Cache "Less for More" in Information-Centric Networks

W. K. Chai, D. He, I. Psaras and G. Pavlou (presenter)

Department of Electronic & Electrical Engineering
University College London
London WC1E 6EA
United Kingdom

Background: Caching in Information-Centric Networking (ICN)

- In Information-Centric Networking (ICN), contents are ***named***.
- Content requests can be served from ***any*** node having the content identified by the content name.
- ICN features ubiquitous in-network caching:
  - potentially every router caching indiscriminately *all* content fragments that traverse it
  - if a matching request is received while a fragment is still in its cache store, it will be forwarded to the requester from that element, avoiding going to the hosting server.
- Specifically, NNC* proposes:
  - A router **caches every content** chunk that traverses it with the assumption that routers are equipped with (large) cache stores.
  - A **least recently used (LRU)** cache eviction policy is used.

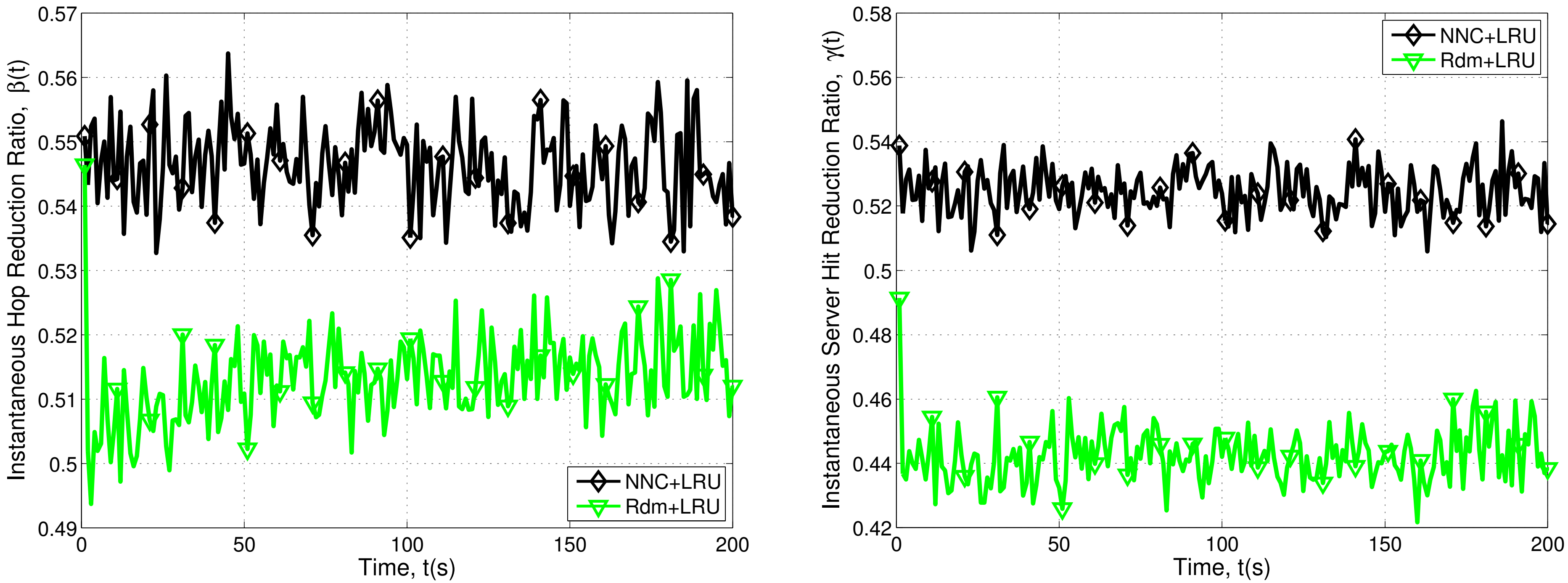
*V. Jacobson, et al., "Networking Named Content," Proc. ACM CoNEXT, 2009, pp. 1-12.

Argument & Objective

- The **Argument**: indiscriminate universal caching strategy is unnecessarily costly and sub-optimal
  - High content replacement frequency may result in content being replaced before getting a hit.
- The **Objective**: to study alternative in-network caching strategies for enhancing the overall content delivery performance.
  - We address the central question of whether caching only at specific sub-set of node(s) en route the delivery path can achieve better gain.
  - If yes, which are these nodes to cache and how can we identify them?

"Cache All" vs "Random" Strategies

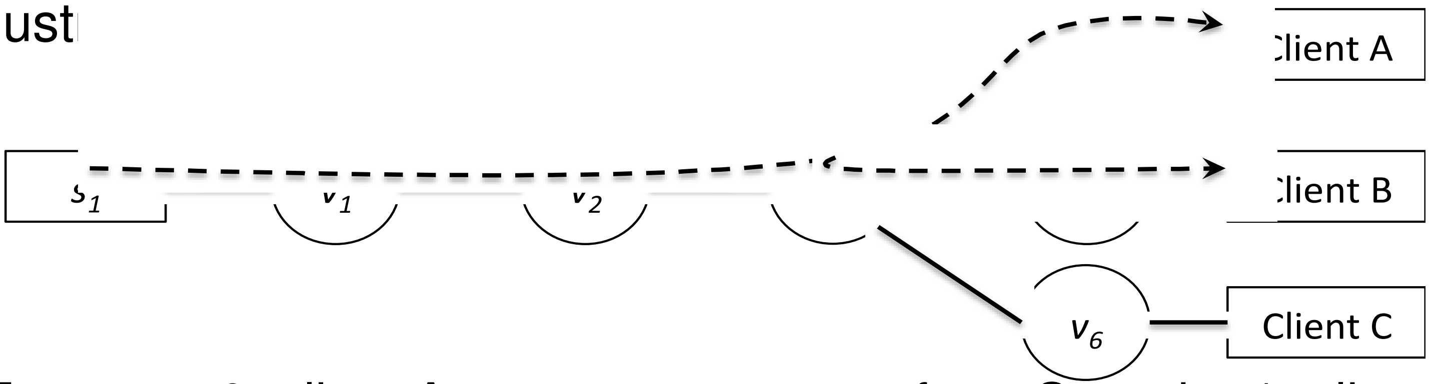
- Ubiquitous caching (***NNC+LRU***): caches every content in every node along the delivery path and uses LRU eviction strategy
- Random caching strategy (***Rdm+LRU***): caches at one randomly selected node along the delivery path and also uses LRU eviction



Simple random caching outperforming ubiquitous caching in the number of hops to hit the content (left) and reduced server hits (right).

Approach & Rationale

- The **approach**: to cache content only in selected node(s) for each content request
- The **rationale**: some nodes have higher probability of getting a cache hit than others.
  - By strategically caching the content at "**better**" node(s), we can decrease the cache eviction rate and increase the cache hit.
  - Illust



  - Events: t=0, client A requests a content from S_1 and t=1, client B requests a content from S_1
  - If NNC, ubiquitous caching at v_1, v_2, v_3 and v_4 but only v_3 *necessary for the request from client B*
  - v_3 being the "**important**" node in this case.

(Ego Network) Betweenness Centrality

- The **betweenness** centrality (***Betw+LRU)***: measures the number of times a specific node lies on the content delivery path between all pairs of nodes in a network topology.

$$betweenness\ centrality, C_B(v) = \sum_{i \neq v \neq j \in V} \frac{\sigma_{i,j}(v)}{\sigma_{i,j}}$$

  - The **basic idea** is that if a node lies along a high number of content delivery paths, then it is more likely to get a cache hit.
  - Caching only at those "*important*" nodes reduces the cache replacement rate while still caching content where a cache hit is most probable to happen.
- The **Ego Network Betweenness** centrality (***EgoBetw-LRU***):
  - The ego network consists of a node together with all its immediate neighbours and all the links among those nodes.
  - The idea is for each node, v to compute its $C_B(v)$ based on its ego network rather than the entire network topology.

Operation

- *Betw+LRU* and *EgoBetw+LRU* caching:
  - Operates at ***per request*** level.
  - The C_B value is pre-computed ***offline*** and configured to every router by the network management system.
  - The content request message records the highest centrality value among all the intermediate nodes.
  - This value is copied onto the content messages.
  - On the way to the requesting user, each router matches its own C_B against the attached one
    - The content is cached only if the two values match.
    - If more nodes have the same highest centrality value, all of them will cache the content.

Features

- *Betw+LRU* and *EgoBetw+LRU*:
  - cache content at nodes which have ***high probability of getting cache hits***
  - both approaches always ***spread*** content towards users
    - If a content is popular at a specific region of the Internet, then this content will eventually be cached in that region
  - is ***lightweight*** - each node independently makes its caching decision solely based on its own C_B
    - neither requiring information exchange with other nodes nor inference of server location or of traffic patterns.

Metrics

- In-network caching aims to:
  1. Reduce the content delivery latency whereby a cached content near the client can be fetched faster than from the server
  2. Reduce traffic and congestion since content traverses fewer links when there is a cache hit
  3. Alleviate server load as every cache hit means serving one less request.

$$Hop\ reduction\ ratio, \beta(t) = \frac{\sum_{r=1}^{R} h_r(t)}{\sum_{r=1}^{R} H_r(t)}$$

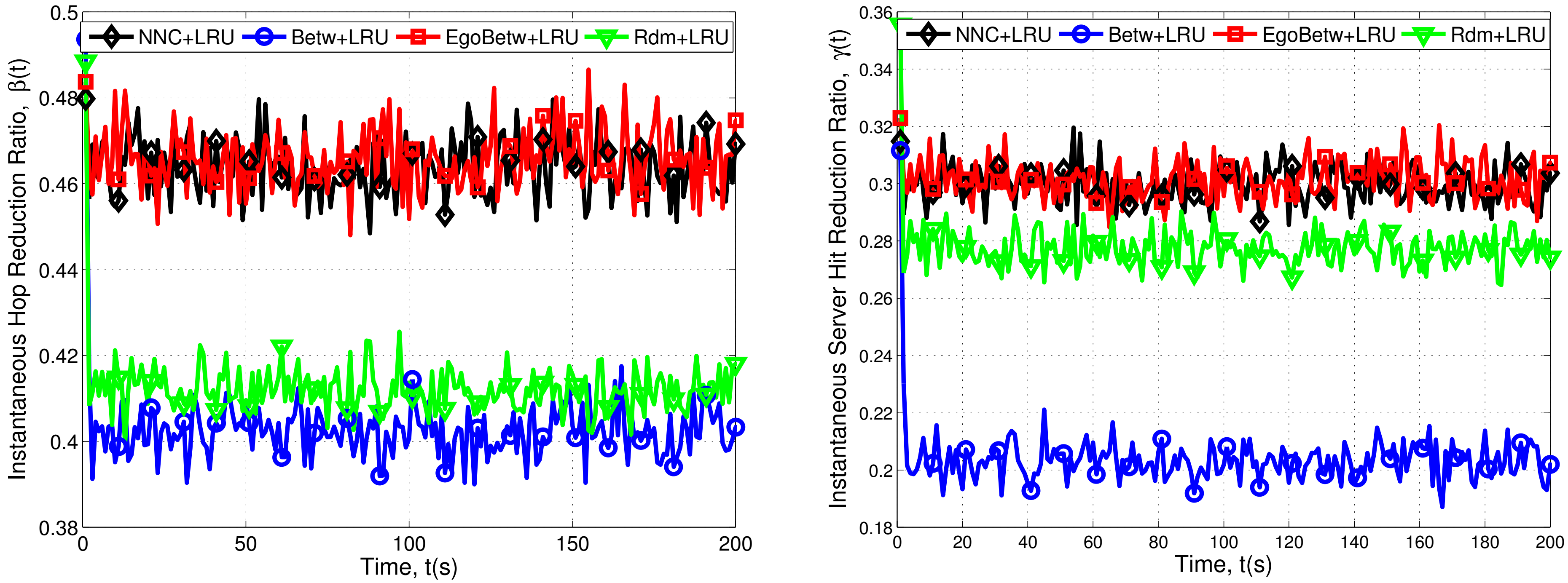
– where $H_r(t)$ = the hop count from client to server requesting f_r from time t-1 to t and $h_r(t)$ = the hop count from the content client to the first node where a cache hit occurs for f_r from t-1 to t.

$$Server\ hit\ reduction\ ratio, \gamma(t) = \frac{\sum_{r=1}^{R} w_r(t)}{\sum_{r=1}^{R} W_r(t)}$$

– where $W_r(t)$ = the number of request for f_r from t-1 to t and $w_r(t)$ = the number of server hits for f_r from t-1 to t.

Performance Evaluation on *k*-ary Trees

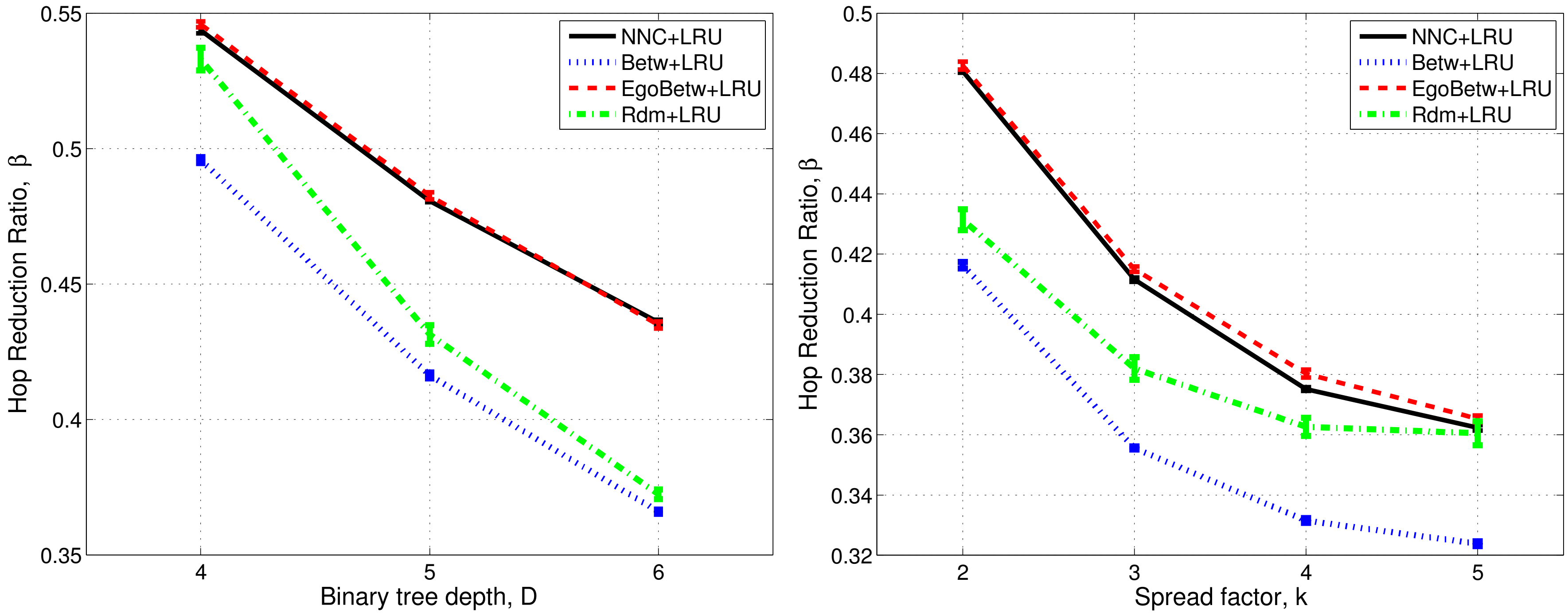
- Instantaneous performance for a 5-level binary tree (*k*=2, *D*=4)



Instantaneous behavior of the caching schemes for a binary tree; (left) β, (right) γ.

Performance Evaluation on *k*-ary Trees

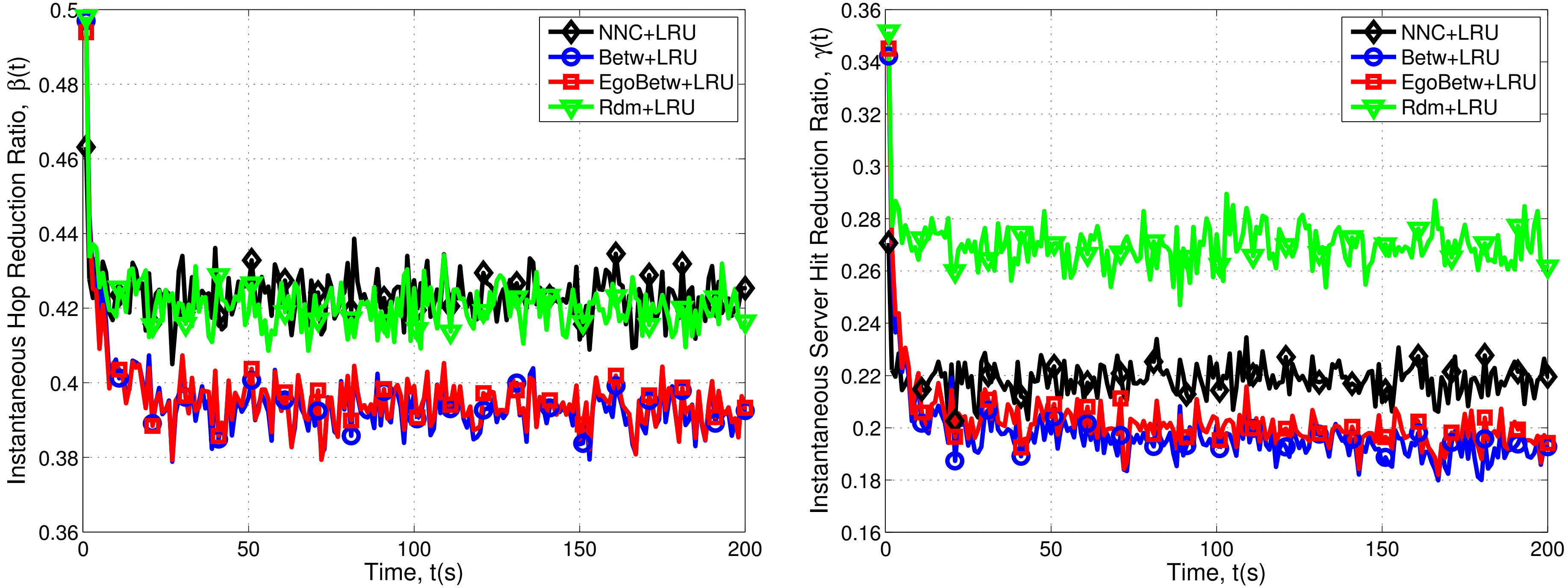
- *k-ary* trees with varying *k* and *D.*



Betw+LRU consistently outperforms the rest over different *D* (left) and *k* (right).

Performance Evaluation on Scale-free Graphs

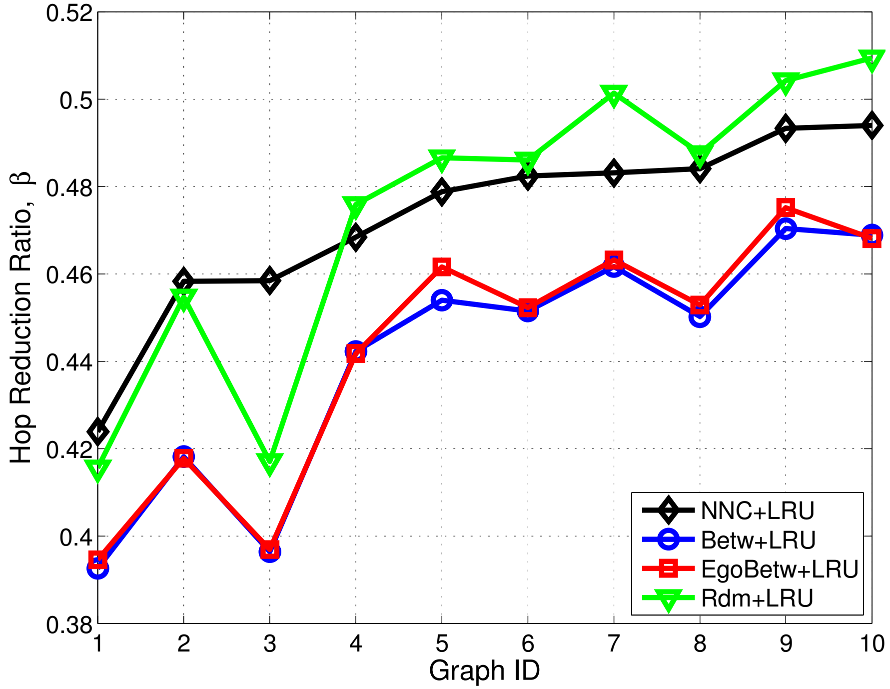
- Scale-free graphs
  - Barabasi-Albert (BA) graphs → preferential attachment
  - *N* = 100 and mean valence = 2

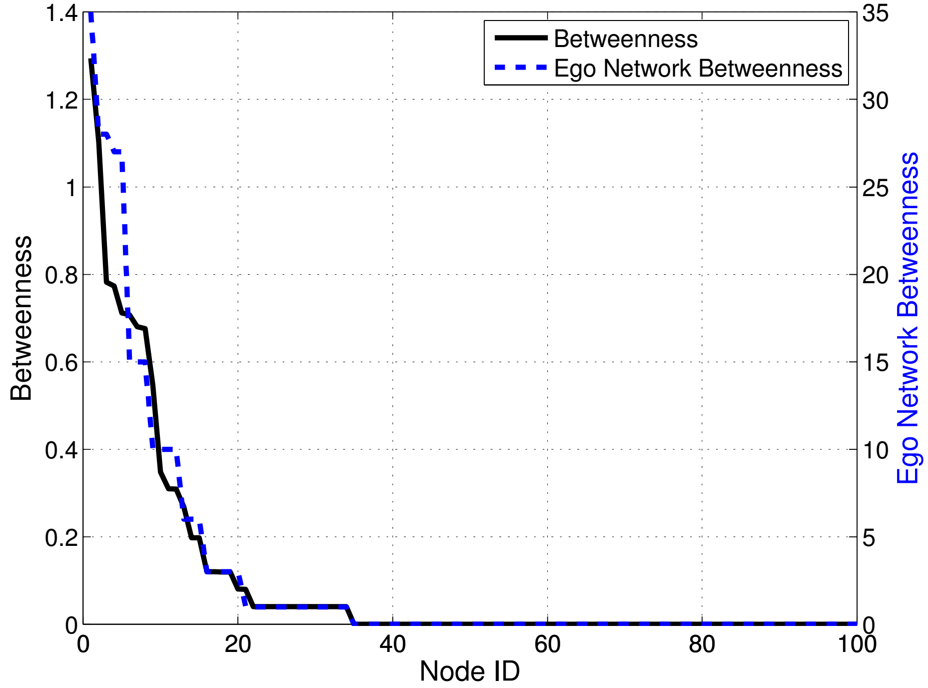


Instantaneous behavior of the caching schemes in a B-A graph; (left) *β*, (right) γ.

Performance Evaluation on Scale-free Graphs

- 10 generations of B-A graphs with N = 100 and mean valence = 2.

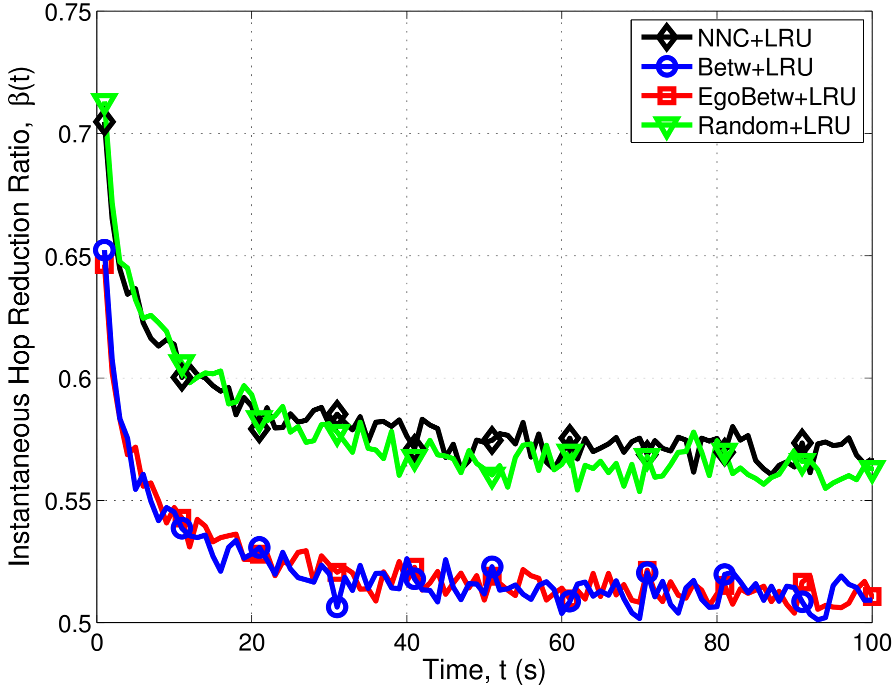


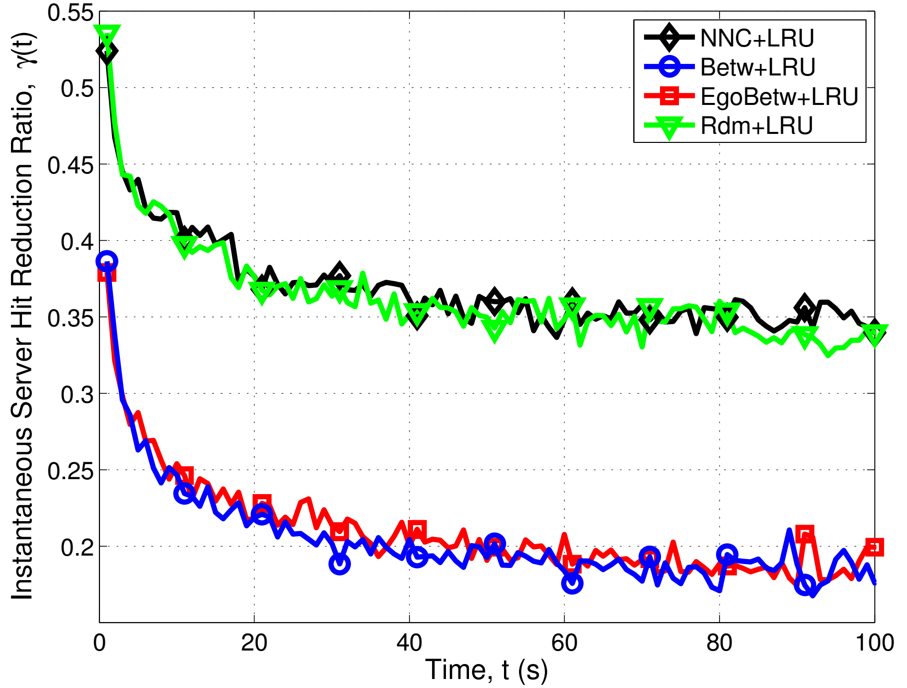


Caching performance with different 100-node B-A graphs (left) and a sample ego network betweenness and betweenness values of the nodes in a B-A graph (right).

Evaluation on Real Internet Topology

- Real-world Internet topology
  - Large domain-level topology extracted from the CAIDA dataset
  - Root at tier-1 ISP (AS7018)
  - 6,804 domains and 10,205 links





Instantaneous behavior of the caching schemes in a large-scale real Internet topology; (left) β, (right) γ.

Summary and Conclusions

- We **argue against the necessity of a ubiquitous caching** strategy in ICN and investigate the possibility of caching less to achieve higher performance gain.
- We demonstrate that even a **simple random caching strategy** (*Rdm+LRU*) can **outperform** (though inconsistently) the current **pervasive caching** paradigm under the conditions
  - that the network topology has low number of distinct delivery paths
  - high average delivery path length.
- We propose a **caching** strategy **based on** the concept of **betweenness centrality** (*Betw+LRU*)
  - content is only cached at the nodes having the highest probability of getting a cache hit along the content delivery path.
  - also proposed an approximation of it (*EgoBetw+LRU*) for scalable and distributed realization in dynamic network environments.

Summary and Conclusions (cont'd)

- We compare the performance of our proposals against the ubiquitous caching of the NNC proposal (*NNC+LRU*).
- Observations:
  - ***Betw+LRU* consistently achieves the best hop and server reduction ratios** across topologies having different structural properties without being restricted by the operating conditions required by *Rdm+LRU*.
  - *EgoBetw+LRU* approximates closely *Betw+LRU* in non-regular topologies (e.g., B-A graphs, real Internet topology)
    - a practical candidate for the deployment of this approach.
  - Besides synthetic topologies (i.e., *k*-ary trees and B-A graphs), the observations are further verified with a large-scale real Internet topology.
- We conclude that indeed **caching *less* can achieve *more*** and our proposed *(Ego)Betw+LRU* is a candidate for realizing this promise.

Related Publications

- Work done in the context of the EU FP7 project COMET which we lead technically - http://www.comet-project.org/

- W. K. Chai, I. Psaras, G. Pavlou, et. al., "CURLING: Content-ubiquitous Resolution and Delivery Infrastructure for Next-generation Services," IEEE Commun. Mag., vol. 49, no. 3, pp. 112-120, 2011.
- I. Psaras, R. Clegg, R. Landa, W. K. Chai and G. Pavlou, "Modelling and Evaluation of CCN-caching Trees," Proc. of IFIP Networking, Valencia, Spain, May 2011.
- I. Psaras, W. K. Chai and G. Pavlou, "To Cache or Not To Cache: Probabilistic In-Network Caching for Information-Centric Networks," to appear in the ACM SIGCOMM ICN Workshop, August 2012.