Design and Analysis of Content Caching Networks

Jim Kurose
Department of Computer Science
University of Massachusetts
Amherst MA USA

Visiting Scientist, Technicolor Paris/Palo Alto Labs
Professeur Invite, LINCS

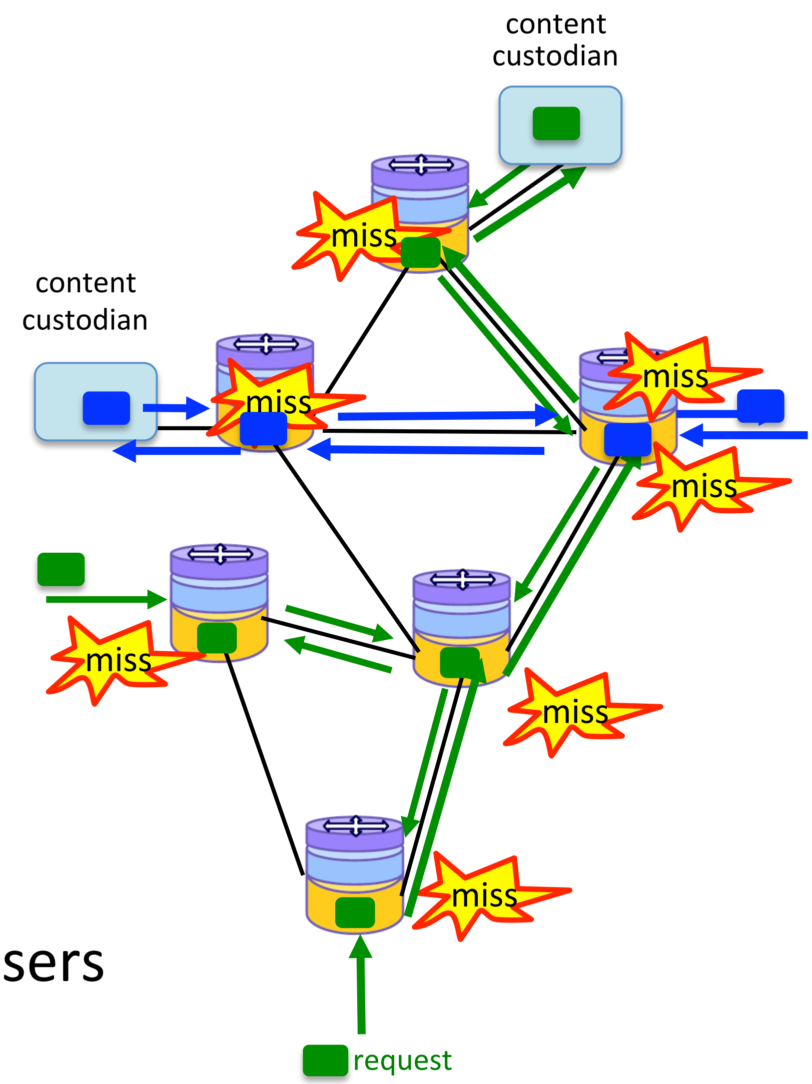
Workshop on Storage, Nov. 2012

Overview

- networks of caches
- breadcrumbs: content location/caching
- analysis of *networks* of caches
  - approximation algorithms
  - network calculus for cache networks
  - ergodicity

Networks of caches: ICN scenario

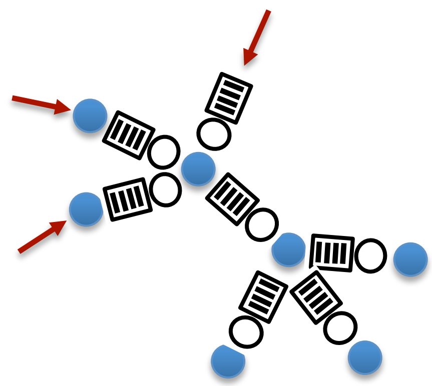
- consumer *requests content*
- request *routed* (e.g., shortest path) to known *content custodian*
- en-route to custodian, *caches* inspect request
  - *hit:* return local copy
  - *miss:* forward request towards custodian
- during download, content *stored* at caches along path
- content requests from different users interact: *cache replacement*



Note: ICN thinks wide area, but CamCube fits this model

Cache *networks*

- *network effect: interaction* among content request/ reply flows from different users:
  - *content replacement*: requested content by one user replaces content previously requested by others



Packet-switching:
queueing networks
(Kleinrock, 1963)

Networks of caches: challenges!

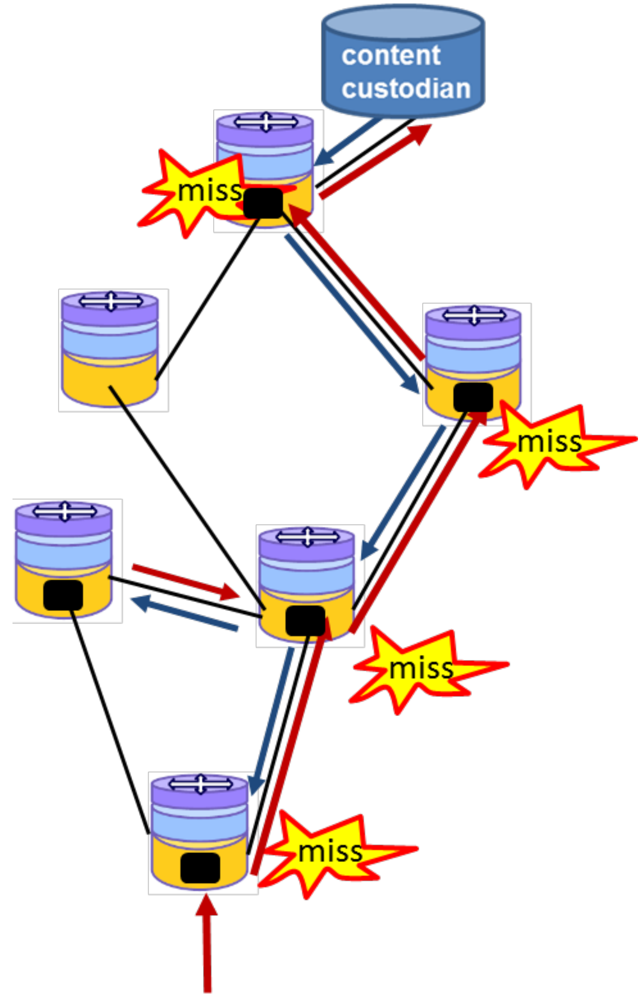
- locating content: known content location; opportunistic caching within network
  - *implicit:* search for content on shortest path to custodian
  - *explicit*: adaptive, state-dependent, and/or randomized search
  - similarities/difference from P2P search
- managing content: cache replacement, cache loading/ prefetching
- analyzing networks of caches

Overview

- introduction: content-centric networks
- caching networks
- *breadcrumbs*: best-effort content location/caching
- analysis of *networks* of caches
  - approximation algorithms*
  - network calculus for cache networks*
  - ergodicity*

Breadcrumbs: motivation

- *goal:* access content, given known custodian location
- explicit or implicit cache search policy
  - *implicit:* search for content on shortest path to custodian
  - *explicit*: adaptive, state-dependent, and/or randomized search
- *Q:* how much state to maintain about network cache contents?

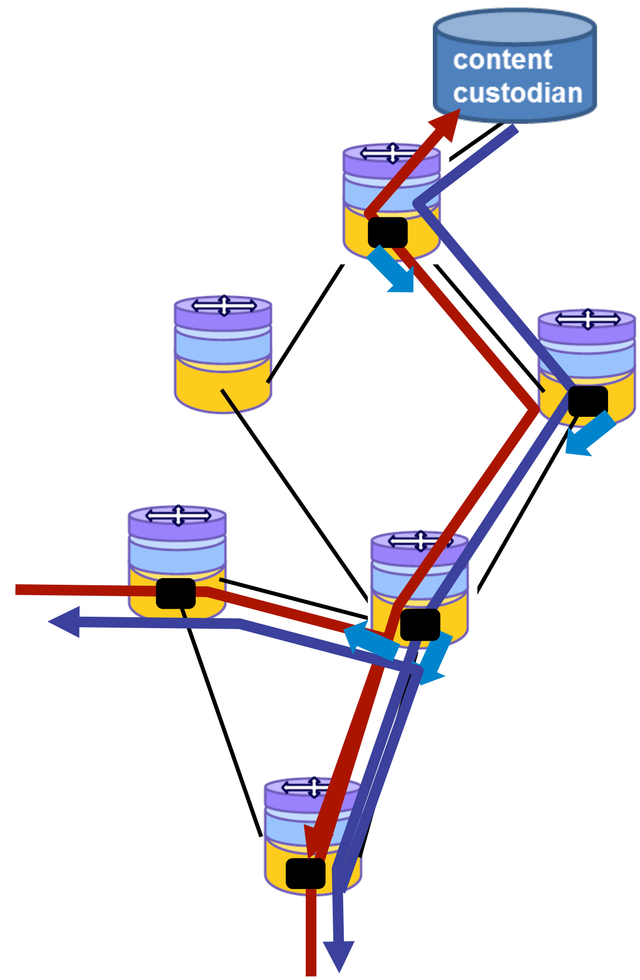


Breadcrumbs

Breadcrumb: per-file soft state - next-hop neighbor where file was last forwarded

Breadcrumbs search

- request starts on route (shortest path) to custodian
- enroute cache holding file sends file to requestor
- cache with recent (breadcrumb can route request following breadcrumb) or to custodian
  - if **dead-end** reached: reroute to custodian
- soft-state timeout

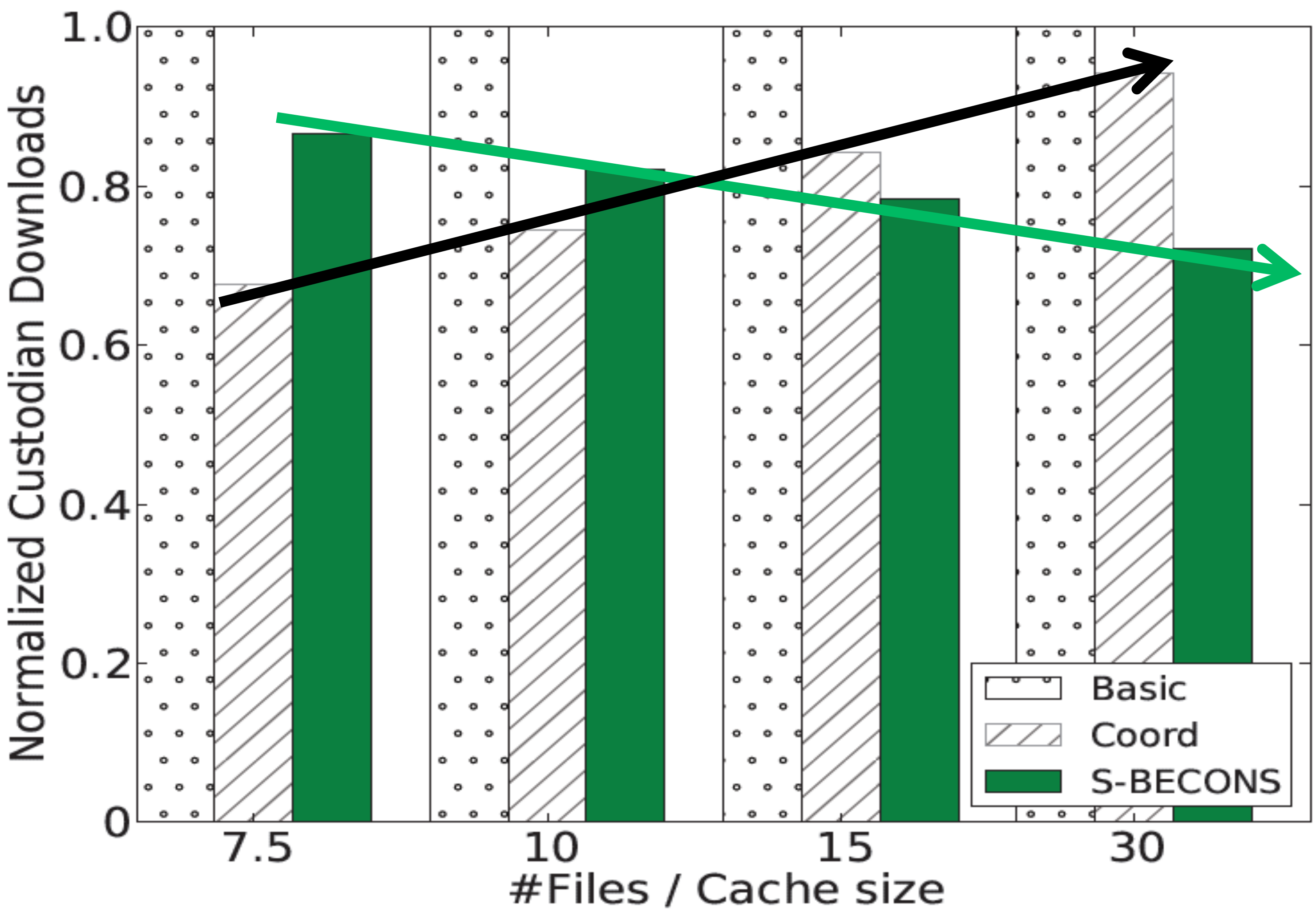


- breadcrumb path properties: stability, invalidation

Breadcrumbs: simulation analysis

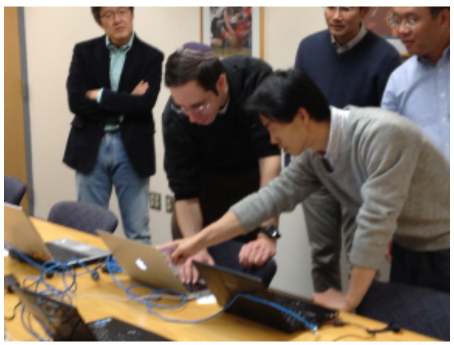
- nxn torus, Zipf and uniform content access probability, 300 files, varying cache sizes
- compared to
  - caching w/ no Breadcrumbs (“Basic”),
  - coordinated caching with upstream neighbor
- performance metric
  - minimize load on custodians
- sample results
  - *Breadcrumbs* performance improves (relatively) as network size grows, and as the files-to-cache size ratio grows

Breadcrumbs: sample results



Breadcrumbs: future work

- tradeoff between time to locate content, cache/ custodian load balancing
- estimating, exploiting cache eviction rates of neighbors



NEC implementation

“Breadcrumbs: efficient, best-effort content location in cache networks,” Elisha J. Rosensweig and Jim Kurose, *2009 IEEE INFOCOM Mini-conference*

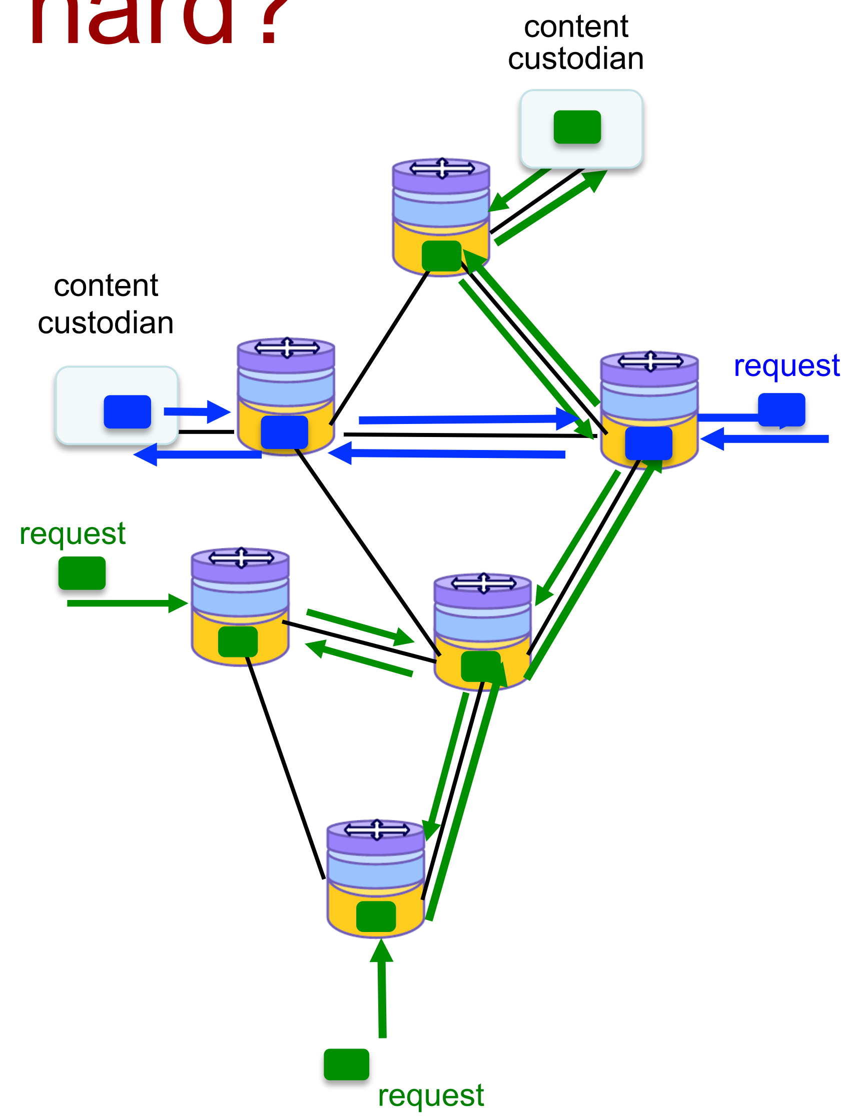
“Performance Evaluation of Partial Deployment of Breadcrumbs in Content-Oriented Networks,” T. Tatsuhiro et al, *2012 IEEE ICC Futurenet*

Overview

- introduction: content-centric networks
- networks of caches
- content location/caching
- analyzing *networks* of caches
  - approximation algorithms
  - network calculus for cache networks
  - ergodicity

Why is the problem hard?

- simultaneous allocation of resources (cache storage) from point where content is located back to requestor
- asynchronous resource release, depending on cross flows
- inter-cache flows complex:
  - miss stream from downstream caches
  - lack of independence

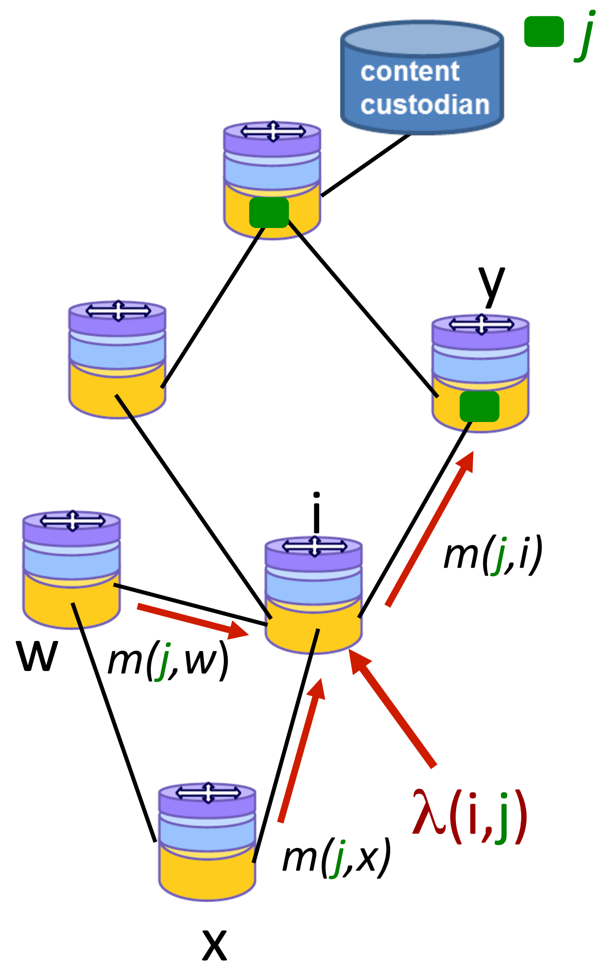


Modeling a network of caches

- node *i*: *exogenous* (external) arrivals for content *j*: $\lambda(i,j)$
- node *i*: *internal* arrivals (miss stream) for content *j* from downstream neighbors *h*: $m(j,h)$
- $r(i,j)$: aggregate rate of arrival requests at *i* for content *j*

$$r(i,j) = \lambda(i,j) + \sum_{\substack{\text{all downstream} \\ \text{neighbors, } h}} m(j,h)$$

- ZDD: zero download delay assumption

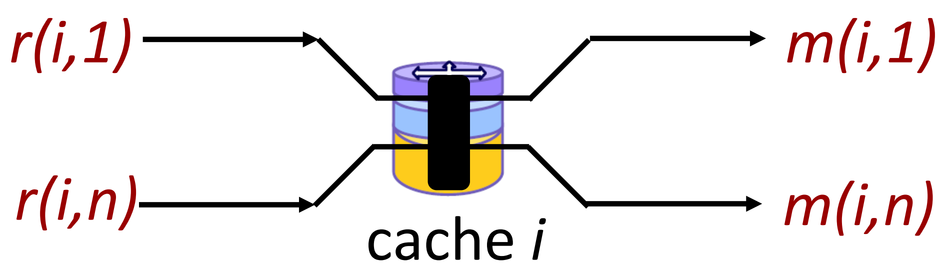


Approximating cache network performance

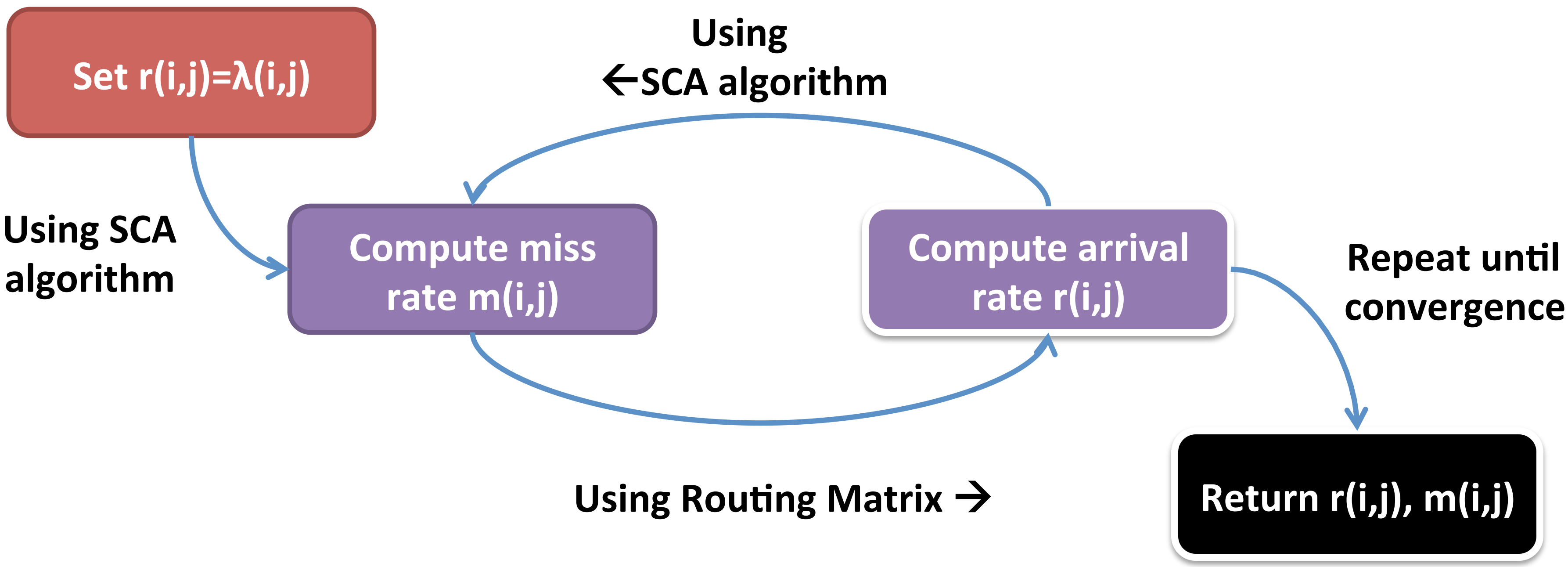
- **SCA:** standalone cache *i* approximation algorithm: given *r(i,j),* compute miss rate for all content j
- Independence Reference Model (IRM) of incoming requests:

$$Pr(X_t = f_j \mid X_1,..,X_{t-1}) = Pr(X_t=f_j)$$

- SCA approximation algorithm for LRU: [Dan 1985]



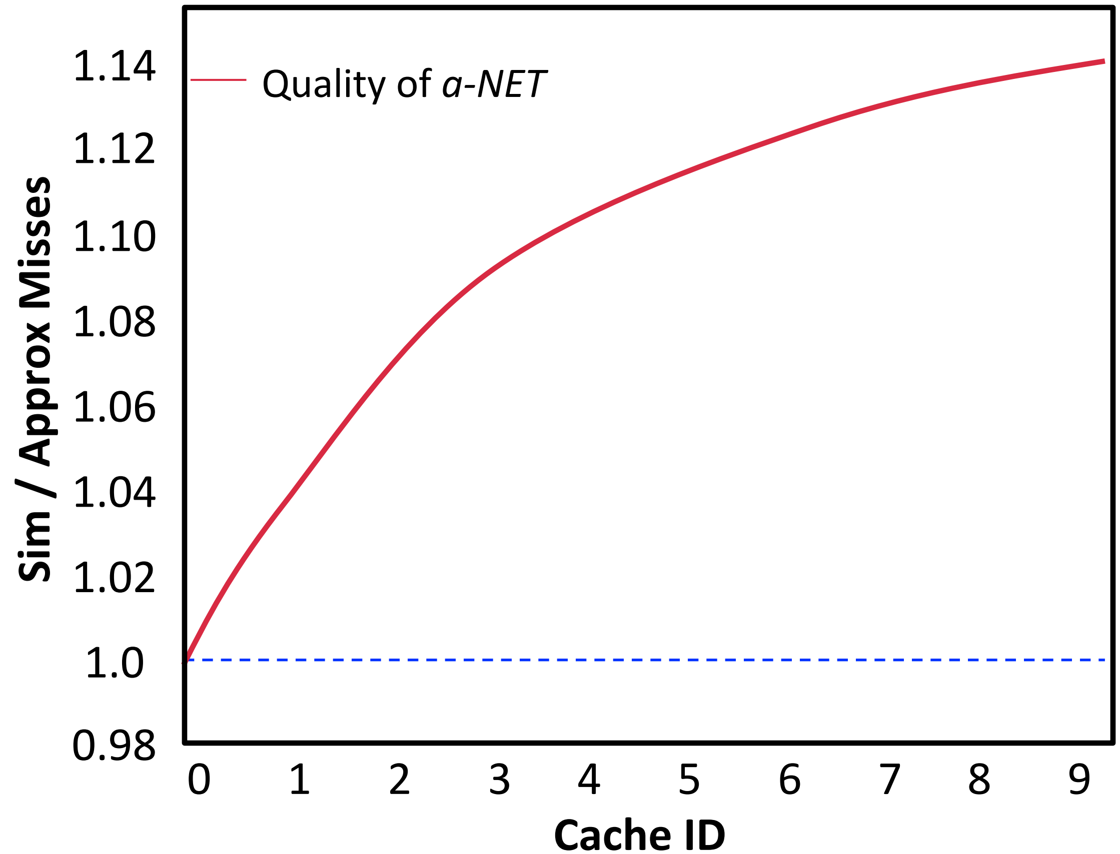
Fixed-point iteration



Note: tree-network (feed-forward) require single iteration

Fixed point approximation error

- line topology with 9 nodes
- errors decrease in networks with high node degree



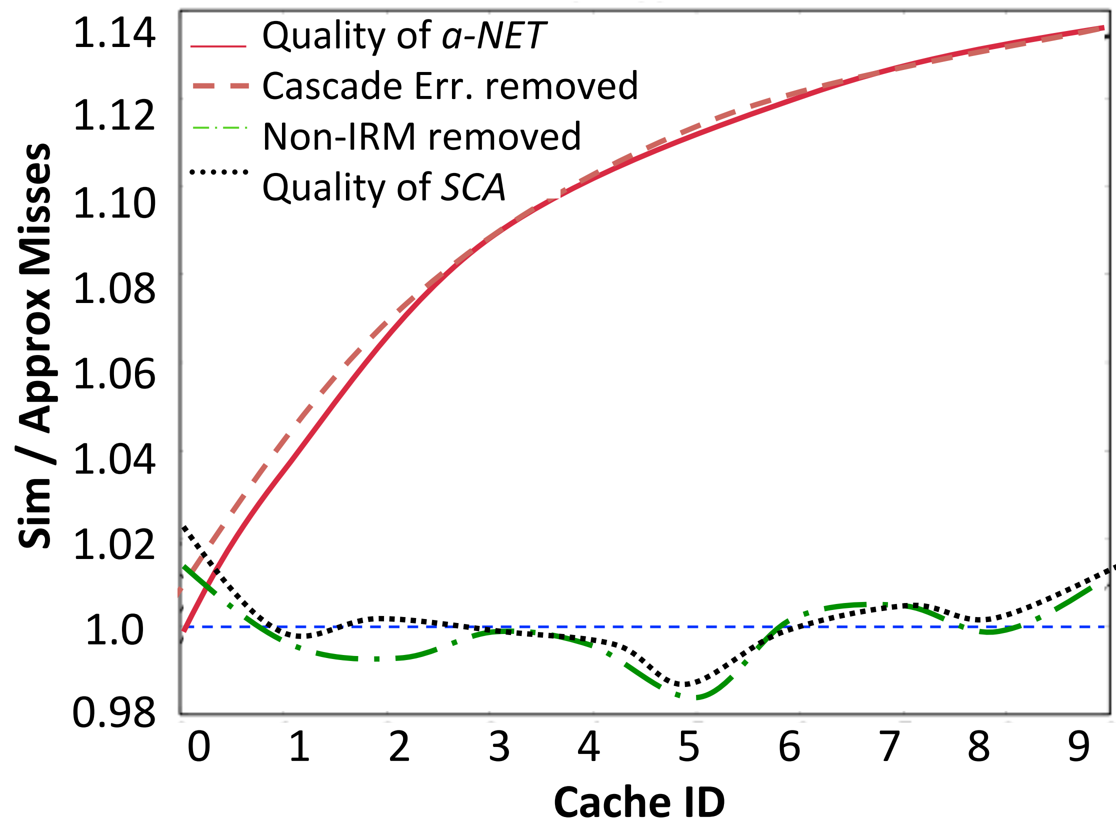
Error factor analysis

sources of approximation errors:

- SCA algorithm inaccuracies?
- cascading errors?
  - approximated output rates of one iteration is input to next iteration
- violating IRM assumed by SCA algorithm?
  - miss process for file *j* negatively correlated

Error factor analysis

- Factor analysis reveals that *non-IRM input* to SCA is main cause of error



"Approximate Models for General Cache Networks,"
Elisha J. Rosensweig, Jim Kurose, Don Towsley, *2010 IEEE INFOCOM*

Overview

- introduction: content-centric networks
- caching networks
- *breadcrumbs*: best effort content location/caching
- analysis of *networks* of caches
  - approximation algorithms
  - network calculus for cache networks
  - ergodicity

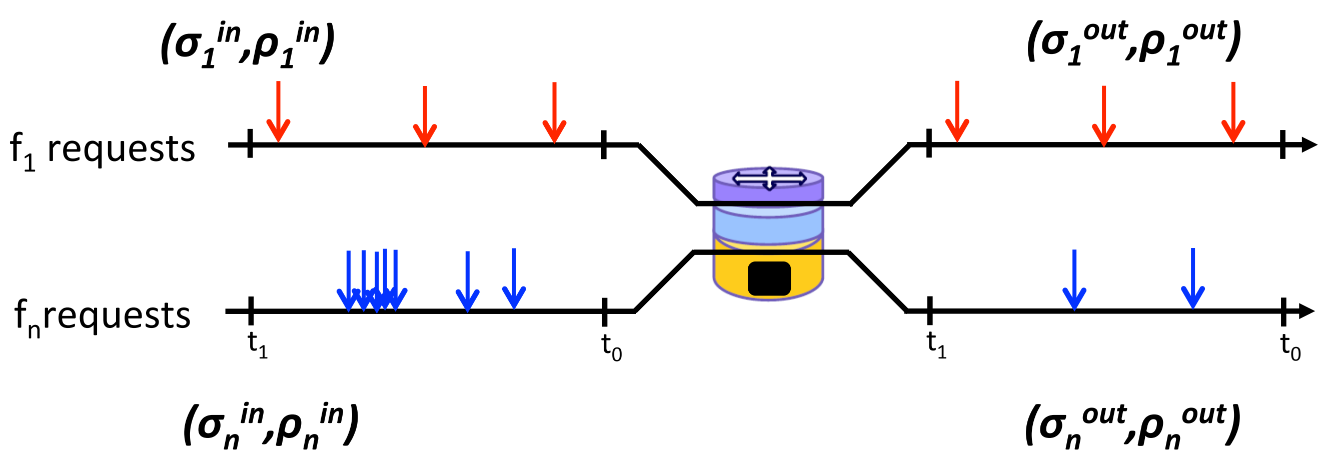
(σ_i,ρ_i) analyses of cache networks

(σ_i,ρ_i) bounds *# requests for f_i* over $[t_1,t_2]$:

$$\int_{t_1}^{t_2} r_i(t)\,dt \le \rho_i(t_2 - t_1) + \sigma_i$$

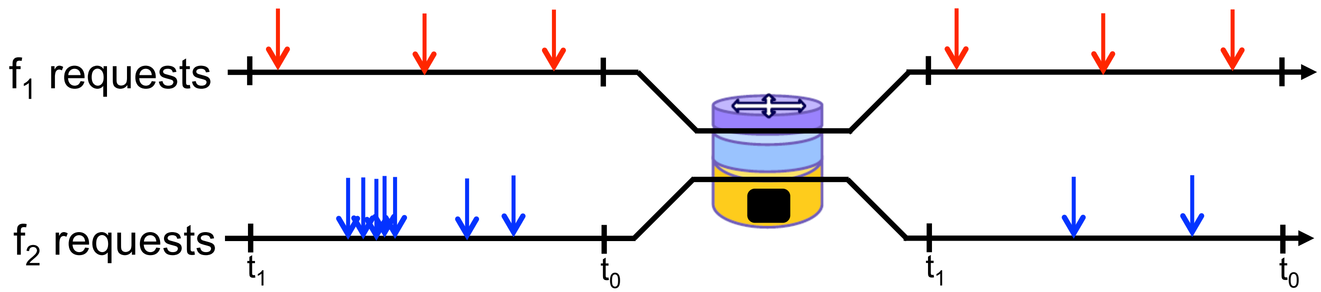
where $r_i(t)$ = request rate for f_i at time t

Goal: a network calculus for cache networks:



(σ_i,ρ_i) cache networks: observations

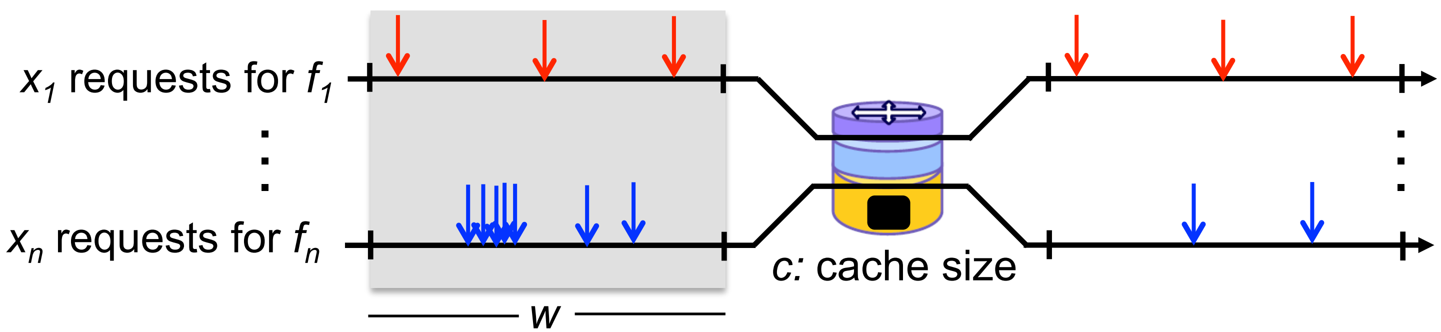
- not all requests arriving at cache will leave (unlike queue)



- stream of input requests for one file only generates no output
  - interactions among files in cache critical
- "burst" of request for same file generates one output
  - different intuition (from queues) about "performance damage" of bursts

Building block: miss set, M_i

miss set for f_i: set of requests for *c* unique files, other than *i*

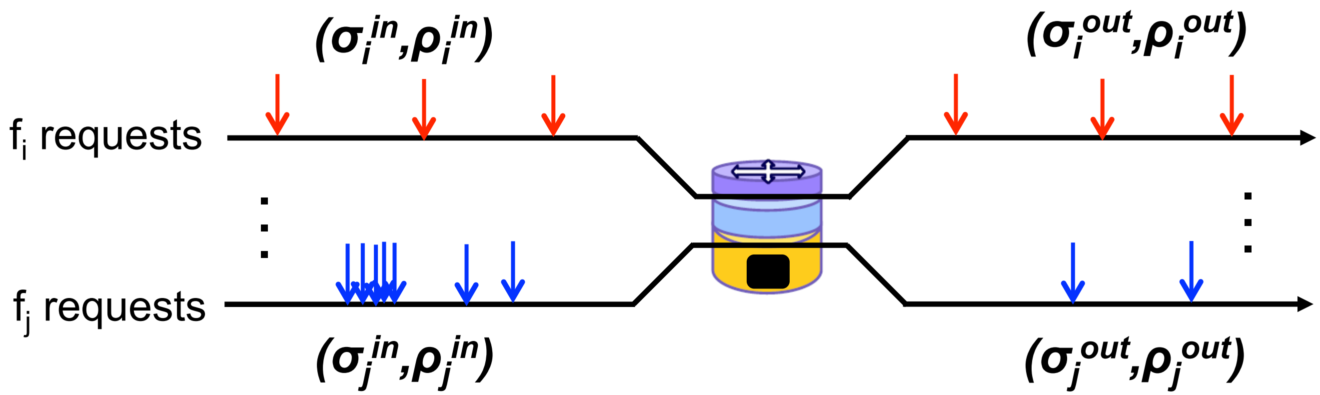


$M_i(x_1, \ldots., x_n, c)$: max number miss sets for file f_i, given $\{x_i^{in}\}$ arrivals, cache of size c.

properties:

- $M_i = min(x_i^{in}, M)$ *!!*
- $x_i^{out} \leq M_i$, and this bound is achievable

From $(\sigma_i^{in},\rho_i^{in})$ to $(\sigma_i^{out},\rho_i^{out})$:

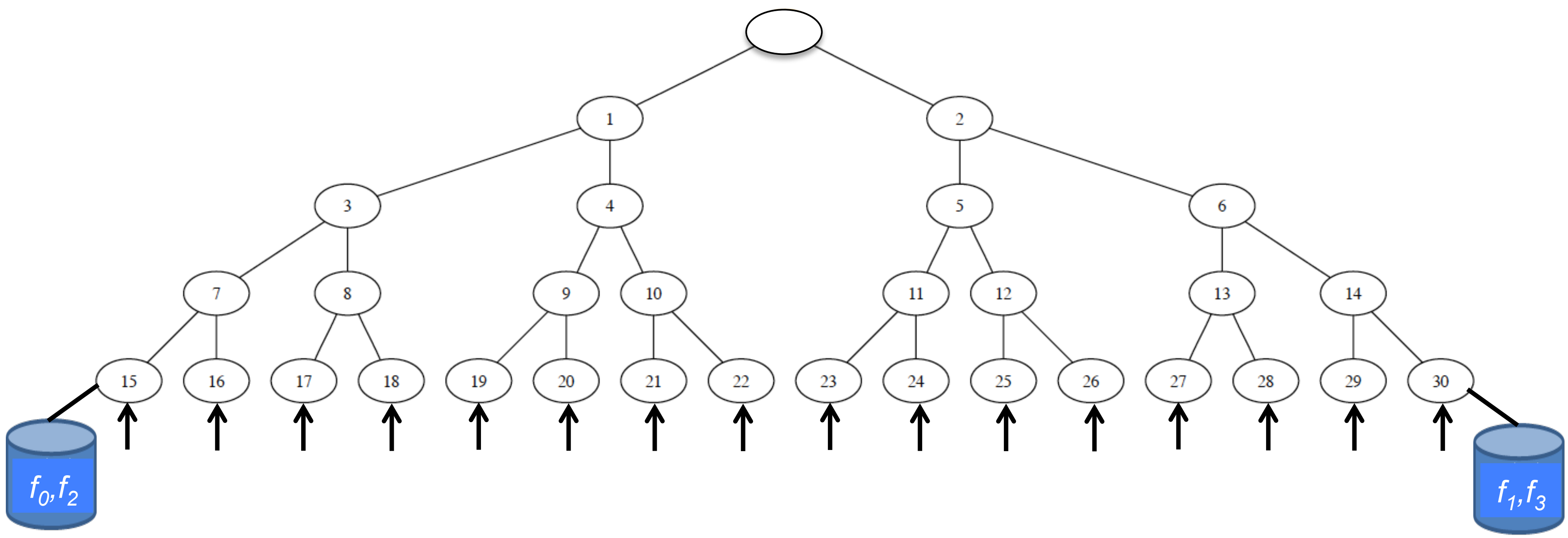


Theorem: For a cache of size *c*:

$$\rho_i^{out} = \min(\rho_i^{in}, M_i(\rho_1^{in}, \ldots, \rho_n^{in}, c))$$

Can calculate ρ_i^{out} from $\{\rho_i^{in}\}$

Numerical example

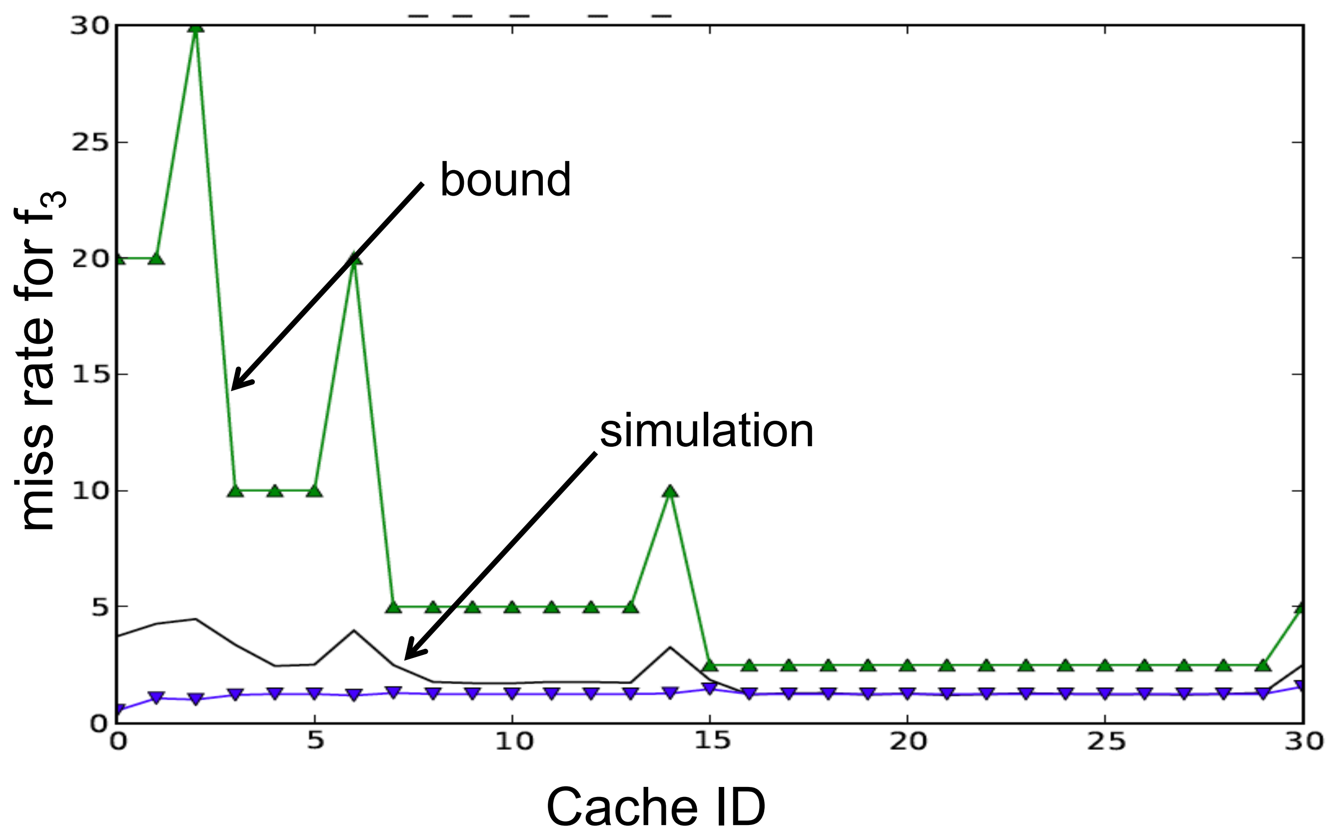


cache size = 2 at each node

homogeneous IRM arrivals, exponential interarrival times

4 files, uniform popularity distribution

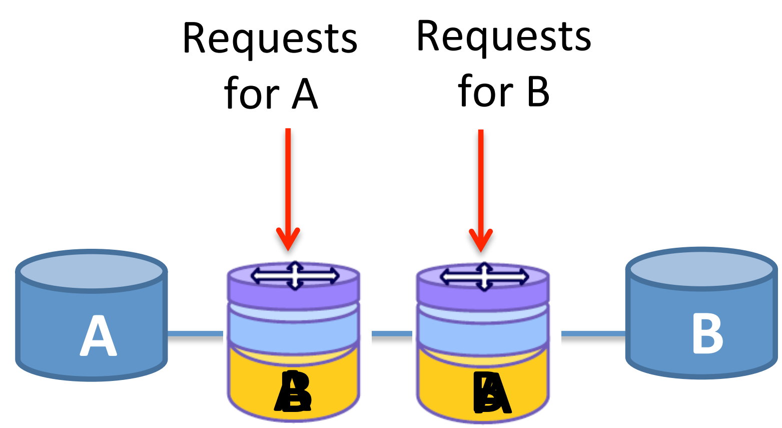
Numerical example: bounding results



E. Rosensweig, J. Kurose, “A Network Calculus for Cache Networks,” to appear in *IEEE Infocom 2013.*

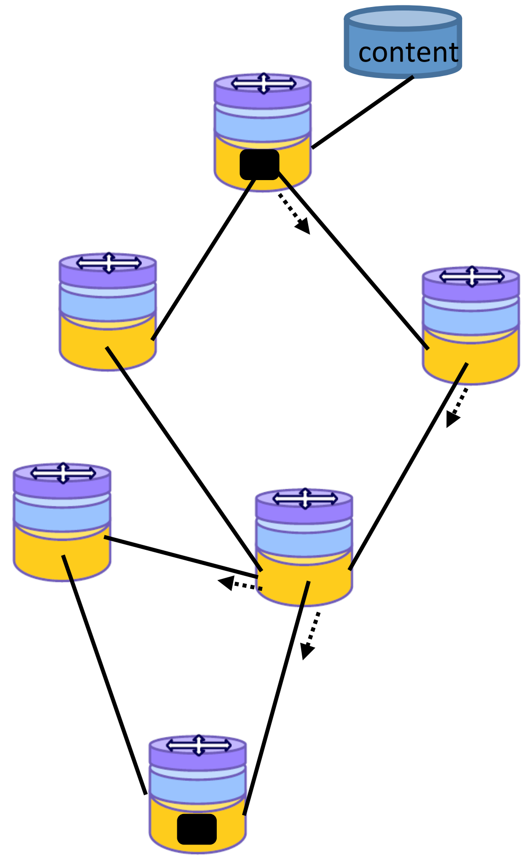
Ergodicity of cache networks

- does steady state performance depend on initial conditions (ergodicity)?
  - shown existence of non-ergodic cases (replacement policy, topology, cache size)
  - derived sufficient conditions for ergodicity
    - topology (single-custodian trees)
    - from individual ergodicity to system ergodicity



Ergodicity of cache networks

- ergodicty (continued):
  - showed random replacement: ergodic
  - defined class of non-protective policies (including LRU): all ergodic



E. Rosensweig, D. Menasche, J. Kurose, “On the Steady-State of Cache Networks,” to appear in *IEEE Infocom 2013.*

Future work

- cache network protocols
  - multipath: download from multiple custodians
  - mobility
  - proactive caching, replacement in video-content distribution nets
- network calculus
  - continued development of deterministic bounds
  - stochastic bounding techniques
- approximate analysis of cache networks
  - asymptotic (Kelly, Bonald) results

The end

?? || /* */