Coupling Caching and Forwarding: Benefits, Analysis, and Implementation

Giuseppe Rossini and Dario Rossi
Telecom ParisTech, Paris, France
first.last@telecom-paristech.fr

ABSTRACT

A recent debate revolves around the usefulness of pervasive caching, i.e., adding caching capabilities to possibly every router of the future Internet. Recent research argues against it, on the ground that it provides only very limited gain with respect to the current CDN scenario, where caching only happens at the network edge.

In this paper, we instead show that advantages of ubiquitous caching appear only when meta-caching (i.e., whether or not cache the incoming object) and forwarding (i.e., where to direct requests in case of cache miss) decisions are tightly coupled. Summarizing our contributions, we (i) show that gains can be obtained provided that ideal Nearest Replica Routing (iNRR) forwarding and Leave a Copy Down (LCD) meta-caching are jointly in use, (ii) model the iNRR forwarding policy, (iii) provide two alternative implementations that arbitrarily closely approximate iNRR behavior, and (iv) promote cross-comparison by making our code available to the community.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications, Packet-switching networks

General Terms

Algorithms; Performance; Design;

Keywords

Information Centric Networking; Caching; Forwarding

1. INTRODUCTION

With the advent of Information Centric Networking (ICN) [2], the network evolves from a simple interconnections of pipes and buffers, and rather becomes a network of caches. This induces a radical change in network operations: as opposite to IP networks, where routers transfer and discard IP packets in the shortest possible time, ICN routers instead aim at storing content chunks for the longest useful time. In turn, new challenges arise for ICN performance evaluation, on both modeling [6–8, 10, 17, 23, 31] and algorithmic aspects [9, 11–13, 15, 24, 25, 29, 30, 35, 39].

At high level, a cache network can be modeled as a triple $\langle \mathcal{F}, \mathcal{D}, \mathcal{R} \rangle$, where: $\mathcal{F}$ represents the forwarding policy, determining the next hop for each content request, whereas content items travel back along breadcrumbs left by the requests [22, 30]; a meta-caching algorithm $\mathcal{D}$ lets node decide whether to store any new content item passing by; a replacement algorithm $\mathcal{R}$ selects, in case of positive decision in the previous step, which cache element should be evicted to make room for the new one.

Given the pervasiveness of caches in ICN, meta-caching is considered a crucial element to differentiate content of individual caches. Forwarding is instead essential to extend the reach beyond caches that lay on the path toward the repository, possibly reaching off path copies. Yet, while ICN performance are dependent on the triple $\langle \mathcal{F}, \mathcal{D}, \mathcal{R} \rangle$, with few exceptions research has so far limitedly considered a single of the above aspect in isolation – implicitly assuming either Shortest Path Routing (SPR) forwarding or Leave a Copy Everywhere (LCE) meta-caching.

Most importantly, a debate has been recently ignited around the usefulness of ubiquitous caching [16, 19]. While it is well understood that systematically caching the same object everywhere is not necessarily beneficial for system performance, however conclusive results have yet to emerge from the discussion. In particular, very recent work [16] shows that the most of the caching gain is attainable by simply (and painlessly) caching at the edge of the network. Yet, we argue that [16] misses a crucial point: i.e., that the interaction of the above policies concurs in determining the global ICN performance. While authors of [16] correctly select an ideal forwarding policy $\mathcal{F}$, that achieves (locally) optimal forwarding decisions, their (implicit) selection of the $\langle \mathcal{D}, \mathcal{R} \rangle$ pair (and especially of the LCE meta-caching policy $\mathcal{D}$ that, as we will see, plays a paramount role) yields to an underestimation of ICN performance.

In the reminder of this paper, we overview related work in Sec. 2, and especially highlight the simulation [16] and modeling [31] work we directly compare with. Sec. 3 then explores benefits of $\langle \mathcal{F}, \mathcal{D}, \mathcal{R} \rangle$, showing that significant gains can be obtained when ideal Nearest Replica Routing (iNRR) forwarding and Leave a Copy Down (LCD) meta-caching are jointly in use: indeed, LCE nullifies benefits of iNRR by forcing multiple synchronous evictions in spatially dis-

ICN'14, September 24–26, 2014, Paris, France.

http://dx.doi.org/10.1145/2660129.2660153.

joint caches, while this can be avoided by LCD (or even simple probabilistic) meta-caching. Sec. 4 then carries on an extensive simulation comparison with edge caching techniques proposed in [16]: we gather that [16] underestimates ICN gain due to (i) a limited focus on $\mathcal{F}$ forwarding policy neglecting meta-caching $\mathcal{D}$, coupled to a (ii) oversimplified network scenario with poor path diversity, so that iNRR potential cannot be fully exploited. Sec. 5 introduces our iNRR model, that builds over [31]: while [31] only considers shortest path routing toward permanent content stored at some custodian (modeling on path caching with SPR), we extend it with the ability to look for nearby temporary content replicas (modeling off path caching with iNRR). Finally, we remark that iNRR is however an *ideal* forwarding strategy: therefore, we propose and evaluate two practical implementations that, trading off delay vs distance, achieve arbitrarily close performance to iNRR in Sec. 6. To promote cross-comparison, all our code is available to the scientific community at [1].

Table 1: Taxonomy of related work: Meta-caching and forwarding have been, so far, separately studied.

Meta-caching $\mathcal{D}$	Type	Knob	Ref
LCE	[6–13, 15–17, 23–25, 29–31, 35, 39]		
Fix	Prob.	p	[5, 25, 35]
ProbCache	Prob.	Distance	[29]
LCD	Det.	Distance	[24, 25, 35]
WAVE	Prob.	Distance	[13]
Btw	Det.	Centrality	[9]

Forwarding $\mathcal{F}$	Type	Knob	Ref
SPR	[6–13, 15–17, 23–25, 29–31, 35, 39]		
Source routing	Prob.	-	[35]
Flooding	Prob.	*Distance*	[12]
INFORM	Prob.	*Delay*	[11]
CATT	Det.	*Distance*	[15]
NDN	Det.	*Dist, Delay*	[39]
iNRR	Det.	-	[6, 16]

2. BACKGROUND

Taxonomy. Tab. 1 reports a taxonomy of related work addressing ICN evaluation. The table is split in two portions, meta-caching (top) and forwarding (bottom): it clearly emerges that $\mathcal{F}$ and $\mathcal{D}$ aspects have been so far studied separately. Work focusing on meta-caching [9, 13, 24, 25, 29] usually assumes Shortest Path Routing (SPR) as underlying request forwarding strategy. In this context, many policies have been proposed that are either deterministic (LCE, LCD [24, 25], Betweenness [9]) or probabilistic (Fix [5, 25], ProbCache [29], WAVE [13]). These policies exploit different information (ranging from simple distance [25, 29] to more complex topological properties [9]) and possibly explicitly take into account ICN chunking [13].

Similarly, work focusing on forwarding policies [6–13, 15–17, 23–25, 29–31, 35, 39] usually assumes that new contents are always cached, which is commonly referred to as Leave a Copy Everywhere (LCE) in meta-caching terms. The interest of alternative strategies to SPR is that there may be closer cached copies laying *off path* between the requester and the custodian of the permanent copy, that thus SPR is unable to reach. To achieve this purpose, the ICN community has tested several forwarding approaches, ranging from multiple disjoint source routed paths [35], to dynamic approaches based on flooding [12], learning [11, 39], or routing using potential [15]. Of particular interest, [16] considers an ideal Nearest Routing Replica (iNRR) scheme that allows to reach the closest, possibly off path, cached copy. While iNRR is not a practical scheme, as it requires instantaneous knowledge of the status of all caches in the network, however it provides an ideal upper-bound to $\mathcal{F}$ performance, and as such is worth considering. Additionally, we offer two distributed NRR implementations in Sec. 6, that can attain performance arbitrarily close to that of iNRR.

Modeling. Concerning modeling work, separation of $\mathcal{F}$, $\mathcal{D}$ and $\mathcal{R}$ is easily understood: due to the complexity in analysing caching networks, studies have tackled each aspect in isolation. In particular, considering simple topologies (e.g., cascades or trees), [24] models LCD meta-caching ($\mathcal{D}$ policy), while [17] addresses LRU and random replacement ($\mathcal{R}$ policies), and [7, 8] explicitly account for the fact that objects are split in chunks. Considering instead more complex networks, [31] models object-level cache hit of Shortest Path Routing (SPR) on arbitrary topologies, while in the context of wireless networks, an asymptotic analysis of SPR vs iNRR (under LCE) is provided in [6].

We point out that while ICN introduces a number of new challenges (e.g., chunk vs object level, pervasive caching, request routing over complex topologies, etc.), caching is not a new problem. As such, in terms of modeling techniques, the above work possibly extends to the ICN context previous seminal work. More precisely, [17, 24] build over the Che [10] approximation, while model in [7, 8] extend Jelenkovic's [23] to the case of multiple chunk and [31] extends the Dan and Towsley [14] LRU approximation from a single cache to a network of caches operating according to SPR forwarding. In Sec. 5 we extend [31] to model iNRR forwarding, in reason of its performance as we will see shortly.

Simulation. Separation of $\mathcal{F}$, $\mathcal{D}$ and $\mathcal{R}$ is instead less justified in simulative work. In part, this is due to the fact that a natural choice for $\mathcal{R}$ is the Latest Recently Used (LRU) policy, though it has been pointed out that random replacement (i) exhibits similar performance at a lower complexity [17, 18, 35] (ii) it may be preferable to LRU due to line rate constraint [5, 28]. We further point out that, while the joint impact of meta-caching $\mathcal{D}$ and replacement $\mathcal{R}$ policies has gained limited attention (among others, by our own work [35]), *to the best of our knowledge, the forwarding $\mathcal{F}$ and meta-caching $\mathcal{D}$ policies have not been jointly considered so far.* As the performance impact of the $\langle \cdot, \mathcal{D}, \mathcal{R} \rangle$ couplet is limited with respect to that of $\langle \mathcal{F}, \mathcal{D}, \cdot \rangle$, in this work we mostly focus on the latter. We start by showing this impact in Sec. 3. Then, in Sec. 4, we critically contrast recent results that (too) quickly dismiss ubiquitous caching [16].

3. COUPLING BENEFIT

We start by showing that, provided that forwarding and meta-caching decisions are jointly considered, sizeable gains appear. Simulation results are obtained with ccnSim [34], an highly scalable chunk-level[1] simulator that we have developed and optimized over the last few years. To give an idea of ccnSim scalability, the large-scale scenario reported

[1]To facilitate comparison with [16, 31] that consider object-level caching, in this work we use ccnSim at object level.

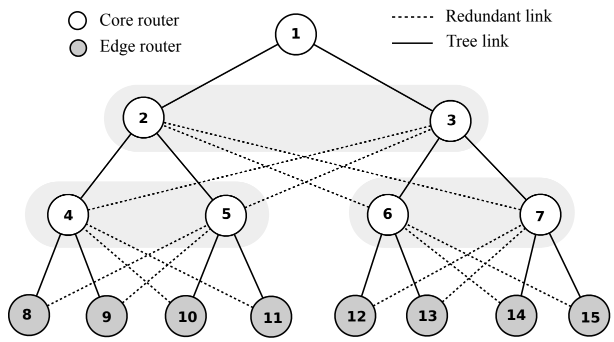


Figure 1: Redundant 4-level binary tree. Dashed links are present with probability μ. Shadowed blocks represent aggregate caches seen by lower level nodes in presence of redundancy ($\mu > 0$).

in Sec. 4, corresponding to one billion worth of object requests, out of a 100 million object catalog, with caches storing 100,000 objects, can be simulated by a common off-the-shelf PC equipped with 8GB of RAM memory in few hours [34]. For this work, we extended ccnSim to include a number of meta-caching (e.g., ProbCache [29], Btw [9]) and forwarding (e.g., iNRR [16]) algorithms, that we make available, along with the scenarios and scripts used to gather results in this paper, at [1].

3.1 Scenarios

To facilitate comparison with [16] in Sec. 4 and with [31] in Sec. 5, we consider network scenarios as similar as possible to those introduced there, namely access tree [16] and grid [31] topologies. We point out that [16] additionally considers access trees to be attached to PoP of realistic backbone networks (gathered with Rocketfuel as in our previous work [33,35]). Despite great effort is made in [16] to describe the scenario, however the lack of crucial parameters (e.g., repository placement, content redundancy and allocation to repositories, etc.), makes a 1-to-1 comparison difficult. As such, to promote cross-comparison, we make our scenarios available to the scientific community, under the form of configuration files for ccnSim, so that independent research can confirm (or disprove) our findings.

We argue that it would be possible to use *realistic topologies* and *workload*, to reinforce the realism of the evaluation. Yet, we also point out that considering realistic topologies would let the scenario significantly drift from [31], rendering the cross-comparison task harder. Additionally, while trace-driven evaluation [16, 21, 38] is tempting, we argue that it is not necessary for a *relative* performance comparison. Indeed, CDN request traces from Akamai [16] offer only an aggregated but partial view of the requests served by many ISPs, which can bias the results. Further, while [21, 38] show that real workloads yield to caching results that are more favorable with respect to synthetic workloads where object popularity is stationary over the whole period (due to a temporal request correlation on short time scales [21] and of a finite object lifetime on a longer timescale [38]), at the same time we expect temporal correlation to be beneficial to any ICN strategy. Additionally, an advantage of synthetic workload is to ensure convergence of the results shown in the following, that are thus technically sound, albeit possibly conservative as they neglect temporal request correlation.

Specifically, we consider a 10x10 grid (100 nodes) and a 6-level binary tree ($2^6 - 1$=63 nodes). Since networks are engineered adhering to fault tolerance and resilience principles, it is extremely unlikely for an access topology to have exactly a single physical link between any pair of parent and child nodes as in [16] – as otherwise, cutting a single link up in the hierarchy would cut a whole subtree. As such, we consider that a node may have an additional link to its aunt (i.e., the immediate sibling of its direct parent) that can be used for backup or load balancing. Each additional link, represented with dashed lines in the 4-level tree of Fig. 1, is present with i.i.d. probability $\mu \in [0, 1]$.

For simplicity, we consider topologies with uniform delay (1ms), as heterogeneity plays a minor role [33, 35], and consider to operate below congestion (links have infinity capacity). As in [16], that offers fitting over global Akamai dataset, we consider object popularity to follow a Zipf distribution with $\alpha \approx 1$. We use homogeneous size caches, with a cache to catalog size ratio of 0.1% (much more conservative that 5% in [16]) instantiated in a small (large) scenario where caches are able to store 100 (100,000) objects out of a 100,000 (100,000,0000) objects catalog. Small vs. large scale scenarios allow us to respectively explore wide parameter settings vs. gather performance on a more realistic use case. Simulations start from empty caches, and statistics are gathered after the hit ratio reaches steady state. Results reported in the following are averaged over 20 runs.

3.2 Performance

As performance metric, we consider the average distance that the content has traveled in the ICN network. This metric has the advantage of being very insightful and compact at the same time, as it directly relates to user QoE (i.e., delay) as well as network QoS (i.e., load and cache hit). Moreover, while [16] additionally expresses cache hit and repository load, it however mostly reports relative error between iNRR and alternate strategies: as direct comparisons are de facto impossible, and to limit redundancy given space constraints, we hence avoid reporting additional metrics beyond the content distance.

In terms of $\mathcal{F}$, instead of being limited by implementation (and configuration) details of the numerous proposed ICN forwarding policies [11, 12, 15, 35, 39], we consider (i) iNRR [16] as upper-bound of the achievable performance for off-path caching, and (ii) SPR that can limitedly hit on-path copies. In terms of meta-caching $\mathcal{D}$, we instead implement (and make available in [1]) several of the proposals in Tab. 1: we prefer to include a relatively large list (to the risk of annoying the reader), as we believe a systematic investigation of coupled forwarding/meta-caching to be necessary (in reason of the previously shown gap in the current literature). We include LCE as a term of comparison, that we instead expect to provide a performance lower-bound as it provides poor cache diversity and forces high eviction rates over the whole network. Finally, in terms of replacement $\mathcal{R}$ we experiment with LRU and uniform probabilistic replacement [5] (though we mostly report results concerning the former due to secondary $\mathcal{R}$ impact).

Fig. 2 reports the average distance at which content is found in the ICN network as function of the meta-caching

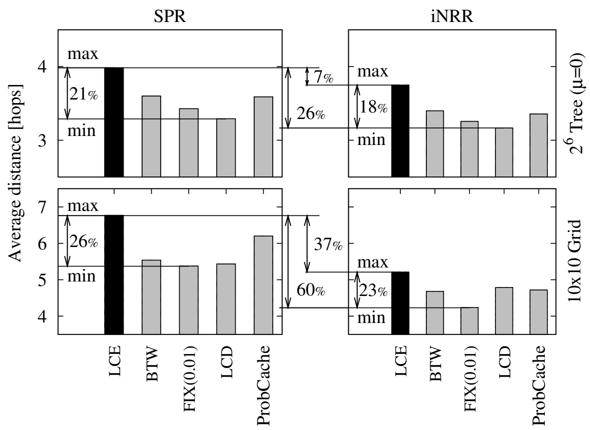


Figure 2: ⟨F,D⟩ performance at a glance: average content distance as a function of meta-caching policies, for SPR (left) and iNRR (right) forwarding, on tree (top) and grid (bottom) topologies.

policies, for SPR (left) and iNRR (right) forwarding, on tree (top, without additional links $\mu = 0$) and grid topologies (bottom). The plot is annotated with percentage gain that could be achieved by moving from the ⟨SPR,LCE⟩ worst-case to other, more sensible, ICN configurations.

First, recall that the top plots of Fig. 2 report the scenario of [16]: in this case, [16] correctly points out that the difference between ⟨SPR,LCE⟩ and ⟨iNRR,LCE⟩ is below 10%. Yet, as authors limitedly experiment with a naive LCE meta-caching, they ignore potential gain due to LCD (about 21% even considering SPR) or the joint use of LCD and iNRR (about 26%). Additionally, we point out that gains in this scenario are limited by the poor path diversity that the tree offers to iNRR. For instance, in the example provided in Fig. 1 for a 4-level tree, starting at node 15, on-path caching with SPR traverses 4 caches from the edge to the root (i.e., 15, 7, 3, 1): iNRR disposes of only 2 additional off-path nodes when $\mu = 0$ (i.e., 6, 14), but of 6 nodes when $\mu = 1$ (i.e., 4, 5, 6, 12, 13, 14).

Hence, gains of ⟨iNRR,LCD⟩ are potentially higher on more meshed topologies, that allow iNRR to explore a larger (and closer) neighborhood. This clearly reflects in the bottom plot of Fig. 2, obtained on a 10x10 grid: in this case, the difference between ⟨SPR,LCE⟩ and ⟨iNRR,LCE⟩ is about 37%. Additional gain could be attained by coupling iNRR forwarding to LCD or fixed probabilistic FIX decisions, for a reduction of the average distance of about 60%. Clearly, as content travels less than half the path in ⟨SPR,LCE⟩, the network load also divides by over a factor of two, and similarly happens for user latency. – shortly, as opposed to findings in [16], ubiquitous caching cannot be dismissed without a second look.

4. $\langle \mathcal{F}, \mathcal{D}, \mathcal{R} \rangle$ SIMULATION

4.1 Consistency of coupling gains

Fig. 3 reports a sensitivity analysis of the gains achievable by coupling meta-caching policies to forwarding policies such as iNRR, gathered via simulation over smoothly varying network redundancy $\mu \in [0, 1]$. The plot is annotated with gain

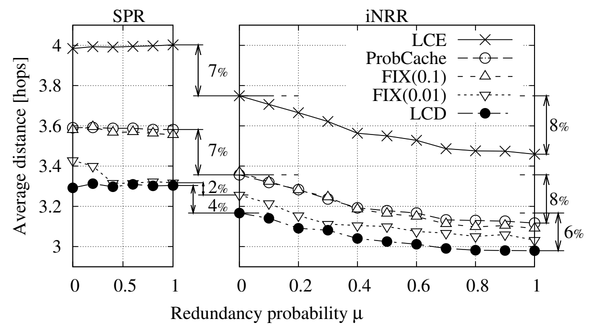


Figure 3: Sensitivity analysis of $\langle \mathcal{F},\mathcal{D} \rangle$ when $\mathcal{R}$=LRU: 6-level binary tree topology, with varying redundancy probability μ.

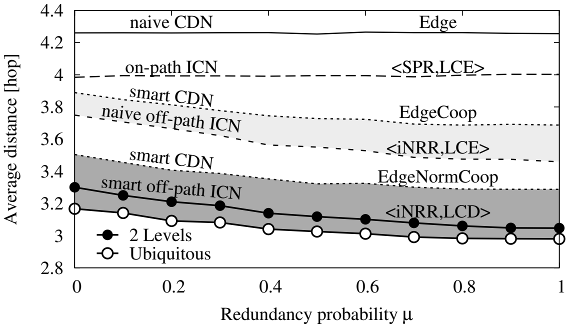
from ⟨SPR,·⟩ to ⟨iNRR,·⟩, as well as with gain due to the redundancy (from $\mu = 0$ to $\mu = 1$ for any given $\langle \mathcal{F},\mathcal{D} \rangle$ setting). As it can be expected, redundancy plays a negligible role for SPR (though in case of multiple equivalent paths, SPR chooses between them at random, possibly traversing different caches). Unsurprisingly, deterministic LCD decisions consistently achieve best performance for trees [25], exhibiting furthermore a good interplay with iNRR. Next comes simple probabilistic decisions FIX($\frac{1}{100}$), while complex probabilistic strategies driven on either distance (ProbCache [29]) or topological properties (e.g., Btw [9]) achieve intermediate gain. In reason of the added complexity (as it is often pointed out, simpler solutions are preferable due to line rate constraints [5, 28]) and limited gain, we thus disregard the latter meta-caching policies, while we point out simple probabilistic decisions to be a good-enough candidate for ICN.

Overall, the average path length increases from slightly less than 3 hops for ⟨iNRR,LCD⟩ to about 4 hops for ⟨SPR,LCE⟩, i.e. a sizeable 33% increase (though gain may be larger for more meshed topologies). Finally, we experiment with different Zipf skews: while we do not report pictures for reason of space, we observe that gain increases for growing α.

4.2 Comparison with edge-caching

We perform an exhaustive comparison of ICN vs Content Distribution Network (CDN) strategies. In particular, we consider some of the edge-caching techniques that [16] offers as "good enough" replacement for ICN. We again focus on the access tree topology, to mimic scenario in [16], and additionally consider that networks are possibly engineered with fault tolerance (i.e., redundancy probability μ).

As far as CDN is concerned, we include the *Edge* strategy where only leaf nodes have caching space, corresponding to a naïve CDN scenario. We then implement *EdgeCoop*: as the terse description ("CDN routes can do a scoped lookup in sibling cache") in [16] does not allow to understand whether only caches having a common parent can cooperate, we opt for an approach that is as favorable as possible to EdgeCoop, to avoid any bias toward ICN. Our implementation of EdgeCoop allows caching only at leaf nodes, but exploits iNRR routing strategy: thus, any temporary copy that is cached at a distance shorter than or equal to that of the permanent copy stored at the custodian above the root of the tree is possibly accessed (in practice, only half of the leaf



μ	$\langle SPR, LCE\rangle$	Edge Coop	$\langle iNRR, LCE\rangle$	Edge Norm Coop	$\langle iNRR, LCD\rangle$ 2-Levels	$\langle iNRR, LCD\rangle$ Ubiquitous
0	2%	4%	8%	14%	19%	22%
1	2%	10%	15%	19%	25%	27%

Figure 4: Comparison of several ICN vs CDN [16] strategies: average distance $E[d_X]$ of strategy X, as a function of the redundancy probability μ. Additionally, the figure tabulates the gain of strategy X over Edge, measured as $\big(E[d_{\mathbf{Edge}}] - E[d_X]\big)/E[d_{\mathbf{Edge}}]$.

nodes are accessible when $\mu = 0$, while all leafs are accessible for $\mu = 1$). Under homogeneous cache size, as in CDN-like scenarios only leaf nodes are equipped with caches, it follows that Edge and EdgeCoop scenarios have about half the cache space of ICN scenarios. To perform a fair comparison, we thus consider as in [16] an *EdgeNormCoop* scenario where individual caches are twice as large as in the previous case, so that the overall cache space is the same as in ICN scenarios. Since distance is our main performance metric, and since EdgeNormCoom allocates all cache space as close as possible to users, we expect to get a conservative estimate of ICN benefits, if any, from our comparison.

As far as ICN is concerned, we first consider a naïve ⟨SPR,LCE⟩ strategy where caching is ubiquitous but, due to SPR forwarding, only on-path caches can be exploited. We further include ⟨iNRR,LCE⟩ that [16] identifies as ICN best-case, and finally include the ⟨iNRR,LCD⟩ configuration, representing an even better alternative. In particular, to assess at a more fine-grained the value of ubiquitous caching, in the ⟨iNRR,LCD⟩ case we further consider (i) an *Ubiquitous* case where the total cache budget is allocated evenly across caches of all 6-levels of the tree and (ii) a *2-Levels* case (also considered in [16]) where the total cache space is allocated evenly across the last two levels of the tree, while nodes up in the hierarchy are not equipped with caching functionalities.

Results of the comparison are shown in Fig. 4. To confirm that we do not aim at exagerating ICN gains, consider the CDN EdgeNormCoop vs ICN ⟨iNRR,LCE⟩ strategies. While the comparison is favorable to ICN in [16] (in terms of latency, congestion and origin load performance), the reverse holds in our conservative settings (where thus distance is lower for CDN EdgeNormCoop than for ICN).

Next, to facilitate the ICN vs CDN comparison, two shaded regions are shown in the plot. On the basis of the light-gray region separating ⟨iNRR,LCE⟩ from EdgeCoop, [16] concludes that ICN does offer only minimal performance improvement over sensibly configured CDN scenarios, so that

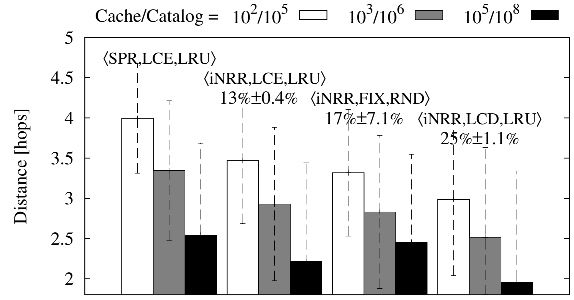


Figure 5: Small- vs medium- and large-scale scenarios. Relative gain ($\mu \pm \sigma$) over ⟨SPR,LCE,LRU⟩ remains similar over all scenarios.

(most of the) ICN gain is within reach of (less painful) CDN solutions. The dark-gray region between ⟨iNRR,LCE⟩ and ⟨iNRR,LCD⟩ instead represents the potential gain due to joint meta-caching and forwarding, missed so far by related work (including [16]), in reason of a narrow focus on specific aspects of the whole algorithmic space (recall Tab. 1).

Another interesting considerations can be made comparing the CDN-EdgeCoopNorm vs ⟨iNRR,LCD⟩ in the 2-Levels (black points) and Ubiquitous (white points) cases. Recall that these three strategies have the same cache space, but differ in the cache placement strategy. As we previously pointed out, CDN places all cache budget to the leaf, close to the users, which should be beneficial in terms of the distance to the hit. Yet, cooperation via scoped lookups with iNRR forces in this case to possibly longer paths up and down the tree. In the ICN case instead, paths to cached content can be shorter due to the statistical multiplexing gains that arise due to aggregation of requests coming from multiple leafs. At the same time, as it is still beneficial to cache the most popular content close to the users, these gain exhibit diminishing returns for an increasing number of levels of aggregation. From Fig. 4 we see indeed that *Ubiquitous* caching (6-level in this example) further reduces the distance with respect to *2-Levels*, albeit the gain reduces.

Finally, the figure tabulates gain of a strategy X over Edge, computed as $\big(E[d_{\mathrm{Edge}}]-E[d_X]\big)/E[d_{\mathrm{Edge}}]$ where $E[d_X]$ represents the average distance of strategy X. It can be seen that ⟨iNRR,LCD⟩ gain is sizeable, for both binary trees ($\mu = 0$) and trees with full redundancy ($\mu = 1$). Yet, we point out the ultimate goal for an ISP to deploy ICN is to ameliorate the service delivered to users, while possibly reducing the delivery costs [4]. Under this light, it is hard to assess whether the technical gains shown in this section translate into economic gains that are substantial enough to justifying ICN deployment – which is outside the scope of this paper and rather calls for technico-economic studies.

4.3 Small to large-scale scenarios

Small, medium (or large) scale scenarios allow us to respectively explore wide parameter settings, and gather performance on a more realistic (or extreme) use case. We fix $\alpha = 1$ and the cache to catalog size ratio C/N to a conservative 0.1%, and let the cache C and catalog sizes N vary. Precisely, we instantiate a small-scale scenario with $C/N = 10^2/10^5$, a medium-scale with $C/N = 10^3/10^6$ and a large-scale with $C/N = 10^5/10^8$. As video is preeminent, and given an average size of YouTube videos of 10MB [20], medium and large scale cache sizes vary from the feasible 10GB [5, 28] to the challenging 10TB [36] range. Catalog size of the medium scenario is of the same order of magnitude as in [16], whereas the large-scale scenario models a more challenging YouTube scenario.

Average distances (and coefficient of variation) are reported in Fig. 5 for naïve on-path caching ⟨SPR,LCE,LRU⟩, naïve off-path caching ⟨iNRR,LCE,LRU⟩, simple probabilistic off-path meta-caching and replacement ⟨SPR,FIX,RND⟩, and the best off-path strategy ⟨SPR,LCD,LRU⟩. Each strategy is annotated with the average gain over ⟨SPR,LCE,LRU⟩ (± standard deviation across different scales).

We see that performance improve (i.e., distance decreases) for large catalogs. This can be explained considering that, for fixed Zipf $\alpha = 1$ and fixed cache to catalog ratio C/N, a larger cache C can accommodate a larger fraction of top content out of the entire catalog N. Formally, $\sum_i^C i^{-\alpha} / \sum_i^N i^{-\alpha}$ increases from small to large catalog, so that C=100 (100,000) most popular cached objects corresponds to the 43% (63%) of the whole requests for a N=100,000 (100,000,000) catalog.

Hence, we gather that small scale scenario (i) corresponds to conservative cache hit results and (ii) allows a reliable estimate of the relative gain of ubiquitous caching over on-path caching – as the relative gain over ⟨SPR,LCE,LRU⟩ is the same for all scenarios (except the simplistic ⟨SPR,FIX,RND⟩ case we disregard in the following).

5. MODELING iNRR

As shown in the previous section, iNRR achieves interesting performance with respect to SPR forwarding. Furthermore, iNRR benefits are especially apparent with topologies having redundant links. As such, it would be useful to have an approximate iNRR model valid for arbitrary network of caches. We tackle this challenge by extending the aNET model proposed in [31], that unlike other caching models is applicable to any topology but is limited to Shortest Path Routing. In this section, we first recall aNET (Sec. 5.1) and introduce the relevant notation, then present our iNRR extension (Sec. 5.2) before comparing their accuracy (Sec. 5.3).

5.1 aNET model and notation

According to our terminology, a ⟨SPR,LCE,LRU⟩ network is modeled by aNET [31]. aNET approximates network behavior by decomposing the problem and computing the LRU approximation [14] for each cache in the network. The network itself is represented as a graph $G = (V, E)$ with $v \in V$ a vertex node having a cache of size $|v|$ objects. We denote the content catalog with $\mathcal{N}$, with size $N = |\mathcal{N}|$. As ⟨SPR,LCE,LRU⟩ forwards the miss stream of each cache along the SPR toward the permanent replica, it follows that the incoming request stream at each cache accounts for both exogenous user request, as well as the miss stream of neighboring caches. aNET takes into account this incoming stream by iterating the solution of individual caches, and reevaluating the miss stream until the stabilization of the whole system. aNET iteratively solves the following set of equations reported in Fig. 6.

Incoming requests at node v for content $i \in \mathcal{N}$ are expressed in (1). The first term in (1) represents the exogenous arrival rate $\lambda_{i,v}$ for content i, and the second term accounts for the miss stream $m_{i,u}$ coming from neighboring nodes u having v as their next hop $R(u, \mathcal{S}(i))$ in the shortest path

$$r_{i,v} = \lambda_{i,v} + \sum_{u:R(u,\mathcal{S}(i))=v} m_{i,u} \quad (1)$$

$$p_{i,v} = \frac{r_{i,v}}{\sum_{j=1}^{N} r_{jv}} \quad (2)$$

$$\vec{\pi}_v = LRU(\vec{p}_v, |v|) \quad (3)$$

$$m_{i,v} = r_{i,v}(1 - \pi_{i,v}) \quad (4)$$

Figure 6: aNET model

toward the repository $\mathcal{S}(i)$ for content $i \in \mathcal{N}$. The local popularity $p_{i,v}$ is expressed by (2), representing the relative proportion of request of content i at node v. Given the steady state local request distribution over all contents $\vec{p}_v$ and a cache size $|v|$, each cache v applies in (3) the LRU algorithm [14] to determine the probability $\vec{\pi}_v$ that any given content $i \in \mathcal{N}$ is present in its cache. Finally, the miss stream $m_{i,v}$ is computed as in (4).

Two crucial points in the above set of equations are worth stressing. First, (4) was only proven to hold for an Independent Reference Model (IRM) [31]. Second, the approximate LRU algorithm (3) was designed only for IRM streams [14]. However, as the request stream also consists of miss stream of the neighbors as per (1), the aggregate request stream is not IRM: hence, steps (3)-(4) consist in an IRM violation, and are potential sources of error in the approximation.

5.2 iNRR model

We extend the set of aNET equations to model iNRR forwarding strategy. Under SPR forwarding, content can be possibly found only along the shortest path toward a custodian of permanent content replicas: hence, the miss stream (1) aggregates requests of shortest paths passing through v. The crucial difference from aNET is that, under iNRR forwarding, any *valid* path is possibly followed. By valid path, we imply that (i) paths are loop free, (ii) in case multiple copies are stored at several nodes along any given path, the closest copy is accessed. Additionally (iii) in case of multiple copies having equal distance over multiple paths, each copy is equally likely to be chosen.

To model the above observations (i)–(iii), we introduce the following notation. As in aNET, the SPR routing matrix for the network $R(v, u), v, u \in V$ indicates v's next hop to reach node u. Nodes are directly connected to v when $R(u, v) = v$, and we indicate with $N(v) = \{u : R(u, v) = v\}$ the set of v's neighbors. For convenience, $\mathcal{S} = \mathcal{S}(i)$, $\forall i \in \mathcal{N}$ indicates the unique repository in the network (the model can be easily extended to the case of multiple repositories), so that $R(v, \mathcal{S})$ represents the FIB information used by SPR to reach it.

In addition to SPR FIB information (possibly hitting content cached on-path to $\mathcal{S}$ as in aNET), iNRR is able to find any off-path content that is not located further than the repository (so that caches as close as the repository, can offload the latter). To identify such content, we define $D(v, u)$ as the SPR distance between any two nodes $v, u \in V$. We next define $B(v, u)$ as the ball centered in v having ray $D(v, u)$, i.e., $B(v, u) = \{x \in V : D(v, x) \le D(v, u)\}$. Thus, $B(v, u)$ represents the set of nodes that are not further away than u from v. For convenience, we also define the border and interior of $B(v, u)$ as $B_b(v, u) = \{x \in V : D(v, x) = D(v, u)\}$ and $B_i(v, u) = \{x \in V : D(v, x) < D(v, u)\}$ re-

$$r_{i,v} = \lambda_{i,v} + \sum_{u:u\in N(v)} m_{i,u,v} \tag{5}$$

$$p_{i,v} = \frac{r_{i,v}}{\sum_{j=1}^{N} r_{jv}} \tag{6}$$

$$\vec{\pi}_v = LRU(\vec{p}_v, |v|) \tag{7}$$

$$m_{i,v} = r_{i,v}(1-\pi_{i,v}) \tag{8}$$

$$s_{i,v,u} = \sum_{\substack{x:R(v,x)=u\\ \wedge\, x\in B(v,\mathcal{S})}} \left[\prod_{y\in B_i(v,x)} (1-\pi_{i,y})\right] \frac{\pi_{i,x}^2}{\sum_{z\in B_b(v,x)} \pi_{i,z}} \tag{9}$$

$$m_{i,v,u} = \begin{cases} m_{i,v}s_{i,v,u} & u \neq R(v,\mathcal{S}) \\ m_{i,v}(1-\sum_{w\neq u} s_{i,v,w}) & u = R(v,\mathcal{S}) \end{cases} \tag{10}$$

Figure 7: iNRR model

spectively. For instance, $B_b(v,\mathcal{S})$ represents the set of nodes that are as far from v as the server $\mathcal{S}$, while $B_i(v,u)$ represents the set of nodes closer than u to v. Finally, we denote with $m_{i,u,v}$ the proportion of miss stream for content i coming from u to v. Then, our iNRR model iteratively solves $\forall i \in \mathcal{N}, v \in V$ the set of equations reported in Fig. 7. Shortly, while (6), (7) and (8) perform the same steps as in aNET, iNRR modifies (5) to account for a proportion of miss stream of neighboring nodes, and further adds equations (9) and (10) to quantify this proportion.

As per observation (iii), any node u will split its miss stream equally among its neighbors $N(u)$. This is modeled by (5), where all v's neighbors $N(v)$ contribute to request arrival at v, with $m_{i,u,v}$ the proportion of miss stream for content i coming from u. Observations (i) and (ii) are instead expressed through (9) and (10). More precisely, (9) defines the split ratio $s_{i,v,u}$ among neighboring nodes, and (10) applies the split ratio to the miss stream $m_{i,v}$, depending on whether u lays on the shortest path to the server $u = R(v,\mathcal{S})$ or not.

Especially, (9) bares additional discussion. The term $s_{i,v,u}$ represents the proportion of the miss stream of node v sent through v's immediate neighbor u to reach node x for content i. iNRR forwards such requests iff:

- Next hop for x from v passes through $u = R(v,x)$, and the distance $D(v,x)$ is shorter than or equal to the distance toward the server $R(v,\mathcal{S})$, i.e., x falls in the ball $B(v,\mathcal{S})$ (external sum).
- Any node y closer than x to v, i.e., laying in the interior ball $B_i(v,x)$, does not own the content i, which happens with probability $1-\pi_{i,y}$ for each node (internal product).
- The selected node x owns the item i (with probability $\pi_{i,x}$), and it is chosen among all the nodes $z \in B_b(v,x)$ at the same distance from v (terms $\pi_{i,x}/\sum_z \pi_{i,z}$).

Finally, by means of (10), we differentiate the case in which the neighbor $u = R(v,\mathcal{S})$ is the immediate next hop toward the repository or not, giving preference to cached copies to offload the repository. Hence, the miss stream that finds objects in the ball $B(v,\mathcal{S})$ flows through off-path

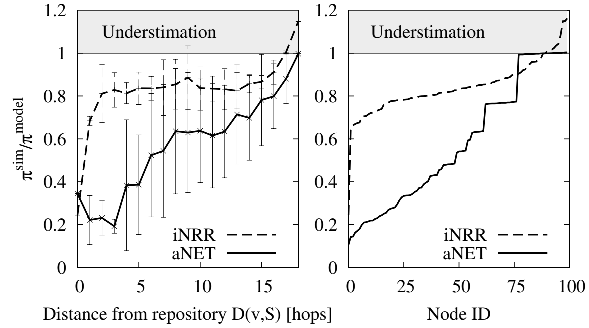


Figure 8: iNRR vs aNET hit rate π accuracy, on a 10x10 grid: per-node (left) and as function of the distance from the repository (right).

neighbors, whereas the rest of the miss stream flows through the next hop $u = R(v,\mathcal{S})$, thus on-path to $\mathcal{S}$. As in aNET, we iterate until convergence (average distance between two consecutive steps of (5) to be $< 10^{-5}$).

5.3 iNRR vs aNET accuracy

Our model inherits IRM assumption of aNET, hence it also inherits possible inaccuracy due to IRM violation. As aNET vs. iNRR model different ICN architecture, namely on-path vs. off-path caching, their result cannot be directly compared. Thus we evaluate their accuracy against simulation of ⟨SPR,LCE,LRU⟩ vs. ⟨iNRR,LCE,LRU⟩ respectively, and consider a 10x10 grid, where the iNRR gain over SPR is visible (recall Fig. 2). We compute accuracy with respect to simulation for (i) each node individually, as well as for (ii) all nodes having the same distance $\{x : D(x,\mathcal{S}) = d\}$ from the repository. More precisely, indicating the average hit probability for node v as $\bar{\pi}_v$, we evaluate accuracy in Fig. 8 as the ratio $\bar{\pi}_v^{sim}/\bar{\pi}_v^{model}$.

As for aNET, we know from [31] that the impact of IRM violation grows with the size of the network under study (or, equivalently, decreases with the density of repository in the network). This is because the IRM assumption does not hold especially for long paths, as miss stream prevails over the exogenous arrivals. Consequently, we expect aNET to be negatively affected by the large topology size, as the SPR distance to $\mathcal{S}$ can grow quite large. We instead expect iNRR forwarding to lessen the impact of IRM violation with respect to SPR. First, this is due to the fact that iNRR find closer copies (see Sec. III-A of [31]). Second, and most important, under iNRR nodes split their miss stream across each neighbor: as this mixes independent miss stream flows, it results in a more IRM-like miss flows with respect to SPR routing (similarly to what happens by increasing the k-arity of the SPR tree in Sec. III-A of [31]). Hence, we point out comparison on the same scenario to be unfair, as aNET and iNRR are neither operating on the same distance, nor on the same neighbor fanout. To partly compensate for this bias, we attach clients to each grid node, i.e., $\lambda_{i,v} > 0, \forall v$, so to reinforce the IRM component of the request arrival, in an attempt to make the comparison more favorable to aNET.

For the sake of readability, in the left plot of Fig. 8 nodes are ranked for increasing $\bar{\pi}_v^{sim}/\bar{\pi}_v^{model}$ ratios. In the right plot of Fig. 8, we complement the average ratio with standard deviation bars. First, results confirm that iNRR error

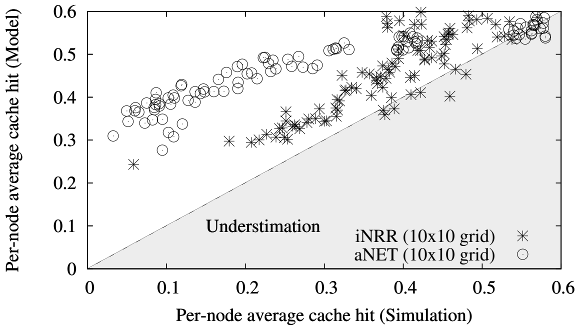


Figure 9: Scatter plot of the average cache hit per node $\bar{\pi}_v$ obtained via simulation vs model, for aNET and iNRR, on a 10x10 grid.

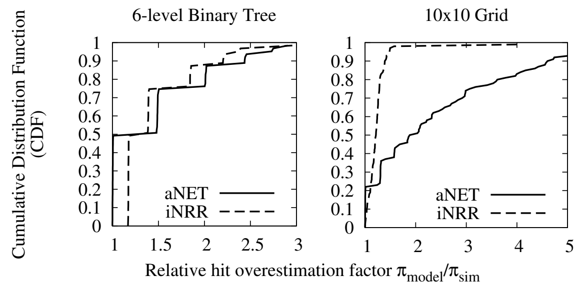


Figure 10: CDF of the cache hit overestimation for aNET vs iNRR model, on a 6-level binary tree (left) and a 10x10 grid (right).

is significantly lower than aNET. We can further observe that the iNRR error is less affected by the topological position (essentially, SPR distance) from the repository with respect to aNET. In the aNET case, the ratio becomes closer to 1 as the distance from the repository increases: notice the large plateau of about 20 nodes (i.e., leaves of the SPR distribution tree rooted at $\mathcal{S}$) having unity ratio in top of Fig. 8, that are aggregated at $d = 18$ in bottom of Fig. 8. We further show a scatter plot of the average cache hit per node $\bar{\pi}_v$ obtained via simulation vs model in Fig. 9, showing that under iNRR model overestimation reduces especially for nodes with low cache hit.

To further exacerbate difference between iNRR and aNET, we consider additional scenarios that reinforce the soundness of the above reasoning. Specifically, we contrast a 6-level binary tree scenario (where clients are attached only at leaf nodes) to the 10x10 grid in Fig. 10. The figure depicts CDFs of the cache hit overestimation (computed as $\bar{\pi}_v^{model}/\bar{\pi}_v^{sim}$, inverse w.r.t. metric shown in Fig. 8): as expected, performance are very close in the tree but very far apart in the grid, confirming our reasoning. Aside, notice the perfect match of LRU [14] for the SPR case in the tree topology, that no longer holds for iNRR, where leaf nodes also possibly receive a non-IRM miss-stream component of other nodes.

6. APPROXIMATE iNRR IMPLEMENTATION

6.1 Framework

It should be clear that iNRR is an ideal forwarding policy, requiring an oracle or, equivalently, the knowledge of the state of all caches to instantaneously propagate of in the whole network. We thus propose two practically viable implementations of Nearest Neighbor Routing (NRR). We cast these solutions on the ground of the general framework we develop in [12], that we briefly recall here.

We assume ICN nodes to be equipped with a FIB structure, proactively populated by a SPR routing protocol, containing information that allows to follow the shortest path toward a permanent copy of the repository. Requests forwarded along the FIB have thus the chance to find *on path* cached copies, and in case no cached copy is found, they ultimately access the permanent replica at the custodian.

Additionally, we require ICN nodes to be equipped with a Temporary FIB data structure (TFIB), reactively populated by an *off path* exploration of the ICN network, triggered by user demand on a new request. We assume that the *exploration phase* is carried only for the first (or few) chunk(s) of a new content, and is aimed at dynamically constructing a path toward the closest cached replica. The path is then stored in the TFIB. In the subsequent *exploitation phase*, the forwarding process can use the new TFIB entry for the next chunks requests of the same content (overriding thus FIB entries). While it is outside the scope of this paper, we point out that TFIB is possibly managed as a LRU cache, so that TFIB entries span over subsequent requests of different users for the same content.

6.2 Design

In this section, we focus on the *exploration phase*, of which we provide two alternative implementations based on scoped flooding, namely NRR' and NRR", that respectively require one and two phases. Both NRR' and NRR" flood requests over the network, limiting the flooding scope via a TTL field. In modeling terms, NRR limits the radius ρ of the ball centered around v, i.e., $B_\rho(v) = \{u : D(v,u) \leq \rho\}$.

Differences from NRR' and NRR" arise in the way requests are treated during the exploration phase. NRR' floods *regular request* packets, so that it generates possibly multiple data chunks in return – one per each cached copy found in $B_\rho(v)$. Hence, NRR' possibly generates an overhead in terms of load and cache eviction rate, though the duration of the exploration phase is the minimum possible before the closest copy is hit. Conversely, NRR" floods *meta request* packets, with a flag set to indicate that only a binary reply concerning content availability, but not the whole content data, is requested in return[2]. Replies of this first phase populate the TFIB with a negligible load (no actual data is sent), avoiding cache pollution due to eviction (as only meta information about the chunk is sent), but introduces a delay (data downloaded in the second phase).

6.3 Evaluation

Before considering the tradeoff induced by NRR' vs NRR" in terms of load vs delay, let us first analyze their impact on cache eviction. We compare NRR' and NRR" to iNRR by measuring the number of additional hops needed on av-

[2]This technique is already commonly used, e.g., in HTTP GET vs HEAD request methods: in the former case, the HTTP response encapsulates the whole object data, in the latter case, only the headers concerning the object.

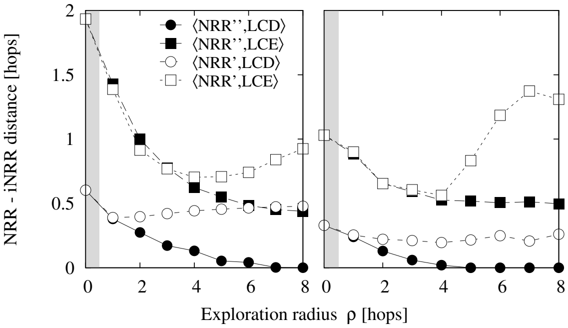


Figure 11: Additional distance of NRR implementations with respect to iNRR: NRR' vs NRR" policies, for LCE or LCD meta-caching, as a function of the exploration radius ρ. 10x10 grid (left) and 6-level μ=1 tree (right).

erage to find the content. For completeness, we consider $\mathcal{D} \in$ {LCE,LCD} and $\mathcal{F} \in$ {NRR',NRR"} with respect to the ideal ICN strategy ⟨iNRR,LCD,LRU⟩. Fig. 11 depicts the number of additional hops as a function of the radius ρ for the grid (left) and tree (right) topologies. The picture reports all ⟨$\mathcal{F}$,$\mathcal{D}$⟩ combinations of NRR" (black) vs NRR' (white) and LCD (circle) vs LCE (square) settings. For $\rho = 0$, NRR degenerates in SPR routing (shadowed region).

Several interesting insights are gathered from Fig. 11. First, performance of ⟨iNRR,LCD,LRU⟩ can be approximately arbitrarily close with ⟨NRR",LCD,LRU⟩, as the additional distance goes to zero for $\rho \geq 6$ on trees and grids. Second, cache eviction due to LCE implies an important performance penalty for both NRR' and NRR" (as expected due to results in previous section). Third, even under LCD, cache eviction due to flooding possibly translate into an important performance penalty as well: this is due to the use of regular request packets in NRR', generating data in return that possibly yields multiple cache evictions. Fourth, notice that additional distance decreases for growing ρ only in the case of NRR": this means that NRR" exploration is not only effective but also robust. Conversely, in the NRR' case, whenever ρ increases, eviction increases as well due to both higher chance to find the content on the one hand, and longer paths up to ρ on the other hand. This phenomenon is especially evident for the tree under LCE meta-caching: as soon as ρ becomes comparable with the distance to the repository, this allows a significantly larger portions of the tree to be explored, with consequent massive eviction[3]. We thus conclude that ⟨NRR",LCD⟩ with (arbitrarily large) ρ is able to (arbitrarily close) approximate iNRR.

We now comment on the load and delay induced by NRR' and NRR". As far as load is concerned, NRR" is clearly more lightweight than NRR'. Indeed, while the number of requests sent by NRR' and NRR" is the same, the amount of data chunks sent in return equals either (i) the number of cache hits for NRR', or (ii) the single closest hit for NRR". As chunks travel multiple links, NRR" significantly reduces the load not only because it sends a single chunk (major impact on load), but also because it sends the closest among all cached chunks (second order impact).

As far as delay is concerned, NRR' is possibly faster than NRR" due to the fact that whenever the data is found, it is immediately sent back, whereas NRR" requires an additional phase. While at a first glance it may seem that delay under NRR" would be roughly double with respect to NRR', however this is not the case. Observe first that exploration delay only affects the first chunk, and not subsequent chunks that instead exploit readily available TFIB information. Hence, the delay penalty of the first chunk diminishes weighted over the whole content transmission. Additionally, Fig. 11 shows that content is closer in NRR" than in NRR': for instance, in the 10x10 grid, the median number of hops is $d'' = 2$ under NRR" and $d' = 3$ under NRR'. Denoting with δt the average link delay, the median duration of the two phases in NRR" takes $2(2d''\delta t)$: compared to a median duration of $2d'\delta t$ for single-phased NRR', this accounts for a modest 25% increase, that moreover applies to the first chunk only.

7. DISCUSSION AND CONCLUSION

This paper offers new arguments to the debate about gain vs pain of ubiquitous caching. Our contributions can be summarized as follows. First, we show that gains of ubiquitous caching only appear by jointly considering $\mathcal{F}$ forwarding and $\mathcal{D}$ meta-caching policies. Specifically, we show that, in both ideal and practical settings, meta-caching policies (such as LCD or even simple random policies) are necessary to enable potential gains offered by smart forwarding policies (such as iNRR and variants) – as otherwise these potential gains are completely offset by cache pollution dynamics.

Under this light, it appears that while [16] dismisses ubiquitous caching due to its limited gains, the comparison has however missed the actual best-case for ICN performance. Indeed, our results show that ⟨iNRR,LCD,·⟩ obtains significant gains beyond the ⟨iNRR,LCE,·⟩ strategy identified in [16] as the ICN optimum. Yet, this work is by no means complete, as gains are obtained over a limited set of synthetic topologies, with a temporally stationary and spatially uniform catalog. Since LCD has been designed for hierarchical topologies, alternative meta-caching policies, as 2-LRU [26], may be preferable in the general case. Similarly, the benefits of aggregation may be exacerbated by workload where requests are spatially correlated [33]. Finally, results need to be confirmed on realistic workload, such as real traces [16, 21] or synthetic workload fit on real traces [38].

Consequences of these findings can be discussed from multiple viewpoint. From a technical viewpoint, it follows that future ICN literature should not limitedly consider a naïve ⟨SPR,LCE,LRU⟩ strategy, as it offers a too weak candidate for comparison, but also consider ⟨iNRR,LCD,LRU⟩ or even better alternatives [37]. Comparison with optimal strategies [3] is still missing: gauging the distance from optimum would help in understanding the extent of gains that are still possible beyond ⟨iNRR,LCD,LRU⟩. From an economic viewpoint, business considerations will answer whether such gains are economically worth the deployment of ICN – yet, business considerations should be taken on the ground of all relevant technical information. On this regard, caching is likely to play an important role but surely not the only one. ICN are indeed appealing also to solve the curse of mobility, and additionally offer an appealing model where security is bound to content, as opposed to the channel used for its

[3]Intuitively, under LCE cache pollution extends to the other side of the tree. Under LCD, as popular content is pulled toward the edge of the network, requests do not explore the whole network, successfully limiting cache pollution.

transmission. All these aspects are outside the scope of this work.

We then model performance of iNRR under arbitrary cache networks by extending aNET [31]. We observe that iNRR is far more accurate than aNET: this follows from the fact that, due to a reduced average distance with respect to SPR, as well as an increased mixing of miss streams typical of iNRR, the IRM model violations are less violent for iNRR than they are for aNET. Still, we notice that due to a systematic cache hit overestimation, there is room for improvement in iNRR, e.g., following the approach in [27]. Additionally, iNRR currently models an LCE policy: hence, extending the model to the LCD policy seems a logical next step.

Finally, we evaluate two practically viable implementations of NRR based on scoped flooding. We start by observing that the exploration approach proposed in [16] is a necessary ingredient to reach off-path content. To put our contribution in perspective, we may say that this work finds the remaining two necessary ingredients. Indeed, exploration has possible unwanted consequences: since popular content is possibly hit at multiple caches, data sent back in return may unnecessarily replicate at multiple caches of these paths [12]. We identify meta-caching as the second ingredient, necessary to limit the proliferation of the same content on routers along each of these return paths. We finally identify meta-interests [32] as the last ingredient, necessary to avoid proliferation over multiple paths. Meta-interests let NRR" attain (i) the shortest possible distance, as it achieves an arbitrarily close approximation of iNRR, (ii) the lowest possible data overhead, as it avoids multiple parallel requests for the same chunk, at the price of (iii) a tolerable increase for the delay of the first chunk.

Acknowledgements

This work has been carried out at `http://www.lincs.fr`. We thank our shepherd, Cedric Westphal, and the anonymous reviewers whose useful comments have contributed in significantly ameliorating the quality of this paper.

8. REFERENCES

[1] http://www.infres.enst.fr/~drossi/ccnSim.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7), 2012.

[3] A. Araldo, M. Mangili, F. Martignon, and D. Rossi. Cost-aware caching: optimizing cache provisioning and object placement in icn. In *IEEE Globecom*, 2014.

[4] A. Araldo, D. Rossi, and F. Martignon. Design and evaluation of cost-aware information centric routers. In *ACM ICN*, 2014.

[5] S. Arianfar and P. Nikander. Packet-level Caching for Information-centric Networking. In *ACM SIGCOMM, ReArch Workshop*, 2010.

[6] B. Azimdoost, C. Westphal, and H. R. Sadjadpour. On the throughput capacity of Information-centric networks. In *ITC*, 2013.

[7] G. Carofiglio, M. Gallo, and L. Muscariello. Bandwidth and Storage Sharing Performance in Information Centric Networking. In *ACM SIGCOMM, ICN Workshop*, 2011.

[8] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino. Modeling Data Transfer in Content-Centric Networking. In *ITC*, 2011.

[9] W. Chai, D. He, I. Psaras, and G. Pavlou. Cache less for more in information-centric networks. In *IFIP Networking*. 2012.

[10] H. Che, Z. Wang, and Y. Tung. Analysis and design of hierarchical web caching systems. In *IEEE INFOCOM*, 2001.

[11] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini. INFORM: a dynamic interest forwarding mechanism for information centric networking. In *ACM SIGCOMM, ICN Workshop*, 2013.

[12] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino. Exploit the known or explore the unknown?: hamlet-like doubts in icn. In *ACM SIGCOMM, ICN Workshop*, 2012.

[13] K. Cho, M. Lee, K. Park, T. Kwon, Y. Choi, and S. Pack. WAVE: Popularity-based and collaborative in-network caching for content-oriented networks. In *IEEE INFOCOM, NOMEN Workshop*, 2012.

[14] A. Dan and D. Towsley. An approximate analysis of the lru and fifo buffer replacement schemes. *ACM SIGMETRICS*, 1990.

[15] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga. CATT: potential based routing with content caching for icn. In *ACM SIGCOMM, ICN Workshop*, 2012.

[16] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. M. Maggs, K. Ng, V. Sekar, and S. Shenker. Less pain, most of the gain: Incrementally deployable. In *ACM SIGCOMM*, 2013.

[17] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for lru cache performance. In *ITC*, 2012.

[18] M. Gallo, B. Kauffmann, L. Muscariello, A. Simonian, and C. Tanguy. Performance evaluation of the random replacement policy for networks of caches. In *ACM SIGMETRICS*, 2012.

[19] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Information-centric networking: seeing the forest for the trees. In *ACM HotNets'X*, 2011.

[20] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *ACM IMC*, 2007.

[21] C. Imbrenda, L. Muscariello, and D. Rossi. Analyzing cacheable traffic in ISP access networks for micro CDN applications via content-centric networking. In *ACM ICN*, 2014.

[22] V. Jacobson, D. K. Smetters, N. H. Briggs, J. D. Thornton, M. F. Plass, and R. L. Braynard. Networking Named Content. In *ACM CoNEXT*, 2009.

[23] P. R. Jelenković. Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities. *Annals of App. Prob.*, 9(2), 1999.

[24] N. Laoutaris, H. Che, and I. Stavrakakis. The LCD interconnection of LRU caches and its analysis. *Performance Evaluation*, 63(7), 2006.

[25] N. Laoutaris, S. Syntila, and I. Stavrakakis. Meta Algorithms for Hierarchical Web Caches. In *IEEE ICPCC*, 2004.

[26] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. In *IEEE INFOCOM*, 2014.

[27] N. Melazzi, G. Bianchi, A. Caponi, and A. Detti. A General, Tractable and Accurate Model for a Cascade of LRU Caches. *Communications Letters, IEEE*, 18(5):877–880, May 2014.

[28] D. Perino and M. Varvello. A reality check for content centric networking. In *ACM SIGCOMM, ICN Workshop*, 2011.

[29] I. Psaras, W. K. Chai, and G. Pavlou. Probabilistic in-network caching for information-centric networks. In *ACM SIGCOMM, ICN Workshop*, 2012.

[30] E. J. Rosensweig and J. Kurose. Breadcrumbs: Efficient, Best-Effort Content Location in Cache Networks. *IEEE INFOCOM*, 2009.

[31] E. J. Rosensweig, J. Kurose, and D. Towsley. Approximate Models for General Cache Networks. *IEEE INFOCOM*, 2010.

[32] D. Rossi and G. Rossini. Method for managing packets in a network of information centric networking (ICN) nodes, Patent EPO14305866.7/2014.

[33] G. Rossini and D. Rossi. A dive into the caching performance of content centric networking. In *IEEE CAMAD*, 2012.

[34] G. Rossini and D. Rossi. ccnSim: an highly scalable ccn simulator. In *IEEE ICC*, 2013.

[35] G. Rossini and D. Rossi. Evaluating ccn multi-path interest forwarding strategies. *Computer Communications*, 36(7), 2013.

[36] G. Rossini, D. Rossi, M. Garetto, and E. Leonardi. Multi-Terabyte and Multi-Gbps Information Centric Routers. In *IEEE INFOCOM*, 2014.

[37] M. Tortelli, D. Rossi, G. Boggia, and L. A. Grieco. Pedestrian Crossing: The Long and Winding Road toward Fair Cross-comparison of ICN Quality. In *Q-ICN*, 2014.

[38] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini. Temporal locality in today's content caching: why it matters and how to model it. *ACM SIGCOMM CCR*, 43(5):5–12, 2013.

[39] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang. A case for stateful forwarding plane. *Computer Communications*, 36(7), 2013.