Analysis of Latency-Aware Caching Strategies in Information-Centric Networking

Giovanna Carofiglio, Leonce Mekinda, Luca Muscariello

Analysis of Latency-Aware Caching Strategies in Information-Centric Networking

Giovanna Carofiglio
Cisco Systems
gcarofig@cisco.com

Leonce Mekinda, Luca Muscariello
Orange Labs Networks
firstname.lastname@orange.com

ABSTRACT

5G has loudly ambitioned to achieve extremely low latency in mobile networks. To this aim, we have recently introduced two novel latency-aware caching heuristics, LAC and LAC+ and we showed through simulations in Information-Centric Networks their good performance figures. In this paper, we present an insight on their operations: a mathematical analysis of these caching systems led us to novel results that we validate in simulation. The advantages of these algorithms come (i) on one side from the fact they are distributed and lightweight and (ii) from the ability to quickly adapt to content popularity and network congestion, with no signaling nor explicit coordination between the network nodes. In this paper we provide analytical bounds of latency aware caching policies and evaluate their performance by network simulations. The proposed mechanisms can halve the mean and standard deviation of content delivery time with respect to approximations of LFU as leave a copy probabilistically.

CCS Concepts

•**Networks** → **Network performance analysis;**

Keywords

Information-Centric Networks; stochastic modeling; caching

1. INTRODUCTION

Latency reduction objectives, currently emphasized as 5G requirements, imply to solve a number of technical challenges requiring novel solutions in the whole communication network: the physical layer, the MAC as well as the network backhaul and core. In this paper we focus on one specific aspect of the communication path, i.e. caching systems.

Information-Centric Networking (ICN) is a network architecture that embeds caching functions natively. We focus on the Named-data networking architecture and we summarize its characteristics here. A detailed description of the system can be found in [17].

Users retrieve named Data using a pull flow control protocol based on subsequent packet queries, triggering Data packets delivery. Name-based routing and forwarding guarantee that queries are properly routed towards a repository, where a permanent copy of the content is stored, following one or multiple paths. Network nodes maintain three major data structures: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). The CS caches Data packets received, which can be potentially useful to satisfy future Interest packets. The PIT stores Interests that have been forwarded and waiting for matching Data packets to return. The FIB is similar to IP routing table and is maintained by a name-based routing protocol. A strategy module defines the policy for output interface(s) selection at each FIB entry. For each arriving Data packet, a router finds the entry in the PIT that matches the data name and forwards the data to all downstream interfaces listed in the PIT entry. It then removes that PIT entry, and caches the Data in the CS. Indeed, Data may come from the repository, or from any intermediate cache along the path with a temporary copy of the Data packet. Packets of the same content can therefore be retrieved in a multi-path fashion. This means that data packet follow the reverse path build by the queries. This allows fine-grained monitoring of the response delay at any intermediate node in the communication path.

Intrinsic to ICN design are content authentication, multipath forwarding, affordable multicast and ubiquitous caching. Caches must store some valuable objects and evict some others by optimizing some objective in a scalable and fast way. Several content eviction policies have been designed in many technical areas as First-In-First-Out (FIFO), Random, *(evict the)* Least-Frequently-Used (LFU) and several *(evict the)*

CoNEXT CCDWN, December 01-04, 2015, Heidelberg, Germany

Least-Recently-Used (LRU) derivatives like ARC [13] and CAR [2]. policy optimizes a specific objective and most maximize the average hit ratio.

In our work, we consider low network latency objectives that can be met with the help of two novel LRU derivatives, LAC and LAC+. LAC, and its improved version LAC+, are designed on the simple idea that upon content arrival, the larger the retrieval latency, the more favorable the caching decision. Latency includes processing, queuing, transmission and propagation delays.

In this paper, we characterize their miss ratio. Then a quantitative evaluation to corroborate the analysis is provided by means of ICN simulations. Quite impressive results are highlighted in terms of latency reduction under the assumption of a simple, fully distributed approach that self-adapts to varying network conditions. More precisely, we show that our solution outperforms state-of-the-art proposals by achieving significant reduction of average content delivery time and standard deviation up to 50%, along with a very fast convergence to these figures.

2. RELATED WORK

There is a huge literature on caching systems as a means to accelerate the data path of computing systems like ARC [13] and CAR [2]. Some more recent literature has considered web caching to reduce access latency and scale content distribution [11]. In this work we focus on ICN systems like Named-data networking [17] that embeds caching in the data plane of a network layer where congestion and latency are experienced by the transport protocol managing content retrieval. In particular we focus on algorithms that are distributed and lightweight so as to have a feasible implementation at high speed.

Among the fastest approximations of LFU we cite leave a copy probabilistically (LCP) that keeps an object in a node's cache in a data path with probability p. [3] analyzes the p-LRU (LCP + LRU replacement) policy under renewal traffic. The probability p, of keeping an object in the cache after retrieval, is a positive constant smaller than 1. Our contribution extends LCP to dynamic cases, which simply refers to algorithms where p follows a stochastic process.

Much closer to our work, [10] derives through mathematical arguments a dynamically randomized heuristic for LRU caches. The objective was to optimize the storage of variable size documents. A common implementation of LRU, referred to as Move-To-Front algorithm (MTF) consists in moving the most recently used object to the front of a FIFO memory. Though they randomized that MTF rule according to the document size and retrieval cost, they kept it *symmetric* i.e. triggered with the same probability in both hit and miss events. As they ended up providing mathematical justification for a mechanism priorly proposed in [16], we refer to it as Starobinski-Tse-Jelenković-Radovanović's (STJR). On the contrary our approach may be denoted by *asymmetric* as a MTF probability is only considered when a document is freshly inserted into the cache i.e. in case of a miss event. If the document was already in the cache i.e. in case of a hit event, the MTF rule is deterministic as it is applied almost surely.

3. LATENCY-AWARE HEURISTICS

In this work, we analyze two distributed algorithms, LAC[7] and LAC+[6], that aim at minimizing the overall average delivery time in information-centric networks without any coordination among the caches and no signaling. Notice that, fine grained latency measurements are available in ICN as requests sent across an interface pull down data from the same interface. Network wide this enables symmetric routing and latency measurement of the upstream network.

Both work in the following way: *When a client requests at time t a rank-k object, $k \in \mathcal{K}$, that object is either in a cache along the way and consequently returned to the requester, or that cache will download it, then insert it in its local storage with probability $p_k(t)$ or not, with a probability $1 - p_k(t)$ and finally return that object to the requester.* We refer to p_k as decision probabilities.

In LAC, the probability of sending the rank-k object to cache at time t is:

$$p_k(t) \equiv \min\left(\epsilon \frac{T_k(t)^\beta}{\bar{T}(t)^\gamma}, 1\right). \quad (1)$$

LAC+ decision probability $p_k^+(t)$ combines two terms:

$$p_k^+(t) \equiv p_k(t) + (1 - p_k(t))\Theta_k(t) \quad (2)$$

where $T_k(t)$ refers to the monitored latency for content k up to time t and $\bar{T}_k(t)$, $\bar{T}(t)$ to respectively the temporal average for content k and for all cached contents computed up to time t. Averages are estimated using Exponential Moving Average (EWMA) filters. We satisfactory configured the weight of filters past values to 0.9.

ϵ is a small positive real number. β and γ are intensity parameters used in LAC to cleave probabilities between low and high latency retrievals. Higher latency objects will be picked early. Low latencies will get very low decision probabilities but should be eventually picked if the object is popular. For LAC+, since it has a separated latency outlier tracking function $\Theta_k(\cdot)$, we usually set β and γ to 1.

Let μ_t and σ_t be the average and standard deviation of all $\bar{T}_i(t), \forall i \in \mathcal{K}$, at a given node. The n^{th} quantile follows as:

$$Q_n(t) = \mu_t + n\sigma_t. \quad (3)$$

This allows to unfold p_k^+'s second term. $\Theta_k(t)$ is the probability at time t that the rank-k object is a latency

outlier:

$$\Theta_k(t) \equiv \max\left(\frac{\bar{T}_k(t) - Q_n(t)}{\bigvee_{i\in\mathcal{K}} \bar{T}_i(t) - Q_n(t)}, 0\right). \quad (4)$$

We satisfactorily used the first quantile ($n = 1$) throughout the rest of the paper.

To wrap up, LAC+ draws into the cache either highly popular objects sampled using $p_k^+(t)$'s first term or outliers thanks to $p_k^+(t)$'s second term.

4. ANALYSIS

The dynamics of the networked system are complex to capture in a simple model due to the tight coupling between delivery performance and caching functions: the former is certainly affected by network conditions, while clearly network load is a result of caching performance and vice-versa. This is why we focused on the single cache case in developing analytically, some performance bounds expressed in terms of cache miss ratio. In a nutshell, we contribute in showing that the asymmetric design embodied by LAC and LAC+ outperforms known alternatives, typically STJR. LAC and LAC+ are of *asym*-LRU kind as opposed to alternative systems where insertion/replacement operations are symmetrically driven by the same probability (*sym*-LRU). LCP is a special case of asymmetric mechanism where the insertion into the cache is determined by a constant probability p. Refer to Table 1 for the notation used throughout the paper. Variables might be later tagged with the current algorithm in superscript.

4.1 Assumptions

We consider the smallest set of assumptions to have a simple and feasible analytic representation.

Zipf-like popularity: We assume that object popularity follows a generalized Zipf law. Thus $\forall k \in \mathcal{K}, \quad q_k = \frac{c}{k^\alpha}$ with $\frac{1}{c} = \sum_{i\in\mathcal{K}} i^{-\alpha}$ and skewness $\alpha > 0$. This assumption is widely accepted in the literature [4] [14].

Poisson requests: We assume that clients request objects according to a Poisson process of intensity $\lambda > 0$, similarly to [5] [1].

Independent Reference Model: Temporal correlation between object requests, though neglected here like in [16] and [9], is foreseen in future extensions of this work.

LRU replacement policy: we focus on the widely adopted LRU replacement policy whose common implementation consists in moving the most recently served object to the front of a list. This allows to study Move-To-Front algorithm as an LRU scheme [10].

Same object size: For the sake of simplicity, we assume that, like in [8], all retrieved objects have the same size. The model will later be improved to encompass more fine-grained features such as variable object size.

$\text{VRTT}_{k,t}, \forall t \in \mathbb{T}$ are independent and strictly positive.

The characteristic time ("Che") approximation [8] as extended by [9] is a key tool in this work. It states that for LRU caches, the object eviction time is well approximated by a unique constant τ_x.

$t \in \mathbb{T}$	Instant a retrieval occurs. $\mathbb{T} \subseteq \mathbb{R}_+$ denotes the totally ordered set of these times.
x	Local cache size in number of objects.
τ_x	Characteristic time threshold for filling a cache of size x.
λ_k	Request rate of the rank-k object, $k \in \mathcal{K}$.
q_k	Popularity of the rank-k object. $q_k = \lambda_k \lambda^{-1}$.
$\varphi_{k,\tau}$	Probability of receiving at least one request for the rank-k object during τ seconds.
M_k	Asymptotic miss ratio for the rank-k object.
$\{\text{VRTT}_{k,t}\}_{t\in\mathbb{T}}$	Stochastic process modeling the retrieval latency of the rank-k object, as observed from the local cache.
$\{p_{k,t}\}_{t\in\mathbb{T}}$	Caching decision process of the rank-k object.
$\{\pi_{k,t}\}_{t\in\mathbb{T}}$	Miss probability process for the rank-k object.
$\{\mathcal{M}_{k,t}\}_{t\in\mathbb{T}}$	Miss counting process for the rank-k object. It is expected to increase every $\frac{1}{m_k}$ cycle with $m_k = \mathbb{E}[\pi_{k,t}] \quad \forall t \in \mathbb{T}$.

Table 1: Notation.

4.2 Miss ratio

Let $\pi_{k,t}$ be the rank-k object miss probability at time t and $\varphi_{k,\tau}$ be the probability of receiving at least one request for a rank-k object during τ seconds.

PROPOSITION 1. *If we restrict to a countable set of caching decision probabilities, the miss ratio, M_k^{asym}, of asymmetric algorithms such as LAC and LAC+, for the rank-k object approximates to:*

$$M_k^{asym} = \sum_u \mathbb{P}[p_{k,t} = u] \frac{1 - \varphi_{k,\tau_x}}{1 - \varphi_{k,\tau_x}(1-u)}, \ \forall t \in \mathbb{T}. \quad (5)$$

Holding from the Che approximation,

$$\tau_x \text{ is the root of } \sum_{k\in\mathcal{K}} (1 - M_k^{asym}) = x. \quad (6)$$

Note that $\varphi_{k,\tau_x} \triangleq 1 - e^{-\lambda_k \tau_x}$ under Poisson object arrivals. However, accounting for all values of $p_{k,t}$ in Eq.(6) might not be computationally tractable. The following proposition shows that under i.i.d. conditions, values of $p_{k,t}$ can be effectively replaced by a unique expected value.

PROPOSITION 2. *Assuming that $VRTT_{k,t}, \forall t$ are i.i.d. and assuming Poisson object arrivals, the cache miss ratio M_k^{asym} is well approximated using the expected value*

$\mathbb{E}[p]$ *of a unique decision probability* p *when* $\mathbb{E}[p]$ *is very small or when the object popularity is either very small or large as the cache:*

$$M_k^{asym} \approx \frac{1-\varphi_{k,\tau_x}}{1-\varphi_{k,\tau_x}(1-\mathbb{E}[p])}. \quad (7)$$

Note that this approximation still holds for non-Poisson processes when $\mathbb{E}[p]$ is very small. This result is important as it establishes achievable conditions for asymptotic equivalence between the use of a variable decision probability p and the use of its expected value $\overline{p} = \mathbb{E}[p]$. However, the operational drawback of a constant and small $\overline{p}$ is that it postpones considerably the time popular objects are first stored in the cache. LCP suffers from this phenomenon because the expected time to enter the cache is $\frac{1}{\lambda_k \overline{p}}$. Consequently, LCP overall object delivery time converges slowly. LAC+ brings a solution in adequately varying p in order to cache valuable objects earlier.

4.3 Lower bound

Providing a closed-form approximation for *asym*-LRU miss ratio and its characteristic time τ_x^{asym} is hard. Instead, we demonstrate its superiority over the analytically tractable *sym*-LRU mechanism. With some loss of generality, α is assumed greater than one. Let us consider the symmetric mechanism *sym*-LRU where the MTF rules are conditioned by the same probability in both hit and miss cases. By contrast in *asym*-LRU the MTF decision is taken in case of miss only.

PROPOSITION 3. *Assuming* $VRTT_{k,t}, \forall k,t$ *are i.i.d. and large catalog and cache, the steady-state miss probability of symmetric LRU algorithms, for the rank-k object, approximates to:*

$$M_k^{sym} = \exp\left\{-\frac{x^\alpha}{k^\alpha \Gamma(1-\frac{1}{\alpha})^\alpha}\right\}, \quad (8)$$

where $\Gamma(\cdot)$ *is the Gamma function.*

The closed-form expression of Proposition 3 is intrinsically the same as LRU's in [5]. This observation yields the next corollary.

COROLLARY 3.1. *Assuming* $VRTT_{k,t}, \forall k\ \forall t$ *are i.i.d.*

$$M_k^{sym} = M_k^{LRU}$$

i.e. sym-LRU behaves in stationary regime like LRU.

asym-LRU consequently outperforms *sym*-LRU thanks to its convergence to the Least Frequently Used replacement policy[12]. This leads to Proposition 4. Proposition 4 lies on ϵ-permanent accommodation, a notion to be introduced first.

DEFINITION 1. *An object is* ϵ*-permanently accommodated iff its miss ratio is less than a small value* ϵ.

In that context, let $\eta_{mechanism}$ be the number of most popular objects ϵ- permanently accommodated thanks to a caching mechanism.

PROPOSITION 4. *As decision probability's expected value goes small, asym-LRU allows to accommodate* ϵ*- permanently more of the most popular objects than sym-LRU i.e.*

$$\eta_{asym} \geq \eta_{sym}.$$

Let *LAasym* denote LRU equipped for asymmetric latency-aware stochastic caching decision (LAC and LAC+) and let *LAsym* denote LRU modified for symmetric latency-aware stochastic MTF decision (STJR).

COROLLARY 4.1. *As decision probability's expected value goes small,* $\exists \kappa \geq 1 : \eta_{LAasym} \geq \kappa \eta_{LAsym}$.

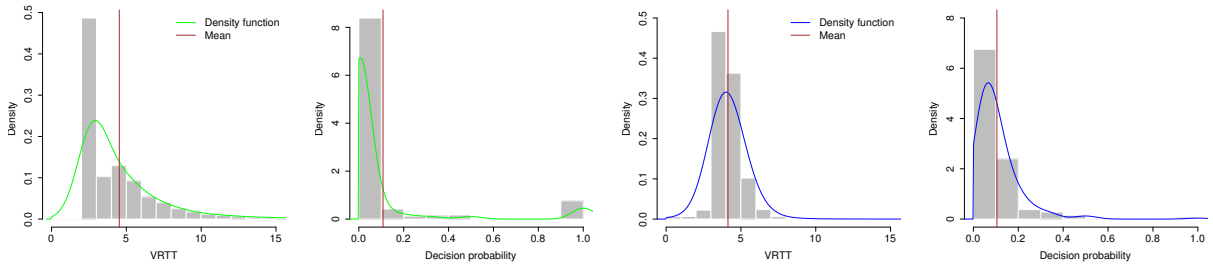
This typically means that the performance of LRU caches equipped with LAC or LAC+ can exceed beyond a given factor κ that of *sym*-LRU, then LRU studied analytically and extensively in previous works [5]. Numerous simulations backed these mathematical results, where often $\kappa > 2$ unleashes tremendous content delivery time decreases.

5. PERFORMANCE EVALUATION

We evaluate LAC and LAC+ against three state-of-the-art caching management mechanisms: LRU + Leave-Copy-Everywhere (LRU), LRU + Leave-Copy-Probabilistically (LCP) and LRU + Leave-Copy-Down (LCD)[11]. This is carried out by means of the packet-level NDN simulator CCNPL-Sim (the code of the simulator as well as the input files to run the scenarios presented in this paper can be found at http://systemx.enst.fr/ccnpl-sim) (i) on a single cache topology, (ii) then on a complex network where core caches are located along a ring. While in (i) the workload is IRM and Zipf skewness $\alpha > 1$, in (ii) we injected some time locality and set $\alpha < 1$ to investigate situations closer to the real world.

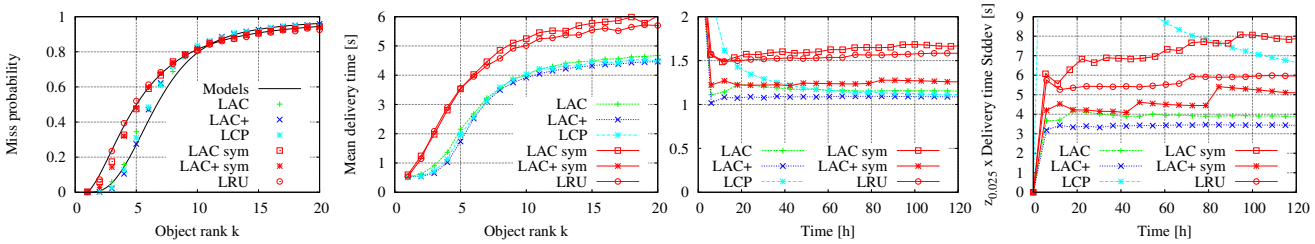
5.1 Single cache topology

The following results are achieved in a simulated ICN with a single caching node between the object consumers and the publishing server. The whole simulation setup is available online. Here are the main configuration parameters. Cache sizes are equal to 80kB. The Poisson process for generating content requests is characterized by a rate of 1 object/s. Objects are requested over a catalog of 20,000 items, according to a Zipf-like popularity distribution of parameter $\alpha = 1.7$. This value of α is still realistic [14]. The two FIFO links from the consumers up to the content publisher have a capacity of 200Kbps and of 30Kbps, respectively. The size of every object conveyed through these links is 10kB, that we also take as fixed packet size. About LAC parameters, $\epsilon = 1$ while $\beta = \gamma = 4.5$ to pick latency



(a) VRTT (LAC) (b) Decision prob.(LAC) (c) VRTT (LAC+) (d) Decision prob.(LAC+)

Figure 1: Single cache: latency and decision probability distributions.



(a) Miss ratio validates models (b) Deliv. time vs rank (c) Mean deliv.time vs time (d) Deliv.time stddev vs time

Figure 2: Single cache: LAC decreases LRU delivery time by 30% and outperforms LCP on convergence.

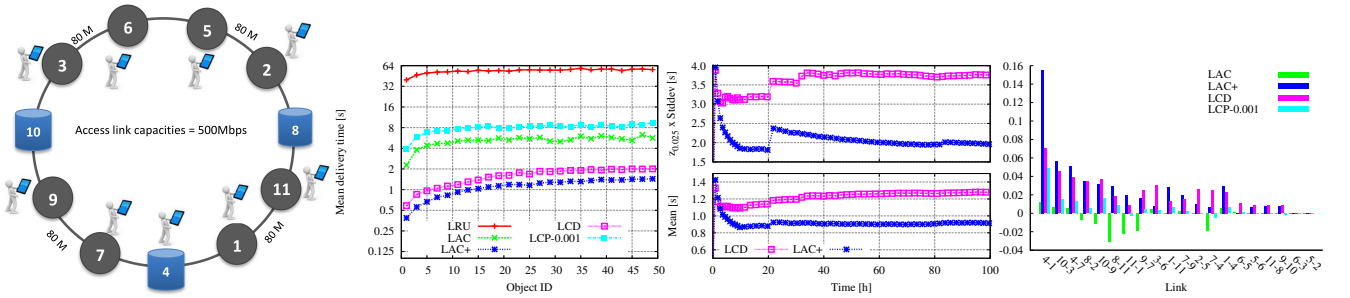
outliers and leave quickly delivered objects to popularity sampling. The function f is the mean latency of all ever-cached objects. LAC+ is configured with $\epsilon = 0.05$ and $\beta = \gamma = 1$, relying on its adjunct outlier tracking function $\Theta_k(\cdot)$. We report the simulation results in Fig.2.

First, it appears clearly in Fig.1 that the decision probability values are predominantly small. The mean decision probability equals 0.1 for both LAC and LAC+. This is what drives their joint popularity sampling / latency screening capabilities. Secondly, we can observe from the plots in Fig.2 that LAC and LAC+ converge to the same steady state as LCP, which approximates the optimal LFU behavior. LCP, LAC and LAC+ miss probabilities coincide even though the former are based on temporal measurements of residual latency, so adapting over time based on the sensed variations in terms of experienced latency. Thirdly, we observe how much LAC/LAC+ latency-aware technique reduces both delivery time mean and standard deviation. It is striking to see how quickly they converge, compared to classical LCP. Observe that LAC+ is so efficient that, even in its symmetric implementation, it captured early the highest popular content and made the delivery time drop. Conversely, the constant decision probability used in LCP is the average of all latency-aware decision probabilities ($p = 0.1$) and this impacts negatively the convergence and the system reactivity to temporal variations of latency, as opposed to our LAC and LAC+ proposals. Finally, we observe that *LAsym* and LRU miss ratio curves coincide in steady state as predicted in [10]. A symmetric filtering of objects to put in and to move to the cache front has the only effect of slowing down convergence while not modifying the dynamics of the underlying Markov chain.

5.2 Ring topology

In this section, we evaluate the consistency of our algorithms in a network scenario where eleven interconnected core nodes form a ring. Every link at the network core offers a 80Mbps capacity. Among the core nodes, three are content producers. Each offers 20,000 Zipf-ranked objects from its own catalog. Objects from the producer at Node 4 are given the prefix */Netflix/*, */Orange/* for those from Node 8 and */Youtube/* for those originating from Node 10. Each object is conveyed in chunks of 3kB and has a total size of 2MB. Each node cache can accomodate up to 40 objects. While the skewness of the Zipf-like popularity distribution, α, remains 0.9 for the whole simulation, we inject some time locality in shuffling every object rank every ten hours. Clients connect to their closest core node to send interests and retrieve data over dedicated 500Mbps links. Client requests reaching every core node follow a Poisson process with intensity $\lambda = 2$ objects/s. Clients are equally interested in every catalog, so that any of them addresses every catalog with probability 1/3. Routing is single path. Fig.3(a) depicts the network setup. LAC+ and ϵ-LCP share a common value of $\epsilon = 0.001$. LAC exploits a different $\epsilon = 1$ but keeps intensity parameters β and γ equal to 1.

The striking results in Fig.3(b) show LAC+ decreasing LRU mean delivery time by up to two orders of magnitude. LAC+ clearly outperforms LAC and LCP in en-



(a) Ring topology (b) Mean delivery time vs ID (c) Delivery time vs Time (d) Load reduction $\times$ load_{LRU}

Figure 3: Ring topology with non-stationary workload

suring a content delivery at least twice faster. Moreover, as witnessed by Fig.2(c) and Fig.2(d), LRU equipped with the Leave-Copy-Down algorithm denoted by LCD is surpassed. LAC+ heuristic minimizes the highest link load in a dynamic way (Fig.3(d)).Thanks to its reactivity to congestion, LAC+ reduces by 30% the mean delivery time inducted by LCD and by 50% the related standard deviation.

6. CONCLUSION AND FUTURE WORK

Throughout the paper, we characterized, bounded and evaluated the performance of latency-aware LRU caches. The theoretical contribution extends the state-of-the-art of probabilistic caching analysis. The novel idea behind ubiquitous latency-awareness is simple, fully distributed and demonstrated powerful by means of extensive simulations. By fully distributed, we highlight the fact that latency-awareness blasting performance is free of any form of signaling. Actually, making early caching decisions based on the latency of retrieved objects will sound increasingly intuitive, especially in the forthcoming 5G era. The task of accurately modeling networks of such caches in order to capture their dynamics is still ongoing.

7. REFERENCES

[1] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda. Congestion-aware caching and search in information-centric networks. In *Proceedings of the 1st International Conference on Information-centric Networking*, ICN '14, pages 37–46, New York, NY, USA, 2014. ACM.

[2] S. Bansal and D. S. Modha. CAR: Clock with adaptive replacement. In *Proceedings of the 3rd USENIX Conference on File and Storage Technologies*, FAST '04, pages 187–200, Berkeley, CA, USA, 2004. USENIX Association.

[3] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi. Check before storing: What is the performance price of content integrity verification in LRU caching? *SIGCOMM Comput. Commun. Rev.*, 43(3):59–67, July 2013.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99*, pages 126–134, 1999.

[5] G. Carofiglio, M. Gallo, and L. Muscariello. On the performance of bandwidth and storage sharing in information-centric networks. *Comput. Netw.*, 57(17):3743–3758, Dec. 2013.

[6] G. Carofiglio, L. Mekinda, and L. Muscariello. FOCAL: Forwarding and caching with latency awareness in information-centric networking. In *IEEE GLOBECOM 2015 Workshop on Information Centric Network Solutions for Real-World Applications*, 2015.

[7] G. Carofiglio, L. Mekinda, and L. Muscariello. LAC: Introducing latency-aware caching in information-centric networks. In *40th IEEE International Conference on Local Computer Networks*, 2015.

[8] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: Modeling, design and experimental results. *IEEE J.Sel. A. Commun.*, 20(7):1305–1314, Sept. 2006.

[9] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for LRU cache performance. In *Proceedings of the 24th International Teletraffic Congress*, ITC '12, pages 8:1–8:8. International Teletraffic Congress, 2012.

[10] P. R. Jelenković and A. Radovanović. Optimizing LRU caching for variable document sizes. *Comb. Probab. Comput.*, 13(4-5):627–643, July 2004.

[11] N. Laoutaris, S. Syntila, and I. Stavrakakis. Meta algorithms for hierarchical web caches. In *Performance, Computing, and Communications, 2004 IEEE International Conference on*, pages 445–452, 2004.

[12] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. *CoRR*, abs/1307.6702, 2013.

[13] N. Megiddo and D. S. Modha. ARC: A self-tuning, low overhead replacement cache. In *Proceedings of the 2Nd USENIX Conference on File and Storage Technologies*, FAST '03, pages

115–130, Berkeley, CA, USA, 2003. USENIX Association.

[14] S. Mitra, M. Agrawal, A. Yadav, N. Carlsson, D. Eager, and A. Mahanti. Characterizing web-based video sharing workloads. *ACM Trans. Web*, 5(2):8:1–8:27, May 2011.

[15] W. L. Smith. On the elementary renewal theorem for non-identically distributed variables. *Pacific J. Math*, 14(2):673–699, 1964.

[16] D. Starobinski and D. Tse. Probabilistic methods for web caching. *Perform. Eval.*, 46(2-3):125–137, Oct. 2001.

[17] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named data networking. *SIGCOMM Comput. Commun. Rev.*, 44(3):66–73, July 2014.

APPENDIX

A. PROOF OF PROPOSITION 1

First, we characterize the arrival process to the front the LRU cache. Given the insertion probability $p_{k,t}$, the Move-to-Front probability at time t, during the time window τ, for object equals:

$$\begin{aligned} F_k(t,\tau) &= ((1-\pi_{k,t}) + \pi_{k,t}p_{k,t})\,\varphi_{k,\tau} \\ &= (1-(1-p_{k,t})\pi_{k,t})\,\varphi_{k,\tau}. \end{aligned}$$

That Move-to-Front probability leads to the cache miss probability in the following way. Under the Che approximation, the rank-k object miss probability for a cache under stochastic caching decision satisfies:

$$\begin{aligned} F_k(t,\tau_x) &= 1-\mathbb{P}_{k,t}[\sharp MTF > x] \\ &= (1-(1-p_{k,t})\pi_{k,t})\varphi_{k,\tau_x}. \end{aligned}$$

$\sharp MTF$ denotes the number of distinct objects moved to the cache front. Upon the assumption that every object gets eventually cached at least once over time, $F_k(t,\tau_x) \approx 1-\pi_{k,t}$. Hence, $\pi_{k,t} \approx \dfrac{1-\varphi_{k,\tau_x}}{1-\varphi_{k,\tau_x}(1-p_{k,t})}$.

If we assume that $\{p_{k,t}\}_{t\in\mathbb{T}}$ is ergodic,

$$\begin{aligned} \mathbb{E}[\pi_{k,t}] &= \int_{[0,1]} \frac{1-\varphi_{k,\tau_x}}{1-\varphi_{k,\tau_x}(1-u)} d\mathbb{P}[p_{k,t}\le u] \qquad (9) \\ &= \frac{1}{\varsigma(\mathbb{T})}\int_{\mathbb{T}} \pi_{k,u}(\omega)d\varsigma(u) \text{ for any outcome } \omega, \\ &\quad \varsigma \text{ denotes the Lebesgue measure} \\ &= m_k. \end{aligned}$$

$\pi_{k,t}$ is an instantaneous miss probability that changes every time a new object arrives to the cache. However, only the miss ratio M_k i.e. long-term average of a miss counter $\mathcal{M}_{k,t}$ can be effectively measured. Since the cache is supposed much smaller than the catalog, $\frac{1}{m_k}$ is finite. Thus, W.L. Smith's elementary renewal theorem for non-identically distributed variables holds [15] and the asymptotic miss ratio for the rank-k object equals: $\lim_{t\to\infty}\frac{1}{t}\mathcal{M}_{k,t}(\omega) = M_k = m_k$ for any outcome ω □

B. PROOF OF PROPOSITION 2

Let $p \overset{d}{=} p_{k,t} \quad \forall k, \forall t$. Given τ_x, the miss probabilities are a convex function of the caching decision probabilities as: $\frac{\partial^2}{\partial^2 p}\pi_{k,t} = \frac{2\left(e^{\lambda_k\tau_x}-1\right)^2}{\left(1+\left(e^{\lambda_k\tau_x}-1\right)p\right)^3} \ge 0$. By Jensen's inequality, $\mathbb{E}[\pi_{k,t}] \ge \frac{e^{-\lambda_k\tau_x}}{1-(1-e^{-\lambda_k\tau_x})(1-\mathbb{E}[p])}$.

Let $\mathcal{D} = \mathbb{E}[\pi_{k,t}] - \frac{e^{-\lambda_k\tau_x}}{1-(1-e^{-\lambda_k\tau_x})(1-\mathbb{E}[p])}$ be the gap between the two sides of the inequality. It is easy to verify that $\lim_{\lambda_k\tau_x\to\infty}\mathcal{D} = \lim_{\lambda_k\tau_x\to 0}\mathcal{D} = 0$. Also, $\frac{\partial}{\partial x}\tau_x > 0$ holds from [12] Appendix A, which applies the implicit function theorem over Eq.(6). Furthermore, as p is a strictly positive random variable, Markov's inequality holds and $\lim_{\mathbb{E}[p]\to 0}\mathcal{D} = \lim_{p\to 0}\mathcal{D} = 0$. Using the fact that $\mathcal{D}$ is differentiable on its domain and non-negative, it gets minimal as either $\lambda_k\tau_x \uparrow \infty$ or $\lambda_k\tau_x \downarrow 0$ or $\mathbb{E}[p] \downarrow 0$. All imply that Eq.(7)'s underestimation of rank-k content miss ratio shrinks as either the rank-k content request rate or $\mathbb{E}[p]$ get very small, or cache size and rank-k content request rate get very large □

C. PROOF OF PROPOSITION 3

Let $\overline{p} = \mathbb{E}[p_{k,t}], \forall k \in \mathcal{K}$ at steady state. Let $\sharp k$ denote the number of times a rank-k object is moved to the cache front during a time interval. The mean number of distinct objects moved to the front of the LRU cache during τ, as $|\mathcal{K}| \uparrow \infty$ and $\tau \uparrow \infty$, is:

$\sum_k \mathbb{E}\left[\mathbb{1}_{\{\sharp k>0\}}\right] = \sum_k (1-e^{-\lambda_k\tau\overline{p}}) \sim (\lambda\tau c\overline{p})^{\frac{1}{\alpha}}\Gamma\left(1-\frac{1}{\alpha}\right)$ in virtue of Lemma 5 of [10]. Hence, the power of α-magnified mean number of distinct objects moved to the front of the LRU cache during characteristic time τ_x^{sym}:

$x^\alpha = \lambda\tau_x^{sym}c\overline{p}\Gamma\left(1-\frac{1}{\alpha}\right)^\alpha \Rightarrow \tau_x^{sym} = x^\alpha(\lambda c\overline{p})^{-1}\Gamma\left(1-\frac{1}{\alpha}\right)^{-\alpha}$.

The rest follows by using the exponential inter-arrival distribution for an object with rank k □

D. OPTIONAL PROOF OF PROP. 4

Let the miss ratio of all permanently stored objects admit a sufficiently small value ϵ as upper bound. Then:

$$\eta_{asym} = \frac{(\lambda c\tau_x^{asym})^{\frac{1}{\alpha}}}{(\log(1+\frac{1}{\mathbb{E}[p]}(\frac{1}{\epsilon}-1)))^{\frac{1}{\alpha}}} \qquad (10)$$

and

$$\eta_{sym} = \frac{x}{\Gamma(1-\frac{1}{\alpha})(-\log\epsilon)^{\frac{1}{\alpha}}}. \qquad (11)$$

Since a first-order Taylor series expansion of ϵ for $asym$-LRU, when $\mathbb{E}[p]\to 0$, yields:

$\eta_{asym} \underset{\mathbb{E}[p]\to 0}{\sim} \dfrac{x}{\Gamma(1-\frac{1}{\alpha})\mathbb{E}[p]\log(1+\frac{1}{\mathbb{E}[p]}(\frac{1}{\epsilon}-1))^{\frac{1}{\alpha}}}$, the ratio of both numbers satisfies:

$$\lim_{\mathbb{E}[p]\to 0}\frac{\eta_{asym}}{\eta_{sym}} \ge (-\log\epsilon)^{\frac{1}{\alpha}} > 1 \ \square$$