



ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ
ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS



Object-oriented Packet Caching for ICN

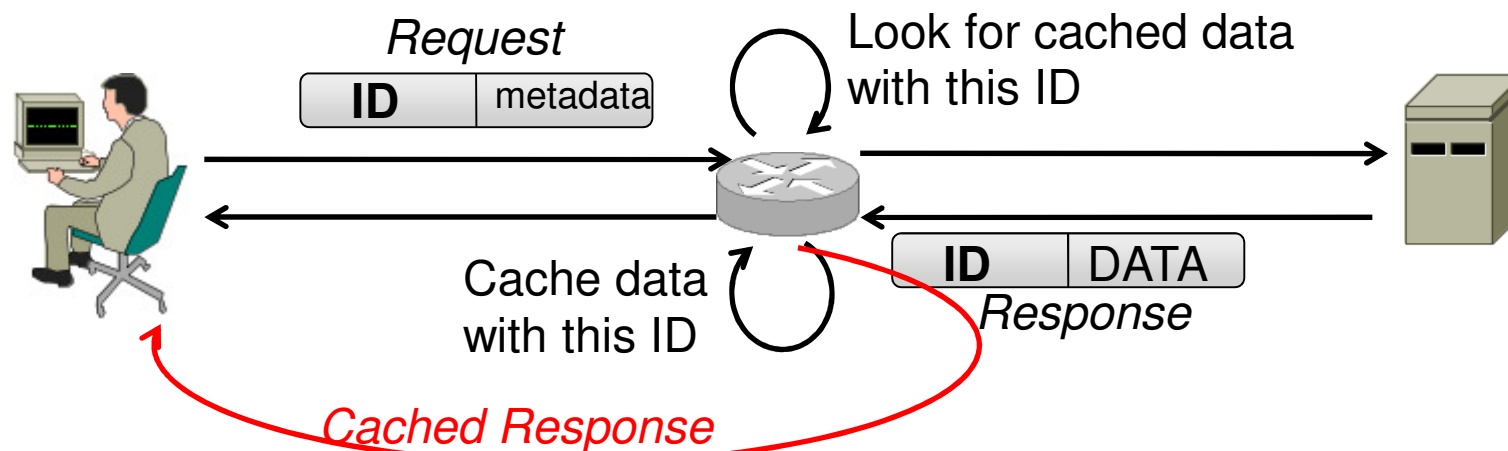
*Yannis Thomas, George Xylomenos, Christos Tsilopoulos,
George C. Polyzos*

Mobile Multimedia Laboratory
Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
47A Evelpidon, 11362 Athens, Greece

2nd ACM Conference on Information-Centric Networking (ICN 2015)
San Francisco, CA, USA

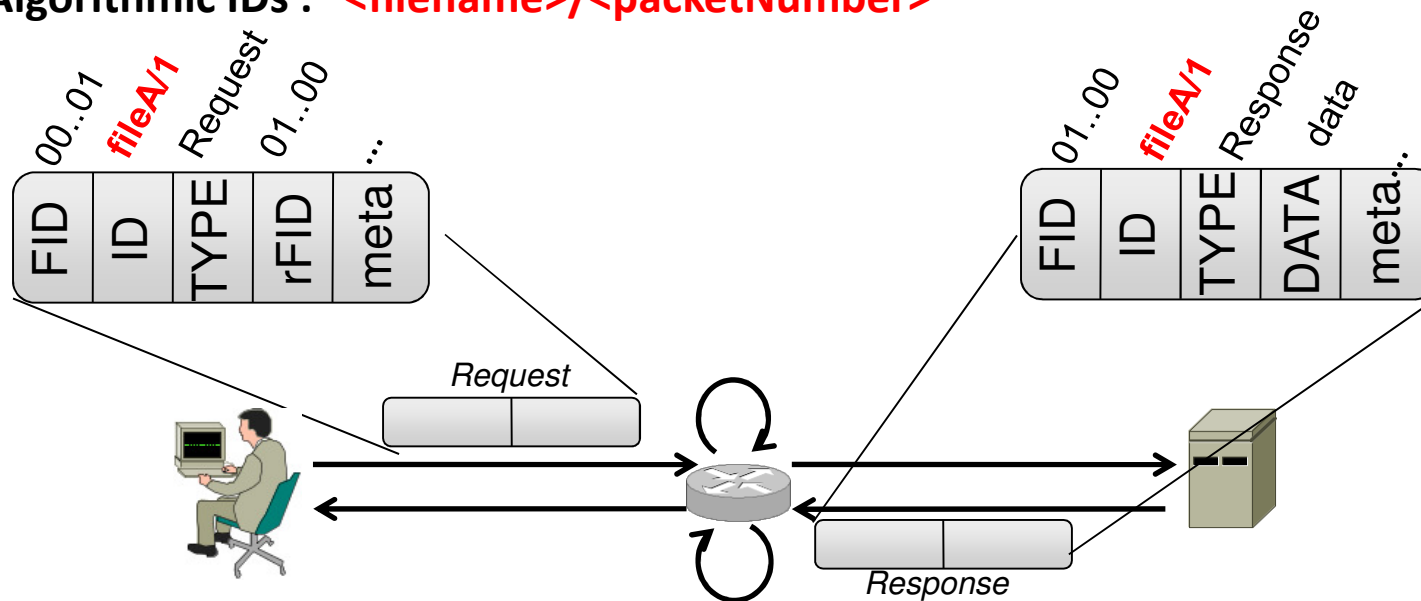
On-path packet-level caching in ICN

- **Self-identified** (data) packets (= network transfer units)
- Receiver-driven transport
 - each (data) packet is **explicitly requested**
- Network storage
 - exploit router memory as **cache**
 - **store** incoming (data) packets (opportunistically)
 - respond immediately to received requests for (cached) packets
 - rather than forwarding the request



On-path caching in *Publish-Subscribe Internetworking* (PSI) with mmTP

- Multipath Multisource Transfer Protocol (mmTP) [1]
- Multiflow receiver-driven transfer protocol for PSI
- Slow path Rendezvous
 - Resolution system locates sources
 - Creates path(s) for each requestor-source pair (multi-source and multi-path)
- Fast-path Rendezvous
 - Requests sent directly to sources - Sequential order of requests
 - **Algorithmic IDs : “<filename>/<packetNumber>”**



[1] Y. Thomas et al., “Multisource and multipath file transfers through publish-subscribe internetworking,” Proc. 3rd ACM SIGCOMM workshop on Information-Centric Networking, 2013.

Caching dimensions

1. Cache management – item replacement policy (micro)
 - for each (& every) cache
 - **When** to insert and evict a packet
 - LRU, FIFO, FILO...
2. Content placement – cache selection policy (macro)
 - **Where** (in the network) to store a packet
 - Everywhere (universal)
 - Betweenness Centrality [7], Probabilistic caching [8], ...

Cache selection policies use cache replacement policies

- e.g., Betweenness Centrality & Probabilistic caching: based on LRU

[7] W.K. Chai, D. He, I. Psaras and G. Pavlou, "Cache 'less for more' in information-centric networks," Proc. IFIP Networking, 2012.

[8] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," Proc. 2nd ACM SIGCOMM workshop on Information-Centric Networking, 2012.

Extra caching dimension

Interplay between **objects** and **packets**

- most caches (proposed, studied...) operate at the **packet level**
- **packet**: main network and cache entity
- **object**: user-level entity
 - target for **popularity** statistics
- chunk...

Sequential access of packets from the start: important

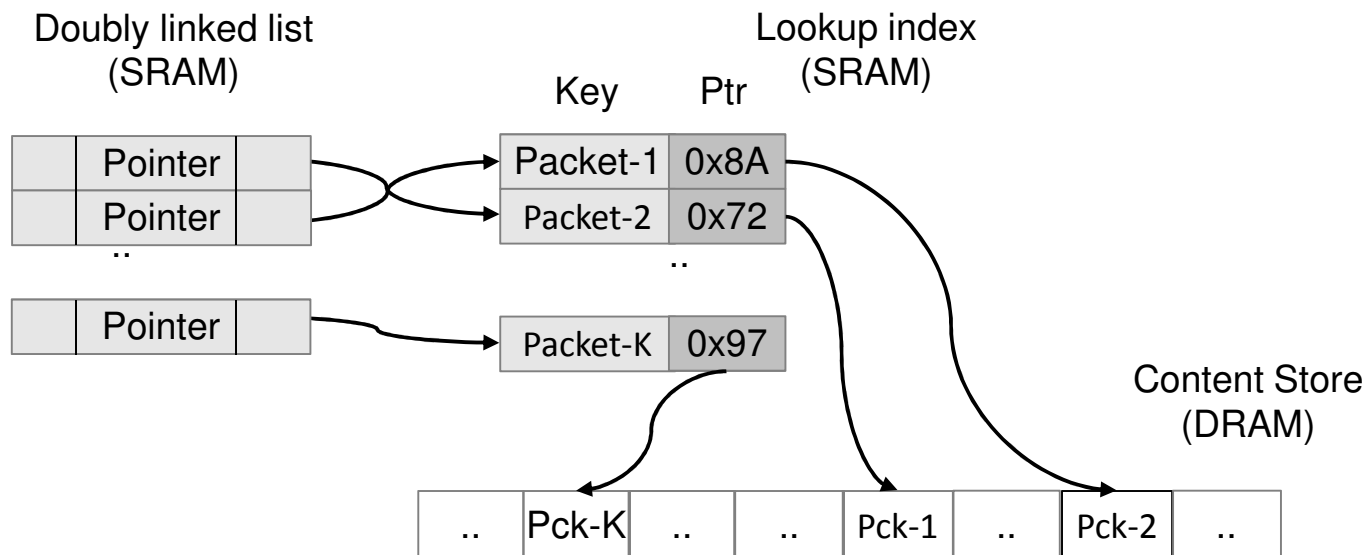
- e.g., for video
 - > 50+% of the network traffic and growing

Main idea: combine

- **object-oriented** cache **lookups**
- with **packet-oriented** cache **replacement**

ICN router packet-cache design

- **Wire-speed** operation (... of large caches)
- Content store
 - **DRAM** - slow but cheap
- Index table to access the store [2]
 - **SRAM** - fast but expensive
- LRU: most commonly used replacement policy



[2] S. Arianfar, P. Nikander and J. Ott, "On content-centric router design and implications," Proc. ACM Workshop on Re-Architecting the Internet, 2010.

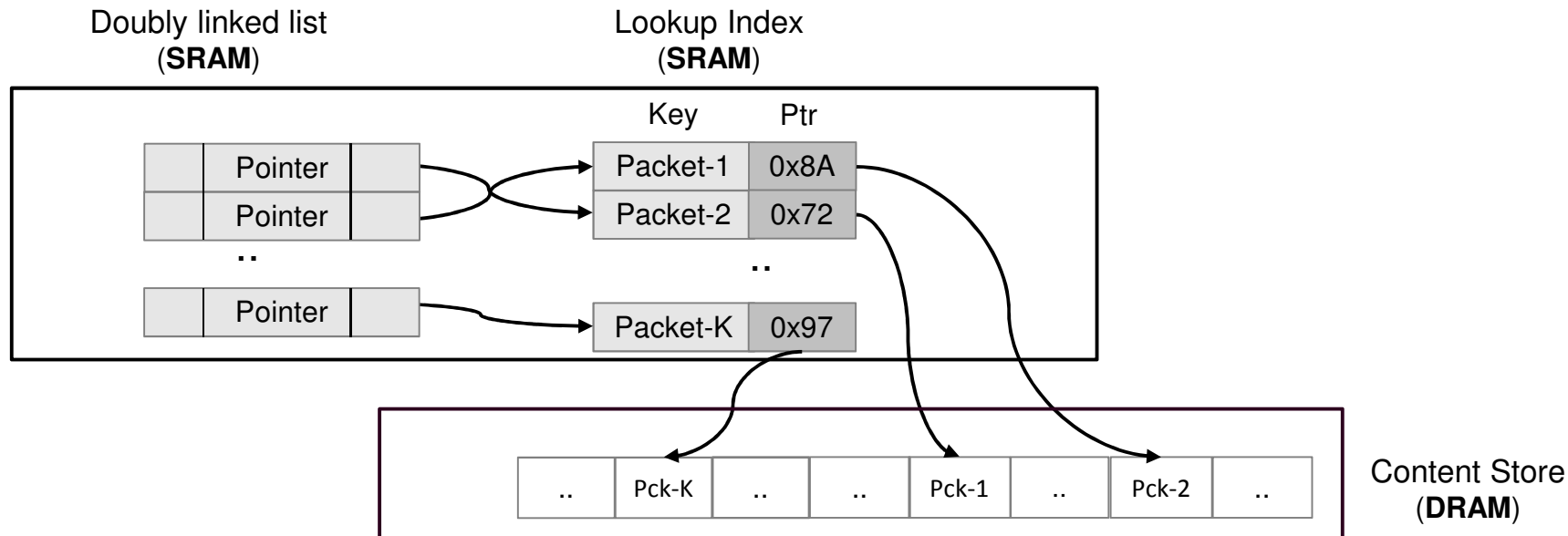
Issues in ICN packet-caches

1. SRAM-DRAM size ratio leads to poor resource utilization
 - 1-to-1 mapping between SRAM-DRAM
 - 1 SRAM entry points to a 1 packet in DRAM
 - SRAM too small to index entire DRAM store
2. Large Object Poisoning
 - Object size outshines object popularity in LRU packet-caches and mitigates caching efficiency
3. Looped Replacement Effect
 - Sequential packet requests of partially stored objects does not work well with replacement policies that ignore (the existence/role of) objects, such as LRU, FIFO and FILO [3]

[3] Z. Li and S. Gwendal, "Time-shifted TV in content centric networks: The case for cooperative in-network caching," Proc. IEEE ICC 2011.

Issue #1: SRAM-DRAM size ratio

- 1-to-1 mapping of **SRAM-indexed, DRAM-stored** packets
 - 210 Mbit SRAM, 40-byte entries: ~688k entries [4]
 - 10 GB DRAM, 1500-byte data packets: ~7.1M packets [4] (!)
 - SRAM can index ~10% of stored packets
- ~90% of DRAM left un-indexed (= unused)



[4] D. Perino and M. Varvello, "A reality check for content centric networking," Proc. 1st ACM SIGCOMM workshop on Information-Centric Networking, 2011.

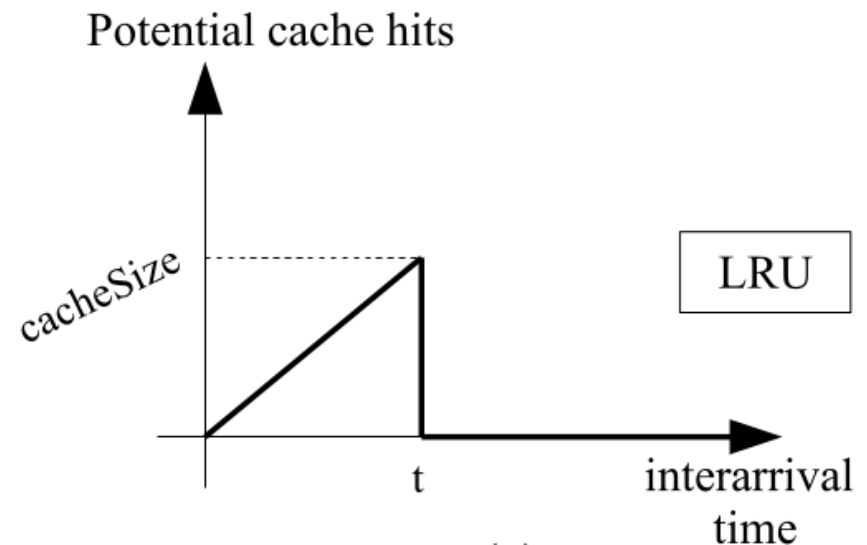
Issue #1 (SRAM/DRAM) : Possible solutions

- Increase (data) packet size
 - Impacts caching granularity → reduces gains [5]
 - Requires changing network's MTU to preserve the self-identification of network units
 - Even with jumbo Ethernet frames (9000 bytes), 20% of 10GB DRAM is still left unused (e.g., what about 40GB DRAM?).
- Split index between SRAM and DRAM
 - Induces *false-positive* accesses to DRAM during packet search
 - Accessing DRAM per packet – too slow! [2]
- **Break 1-to-1 mapping** of SRAM to DRAM entries
 - **Object-oriented Packet Caching (OPC)**

[5] N.T. Spring and D. Wetherall, 'A protocol-independent technique for eliminating redundant network traffic,' ACM SIGCOMM CCR, vol. 30, no. 4, pp. 87-95, 2000.

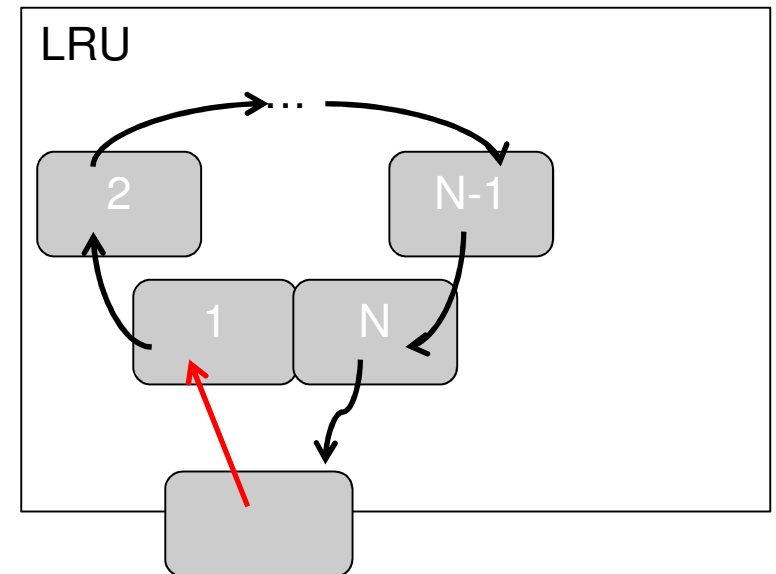
Issue #2: Looped replacement effect

- Sequential, ascending requests (from the start)
 - e.g., video streaming
- LRU
 - **first** packets are evicted before the **last** ones come...
 - first packets are evicted while other packets of the object are present, but are (basically) useless ...
- When 1st packet is evicted, hit prob. (from new streams) becomes **zero**



Issue #3: Large object poisoning

- Object-level (LRU) **popularity** criterion is outshined by **size**
 - New packets always enter at LRU's head
 - Traverse the entire LRU chain before exiting
- **Large & unpopular** objects poison the cache
 - Occupy a great part of the cache
 - Do not provide any gain



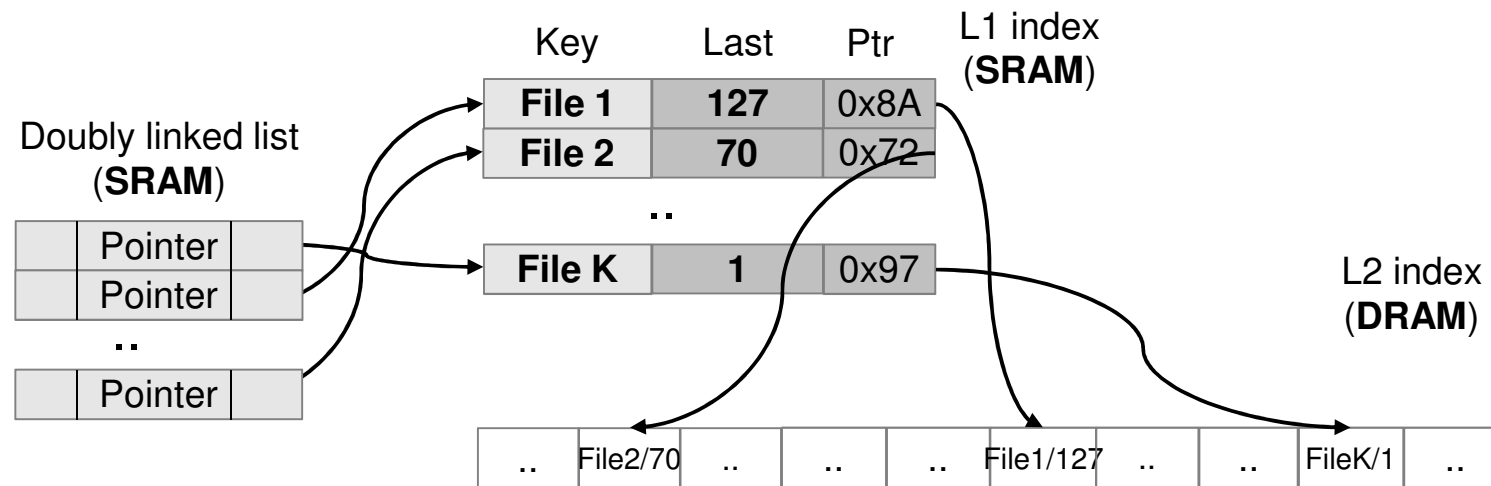
Object-oriented Packet Caching (OPC)

- Two levels of management
 - L1. Object-level content indexing
 - L2. Packet-level content storage
- Assumptions
 - Clients request packets in sequential (ascending) order (e.g., video streaming)
 - Packet names indicate packet position in object
 - “<filename>/<packetNumber>”
- Advantages
 1. Addresses SRAM-DRAM size ratio
 2. Avoids looped replacement effect
 3. Reduces large object poisoning
- Does not require different hardware than (ordinary) LRU

OPC design

Store the initial part of an object from the 1st to the nth packet, with no gaps.

- SRAM holds the index
 - **Key:** object, **Last:** Last packet ID, **Ptr:** @{last packet in DRAM}
 - Object-level LRU -> exploits object popularity
 - **1 entry per object** -> overcomes SRAM bottleneck
- DRAM holds the data packets



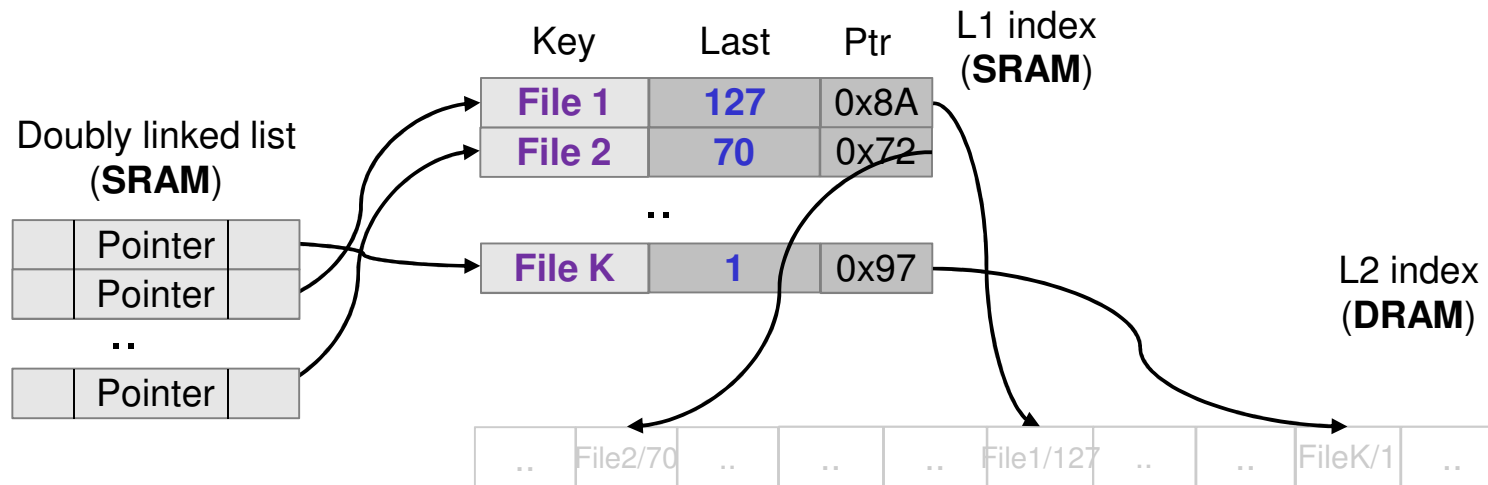
OPC: Lookups

- SRAM lookup, **avoid DRAM reads**
 - Example: request for packet <“file1/23”>

```

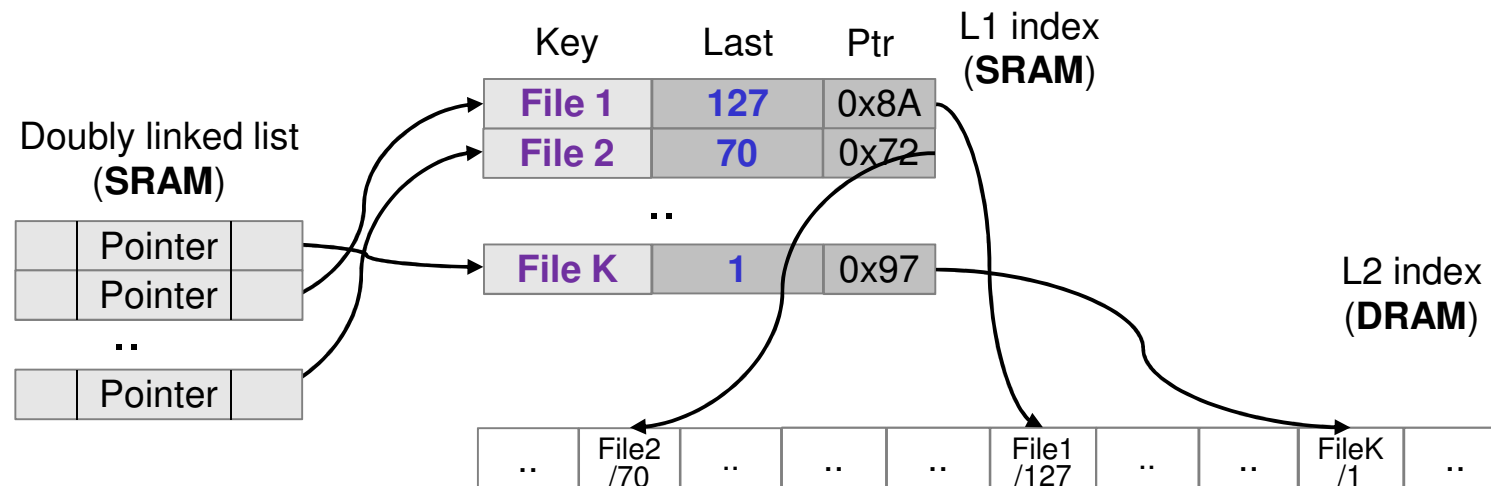
IF (file1 in key && 23 <= Last)
    packet is cached @ <follow Ptr>
ELSE
    packet is not cached
END_IF
    
```

- *Ptr* is the address of the object's last stored packet



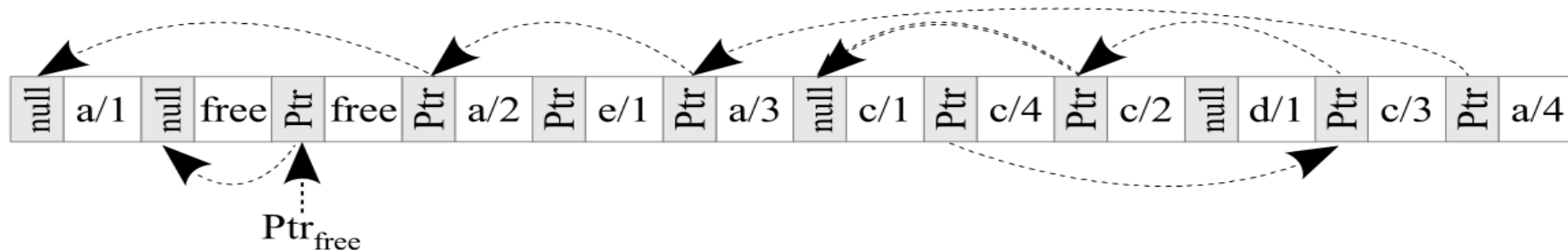
OPC: Replacement policy

- **Insertions**
 - Always start with the 1st packet of a file
 - nth packet only if (n-1)th is already cached
- **Evictions**
 - If SRAM is full: all packets of the LRU object are evicted (remove one entry from the index)
 - If DRAM is full: remove the last packet of the LRU object



OPC: DRAM organization

- **DRAM Entry: Pointer (8 bytes) + (Data) Packet (1500 bytes)**
- 1 single linked-list per object
 - pointers start from tail and point backwards
 - $O(1)$ insertions at the back
- 1 linked-list of available/free slots via Ptr_{free}
- On insertion
 - Packet is stored @ Ptr_{free} and is linked to the appropriate object list
 - Ptr_{free} points to the next free slot
- On eviction
 - Object eviction: all object's packets are linked to the *free-list*
 - Packet eviction: packet is linked to the *free-list* and object's Ptr (SRAM) points to previous packet



DRAM overhead

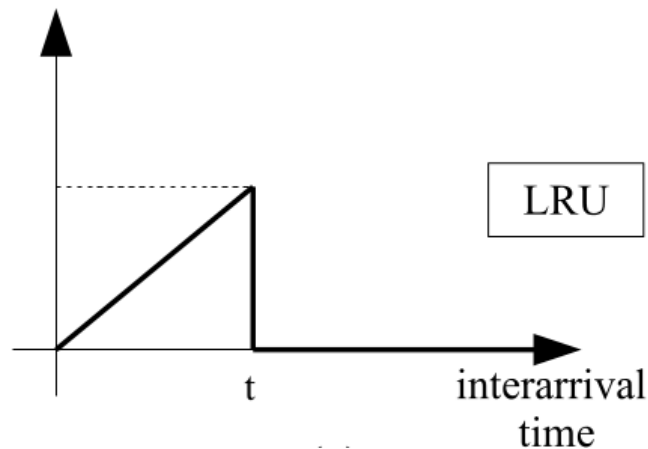
- DRAM reads
 - Packet insertion or eviction: **1 access**
 - Object eviction: **n accesses**, where $n = (\text{\#stored packets})$
 - Packet fetch: **m accesses**, where $m = n - \text{packet_Number}$
- Minimize cost for packet-insertion
- Cost of packet fetch (hit) to be compared to cost of miss
= delay to get packet upstream (>>)

Looped replacement effect in OPC

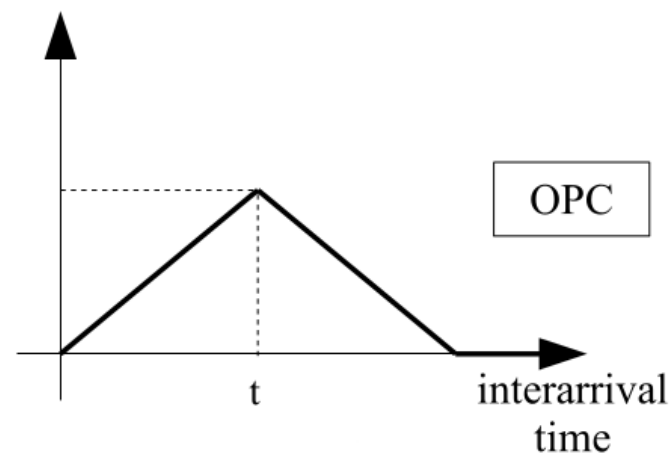
At all times, OPC keeps in the cache the initial part of an object, from the first to the n -th packet, with no gaps

- In OPC potential hits decrease gradually
 1. $T=t$, cache is full and last packet ($t-1$) is evicted, remaining pcks: $[1,t-2]$
 2. LRU would remove the 1st packet, remaining packets: $[2,t-1]$
 3. $T=t+1$, a second object request would get $t-2$ hits
 4. LRU would get ZERO
- **OPC** (theoretically) can provide **200% better hit-ratio** by only addressing this issue

Potential cache hits

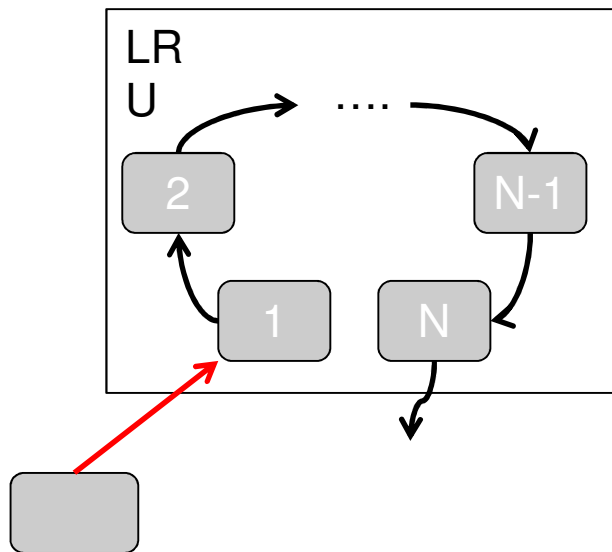


Potential cache hits



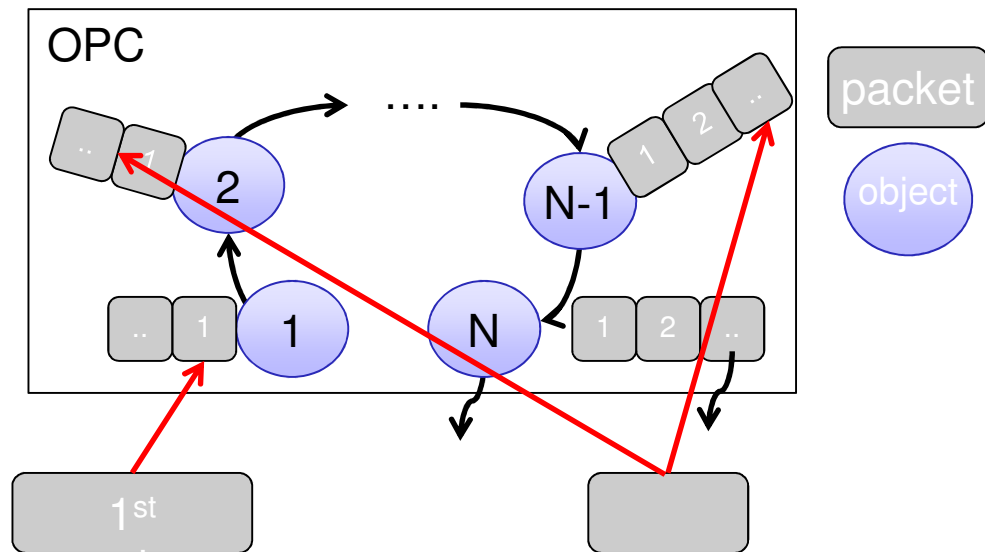
Large object (not) poisoning OPC

- Incoming packets placed at “LRU” position of the **object**
 - In ordinary LRU each incoming packet is placed at the head
- Cached objects get at LRU’s head only in case of cache hit
- Packets of unpopular objects are placed closer to the exit



Entering packets:

- are placed at LRU’s head



Entering packets:

- objects’ 1st packets are placed at LRU’s head
- other are placed at object’s position in LRU

Evaluation setup

CCN/NDN implementation in NS-3

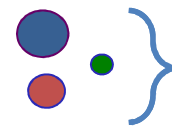
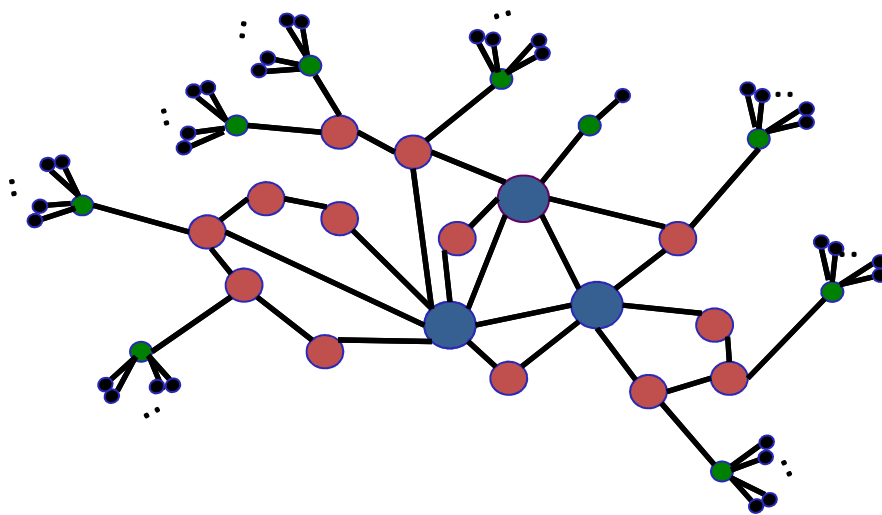
- “Realistic” workload by GlobeTraff (traffic generator)[6]

		Web	P2P	Video	Other
objects		195386	1	176	10485
Size (KBs)	Mean	10.07	100696.00	11053.50	5.00
	Max	29192.10	1006592.0	24867.90	5.00
	Std. dev	82.91	0.00	7706.54	0.00
requests	Mean	3.37	2.00	1.85	2.13
	Max	10984.00	2.00	17.00	1106.00
	Std. dev	53.80	0.00	2.33	15.30
Traffic	GBytes	6.13	1.91	3.50	0.10
	Ratio (%)	52.60	16.40	31.00	0.09

[6] Katsaros et al., “GlobeTraff: a traffic workload generator for the performance evaluation of future Internet architectures,” Proc. IEEE NTMS, 2012.

Topology

- 10 synthetic 'scale-free' topologies 50-100 nodes
- 25 receivers placed at access nodes share a workload served by 1 sender (at a random access node)



Network nodes

- Cache-enabled
- 50-100 in each topology



User nodes

- 25 subscribers per access node
- 1 publisher at a random access node

Evaluation setup (details)

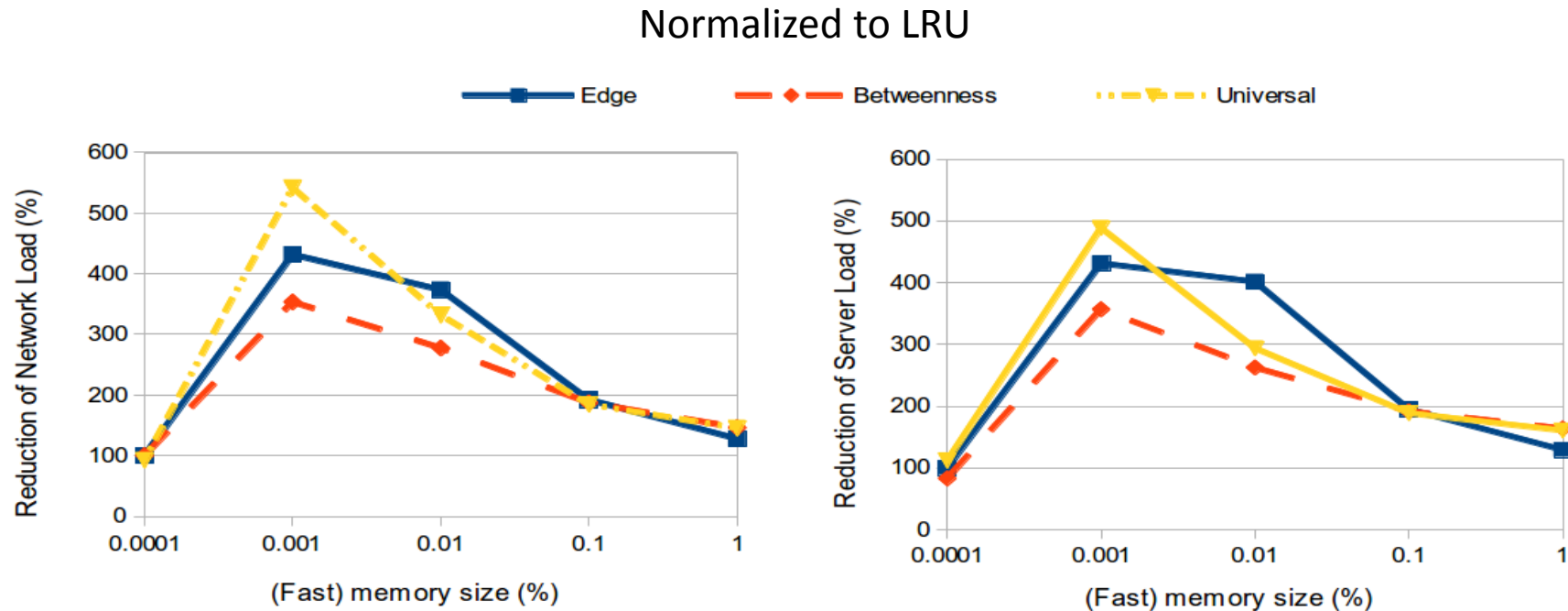
- Network
 - Link delay: 5ms
 - Link capacity: 50Mb/s
 - CCN/NDN implementation
 - PIT, FIB, content store impl.
 - **No losses** or damaged packets
 - Links are **not congested** or stressed
- Cache
 - SRAM latency: 0,45ns
 - DRAM latency: 55ns
- Workload
 - Web traffic
 - Zipf with $a=0.9$
 - Video
 - Weibull, $k=0.513$, $\lambda=6010$
- Application
 - Congest. Control: **stop-and-wait**
 - Receivers start simultaneously
 - request packets in ascending order
 - request next object when last chunk of previous is received

Evaluation parameters

- fast memory (SRAM) size:
 - 0.0001% to 1% of the distinct items in the workload
- Cache placement/selection policy:
 - **Universal caching**: all routers
 - **Edge caching**: only routers at access nodes
 - **Betweenness Centrality** [7]: the most central router on the path
- Metrics:
 - Server load, network load, cache hit-ratio, DRAM accesses, transfer completion time
- The results are illustrated **normalized to LRU**
 - to properly highlight OPC offered gains

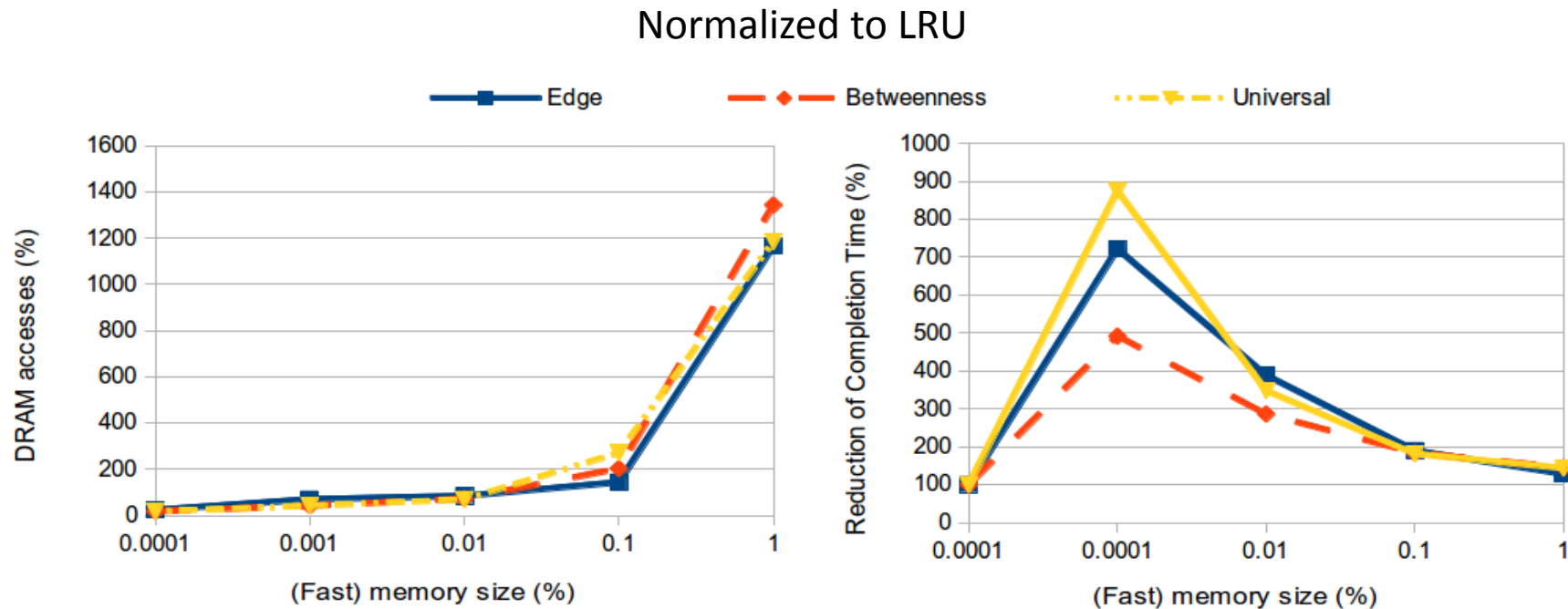
[7] K. Chai, D. He, I. Psaras and G. Pavlou, "Cache 'less for more' in information-centric networks," Proc. IFIP Networking, 2012..

Performance assessment



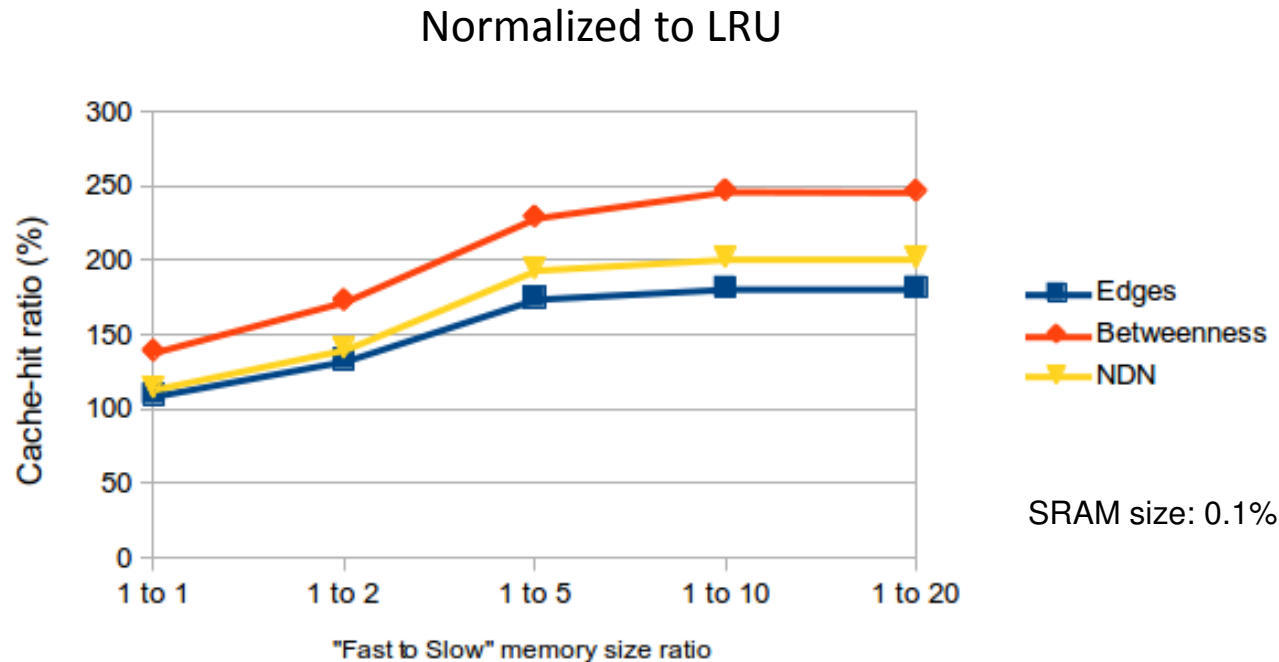
- Substantial improvement (200%-500%) in many cases
 - Roughly **no** gains when memory size is 0.0001%
 - Large (1%) SRAM size: 120%-160% reduction
 - Gains are \sim inversely proportional to the ratio $SRAM_size/Traffic_load$

Performance assessment (cont.)



- DRAM accesses
 - OPC accesses DRAM less than LRU for cache sizes $< 0.1\%$
- Completion time
 - **DRAM overhead is not noticeable**
 - Completion time is mostly dependent on network load

SRAM-to-DRAM ratio



- 1-to-5: Almost reached maximum gains
 - Most popular items in the workload consist of less than 10 packets (Web traffic)
- **1-to-1: OPC** offers 14%-40% better cache-hit ratio
 - By avoiding large object poisoning
 - By addressing looped replacement effect

Conclusions

- ***Object-oriented Packet Caching*** (OPC) design for ICN
 - **object-oriented** cache **lookups** (LRU decisions)
 - **packet-oriented** cache **replacement** (item operations)
- OPC
 - substantially increases resource utilization of caches (with typical parameters)
 - addresses issues associated with packet caches
 - large object poisoning
 - looped replacement effect
 - does not require additional/different hardware
 - performance compared to (simple) LRU
 - up to **4x** further reduction of network and server load
 - up to about **3x** higher hit-ratio



ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ
ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Thank you!

Object-oriented Packet Caching for ICN

*Yannis Thomas, George Xylomenos, Christos Tsilopoulos,
George C. Polyzos*

Mobile Multimedia Laboratory
Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
47A Evelpidon, 11362 Athens, Greece

References

1. Y. Thomas et al., "Multisource and multipath file transfers through publish-subscribe internetworking." Proc. 3rd ACM SIGCOMM workshop on Information-Centric Networking, 2013.
2. S. Arianfar, P. Nikander and J. Ott, "On content-centric router design and implications," Proc. ACM ReArch 2010.
3. Z. Li and S. Gwendal, "Time-shifted TV in content centric networks: The case for cooperative in-network caching," Proc. IEEE ICC 2011.
4. D. Perino and M. Varvello. "A reality check for content centric networking." Proc. ACM SIGCOMM workshop on Information-Centric Networking, 2011.
5. N.T. Spring and D. Wetherall, 'A protocol-independent technique for eliminating redundant network traffic, ACM SIGCOMM CCR, vol. 30, no. 4, pp. 87-95, 2000.
6. K. Katsaros et al., "GlobeTraff: a traffic workload generator for the performance evaluation of future Internet architectures," Proc. IEEE NTMS 2012.
7. W.K. Chai, D. He, I. Psaras and G. Pavlou, "Cache "Less for more" in information-centric networks," Proc. IFIP NETWORKING 2012.
8. I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," Proc. ACM SIGCOMM workshop on Information-Centric Networking, 2012.