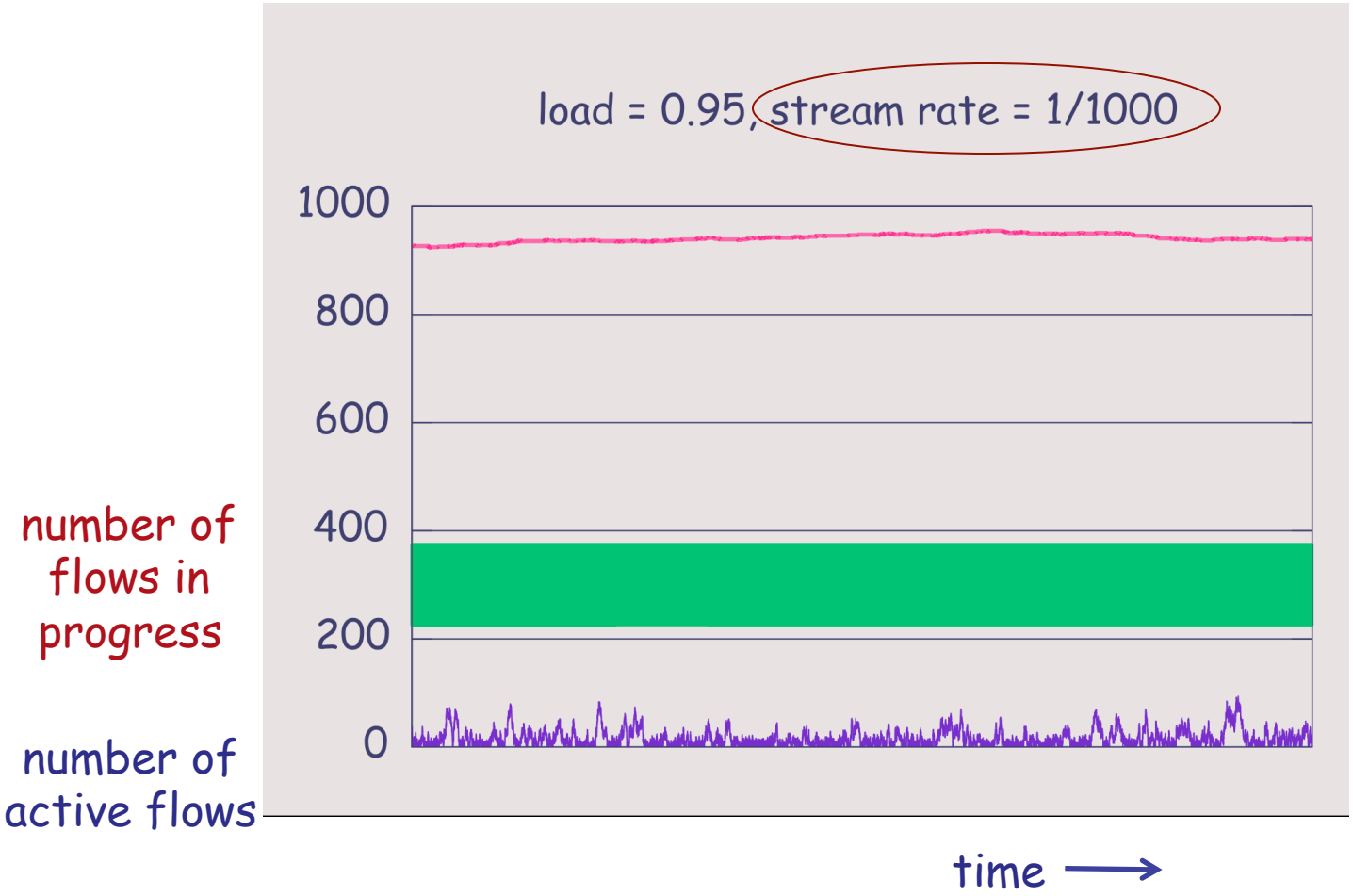
Performance with rate limited flows



Performance with rate limited flows



Performance with rate limited flows



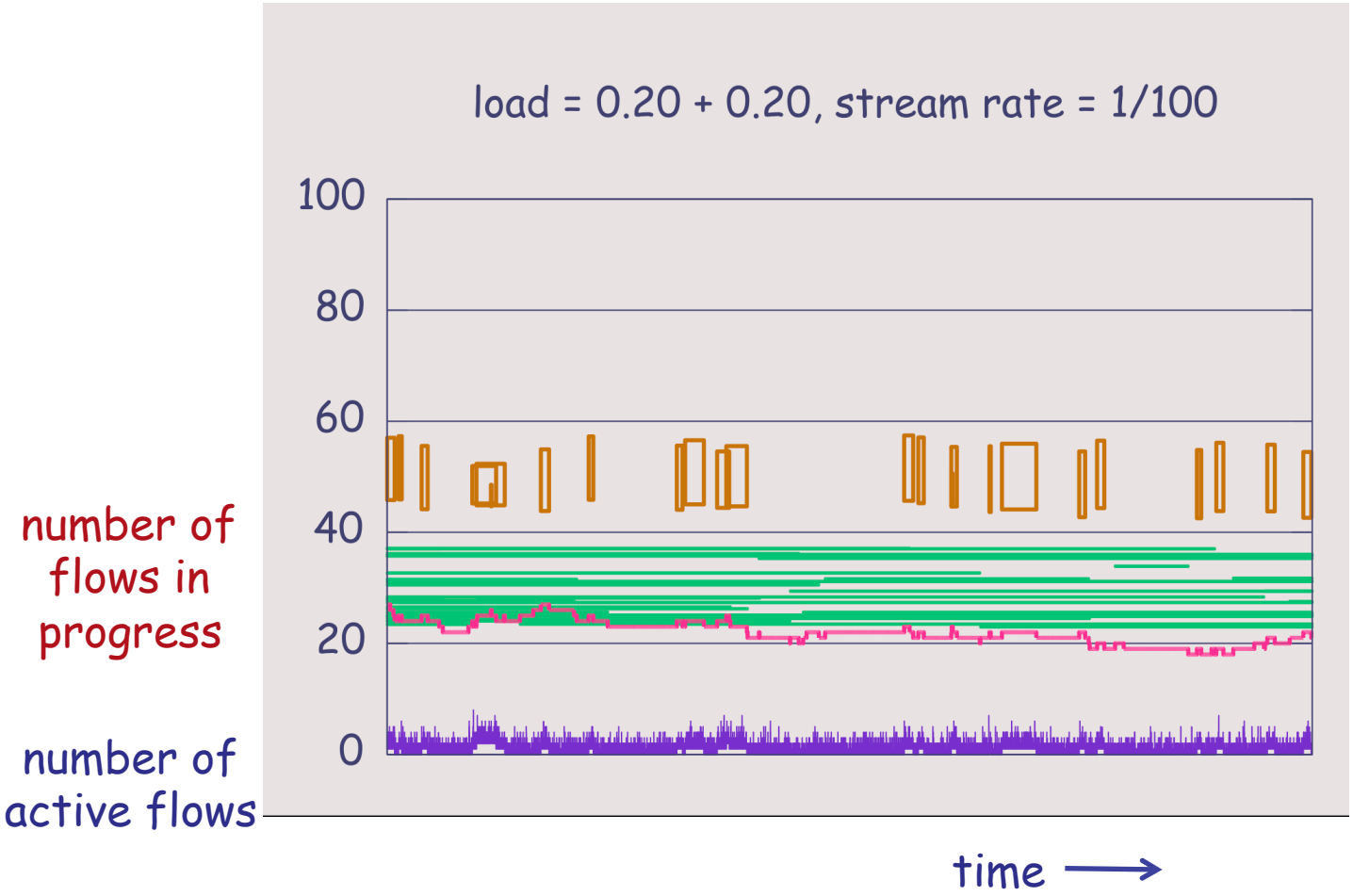
Observations 2

- most flows are not elastic and emit packets at their peak rate
- these flows are "active", and need to be scheduled, only when they have a packet in the queue
- the number of *active* flows is small until load approaches 100%
- fair queuing is feasible and scalable, even when the number of flows *in progress* is very large

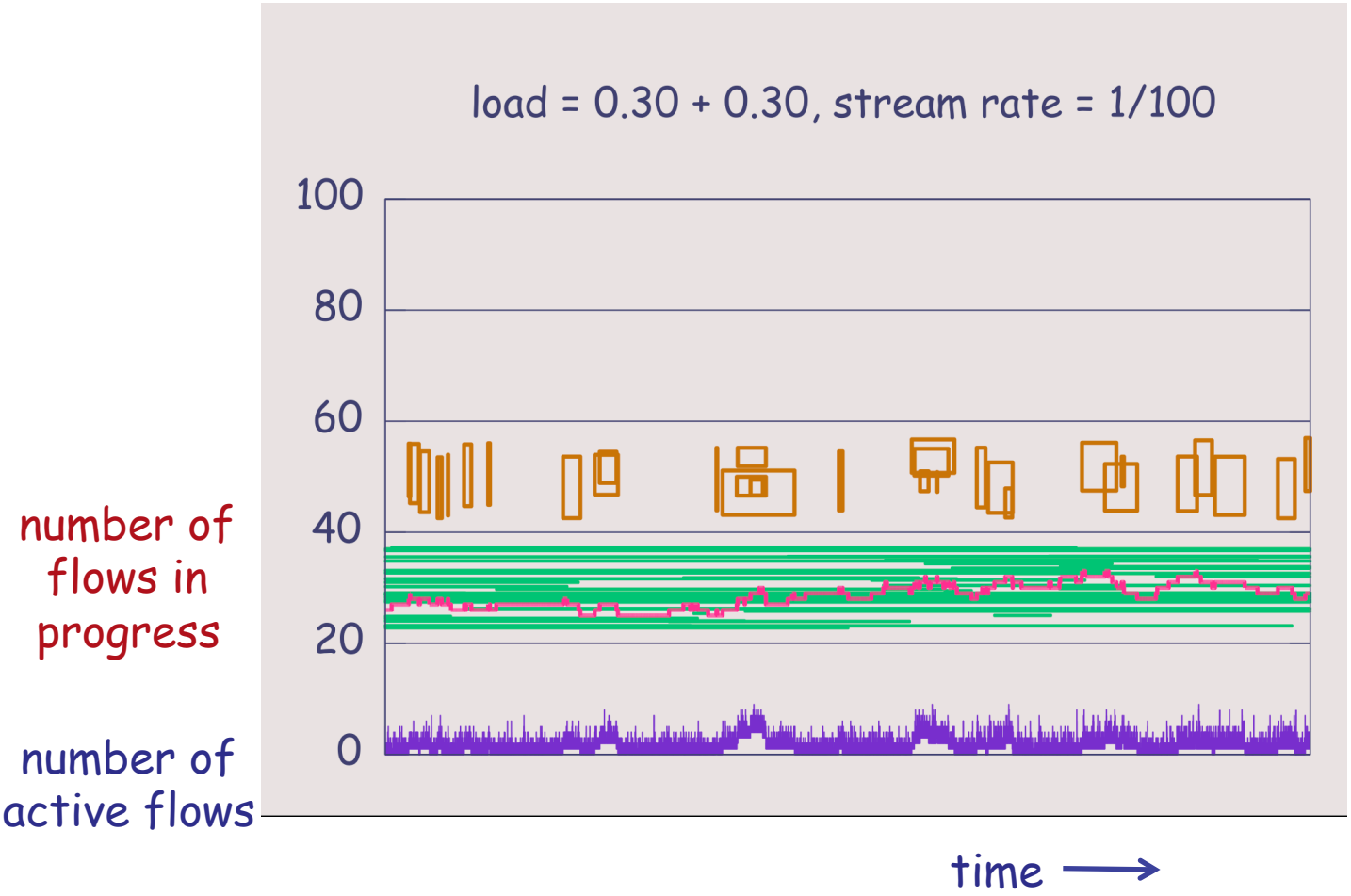
More simulations

- links may be shared by many rate limited flows and a few elastic flows
- consider a link shared by 50% of traffic from flows whose peak rate is 1% of link rate and 50% elastic traffic

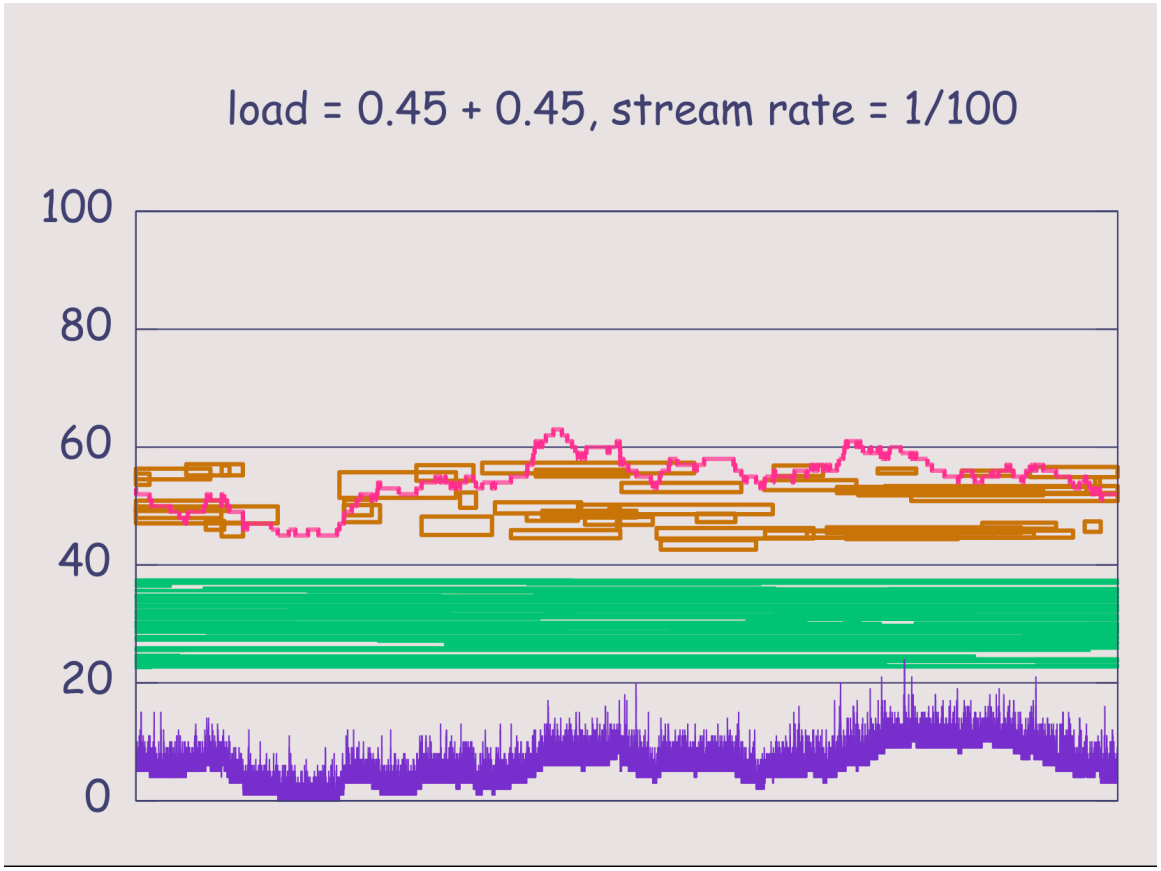
Performance of link with elastic and rate limited flows



Performance of link with elastic and rate limited flows



Performance of link with elastic and rate limited flows



Observations 3

- the number of *active* flows is small (<100) with high probability until load approaches 100%
- therefore, fair queuing is feasible and scalable
- fair queuing means packets of limited peak rate flows see negligible delay:
  - they are delayed by at most 1 round robin cycle
  - this realizes implicit service differentiation since conversational and streaming flows are in the low rate category

Congestion collapse and the "law of the jungle"

- Wikipedia: the Internet was saved in 1987 from congestion collapse by TCP congestion control
- but transport protocols are becoming less and less TCP friendly, so are we again in danger of congestion collapse?
- the answer is
  - "not really", if flow peak rates are limited to ~1-10% of link capacity
  - "not at all", if routers implement fair sharing
- the real cause of congestion collapse is load > 1
  - the number of flows in progress continues to grow while per-flow throughput → 0
- cf. Bonald et al. "*Is the "law of the jungle" sustainable for the Internet?*", Infocom 2009

Recommendation for traffic control

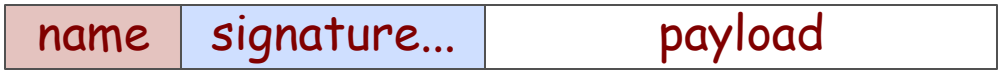
- implement per-flow fair queuing in router queues
  - this is scalable and feasible (though more complex than FIFO)
  - and realizes implicit service differentiation
  - view fairness as an *expedient* not a socio-economic objective
- apply traffic engineering to ensure load is not too close to 100% and overload controls in case this fails
- access networks need more than fair sharing - but see "Bufferbloat" where fair queuing is the preferred solution

Outline

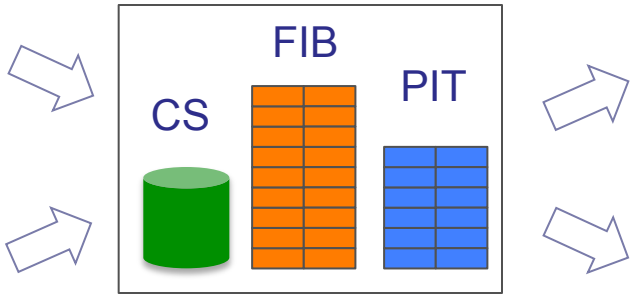
- Internet traffic control
- ~~ICN~~ CCN traffic control
- Impact of in-network caching

Content-centric networking (CCN)

- a new Internet architecture proposed by Jacobson [CoNext 2009]
  - "named data networking": packets have names, not addresses!
- users emit Interests 

- network returns Data 

- using pending Interest table (PIT), forwarding information base (FIB), content store (CS)
  - PIT remembers forwarded Interests and where they came from
  - FIB determines where to forward Interests from name prefix
  - CS is a cache that locally stores popular content

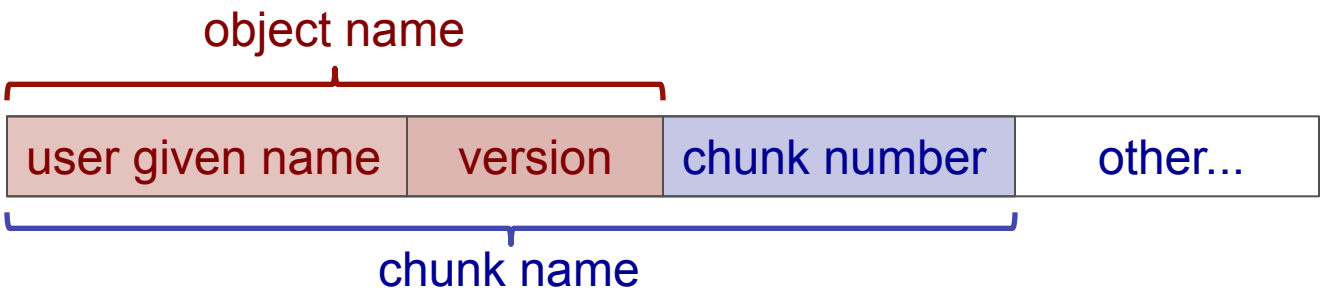


CCN and traffic control

- [CoNext 2009] says
  - "no queues in CCN routers, LRU replaces FIFO"
  - "flow balance removes need for in-path congestion control "
- we disagree ! CCN traffic control is still to be defined
  - a mixture of network mechanisms and end-system protocols
- objectives of traffic control in the core
  - ensure real time flows suffer negligible delay
  - control bandwidth sharing between high rate flows
- we propose a flow-aware traffic control framework
  - borrowing and adapting mechanisms from IP
  - proposing some new, CCN-specific mechanisms

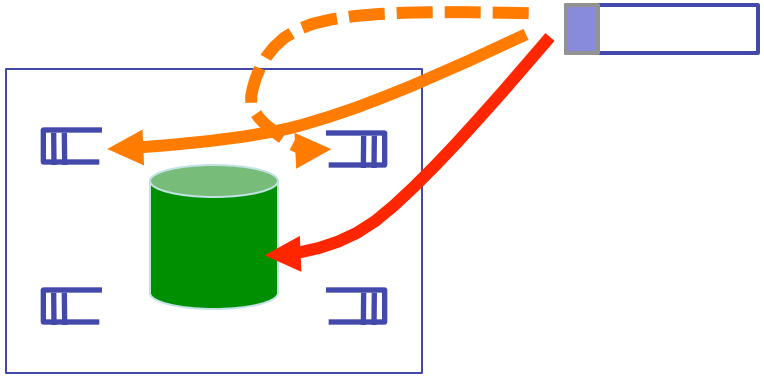
Flow identification in ICN

- flow identified by object name
- possible in CCN where name is in header
  - since hierarchical packet name includes object name
- assume this is possible in ICN in general
  - though PURSUIT, NetInf use flat names for objects and it is not clear that names appear in packet headers
- NB. flow thus includes all downloads of same content object from multiple sources to multiple receivers, as observed at a given point in the network



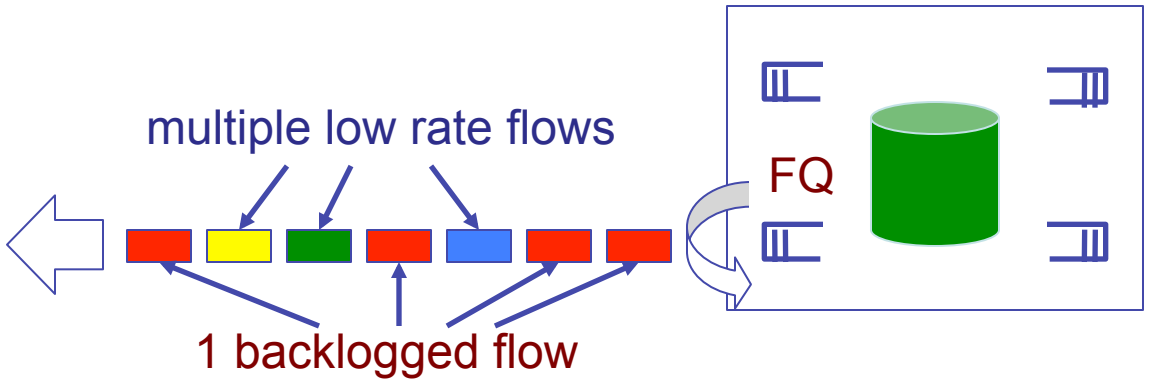
Distinguishing buffers and content store

- to significantly reduce upstream traffic, the content store must have a capacity measured in terabytes
  - eg, 25 TB for a YouTube hit rate of 50%
- an IP buffer is considered large when sized for the bandwidth delay product, ie, less than one gigabyte
  - eg, 10 Gb/s link, 240 ms RTT ⇒300 MB
- buffers and content store have different technologies
  - for different capacity and access speed requirements
- in CCN, on arrival of a Data packet do the following in parallel
  - cache, if appropriate
  - place in buffer on relevant faces
  - discard, applying buffer mgt



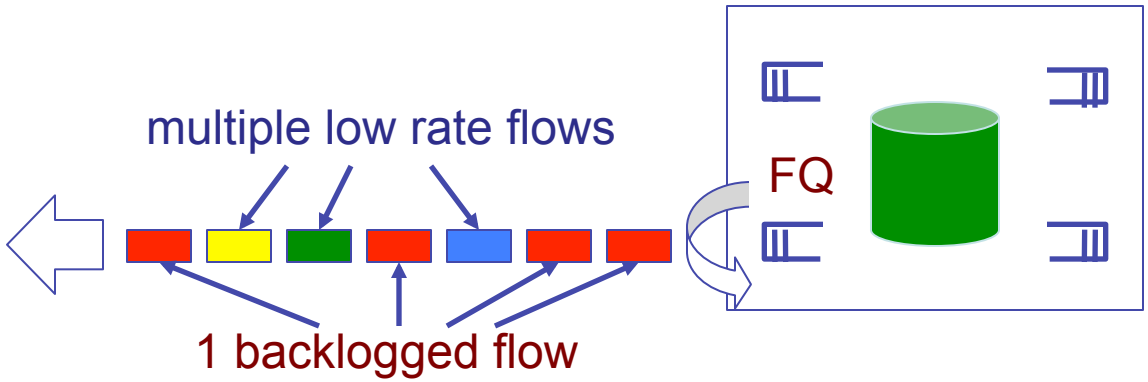
Flow-aware networking

- identify flows by object name…
  - included in chunk name and parse-able
- … on-the-fly, locally, e.g., at a given face
  - all packets with the same object name belong to one flow
- at each face apply per-flow fair queuing with longest queue drop
  - to ensure low latency for real time applications
  - to control bandwidth sharing between elastic downloads
- arguably, the only requirements for a neutral Internet!



Realizing fair queuing

- for example, apply deficit round robin (DRR) and stochastic fairness (SFQ), as implemented in the Linux kernel
- complexity determined by number of "active flows", ie, flows that currently have one or more packets in the buffer
- typically less than 100 flows are active
  - a small number of high rate backlogged flows
  - a small number of packets from many low rate flows
- thus, per-flow fair queuing is feasible and scalable!

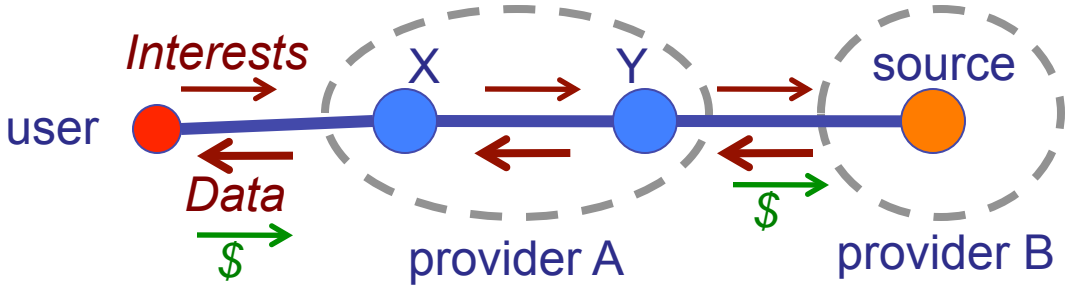


Outline

- Internet traffic control
- ~~ICN~~ CCN traffic control
  - some specific CCN mechanisms
- Impact of in-network caching

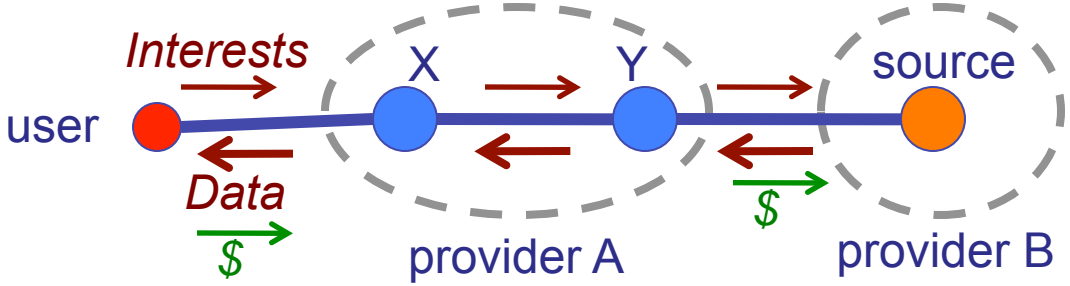
"Interests buy data"

- a proposed direction of charging:
  - user pays provider A, A pays provider B,..., for *delivered* Data
  - not excluding flat rates, peering,...
- brings return on investment and an 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 sold (ie, delivered downstream) due to congestion...



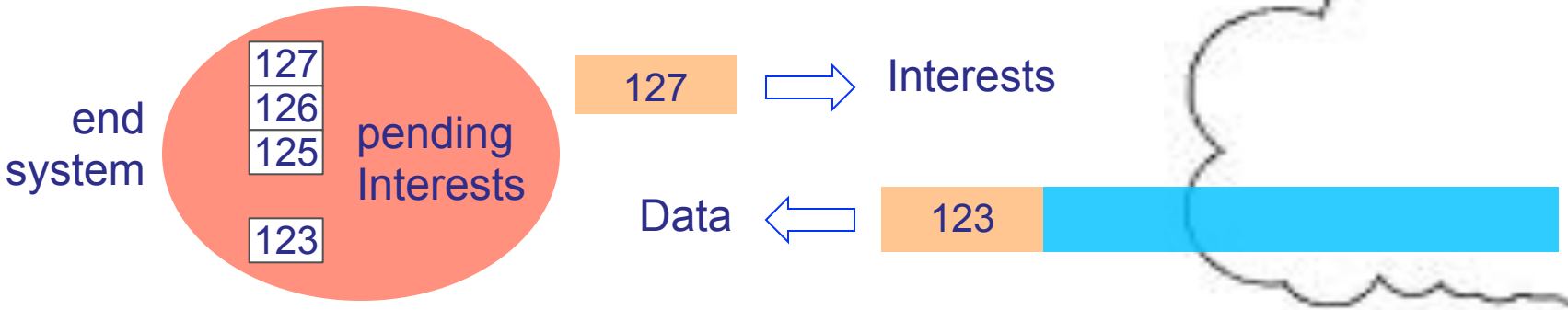
Interest discard

- a particularity of CCN: Interests follow the same path as Data
- a router can therefore participate in congestion control by shaping the flow of Interests
- we propose Interest discard:
  - routers monitor congestion on the downstream link
  - discard upstream Interests in excess of current fair rate
  - all in the same line card
- avoids buying data that cannot be sold (because of insufficient link capacity), and stops abusive emission of Interests



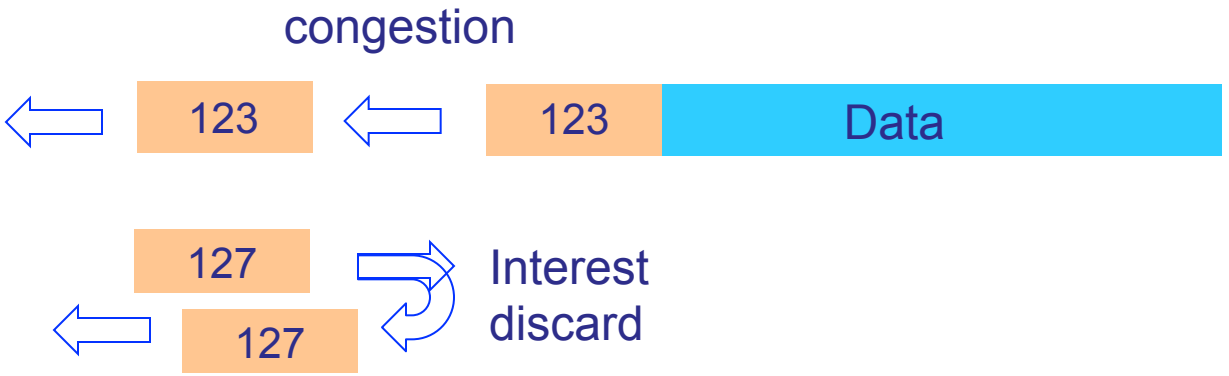
End-system behaviour

- receivers control download rate by Interest emissions
- since fairness is imposed, network performance is broadly independent of user emission strategies
- greedy strategies are OK
  - users can emit Interests as fast as they like without impacting other flows or reducing network capacity
- additive increase, multiplicative decrease (ie, like TCP) is probably better
  - for simpler management of pending Interests in end systems
  - but aggressive parameters are OK (low loss, high throughput)



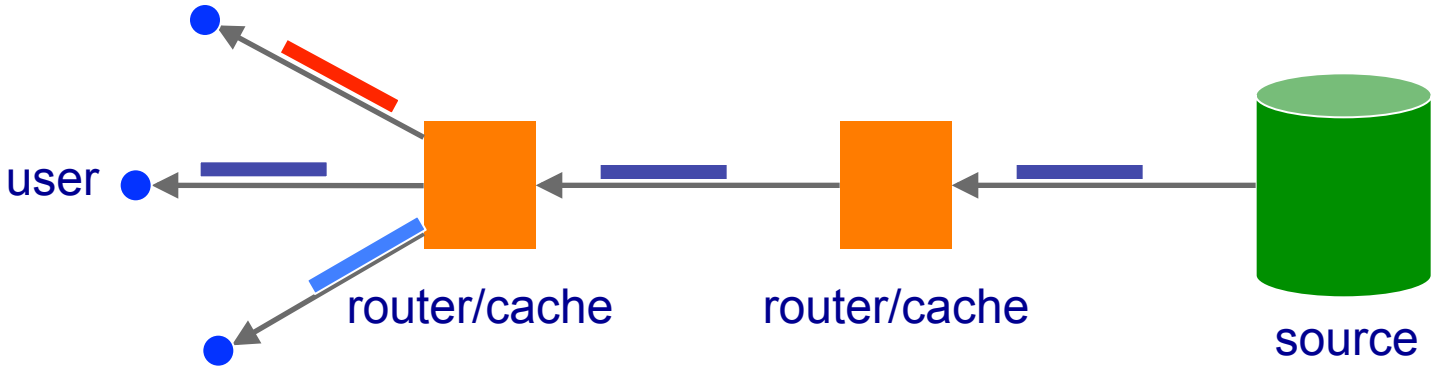
Explicit congestion notification

- for effective AIMD, we require fast congestion notification
  - but sequence is not preserved in CCN
  - triple duplicate ACK won't work!
- we propose an explicit congestion notification:
  - in case of DRR drop or Interest discard, return Data header
- for fast PIT update in routers and user equipment
  - instead of waiting for a time out
- simulations show smaller PIT and better throughput fairness



Multicasting

- fair queuing does not adversely impact multicast in CCN
- eg, same live streaming flow requested by several users
  - multicast by Interest aggregation (and caching)
  - no problem with FQ! streaming flows have low latency, low loss
- eg, same video download by several users asynchronously
  - multiple flows would be treated by FQ as one (possible degradation)
  - but multiple parallel flows for same object are eliminated by caching



Proposed traffic control framework

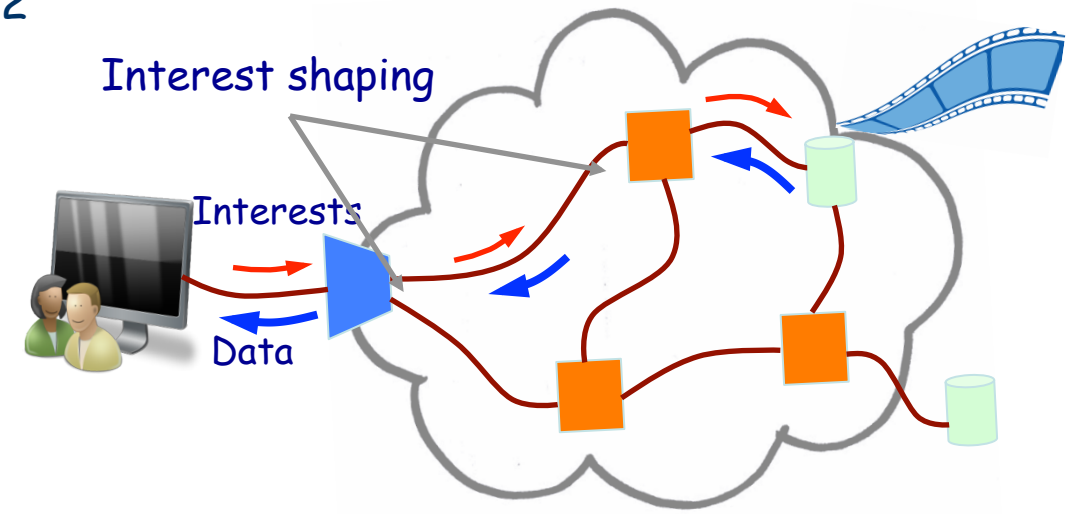
- flow-aware networking is a complete traffic control for CCN
- "Interests buy Data" implies a rational direction of charging
- some requirements
  - explicit object name in packet headers
  - (approximate) fair queuing in face buffers
  - traffic engineering and overload controls to ensure load < 90%
- some enhancements
  - Interest discard to avoid buying excess Data
  - explicit congestion notification by Data payload discard
- see "Flow-Aware traffic control for a content-centric network," S. Oueslati, J. Roberts, and N. Sbihi, in Proc of IEEE Infocom 2012
- there are other proposals...

Outline

- Internet traffic control
- ~~ICN~~ CCN traffic control
  - some specific CCN mechanisms
  - other proposals
- Impact of in-network caching

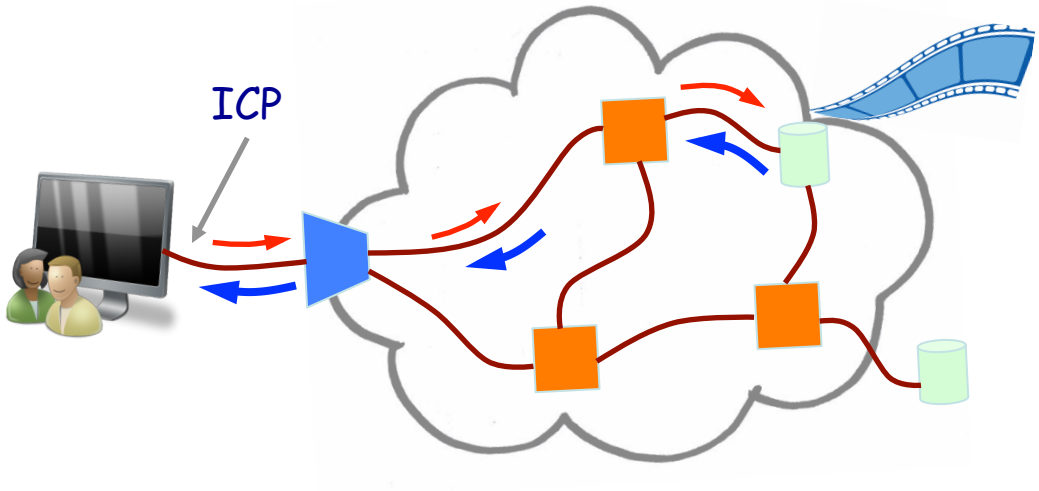
Flow-aware CCN by Interest shaping

- control Data flow sharing by shaping Interest flows
- shape Interests in same line card as controlled Data
  - cf. "An effective hop-by-hop Interest shaping mechanisms for CCN" by N. Rozhnova and S. Fdida, NOMEN 2012
- shape Interests in previous line card, using PIT adaptation
  - cf. "Joint Hop-by-Hop and Receiver-Driven Interest Control Protocol for Content-Centric Networks", C. Carofiglio et al., ICN 2012



Receiver-driven transport protocol

- Interest control protocol
  - additive increase, multiplicative decrease Interest window
  - loss detection by adaptive time-out
  - analysis for stability proof and parameter setting
  - experimental implementation and validation
  - cf. "ICP: Design and evaluation of an Interest control protocol for content-centric networking", G. Carofiglio et al., NOMEN 2012

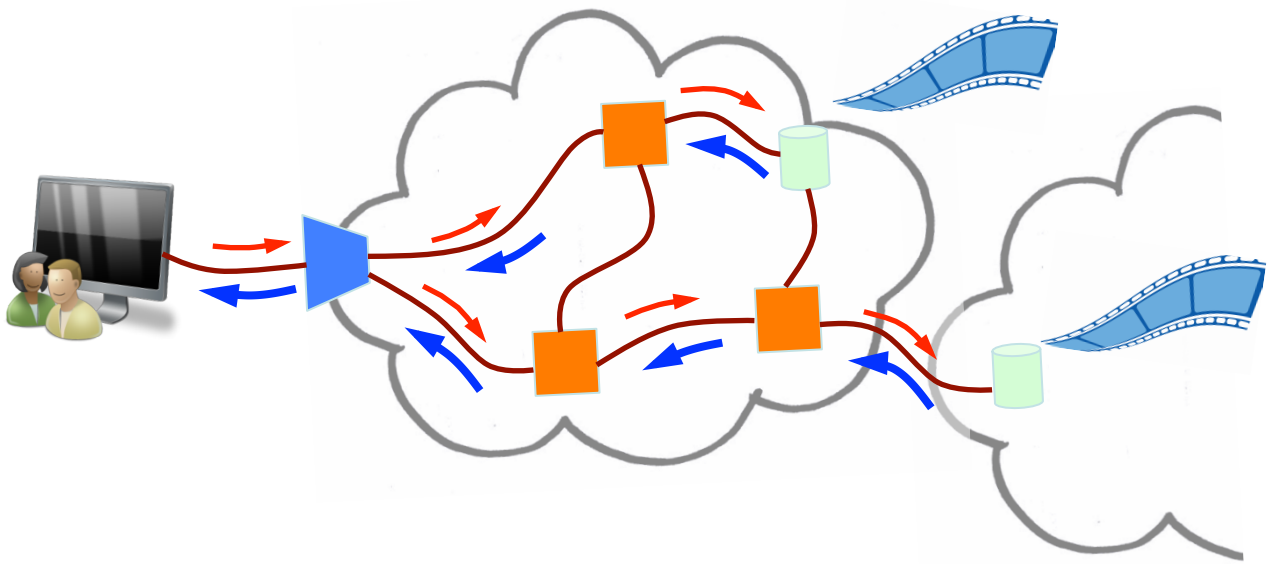


Receiver-driven transport protocol

- Interest control protocol
  - additive increase, multiplicative decrease window
  - loss detection by adaptive time-out
  - analysis for stability proof and parameter setting
  - experimental implementation and validation
  - cf. "ICP: Design and evaluation of an Interest control protocol for content-centric networking", G. Carofiglio et al., NOMEN 2012
- used in conjunction with Interest shaping
  - cf. "Joint Hop-by-Hop and Receiver-Driven Interest Control Protocol for Content-Centric Networks", C. Carofiglio et al., ICN 2012

Multipath forwarding

- adaptive and multipath routing via Interest shaper/PIT coupling
- to use multiple sources of content including on-path caches
  - remote adaptive AQM for RTT dependent probabilistic window decrease
  - unique window, per path RTT estimation
  - cf "Multipath congestion control in content-centric networks", G. Carofiglio et al., NOMEN 2013



"Traffic control" in other ICN architectures

- SAIL/NetInf: use TCP or borrow ICP from CCN…
  - cf. Deliverable 3.2 NetInf Content Delivery and Operations, 2012
- PURSUIT: ConTug is a receiver driven transport protocol
  - cf. Tech Report TR 12-002
- CONET: ICTP is receiver driven and TCP-like
  - cf. "Transport layer issues in information-centric networks", S. Salsano et al., ICN 2012

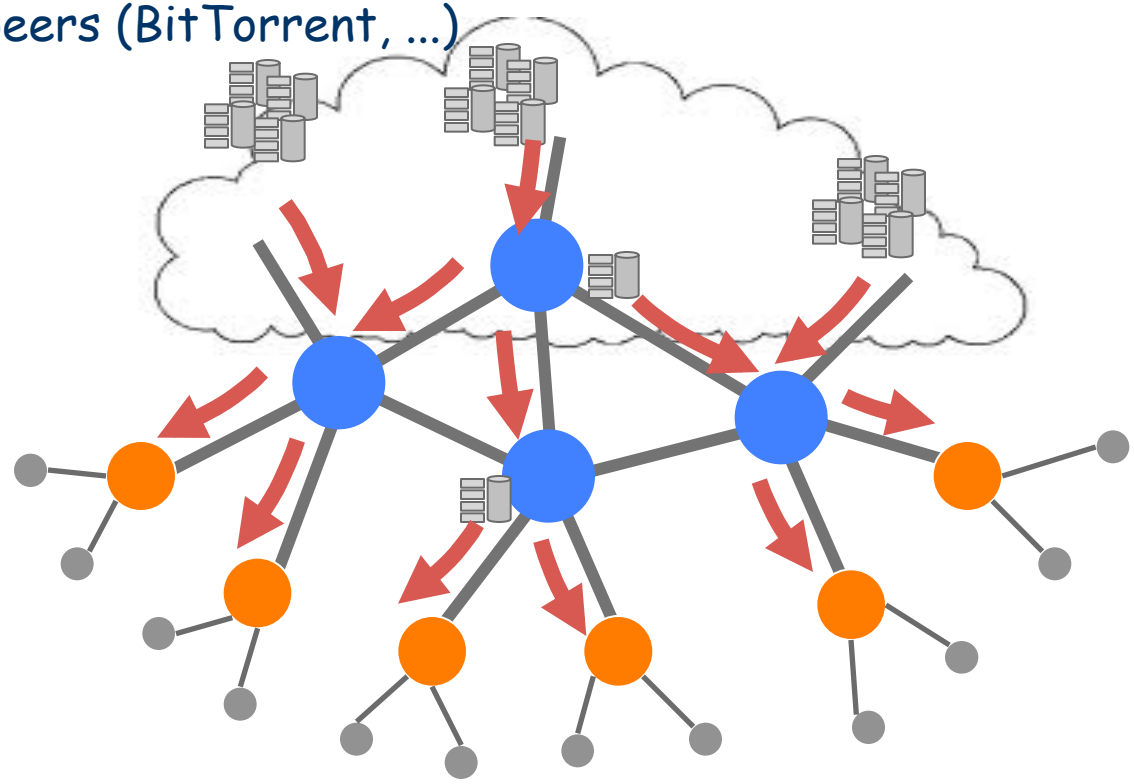
- all seem to rely on end-system congestion control; some use segment names that don't allow flow-awareness
- congestion control is complicated by multi-source delivery (due mainly to in-network caching)

Outline

- Internet traffic control
- ICN traffic control
- Impact of in-network caching

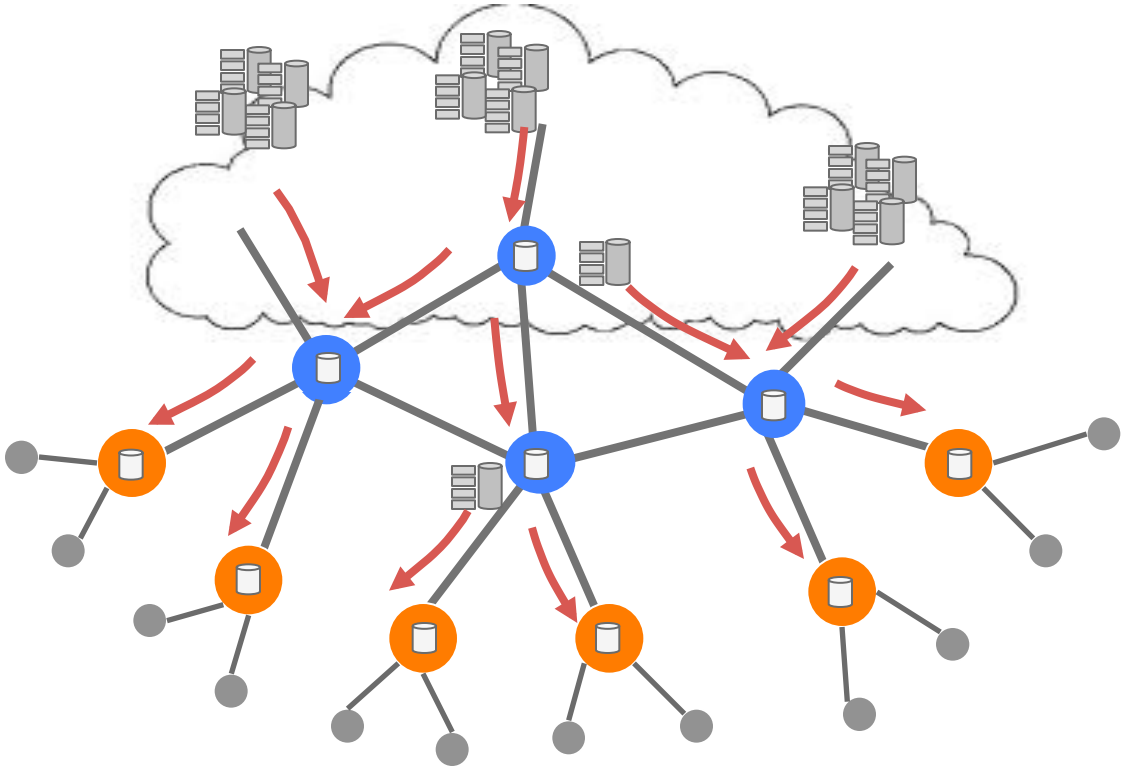
Content traffic in the Internet

- more than 90% of traffic in an ISP network is content delivery
  - from large scale data centers (Google, Facebook,...,)
  - from "content distribution networks" (Akamai, Limelight,...)
  - from peers (BitTorrent, ...)



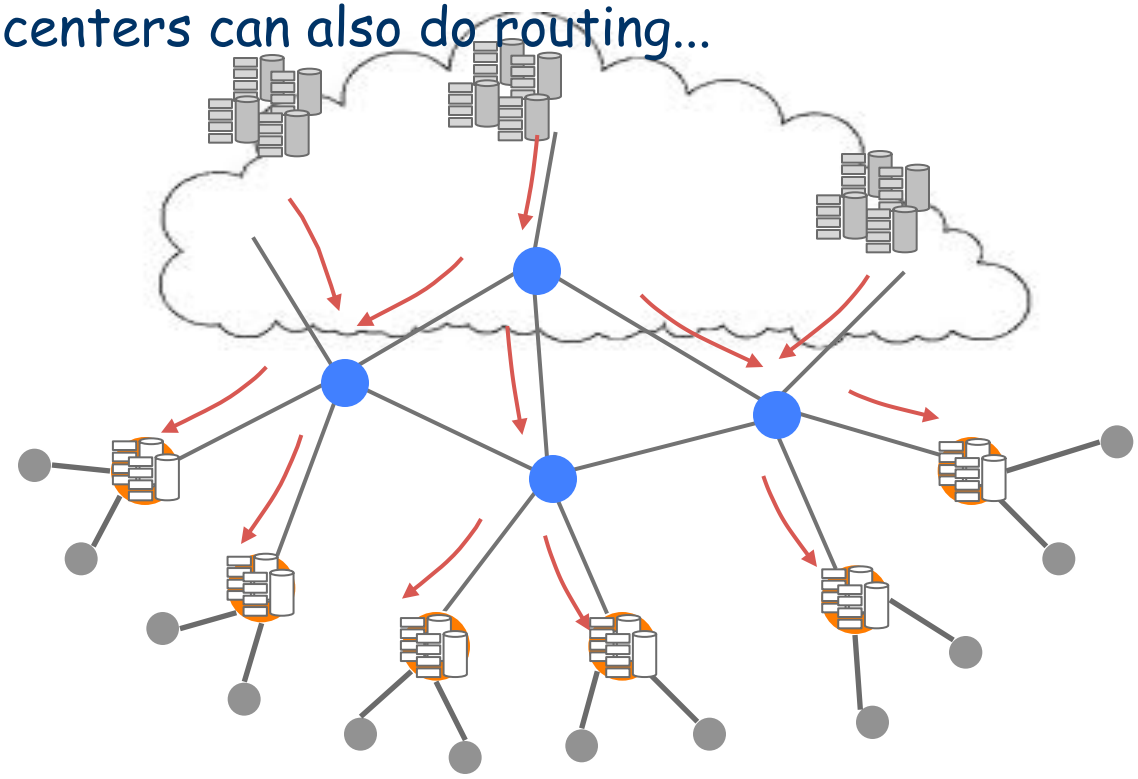
Traffic reduction in an information-centric network

- eg, equip routers with a "content store"
- cache popular content to realize a more favourable memory-bandwidth tradeoff



Or, is this vision flawed?

- router content stores are limited in size by technology while effective traffic reduction requires VERY large caches
- and content delivery is a business needing compute power
- and data centers can also do routing…



Outline

- Internet traffic control
- ICN traffic control
- Impact of in-network caching
  - performance of an LRU cache

LRU replacement

- when a cache is full, some objects must be removed to make room for new ones, eg,
  - least recently used (LRU): replace the object that has not been requested for the longest time
  - random: replace any object chosen at random
  - least frequently used (LFU): only cache the most popular objects
  - ...
- LFU is optimal among policies that cannot see into the future
- LRU appears as a reasonable compromise between complexity and performance
  - objects are indexed by a linked list that evolves at each reqest arrival

Content mix

- Cisco: "96% of traffic is content transfer"
- web, file sharing, user generated content, video on demand
- billions of objects, petabytes of content!

	share	objects	size	volume
web	.18	10^{11}	10 KB	1 PB
file sharing	.36	10^{5}	10 GB	1 PB
UGC	.23	10^{8}	10 MB	1 PB
VoD	.23	10^{4}	100 MB	1 TB

The independent reference model (IRM)

- assume a fixed population of N objects
- requests arrive sequentially and any request is for object n with probability $\propto q(n)$ for $1 \leq n \leq N$

- equivalently, for computing the hit rate,
  assume Poisson arrivals of requests for object n at rate $q(n)$ for $1 \leq n \leq N$

Validity of the independent reference model

- the IRM ignores changes in popularity over time
  - eg, new videos appear, are popular for a time and then disappear
- but, assuming cache content churn occurs at a faster time scale, IRM is reasonable for short periods
  - this is like the stationarity assumption in teletraffic
- accounting for "temporal locality" is more important when estimating the popularity law q(n)
  - measurements over a week may not be relevant for estimating hit rates in the busy period
- q(n) estimation also needs to account for "spatial locality"
  - what's popular in France is not necessarily popular in Norway,...

Content popularity

- Zipf law popularity
  - request rate for n^{th} most popular object, $q(n) \propto 1/n^{\alpha}$
- web page popularity Zipf(α), $.64 < \alpha < .83$
- file sharing Zipf(a), $.75 < \alpha < .82$
- user generated content Zipf(α), $.56 < \alpha < .88$
- video on demand Zipf(α), $.65 < \alpha < 1.2$, Weibull, ...

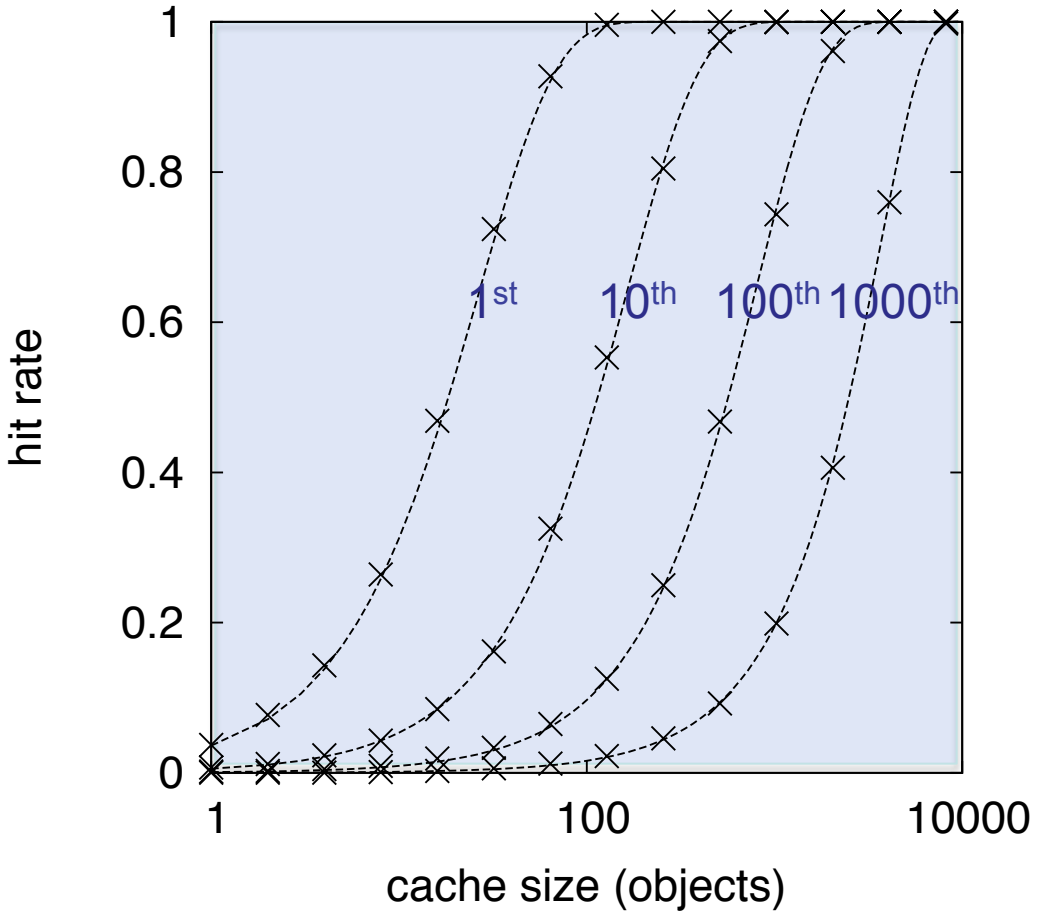
- for our evaluations, we suppose
  - Zipf(.8) for web, file sharing, UGC
  - Zipf(.8) or Zipf(1.2) for VoD

Calculating LRU hit rates (Che, Tung and Wang, 2002)

- cache size C objects, popularity of object $n \propto q(n)$
- assume "independent reference model" or, equivalently, Poisson request arrivals at rate $q(n)$ for object n
- "characteristic time" T_C is time for C different objects to be requested
- assume random variable T_C is approximately deterministic, $T_C \sim t_C$
- then, hit rate for object n is $h(n) = 1 - \exp\{-q(n)t_C\}$
- now, $C = \sum_n \mathbf{1}\{\text{object n is in cache}\}$
- taking expectations, $C = \sum_n h(n) = \sum_n (1 - \exp\{-q(n)t_C\})$
- solving numerically for t_C yields $h(n)$

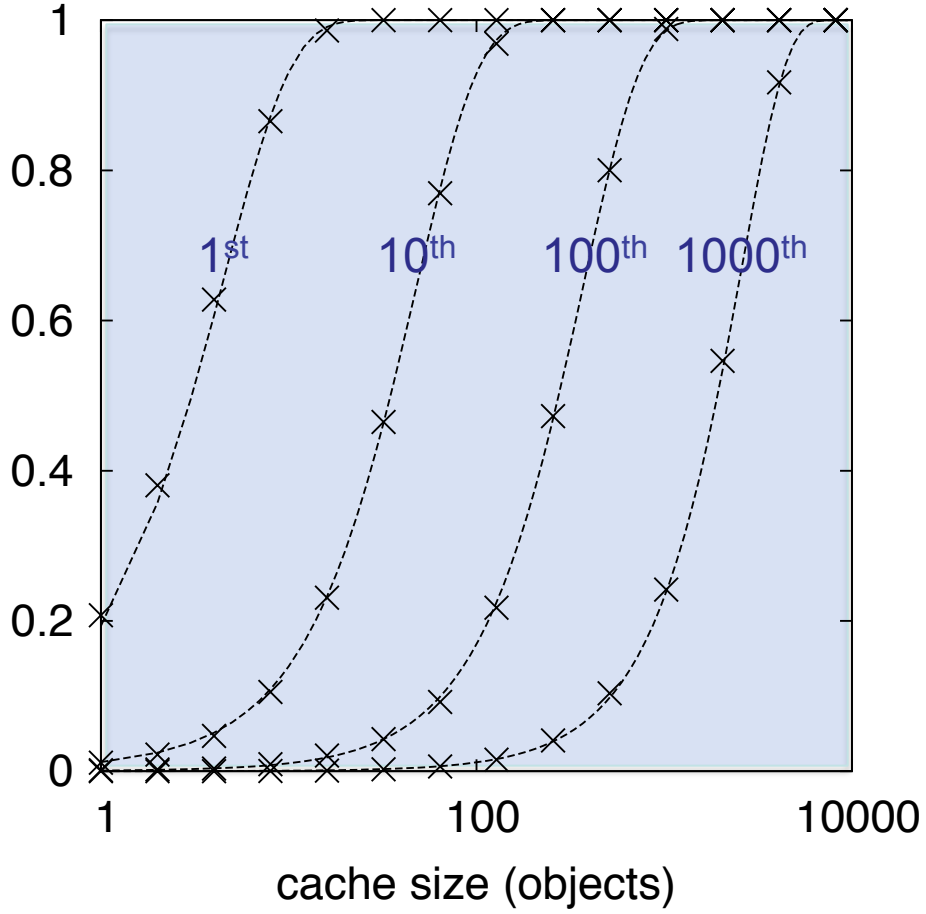
Accuracy of Che approximation

- 10000 objects, Zipf(.8) popularity



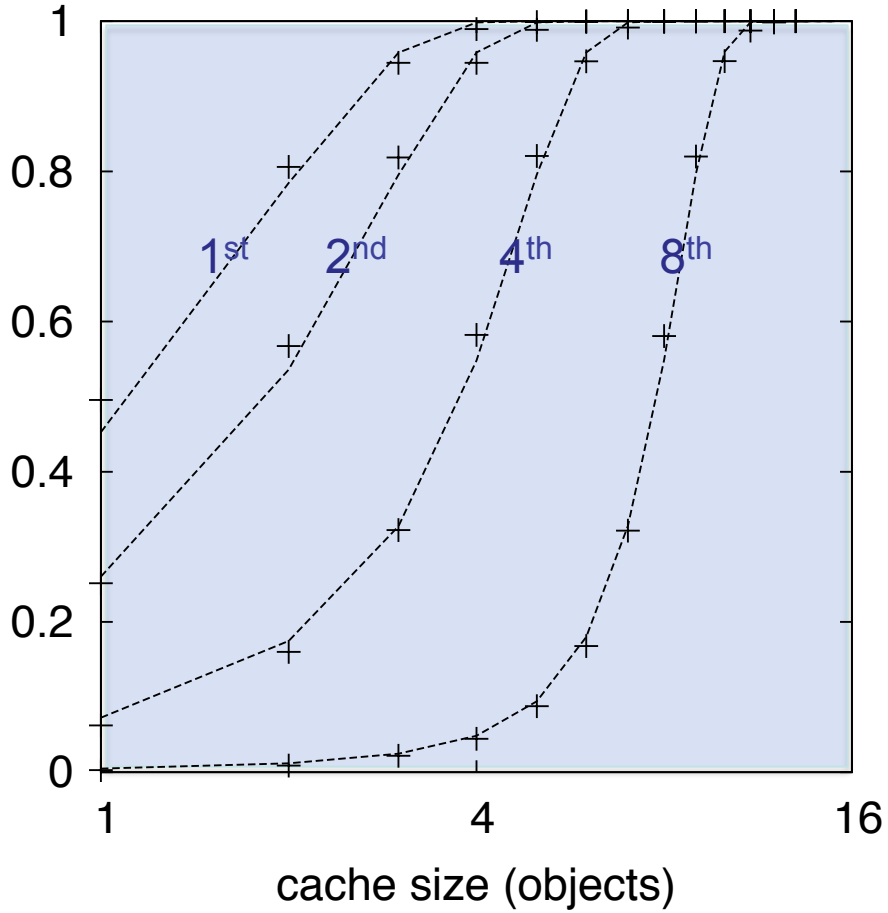
Accuracy of Che approximation

- 10000 objects, Zipf(1.2) popularity



Accuracy of Che approximation

- 16 objects, geometric(.5) popularity

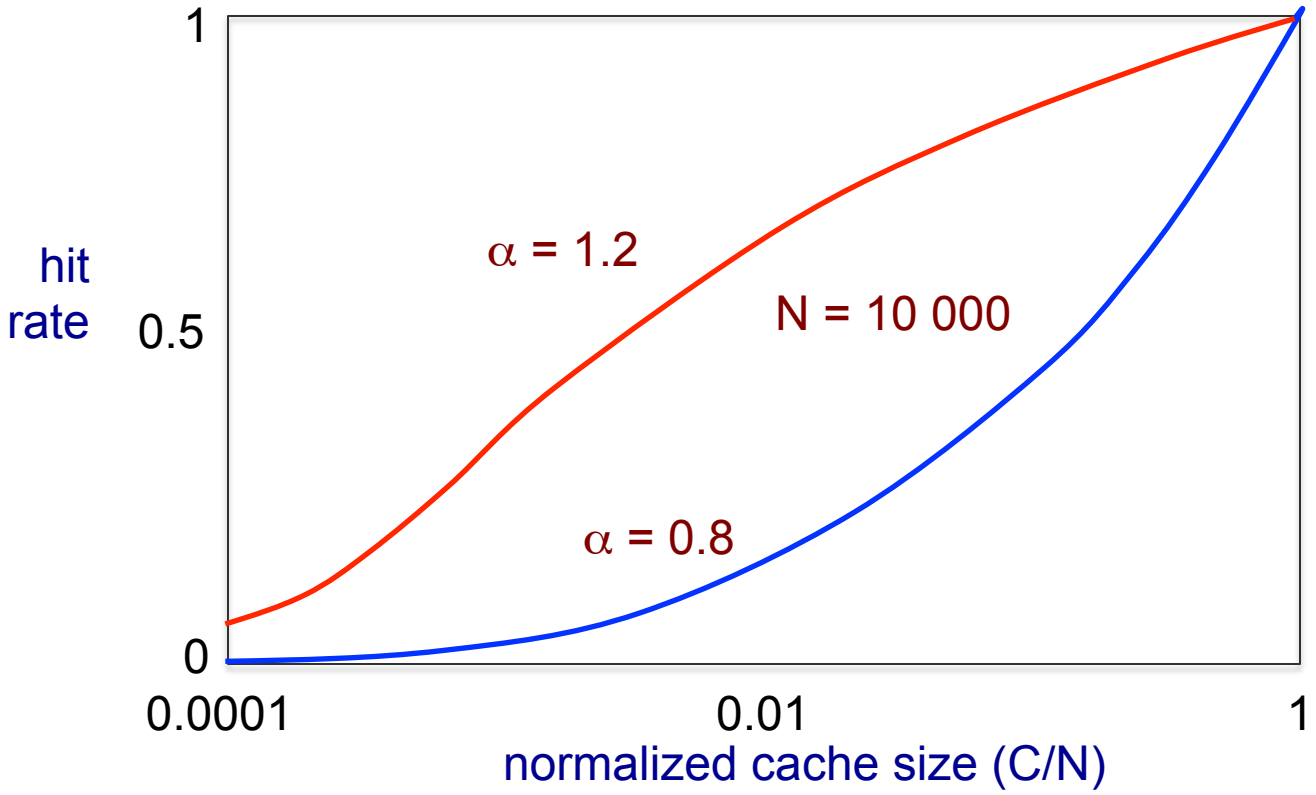


Accuracy of Che approximation

- we can explain accuracy of Che approximation
  - central limit theorem for number of distinct object requests in (0,t)
  - approximate complementary error function (erfc) by a step function
- cf. C. Fricker et al., A versatile and accurate approximation for LRU cache performance. ITC 24, 2012.

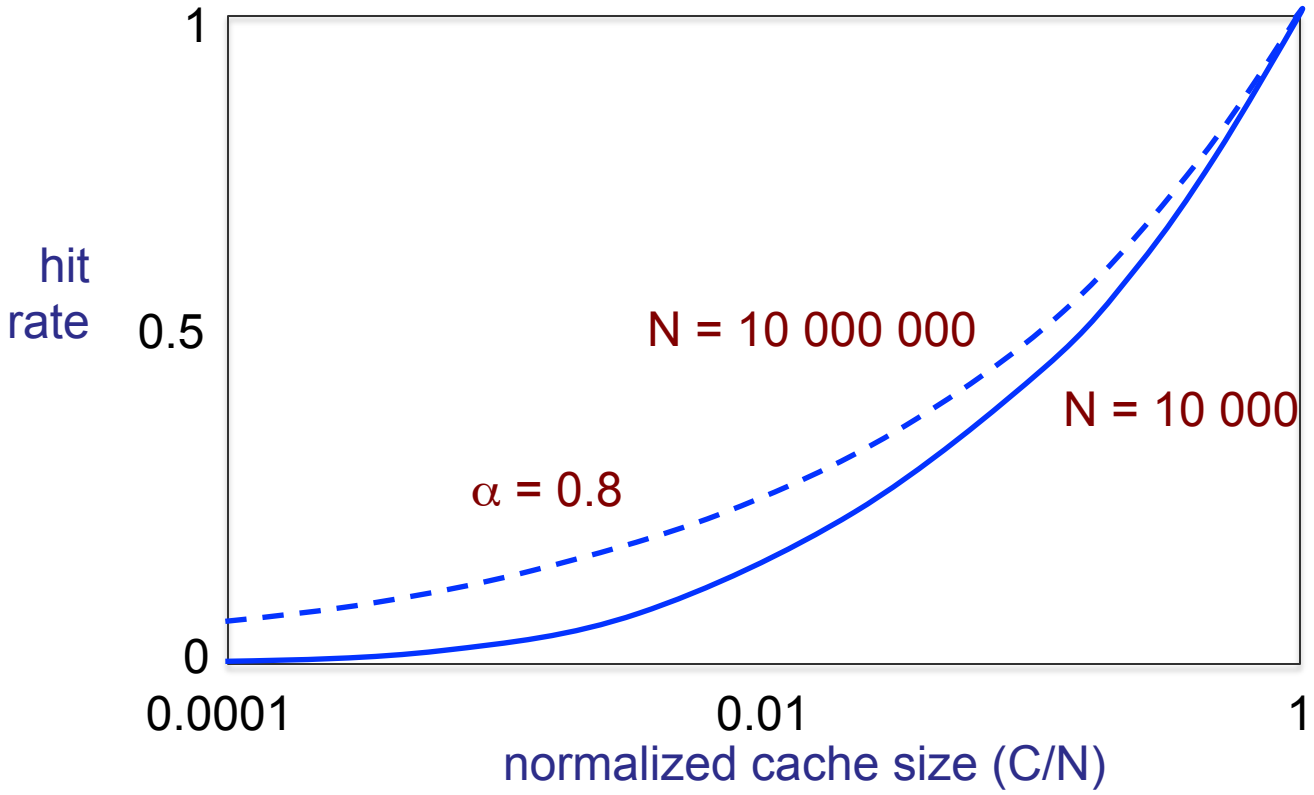
LRU cache performance

- strong impact of Zipf parameter α
- strong impact of object population N



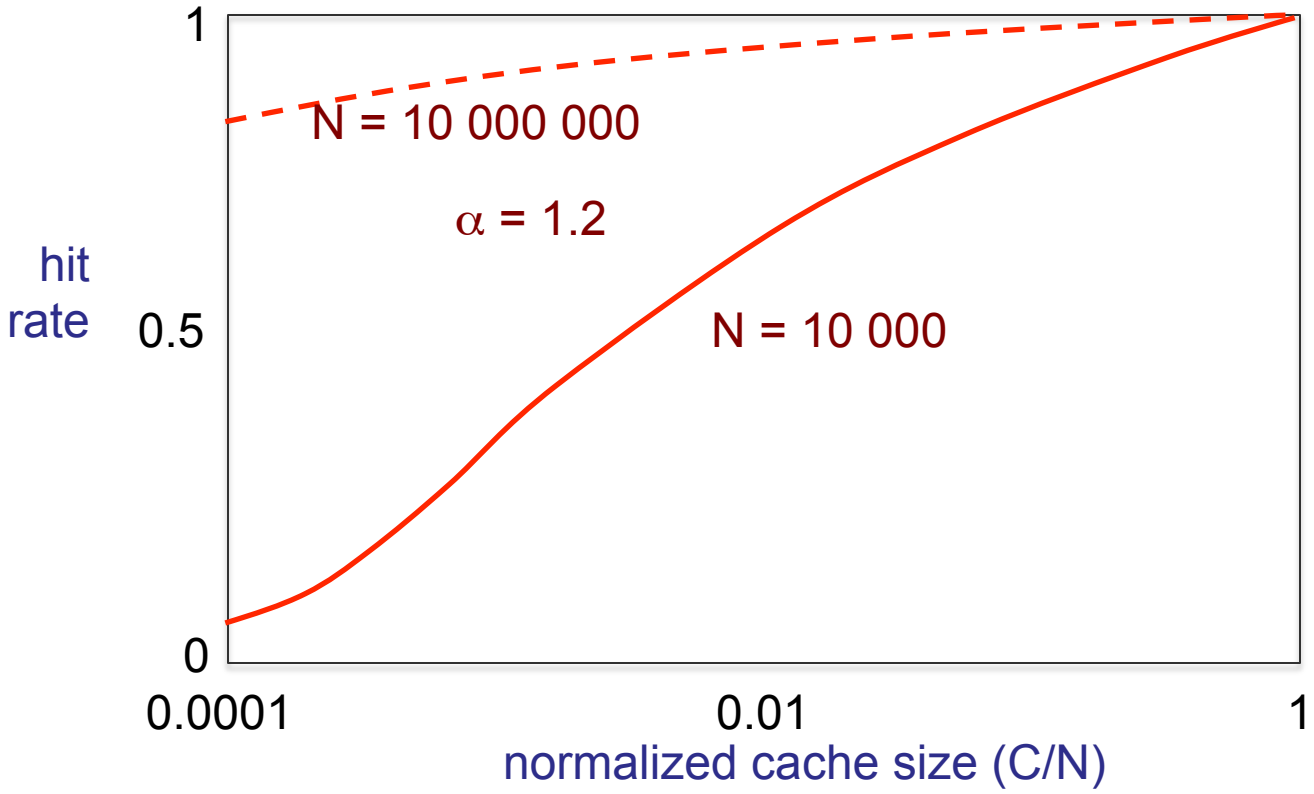
LRU cache performance

- strong impact of Zipf parameter α
- strong impact of object population N



LRU cache performance

- strong impact of Zipf parameter α
- strong impact of object population N



Content mix (recap)

- Cisco: "96% of traffic is content transfer"
- web, file sharing, user generated content, video on demand
- billions of objects, petabytes of content!

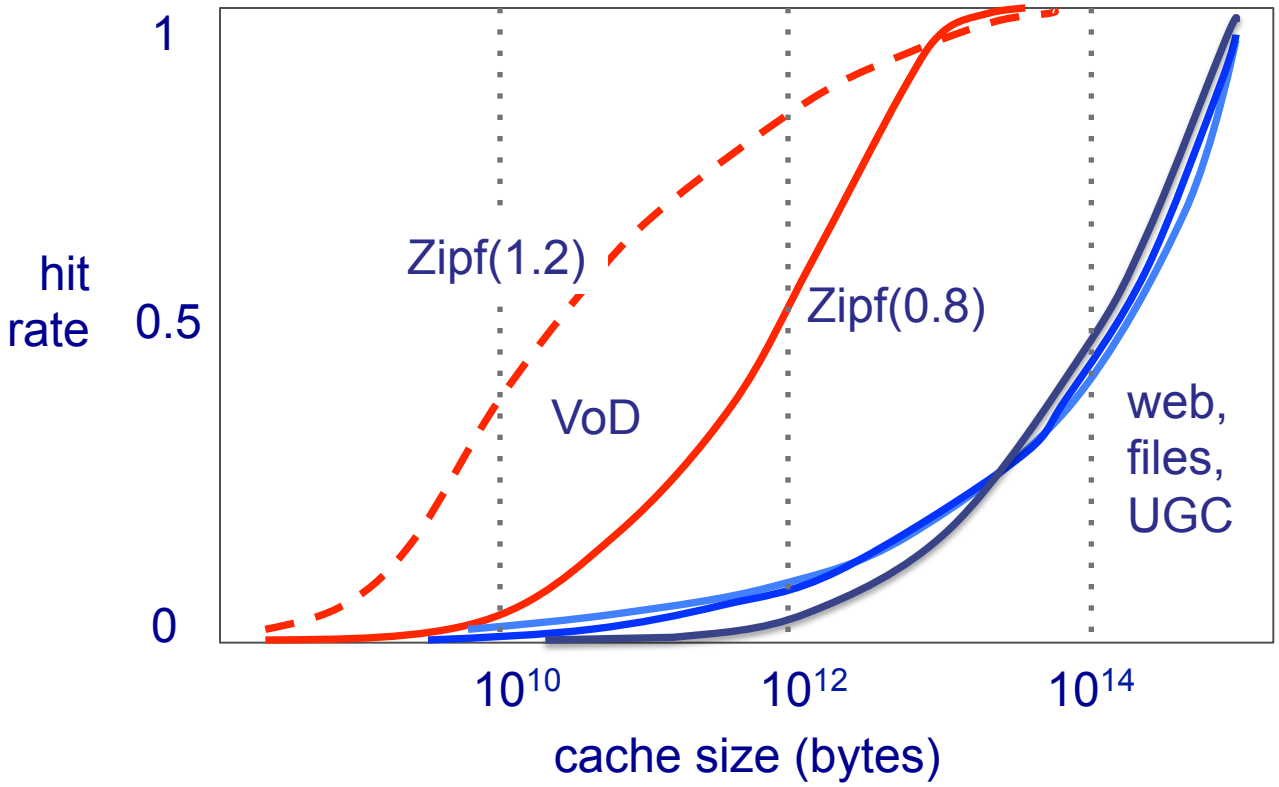
	share	objects	size	popularity
web	.18	10^{11}	10 KB	$1/n^{.8}$
file sharing	.36	10^{5}	10 GB	$1/n^{.8}$
UGC	.23	10^{8}	10 MB	$1/n^{.8}$
VoD	.23	10^{4}	100 MB	$1/n^{.8}$ or $1/n^{1.2}$

Generalized Che approximation

- cache size C bytes, popularity of object n of type i is $q_i(n)$, size of this object is $\theta_i(n)$
- assume "independent reference model »
- "characteristic time" T_C is time for different objects of total size C to be requested, assume $T_C \sim t_C$
- then, hit rate for type i object n is $h_i(n) \approx 1 - \exp\{-q_i(n)t_C\}$
- now, $C = \sum_i \sum_n 1\{\text{type i object \#n present}\}\ \theta_i(n)$
- taking expectations, $C = \sum_i \sum_n h_i(n)\ \theta_i(n)$
  $= \sum_i \sum_n (1 - \exp\{-q_i(n)t_C\})\ \theta_i(n)$
- solving for t_C yields the $h_i(n)$

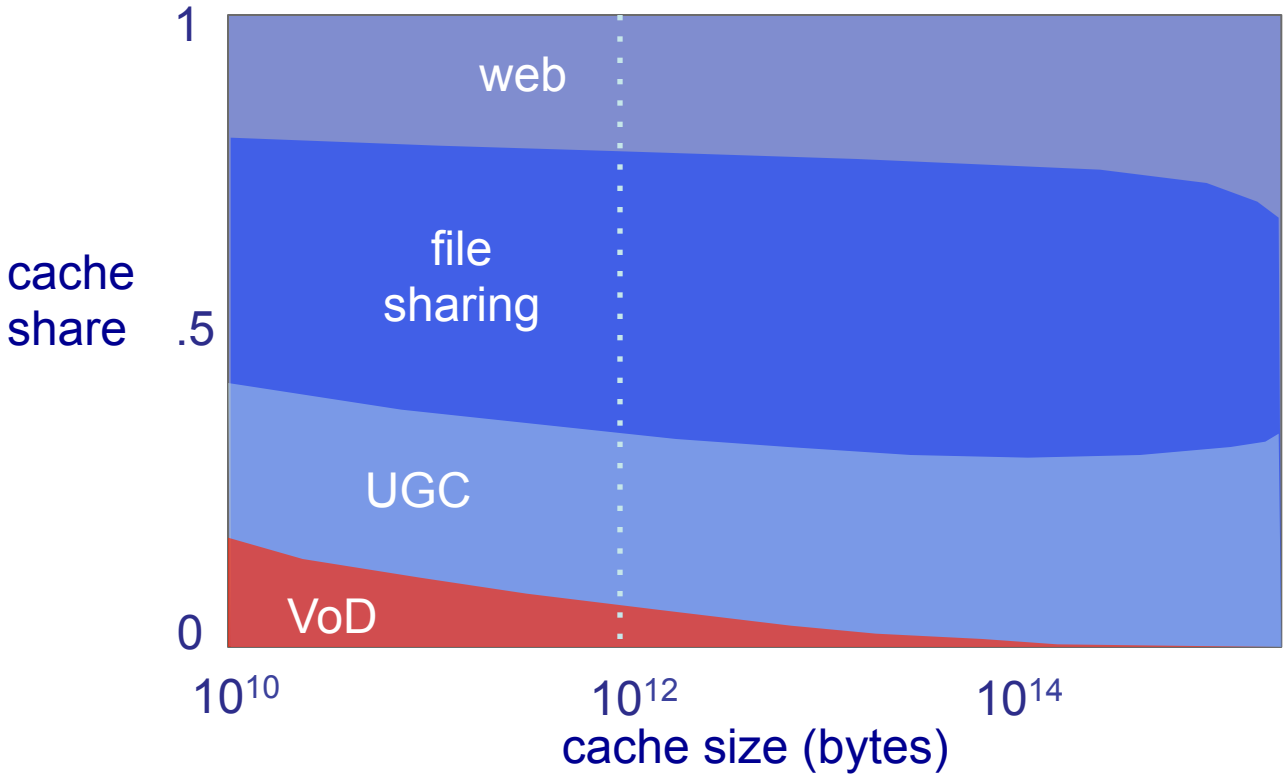
Per content type hit rates

- for web, file sharing and UGC, we need $O(10^{14})$ cache size
- for VoD, $O(10^{12})$ cache size yields high hit rates



Cache occupancy by type of content – VoD popularity is Zipf(1.2)

- disproportionate cache use, e.g., for 1 TB cache:
  - 95% share for web, files and UGC for only 5% hit rate
  - VoD hit rate is 85% but only uses 5% of cache capacity



Is ubiquitous caching really a good idea?

- router cache cannot be very big, $O(10^{12})$ bytes, say
  - for cheap but rapid storage in a large number of ICN routers
- significant VoD hit rate but heavy polution from other types
- it would be more effective to specialize cache for VoD

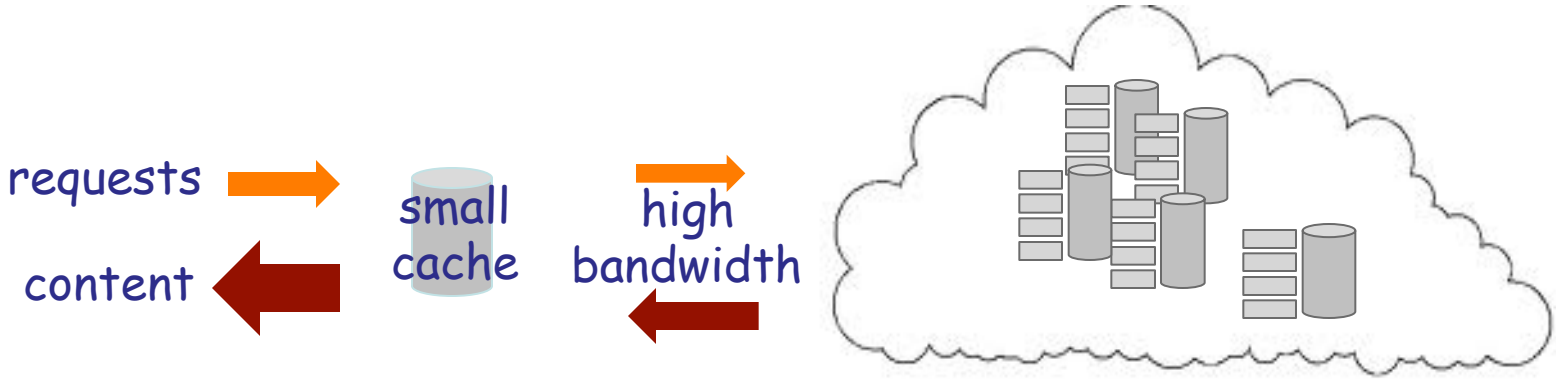
- eg, assume 1 TB cache
  - shared cache hit rate = .17
  - dedicated VoD cache hit rate = .23 (ie, all VoD traffic)

Outline

- Internet traffic control
- ICN traffic control
- Impact of in-network caching
  - performance of an LRU cache
  - the memory-bandwidth cost tradeoff

Evaluating the memory-bandwidth tradeoff

- assuming the "independent reference model"
- accounting for a very large content catalogue...
  - web pages, user-generated content, file sharing, video
  - petabytes of content
- and highly dispersed content item popularities
  - Zipf-like behaviour with exponent < 1
- an example from BitTorrent trackers... [Dan & Carlsson, IPTPS 2010]



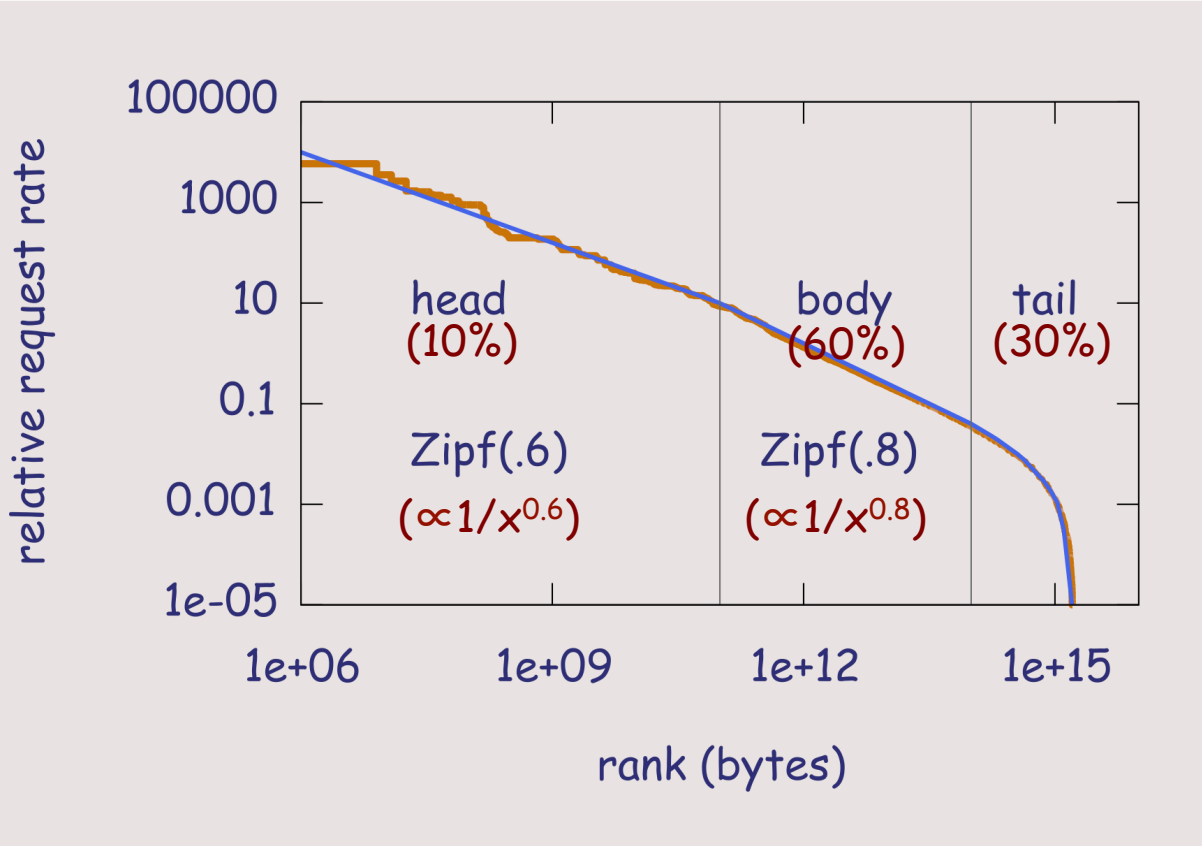
A popularity law for torrents

- the number of BitTorrent leechers is a measure of instantaneous popularity
- assume q(n) is request rate for torrent n; we assume

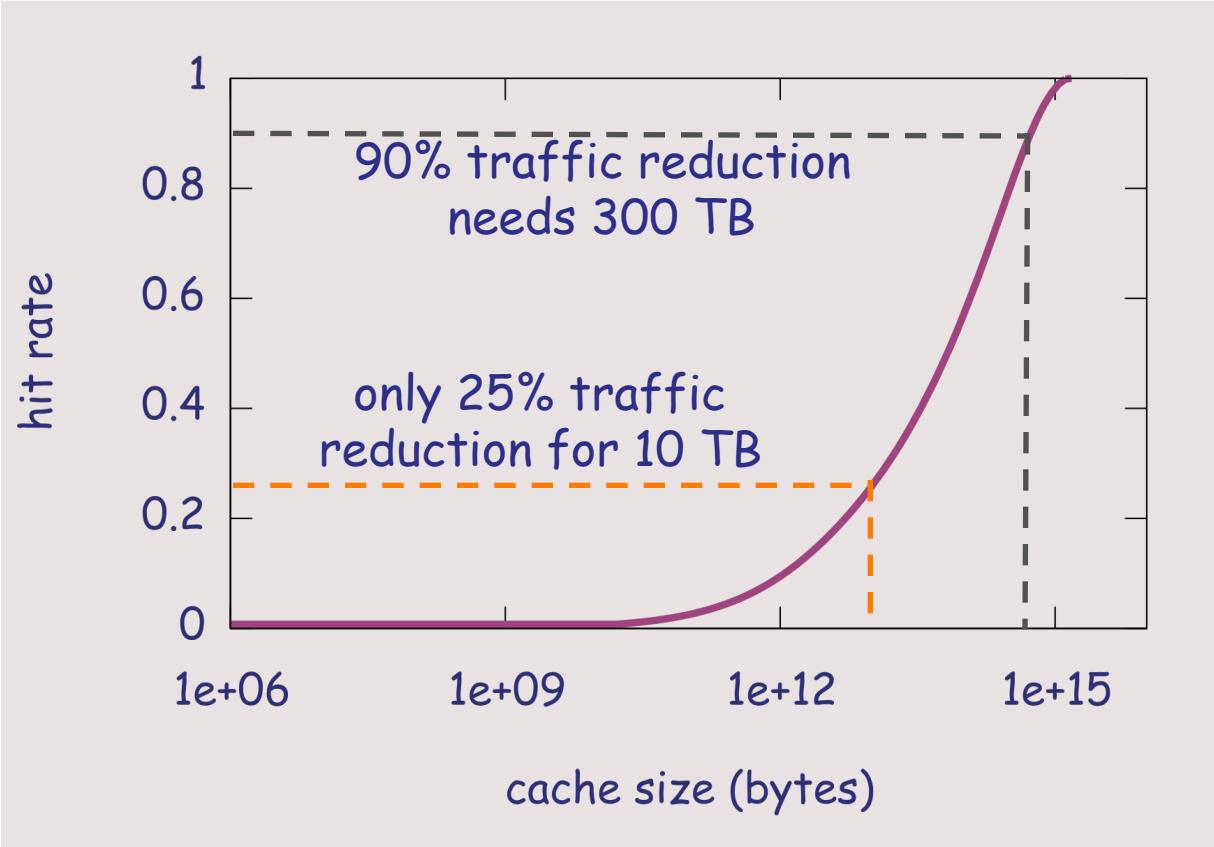
$$q(n) \propto l(n) / s(n)$$

  where l(n) is number of leechers, s(n) is torrent size
- l(n) and s(n) can be derived by "scraping" trackers
  - cf. G. Dan and N. Carlsson. Power-law revisited: large scale measurement study of p2p content popularity, IPTPS 2010
- NB. other popularity measurements are carried out over a period (eg, most watched videos in 1 week) and do not account for temporal locality

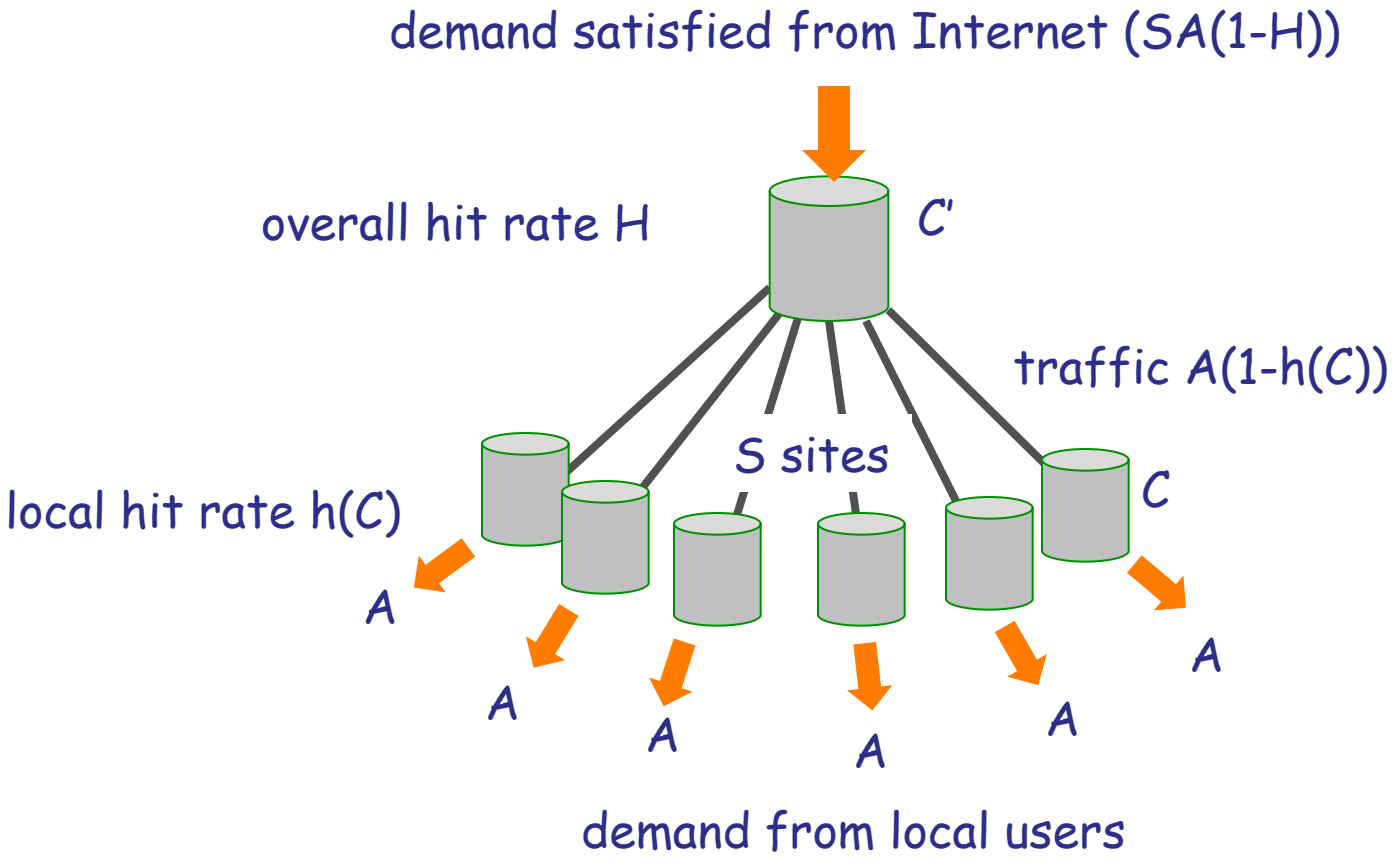
A popularity law for torrents



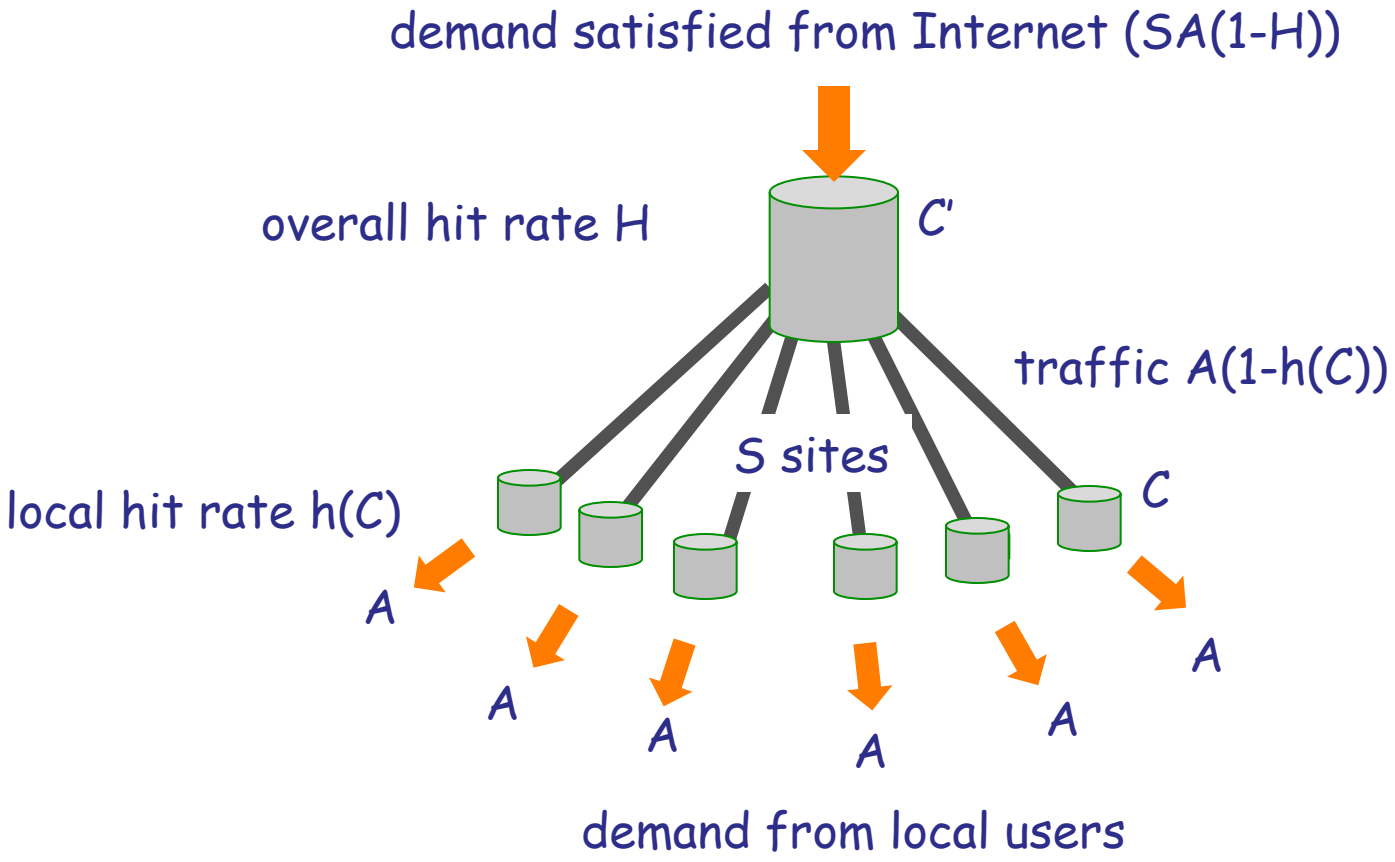
LRU hit rate versus cache size for torrents



A two-layer cache hierarchy



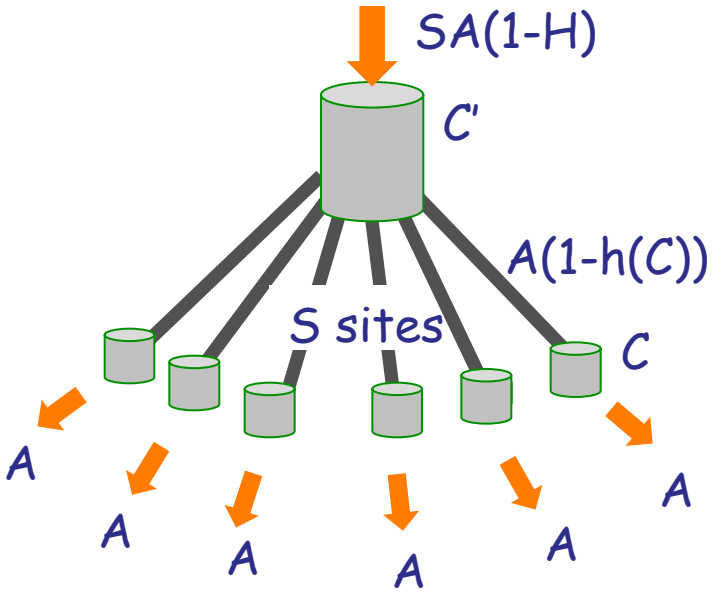
A two-layer cache hierarchy



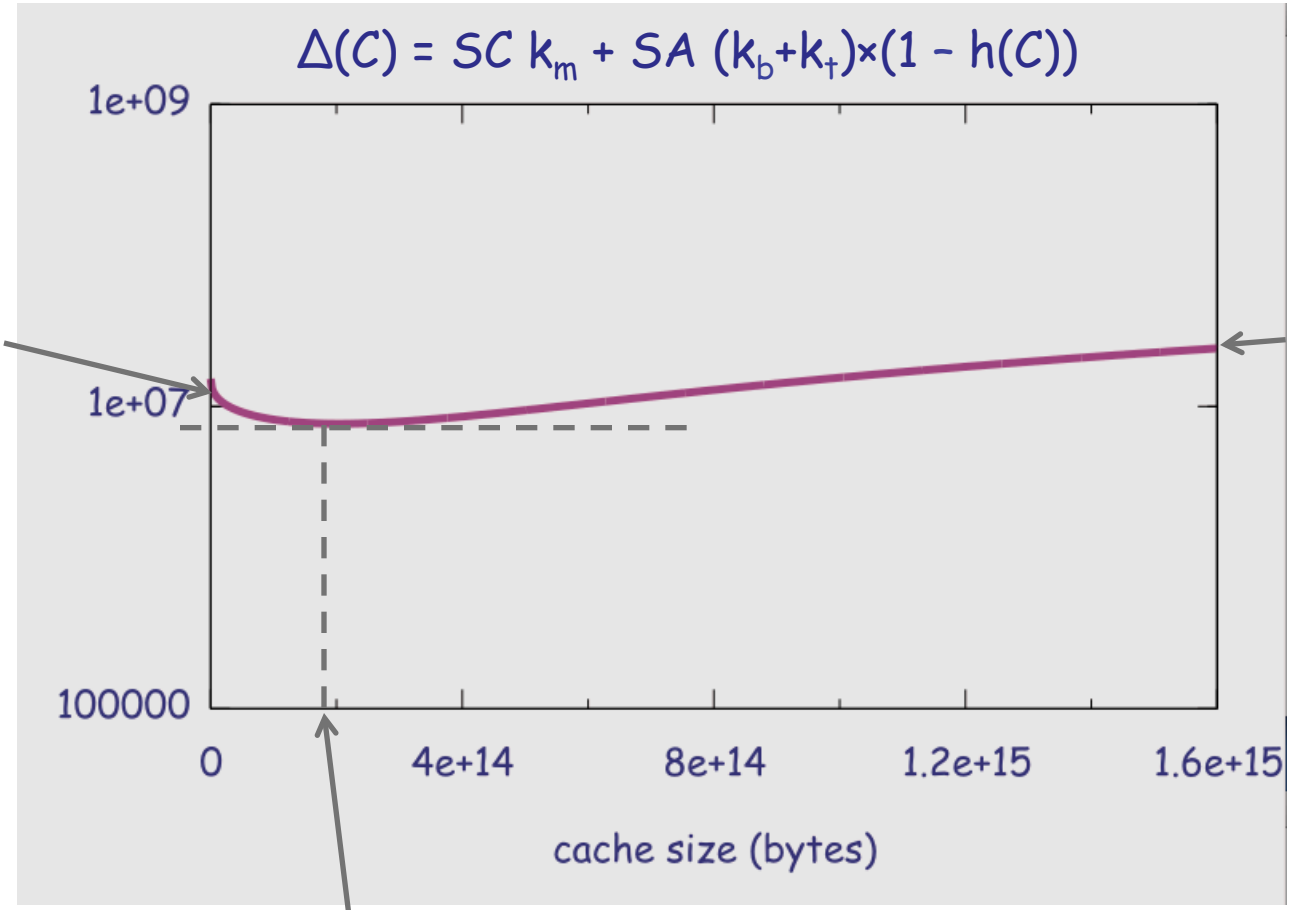
Cost evaluation

- cost = Internet + access + caches + bandwidth
- assume A, S fixed, C' contains all content (ie, $H = 1$)
- a monthly cost of caching
  - memory: k_m per GB $\Rightarrow k_m(SC+C')$
  - traffic: k_t per Mb/s $\Rightarrow k_t(SA+SA(1-h(C)))$
- a monthly cost of bandwidth
  - bandwidth: k_b per Mb/s $\Rightarrow k_b SA(1-h(C))$
- cost variation as C varies:

  $\Delta(C) = SC\,k_m + SA\,(k_b+k_t)\times(1 - h(C))$
- cost "guesstimates":
  - k_m = €0.15 per GB
  - k_t+k_b = €15 per Mb/s



Costing the tradeoff

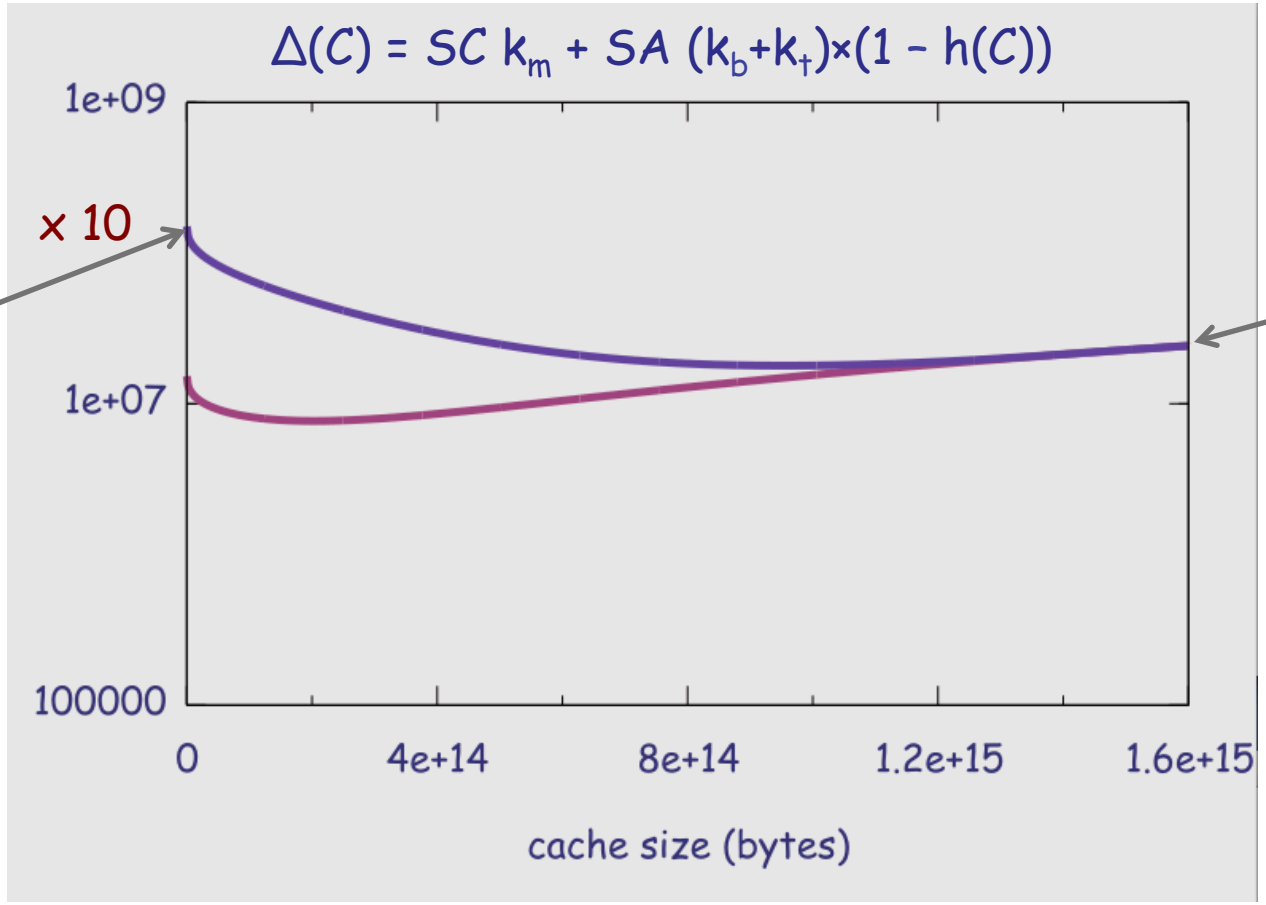


max cost of bandwidth ($C = 0$)

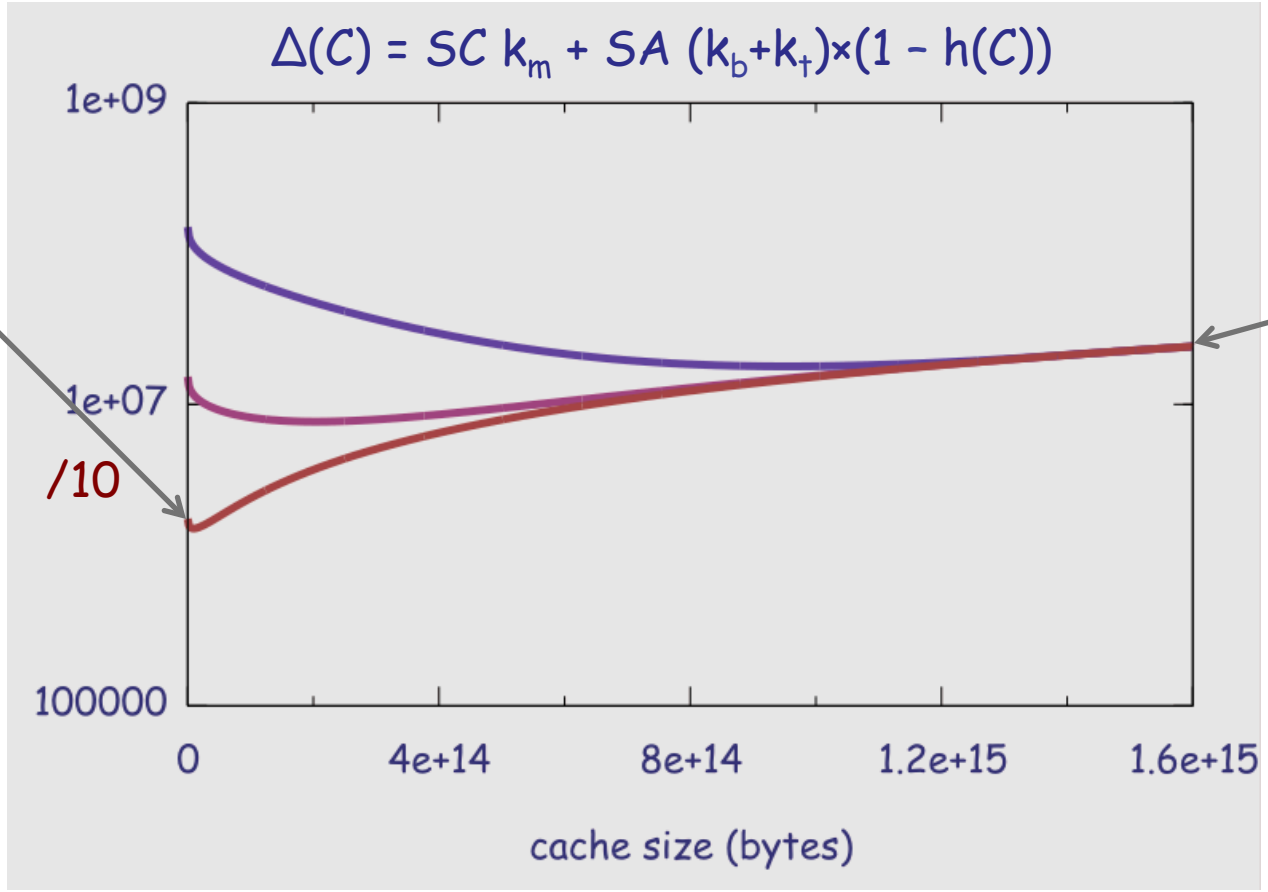
max cost of memory ($C = 1.6\times10^{15}$)

optimal size

Costing the tradeoff



Costing the tradeoff

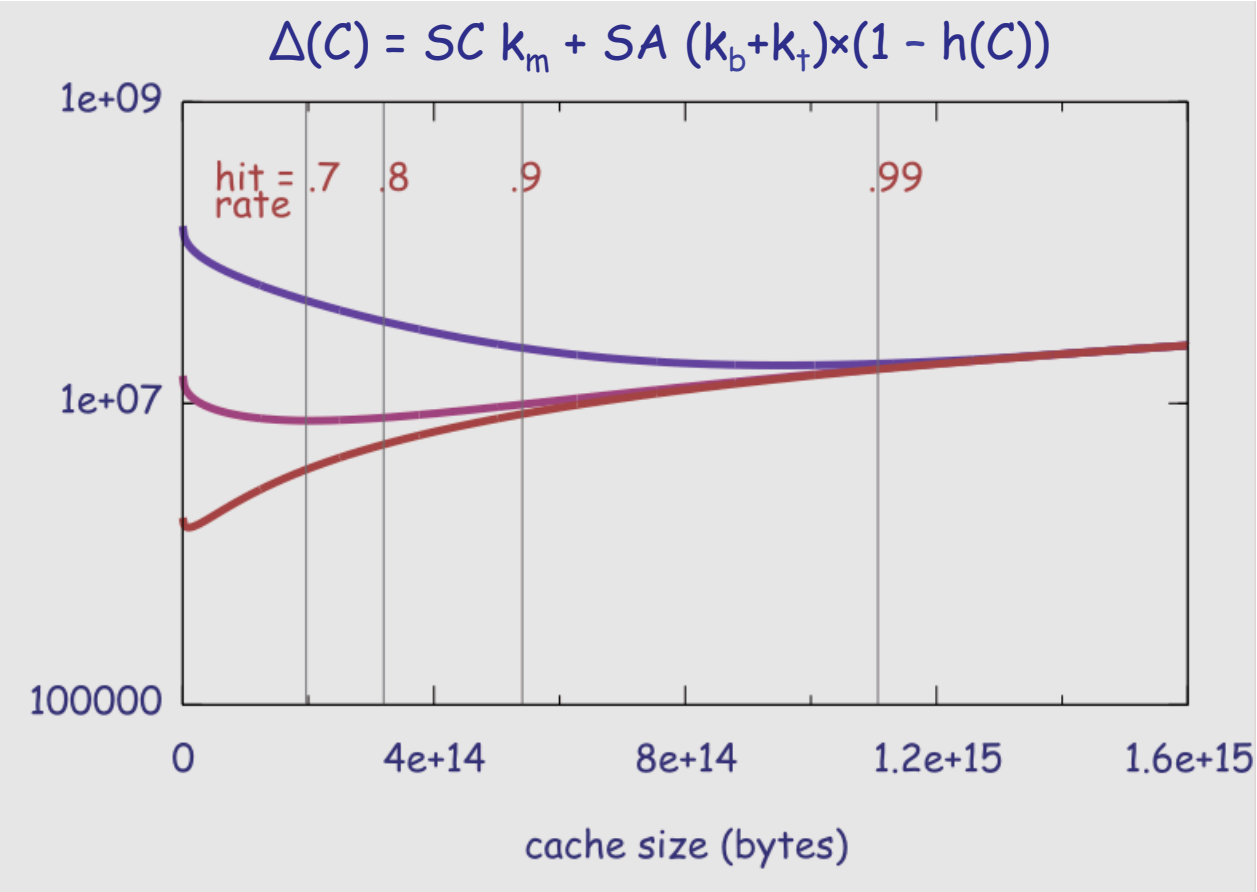


max cost of bandwidth

/10

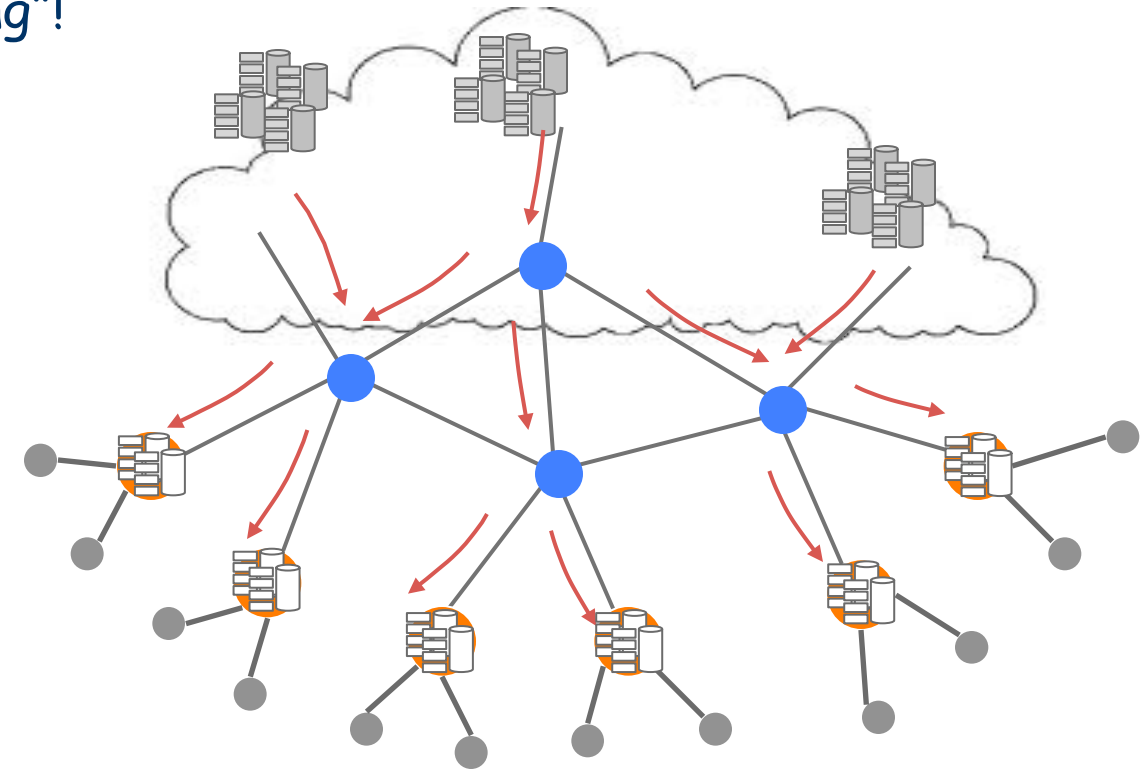
max cost of memory

Costing the tradeoff



Large storage, low bandwidth

- using best guess cost data suggests large (~petabyte) stores capturing more than 90% of content traffic are cost effective
- instead of "routers that do caching", we have "data centers that do routing"!



Outline

- Internet traffic control
- ICN traffic control
- Impact of in-network caching
- Conclusions

ICN as a flow-aware network of data centers

- flow aware traffic control
  - per-flow fair queuing in router buffers
  - implicit service differentiation and no reliance on user cooperation
  - need to ensure link load < 0.9 by TE and overload controls
- transposition of flow-aware control to ICN
  - need flow identifier – object name in packet headers…
  - and "interests buy data", interest discard, ECN…
  - and receiver-based congestion control
- the role of in-network caching
  - reduces traffic by caching popular content close to users
  - but content catalogues are huge and popularity is widely dispersed
  - very large caches are necessary and cost effective
- a future ICN based on data centers that also do routing!