On Popularity- and Volume-based Reduction of Logistic Costs in ICN

Fariborz Derakhshan*, Andreas Timm-Giel*, Ramón Agüero†
*Hamburg University of Technology, Germany
†Universidad de Cantabria, Spain
{fariborz.derakhshan, timm-giel}@tuhh.de
ramon@tlmat.unican.es

***Abstract*—Information-centric networking (ICN) is structured as a distributed caching system in which user requests arrive at edge nodes and traverse the cache hierarchy to access their demanded contents. If an intermediate cache node holds a copy of the desired content, it is promptly delivered to the client within the local domain. Otherwise, it must be retrieved from external sources, causing longer access latencies and higher provisioning costs. To minimize the external logistic costs from the backhaul, we harness the potential of Interaction-based Caching to preserve the most popular and largest contents for a maximum time in the local network. This strategy increases the cache hit rate and reduces the total volume of data that needs to be imported from external sources and the associated logistic costs. Simulative performance evaluations prove the significant gains of the proposed algorithm in terms of hit ratio (over 10.64%), external volume import (over 1.18 times relative to internal volume transport), and minimum content provisioning costs (over 5.1 times relative to internal transport costs) with respect to the Leave Copy Everywhere (LCE) strategy.**

***Index Terms*—Information-centric Networking, Popularity-based Caching, Size-based Caching, Topology-based Caching, Content Volume, Hit Rate, Logistic Costs, Content Provisioning Costs, Interaction-based Caching**

I. Introduction

The advent of continuously evolving data-generating and data-consuming technologies, applications, and networks has ushered in an unprecedented surge in data volume. This surge is chiefly attributable to video and related multimedia applications, with video content alone representing the predominant share of approximately 80% of mobile traffic by 2027 [1]. The forthcoming generations of networks, including 5G, 6G, and beyond, are poised to facilitate the emergence of novel multimedia applications, such as augmented/virtual reality and the Metaverse. These applications demand exceptionally high data rates and remarkably low latencies. An effective strategy to mitigate content delivery delays and alleviate network congestion entails caching large contents often requested in proximity to end-users.

In this paper, we develop a strategy to decrease the volume of content imported from external sources by considering their popularity and size, and leveraging the potential of Interaction-based Caching (IC). In [2], we described how the hit performance rises by permanently saving the n most popular contents in the network, where n is limited by the universal cache size. And in [3], we demonstrated how the caching performance can be maximized by also extending the availability of the rest of the popular contents. The caching performance significantly increases by strictly regulating the diffusion of contents into the cache hierarchy.

Most existing works adopt the popularity of contents as the critical criterion to cache them, and assume the same size for each. However, in scenarios with varying content sizes, prioritizing contents solely on their popularity may not optimize bandwidth usage, as the most requested items are not always the largest. In an extreme case, where the most demanded contents are the smallest ones, the large contents must be recurrently imported from remote sources, deteriorating their retrieval times and increasing the transmission costs and the network congestion probability. To counter this, we also take the contents' volume V into account to maximize the availability of the most popular contents in the network and minimize the volume of data imported from sources outside the network.

The main contributions of this paper are:

- A refined caching strategy that minimizes the external data import and content provisioning costs by jointly considering the contents' popularity P and volume V.
- The seamless integration of P and V into the IC caching strategy, maximizing the availability of prioritized contents in the network.
- A thorough analysis of the caching strategy's effectiveness focused on hit ratio, internal and external data transfer volumes, and the related logistical costs.

The rest of this paper is structured as follows: Section II introduces the related work; Section III details the proposed algorithm; Section IV evaluates its performance, comparing it with alternative solutions; and Section V concludes the paper, outlining our future lines of work.

II. Related Work

The authors of [4] use big data analytics to estimate the popularity of contents, and to indicate the importance of considering the content size for making caching decisions.

Ramón Agüero was partially funded by the Government of Cantabria (FEDER funds) through the project "Enabling Technologies for Digital Twins and their application in the chemical and communications sectors" (GDQuiC) of the program "Grants for research projects with high industrial potential of technological agents of excellence for industrial competitiveness TCNIC.

The authors in [5] propose a strategy based on a usability Time-to-Use (TTU) tag, proportional to the content's popularity and volume. An arriving content is stored only if its average request time is less than its TTU, so it does not expire before the arrival of the subsequent request. If there is not enough space in the cache, LRU (Least Recently Used) is applied to create space. The strategy aims to evict less popular contents and cache more popular ones, however, it ignores the topological properties of the nodes.

The authors of [6] distinguish between a local (l) and a remote (r) network, and defines an average content provisioning cost by $C = p_l C_l + p_r C_r$, where p is the corresponding probability of finding a content in the local or remote network, and C denotes the related cost. It then minimizes C for contents with varying popularity by a caching strategy, lying between a non-cooperative, greedy one, which locally stores as many popular contents as the concerning nodes' storage allows, and a fully cooperative one, which aims to maximize the total number of unique contents in the local network. A new arriving content is compared with the cached contents in their popularity and volume for possible replacement.

Among the few approaches based on both content popularity and content size, the most related to ours is [7], which aims to cache a maximum set of contents at the edge to maximize the hit ratio and minimize the average content provisioning costs, defined as the average price of transferring the demanded contents from their local, as well as backhaul sources, to their clients. Each content is associated with a size-weighted popularity $P' = V^{\theta} P$ with P as popularity, V as volume, and θ as a factor varying between [-2,2]. The proposed strategy stores contents having a higher P' with a larger priority. If contents have the same size, as in most existing works, then $\theta = 0$ and $P' = P$, so the most popular contents are stored at the edge. For $\theta = -1$, $P' = P/V$, so smaller contents are prioritized to larger ones. A factor of $\theta = 1$ gives $P' = VP$, leading to prioritization of larger contents, complying with our policy. If the popularity can be assumed as constant within a period, enabling its estimation, a proactive strategy pre-fetches contents in descending order of P' to the edge server until its storage capacity limit is reached. A reactive approach stores each content fetched from the backhaul, like LCE (Leave Copy Everywhere), at the edge server if it has enough capacity. Otherwise, the P' parameter of the incoming content is compared with that of the contents in the cache, to replace one or many of them that have a lower P'. The scenario, differentiating between edge and backhaul server, uses 10 000 requests for 1000 contents having a uniform size distribution between 1 and 20. The edge server cache has a capacity of 400 units, and the Zipf distribution has a skewness factor of $\alpha = 0.7$. In both proactive and reactive cases, the hit ratio reaches its maximum for $P' = \frac{P}{V}$, since the policy caches the most demanded contents at the edge due their small sizes. However, the lowest average content-provisioning cost is reached for $P' = P$ and $P' = VP$, since the most popular contents or those with the highest PV are stored at the edge. For $P' = VP$, the hit ratio sinks, because contents are cached based on volume-popularity product, not popularity.

Similar to [6] and [7], we introduce a total logistic cost metric consisting of both internal and external components, with a focus on reducing import costs. To this end, we employ the metric $P' = VP$ for optimal content placement.

III. Interaction-based Caching Based on Content Popularity and Content Volume

The demanded contents are provided to the clients either by the local or the backhaul network, giving the total volume of data transferred to clients:

$$V = V_{in} + V_{ex} \; = \; \mathbf{R}_{in}^T \mathbf{V} \; + \; \mathbf{R}_{ex}^T \mathbf{V} \tag{1}$$

with $V_i, R_{in,i}, R_{ex,i} \in \mathbb{N}$ and $i \in \mathbb{N}$, $i \in [1, C']$, where C' denotes the set of the demanded contents. V_{in} denotes the total volume of contents retrieved from internal, and V_{ex} from external sources, and $\mathbf{V}$ is the vector of all volumes of the corresponding demanded contents.

If C' contents, $c_1, c_2, \cdots, c_{C'}$ are requested, then $\mathbf{V}$ represents a column vector $\mathbf{V} = \left(V_1, V_2, \cdots, V_{C'}\right)^T$ of their corresponding sizes, and $\mathbf{R}_{in} = \left(R_{in,1}, R_{in,2}, \cdots, R_{in,C'}\right)^T$ depicts the column vector of the number of requests for the corresponding contents satisfied in the local network. Similarly, $\mathbf{R}_{ex} = \left(R_{ex,1}, R_{ex,2}, \cdots, R_{ex,C'}\right)^T$ is the column vector of the number of requests for externally retrieved contents. All elements of $\mathbf{R}_{ex}$ satisfy $R_{ex,i} \geq 1$, as a demanded content must be first imported if it is not internally available.

Requests find their desired contents either inside or outside the network, yielding their total number: $R = \sum_{i=1}^{C'} \left(R_{in,i} + R_{ex,i}\right)$. Due to this complementarity, a decrease of V_{ex} (1) leads to an increase V_{in}, since contents that are not fetched from outside the network are retrieved from internal sources.

V_{ex} can be minimized by diminishing the frequency of requests for large external contents via keeping the most popular and largest contents locally available for as long as possible. This can be achieved by leveraging the potential of IC [2], which strategically retains a selection of these contents indefinitely within the network and prolongs the accessibility of other contents in descending order of their PV metric.

To identify the optimal contents for the optimization objective, the IC, detailed in [2], first determines their relevant features and meticulously integrates them along with their

TABLE I
Couplings bins

I	g
I_1	$BC_{L-5} \leq g < BC_{L-4}$
I_2	$BC_{L-4} \leq g < BC_{L-3}$
I_3	$BC_{L-3} \leq g < BC_{L-2}$
I_4	$BC_{L-2} \leq g < BC_{L-1}$
I_5	$BC_{L-1} \leq g < BC_{L}$
I_6	$BC_{L} \leq g \leq 1$

respective weights into an aggregated quality metric, such as $q_u^{(c)} = PV$, with both weights set here to 1. It subsequently amalgamates $q_u^{(c)}$ with the attributes of the network topology, like its betweenness centrality levels L, and the properties of nodes, like their cache capacities $\hat{V}$, all within a coupling strength $g_u^{(c)}$. The coupling strength is then transformed by a function $f(\cdot)$ into the betweenness centrality horizon (BCH) of the associated contents c representing their maximum replication radius on their delivery path to their clients u: $BCH_u^{(c)} = f(g_u^{(c)})$.
The concrete expression of $f(\cdot)$ is at the discretion of the network provider and is tailored to align with its strategic objectives. For simplicity and without loss of generality, it can be specified as:

$$BCH_u^{(c)} = g_u^{(c)} \tag{2}$$

The IC replicates a content to all nodes n on its delivery path whose betweenness centrality (BC) degree lies within the BCH of the content: $BC(n) \leq BCH_u^{(c)}$. Consequently, contents with a coupling proportional to their PV metric, $g_u^{(c)} \sim PV$, become replicated to nodes with a centrality degree growing with PV. This approach saves prioritized contents with a large PV metric against eviction by limiting the diffusion of contents with a small PV factor into higher cache levels.

For strictly regulating the diffusion depth of contents, the IC divides the BC domain, $\mathrm{dom}\,(BC) \in [0,1]$, into L intervals, $I_1, I_2, \cdots, I_L$, with L as the number of discrete BC levels in the network. Each interval is defined by a range $I_l = [BC_l, BC_{l+1})$. To ensure the BCH of a content falls within a specific range I_l, i.e., $BCH_u^{(c)} \in I_l$, its coupling strength $g_u^{(c)}$ must originate, due to (2), from the same interval I_l. Thus, the L intervals I_l depict coupling strength containers, with I_1 encompassing the weakest and I_L containing the strongest couplings, as in Tab. I [3].

To provide contents with a coupling they merit, they are sorted in an array $\mathbf{C} = [c_1, c_2, \cdots, c_{C'}]$ in descending PV order and, starting from the beginning of $\mathbf{C}$, are consecutively provided with a coupling taken randomly from the container I_L under the prerequisite that the concerned caches have enough capacity. Thus, the contents' volume V_i is added to a sum $V_{sum} = \sum_{i=1}^{C'} V_i$ as long as $V_{sum} \leq \hat{V}$, with $\hat{V}$ as the unit cache size. Otherwise, V_{sum} is reset, and the couplings of the following contents are taken randomly from the subsequent weaker interval I_{L-1}.

The caching performance can be slightly improved by fully utilizing cache capacities through the 0/1 Knapsack optimization, which combinatorially maximizes the content quantity $N = \sum_{i=1}^{C'} \delta_i$ in each cache while adhering to constraint $\sum_{i=1}^{C'} \delta_i V_i \leq \hat{V}$, with $\delta_i = \{0, 1\}$ denoting whether the content c_i is inserted into the cache or not, and C' as the number of demanded contents from a library with a total of C contents. In this case, caches are first populated with prioritized contents sequentially taken from the PV array, however, their remaining capacity is then filled with a less-prioritized content from a random position in the array. For more clarity in analyzing the impacts of the algorithm on the caching performance, we do not utilize this method.

Generally, an R_i times transfer of a content c_i with the volume V_i over a distance $d_i \in \mathbb{N}$ to its clients leads to a logistic transport expense of:

$$E_i = \kappa \; R_i \, V_i \, d_i \tag{3}$$

with $\kappa \in \mathbb{R}^+$ as provisioning cost factor. If the content is retrieved from internal sources, the logistic cost is:

$$E_{in,i} = \kappa_{in} \; R_{in,i} \, V_i \; d_{in,i} \tag{4}$$

with $\kappa_{in} \in \mathbb{R}^+$ as internal provisioning cost factor. And if the contents is fetched from outside:

$$E_{ex,i} = R_{ex,i} \, V_i \left[\kappa_{in} \, D + \kappa_{ex} \, (d_i - D) \right] \tag{5}$$

where the first component depicts the transport cost inside of the network with the diameter $D \in \mathbb{N}^+$, and the second one that of the outside. The total logistic expense is thus:

$$E = \sum_{i=1}^{C'} \left(E_{in,i} + E_{ex,i} \right) \tag{6}$$

The external costs can be minimized by saving contents with the highest PV at nearest positions, $d_i \to 1$, to their clients for a maximum period of time. A minimum distance $d = D + 1$ yields the minimum external logistic cost:

$$E_{ex,min} = \left(\kappa_{in} \, D + \kappa_{ex} \right) \sum_{i=1}^{C'} R_{ex,i} \, V_i \; = \; \left(\kappa_{in} \, D + \kappa_{ex} \right) V_{ex} \tag{7}$$

To minimize $E_{ex,min}$, the IC reduces the total volume $\sum_{i=1}^{C'} R_{ex,i} \, V_i$ of imported data from exterior sources.

IV. Numerical Performance Analysis

A. Simulation Settings

To evaluate the caching performance, a network topology consisting of a binary tree of height 7 with 128 clients at the leaves, 122 intermediate caches, and a content source at the root (Fig. 1) is composed. The cache nodes depict the local network with a diameter of $D = 6$, and the source represents the entire backhaul network collapsed into a single node.

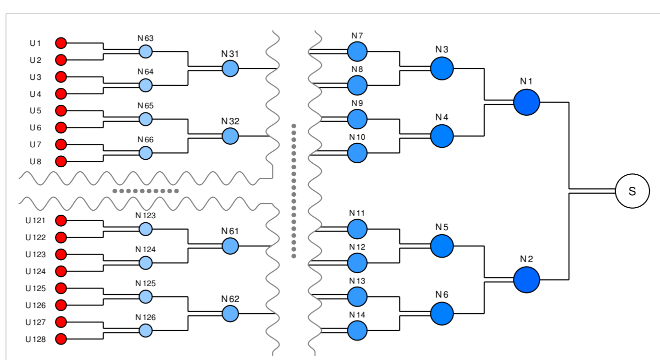

Fig. 1. Network topology with users on the left (red) demanding contents from intermediate caches (blue) and content server S on the right.

Trees possess various advantages for investigating the effects of caching strategies on request propagation and content distribution in networks, rather than system resilience, routing, or congestion. Their structure, lacking multipaths and loops, allows for isolating caching effects from topological complexities. Trees can approximaate large networks on a smaller scale by combining core- and edge-type topologies [8]. Further, complex networks can be regarded as a superposition of trees with common intermediate nodes, allowing for breaking down optimization problems into simpler subproblems, solvable recursively following Bellman's optimality principle [9], [10].

The clients send $R = 12\,800\,000$ Zipf distributed requests with factor $\alpha = 0.8$ for $C = 12\,800$ contents in the library. Half of the requests are used for caching warm-up. The size of the contents ranges from 1 to 1000 units, which may correspond to files of, e. g., 1 MB to 1 GB, or video contents of 10 MB to 10 GB. The average content volume is $\bar{V} = 500.5$ units, which yields with C an average library size of 6 406 400 units. The universal cache size $\hat{V}$ is set to 1% of the library size, i. e., $\hat{V} = 64\,064$ units. Larger cache capacities generally enhance the performance. For the content volumes, we assume a truncated Gaussian distribution with $\mu = 500.5$ and a standard deviation of $\sigma = \mu/4 = 125.125$. We define the transformation function as $BCH_u^{(c)} = g_u^{(c)}$.
Performance evaluations were conducted in a Python simulator, featuring a hierarchical LRU cache network and IC protocol. Table II summarizes the set-up parameters.

B. Performance Evaluation

We repeatedly simulated the scenario to enhance statistical certainty and averaged the resulting metrics, observing notably tight confidence intervals in the performance measurements.

The IC exploits the variation of popularity and volume of contents to increase the caching performance. With progressing equalization of them, the caching efficiency decreases toward that of LCE, which ignores them. Reducing the Zipf parameter α gradually equalizes the demand for contents, so that with $\alpha \to 0$ the PV distribution becomes increasingly dominated by volume distribution. However, the steep exponential decrease of the popularity over content rank causes the popularity distribution to dominate the PV distribution.

Consider Fig. 2 showing the average popularity $\bar{P}$, mean volume $\bar{V}$, and $\bar{PV}$ of the contents stored at different cache levels for a simulation run. Note that contents, which can diffuse into the caches at level l' also diffuse into caches at lower levels $l < l'$ due to IC's replication constraint $BC(n) \leq BCH_u^{(c)}$. Further, all caches, except those at the highest level $l = L$, are exposed to eviction. The rapid decrease of the average values $\bar{P}$, $\bar{V}$ and $\bar{PV}$ from the level-2 to the level-1 caches is caused by the abolition of the caching restriction for the level-1 caches, allowing all contents to diffuse into them. The non-monotonic increase in the average content volume shows that the IC does not consider the contents' volume as the only factor; otherwise the largest contents would be stored in the highest-level caches at $l = 6$. However, the image at the bottom of Fig. 2 manifests how the higher-level caches are filled with contents with a higher $\bar{PV}$ factor.

With falling α, the PV distribution becomes more dominated by V. Hence, the IC saves more of the largest contents in the higher-level caches, so the average volume $\bar{V}$ of cached contents rises in the corresponding nodes (Fig. 3). Conversely, as α grows, it amplifies the influence of the popularity distribution, leading to the ascent of smaller yet more popular contents to higher-level nodes, reducing their average volumes. Consequently, the logistic cost E_{ex} for the import of contents from external sources rises with growing α.

TABLE II
SIMULATION SETUP

Tree topology	branching b	2
	height h	7
Number of users	U	128
Number of caches	N	126
Number of edge caches	N_1	64
Cache size s	$\hat{V}$	64 064
Library size	C	12 800
Content volume	V	$1 \cdots 1000$
Mean content volume	μ	500.5
Standard deviation of content volume	σ	$\mu/4$
Number of requests	R	12 800 000
Number of warm-up requests		$R/2$
Zipf parameter	α	0.8
Transformation function		$BCH = g$
Cache strategy	LCE, IC	
Content replacement strategy	LRU	

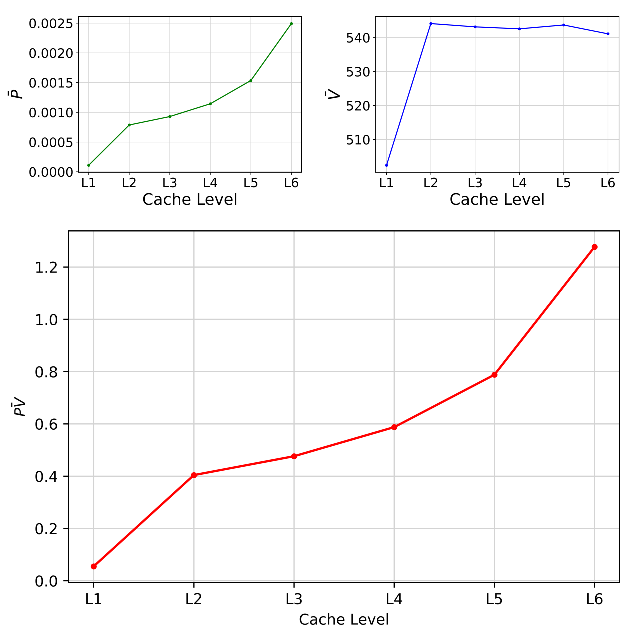

Fig. 2. Average content popularity $\bar{P}$ (above left), content volume $\bar{V}$ (above right), and $\bar{PV}$ (bottom) over the 6 different cache levels in the network.

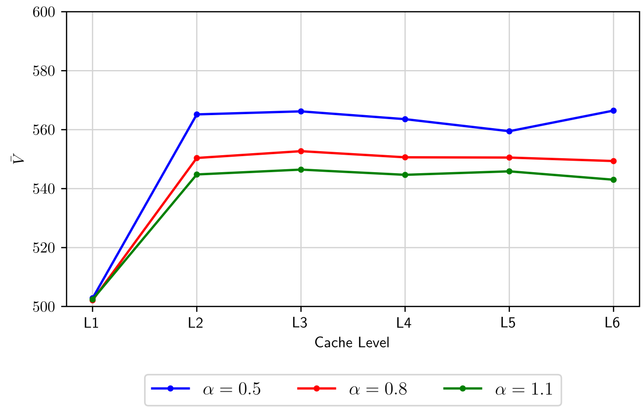

Fig. 3. Average volume $\bar{V}$ of contents cached at different cache levels

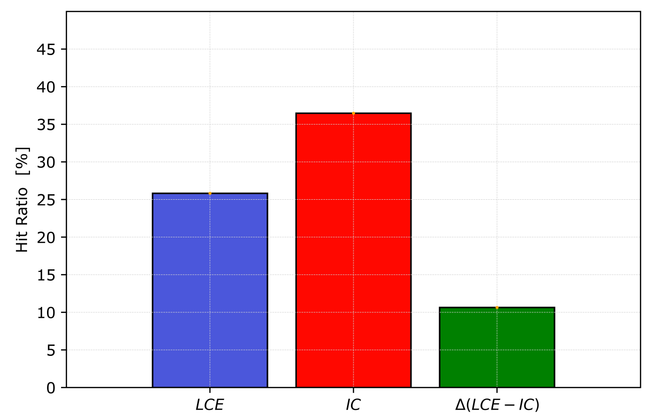

Fig. 4. Hit ratio of LCE and IC and the IC-gain

The integration of the proposed strategy in the IC significantly outperforms LCE in terms of cache hit ratio. It achieves an average cache hit ratio of $36.4775\% \pm 0.0321\%$, whereas the LCE reaches $25.8338\% \pm 0.0222\%$, yielding an IC gain of approximately $10.6437\% \pm 0.0324\%$ (Fig. 4).

Incorporating content volume V alongside popularity P slightly reduces hit performance by about 0.828%, as per scenarios only considering P [3], which aligns with expectations given P's dominance over the PV distribution at $\alpha = 0.8$. Thus, the joint optimization of caching with both P and V produces benefits without substantially affecting the hit rates.

The IC reduces the average total import volume $\bar{V}_{ex}$ by:

$$\Delta\bar{V}_{ex} = \bar{V}_{ex}^{LCE} - \bar{V}_{ex}^{IC} = 361\,044\,900.2 \pm 901593.1455 \quad (8)$$

relative to LCE, equal to about $721\,368.432$ less of the $4\,768\,213.36$ average-size contents imported by LCE, i.e., 15.1287% less average-size contents (Fig. 5). It, thus, decreases the ratio of the average total data volume $\bar{V}_{ex}$ fetched externally to the internally transferred data volume $\bar{V}_{in}$ from $\bar{V}_{ex}^{LCE}/\bar{V}_{in}^{LCE} = 2.8846$ to $\bar{V}_{ex}^{IC}/\bar{V}_{in}^{IC} = 1.7044$, yielding a reduction of 1.1802 (Fig. 6).

TABLE III
SIMULATION RESULTS

	LCE	IC	Δ(LCE, IC)
$\bar{h}$	25.8338%	36.4775%	**-10.6437%**
$\bar{V}_{in}$	827318637.0	1188372454.4	-361053817.4
$\bar{V}_{ex}$	2386490785.8	2025445885.6	361044900.2
$\bar{V}$	3213809422.8	3213818340.0	-8917.2
$\frac{\bar{V}_{ex}}{\bar{V}_{in}}$	2.8846	1.7044	**1.1802**
$\bar{E}_{in}$	1660633368.8	2859632208.6	-119899883980.0
$\bar{E}_{ex,min}$	16705435500.6	14178121199.2	2527314301.4
$\bar{E}$	18366068869.4	17037753407.8	1328315461.6
$\frac{\bar{E}_{ex,min}}{\bar{E}_{in}}$	10.0597	4.9580	**5.1017**

The average volume of all demanded contents is in both cases almost equal: $\bar{V}^{LCE} \approx \bar{V}^{IC}$. Their negligible difference of 8917.2 is due to statistical fluctuations of requested contents.

The general average external logistic cost is defined by:

$$\bar{E}_{ex} = \left(\kappa_{in}\, D + \kappa_{ex}\, \bar{x}\right) \bar{V}_{ex} \quad (9)$$

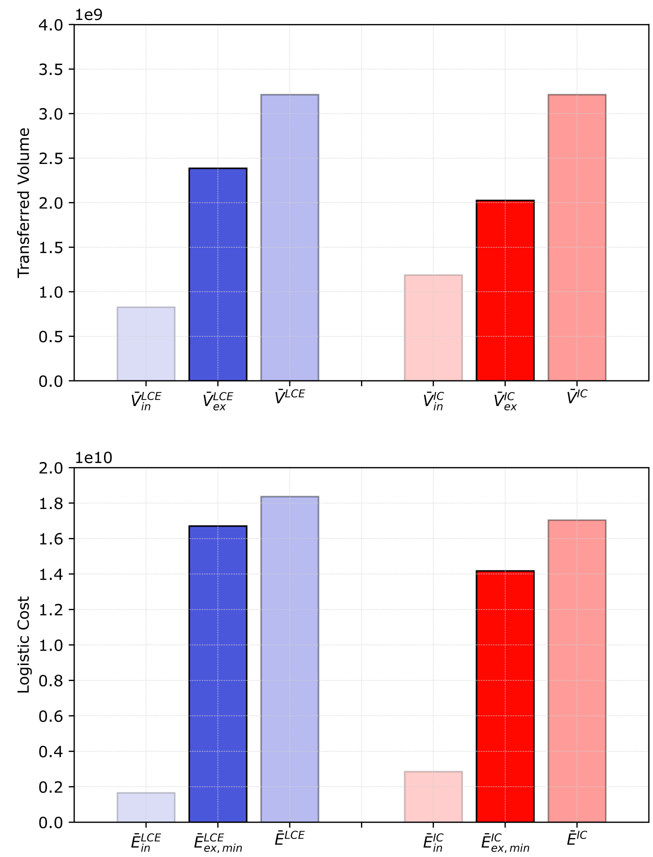

Fig. 5. Top: Average total content volume transferred to clients by LCE and IC from internal and external sources, and their sum (above), and the corresponding average logistic costs (bottom).

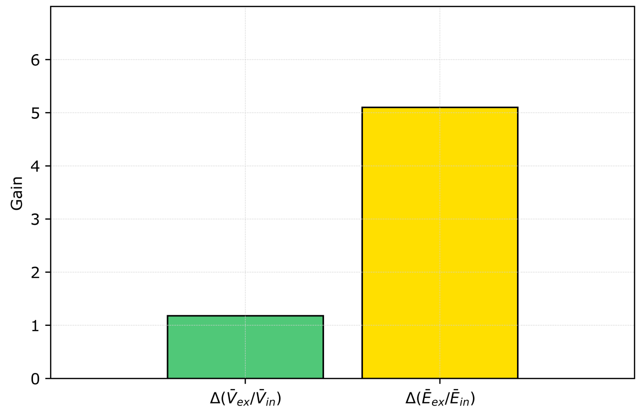

Fig. 6. Gains of IC relative to LCE concerning ratios between average external and internal data transfers, and the associated total logistic expenses.

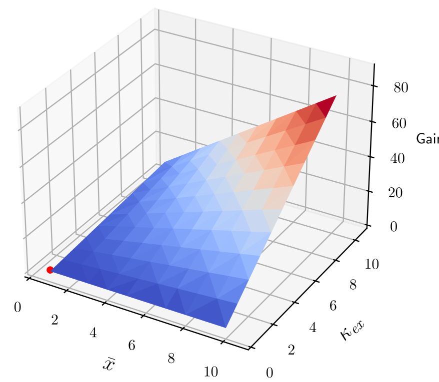

Fig. 7. External logistic cost gains $\Delta(\bar{E}_{ex}/\bar{E}_{in})$ of IC as a function of average distance $\bar{x}$ of external sources from the network and external cost factor κ_{ex}. The minimum gain of 5.1017 is at $\bar{x} = 1$ and $\kappa_{ex} = 1$ (red circle).

where $\bar{d} = D + \bar{x}$ is the average distance of external sources from clients, and $\bar{x} \in \mathbb{N}^+$ is the average distance of the content locations from the local network. The ratio of the average external to internal logistic cost is thus:

$$\frac{\bar{E}_{ex}}{\bar{E}_{in}} = \kappa_{in}\, D\, \frac{\bar{V}_{ex}}{\bar{E}_{in}} + \kappa_{ex}\, \bar{x}\, \frac{\bar{V}_{ex}}{\bar{E}_{in}} \tag{10}$$

IC's gain in costs relative to LCE can be written as:

$$\Delta \frac{\bar{E}_{ex}}{\bar{E}_{in}} = \kappa_{in} D \left(\frac{\bar{V}_{ex}^{LCE}}{\bar{E}_{in}^{LCE}} - \frac{\bar{V}_{ex}^{IC}}{\bar{E}_{in}^{IC}} \right) + \kappa_{ex} \left(\frac{\bar{V}_{ex}^{LCE}}{\bar{E}_{in}^{LCE}} - \frac{\bar{V}_{ex}^{IC}}{\bar{E}_{in}^{IC}} \right) \bar{x} \tag{11}$$

Since in the IC strategy, contents with the largest PV-factor are retrieved from inside, instead of outside the network, the average volume $\bar{V}_{ex}$ imported externally sinks, and the average amount of data $\bar{V}_{in}$ transferred internally rises. Consequently, the average external cost $\bar{E}_{ex}$ decreases and the average internal expense $\bar{E}_{in}$ increases, such that $\bar{V}_{ex}^{IC}/\bar{E}_{in}^{IC} < \bar{V}_{ex}^{LCE}/\bar{E}_{in}^{LCE}$.

The gain $\Delta(\bar{E}_{ex}/\bar{E}_{in})$ further grows with the external data provisioning cost factor κ_{ex}, and the average distance $\bar{x}$ of the content sources from the local network.

Assuming $\kappa_{ex} = \kappa_{in} = 1$ and $D = 6$, the minimum external costs are reached when the content sources are, on average, just $\bar{x} = 1$ hop away from the local network, giving:

$$\Delta \frac{\bar{E}_{ex}}{\bar{E}_{in}} = 7 \left(\frac{\bar{V}_{ex}^{LCE}}{\bar{E}_{in}^{LCE}} - \frac{\bar{V}_{ex}^{IC}}{\bar{E}_{in}^{IC}} \right) = 5.1017 \tag{12}$$

Thus, by reducing the relative external data import $\bar{V}_{ex}/\bar{V}_{in}$ by 1.1802, the IC decreases the relative external logistic costs $\bar{E}_{ex}/\bar{E}_{in}$ by more than 5.1 times.

As κ_{ex} and $\bar{x}$ contribute linearly to the expense gains (11), the benefits grow linearly with them. Figure 7 shows the gains over κ_{ex} for $0 \leq \kappa_{ex} \leq 10$, and $\bar{x}$ for $1 \leq \bar{x} \leq 10$, with the minimum at $\kappa_{ex} = 0$ (no external costs) and $\bar{x} = 1$.

V. Conclusion

To reduce data import volumes and logistics costs, it is crucial to hold contents at more central nodes for longer durations proportional to their popularity-volume (PV) factor. This strategy also enhances the user experience by reducing bandwidth usage, access latency, and network congestion. Regulating the coupling allocation decreases cache pollution and content eviction while increasing local access to the highly demanded largest contents. Integrating more features of caching parties into the IC can further improve its performance. Our future research will explore the IC's potential in reducing content access delays, crucial for 5G/6G's URLLC and eMBB services.

References

[1] Ericsson, "Ericsson Mobility Report, June 2023," https://www.ericsson.com/49dd9d/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-june-2023.pdf, 2023, [Online; accessed 19-July-2008].

[2] F. Derakhshan and A. Timm-Giel, "Interaction-based Caching in Content-Centric Networking," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022, pp. 1–6.

[3] ——, "Optimizing the Performance of the Interaction-based Caching in CCN," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 2104–2109.

[4] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: moving from cloud to edge," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 36–42, 2016.

[5] M. Bilal and S.-G. Kang, "Time Aware Least Recent Used (TLRU) cache management policy in ICN," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 528–532.

[6] M. Taghizadeh, K. Micinski, S. Biswas, C. Ofria, and E. Torng, "Distributed Cooperative Caching in Social Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1037–1053, 2013.

[7] Q. Li, W. Shi, Y. Xiao, X. Ge, and A. Pandharipande, "Content Size-Aware Edge Caching: A Size-Weighted Popularity-Based Approach," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206–212.

[8] J. Pfender, A. Valera, and W. K. Seah, "Easy as ABC: A Lightweight Centrality-Based Caching Strategy for Information-Centric IoT," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, ser. ICN '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 100–111. [Online]. Available: https://doi.org/10.1145/3357150.3357405

[9] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. The MIT Press, 2022.