Energy-Aware Caching at the Wireless Network Edge for Information-Centric Operation

Gürkan Gür

SATLAB
Dept. of Computer Eng., Bogazici University
Bebek, 34342 Istanbul, Turkey

Abstract

Information-centric networking (ICN) has been proposed as a paradigm for overcoming the networking challenges of the current Internet involving the explosion of content consumption and the widening gap between traffic and capacity growth. Although new communication systems and protocols are deployed in the field to meet the broadband traffic surge, the ubiquitous proliferation of mobile broadband access and advanced user devices outpaces implemented countermeasures and aggravates this capacity crunch issue. Moreover, the expanding networking infrastructure is expected to be more energy-efficient conforming to "green communications" concept while serving burgeoning traffic demands. In this paper, we study the application of caching mechanisms to the edge of an infrastructure-based mobile network supporting ICN and explore their impact on the energy consumption of the investigated system. We devise a greedy heuristic cache management strategy for this setting and evaluate its performance. Our scheme incorporates energy reward, popularity, Time-To-Live (TTL) and delay (i.e. chunk loss due to delay sensitivity) factors and provides energy savings with low-complexity operation.

Keywords: energy efficiency, network caching, information-centric networking, wireless content delivery

1. Introduction

The Internet usage has drastically evolved from a point-to-point communication and exchange paradigm to a content dissemination and retrieval context. This circumstance has necessitated a more content-centric rather than a host-centric design. Information-centric networking (ICN) builds on this premise to overcome the shortcomings of address based routing/operation in the emerging era of pervasive and ubiquitous networking. ICN identifies content rather than network locations enabling the addressing schemes facilitated by application-level/social considerations. The incumbent design factor of resource sharing for the conventional IP is translated to a requirement for more service- and content-oriented operation [1].

Another emerging condition is the mobile broadband explosion propelled with new services and content available anytime-anywhere. Hence, the upcoming broadband wireless standards are putting a bigger burden on mobile networks for serving the Internet traffic surge. Moreover, this diverse range of services and modalities bring forth new players and factors such as OTT (Over-The-Top) service providers (e.g. Netflix, Skype and YouTube) and P2P (Peer-to-Peer)-based content sharing, which heavily tax the network resources. However, the network operators generally cannot charge for these high bandwidth services while the network resources are stretched to provide adequate QoS levels. Additionally, administrative partitioning of networks among content and network providers impedes cooperation leading to lack of optimization on the end-to-end path. Therefore, countermeasures and remedies are crucial to mitigate these problems in next-generation IP networks. Information-centric operation is posed as a vital apparatus towards this goal.

In this work, we consider an infrastructure-based wireless network which utilizes ICN paradigm for content (or information) based networking. Caching is a fundamental capability for ICN systems in order to enable scalable and cost-efficient content dissemination [1]. We elaborate on this aspect and propose an energy-aware cache replacement mechanism for improving the system performance. Energy efficiency (EE) at each network component has become more critical with the dwindling energy supplies and the deepening environmental issues. Accordingly, it is paramount to devise widely-applicable algorithms and solutions for energy-efficient network operation [2]. Although the adoption of information-centric approach for network architecture has the potential for enabling energy-efficient content dissemination, this new approach has to be energy-efficient in addition to being an energy-efficiency enabler [3]. Therefore, our main focus as the performance objective is EE in this work. We develop our proposed mechanism considering the prominent factors on caching from the perspective of EE and low complexity.

The intersection of ICN and wireless networks are yet to be explored comprehensively, especially for the prospective 5G systems. Similarly, caching has been typically studied for ad

Email address: gurgurka@boun.edu.tr (Gürkan Gür)

hoc wireless systems and usually dealing with performance metrics other than EE. In that regard, the contributions of our work are as follows:

1. We propose a heuristic cache management scheme for energy-efficient operation of ICN in wireless content dissemination.
2. We devise a system model for caching at the edge of infrastructure-based mobile networks.This model is focused on EE analysis. However, it can be extended for other analytical purposes.
3. We investigate the effect of caching on the energy consumption of these systems. We present multifaceted experimental results on the interplay between different factors such as cache size, object size composition and popularity distribution in this setting.

In the next section, caching for wireless networks is described with a brief overview of related issues. Section 3 presents related work in the literature. In Section 4, we present the system model and system requirements. We also describe *Energy Aware Caching for Wireless ICN (ENACI)* which is a greedy algorithm for energy-efficient cache management problem for this setting. In Section 5, the experimental results are discussed for performance evaluation. Finally, we draw conclusions in Section 6 with a perspective on potential research directions.

2. ICN and Caching at the Edge of Infrastructure-Based Wireless Networks

The challenges faced by the current Internet architecture have led to numerous proposals for Future Internet protocols and architectures. The explosion of video and P2P traffic are among the prominent driving factors in these efforts. Although application-layer solutions, namely CDNs (Content Delivery Networks), P2P overlays and HTTP proxies, have already been deployed through the current Internet ecosystem, more substantial architectural changes are evident. For instance, CDNs are generally effective in shortening transport paths resulting in smaller delays and better throughput [4]. But the deployment cost of CDNs and scalability issues are also prevalent [5]. Therefore, research projects such as SAIL, PSIRP, COMET and 4WARD have proposed various networking models to realize Future Internet concept [6]. ICN has been an active field in that regard with related efforts and proposals such as Data Oriented Network Architecture (DONA), Content-Centric Networking (CCN), and Publish/Subscribe Internet (PURSUIT) [7].

An example ICN network is shown in Figure 1. The ICN approach implies context resolution/service resolution instead of machine resolution [8]. Receiver-driven model and caching are two salient features of ICN. Clearly, this approach benefits the delivery of popular content (e.g. reduced delivery delay) and reduces resource requirements (e.g. bandwidth and server load) in the network [9]. Thus, the loose coupling between content and its originator provides opportunities to facilitate mechanisms for many of the prevalent issues with the current network architecture such as multicast, multipath routing and mobility [10]. However, there are also substantial obstacles such as security and object naming/identification against ICN proliferation. Simply, the content has to be secure with proper confidentiality and credentials while being uniquely identifiable among a huge number of objects present in the network. Deployment of caches and in-network storage at different points in the interconnection, backbone and aggregation levels of a network is critical to improve the system performance for content consumption in ICN[11]. According to [4], main trends and forecasts regarding the Internet traffic envisage the doubling of global IP traffic through 2013 with video becoming the major source of traffic (video traffic will account more than 90% of consumer traffic in 2013). Meanwhile, P2P traffic is expected to grow in volume, however with a decreasing percentage on the overall. ICN provides a pervasive storage infrastructure enabling efficient utilization of network resources in a flexible manner to serve that end. For CSPs (communication service provider) providing mobile broadband services, there is the P2P stress on their infrastructure due to the overlay distribution of content in their networks. In-network storage alleviates this issue especially addressing file sharing and streaming services. However, the energy consumption of these capabilities have to be optimized for enabling green communications and networking.

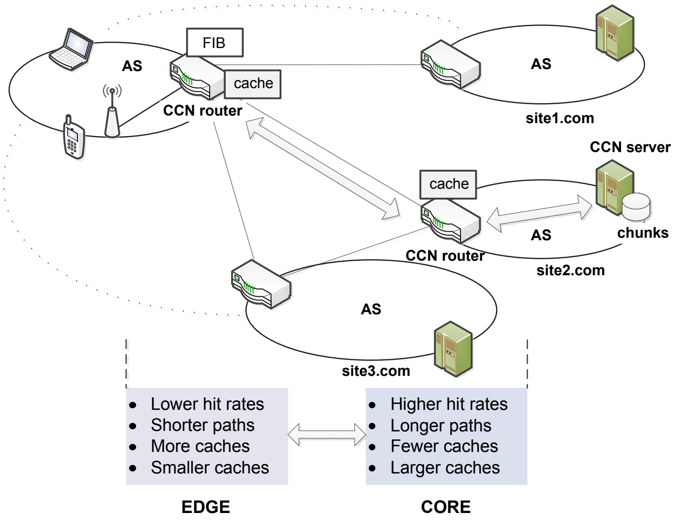


Figure 1: An ICN example considered in CCN (Content-Centric Networking) proposal (AS: Autonomous System, FIB: Forwarding Information Base).

Why caching closer to the edge (smaller localized caches) is important? The tradeoffs related to cache location in an IP network are shown in Figure 1. Caching closer to the content consumer shortens the transmission path providing significant load reduction and smaller latency in the network. The hop count reduction during content dissemination implicitly provides substantial energy savings. Also, for CSPs, there is some popular content which is already identified by the system managers such as operator's web portals. These services will also benefit from caching at the edge, leading to a significant saving on traffic propagation into the core and aggregation networks. This multifactor issue has been explored with different

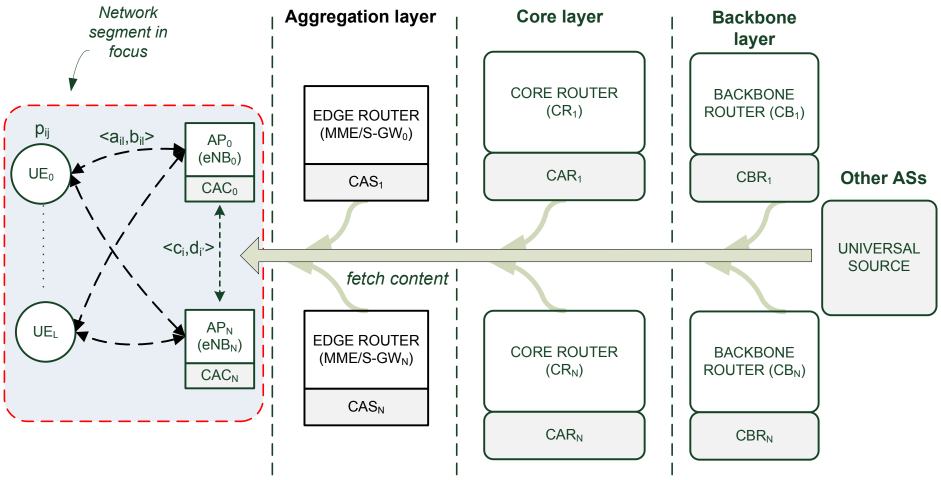


Figure 2: The investigated network setup for ICN in wireless network. The network nodes employ caches (CXY_i's) enabling in-network storage. The node instances closer to the edge follow the LTE terminology (provided in parantheses) but can be any wireless cellular broadband technology with assumed capabilities.

approaches in the literature [8].

In general, the ultimate edge of the network is the networked user equipments. However, user centric approaches (networked caching at the user devices) suffer scalability and feasibility difficulties due to the requirement of software and hardware support among disparate entities managed by individual users. In a wireless cellular network, Access Points (APs) exhibit both advantages of spatial configuration and economies of scale due to centralization, and may act as conjoint devices for storing and forwarding information in addition to their traditional capabilities. Thus, we do not consider the effect of user-resident caching in our analysis but focus on APs [12]. Although caching capability in this type of nodes is not practically available in current wireless cellular networks, it is expected to be feasible with the emergence of 5G networks such as LTE Advanced where wireless APs act as IP routers in addition to radio functionalities [13]. Although spatial and temporal spectrum reuse will substantially improve with these new systems, it is expected that the network backhaul will become a major system bottleneck and necessitate edge-caching in these elements for the exploitation of the inherent spatial and temporal redundancy of user demands [14].

In this work, we elaborate on caching at APs in the network layout as shown in Figure 2. *AP* is a generic term which represents the access point node for a communication device to attach and communicate through the network wirelessly. For instance, this node corresponds to the base station (BS) in GSM, wireless access point (WAP) for IEEE 802.11 or eNB/Home-eNB in LTE standards. We develop a centralized decision logic for cache replacement, i.e. which chunks to keep and which to evict from the cache, based on a greedy heuristic algorithm for this setting and evaluate its performance. Our scheme incorporates Time-To-Live (TTL) and delay (i.e. chunk loss due to delay sensitivity) factors on the energy consumption of the investigated system to improve EE.

3. Related Work

Content cache management has been an important topic especially for ad hoc networks [15]. For instance, a cooperative cache-based content dissemination framework to carry out the cooperative soliciting and caching strategies for the two encountering nodes in a mobile ad hoc networks is proposed in [16]. In [17], Fiore *et al.* focus on diverse cache features, specifically different cache sizes, and design a content replacement strategy for intermittent and peer-to-peer information exchange.

Information dissemination has been under focus especially for wireless sensor networks. Dimokas *et al.* present a cooperative caching solution particularly suitable for wireless multimedia sensor networks (WMSNs) in [18]. The proposed caching solution exploits sensor nodes which reside in positions of the network that allow them to forward packets or communicate decisions within short latency. In [19], optimizations for processing queries by using adaptive caching structures in WSN are discussed and an approximative update policy is presented. In the wired domain, CDNs have been a driving factor to utilize caching for content delivery. CDNs are characterized by robustness in serving huge amounts of requests and content volumes [20]. Caching can be managed using various mechanisms in CDNs. In [21], Chen et al. use an application level multicast tree as a cache replacement policy for each CDN surrogate server. Presti *et al.* [22] determine the caching benefit of content replicas by a non-linear integer programming formulation whereas Bartolini *et al.* decide whether to add a new content replica or remove an existing one using a semi-Markov decision process (SMDP)[23]. Caching can also be affected by the temporal popularity patterns of the contents. In that regard, Famaey *et al.* focus on a cache replacement strategy for multimedia content and presents a generic popularity prediction algorithm which fits a set of functions to the cumulative request pattern of content items [24]. It uses the predicted request patterns to determine the subset of all available content to store in the cache. The emergence of cloud-based storage and green datacenters concept have also supported these endeavors [25]. However, the intersection of wired and wireless networks has been relatively unexplored from the energy-efficient caching perspective.

4. System Model

In this section, we layout the system model analytically and setup the optimization problem for cache management from the perspective mentioned above. The system parameters and notation are shown in Table 1. Throughout the paper, we may restate the definitions for the sake of clarity.

The network setup is shown in Figure 2. The wireless network has an access segment with an aggregation stage and a core part interconnected to other IP networks through a backbone network. For the sake of simplicity, "Universal Source" is a logical shorthand representation for chunk stores/servers in the rest of Internet. As noted in Section 2, we focus on the access segment of the wireless network. There are L UEs consuming content represented as chunks o_j of size s_{oj} through N APs in the system. These distinct objects constitute the global set of chunks I. Each AP_i has a local cache with size M_i containing the set of cached chunks C_i. During content consumption, a chunk not present in the local caches is fetched from the universal source with an average hop-count ($N_{\{br,cr,er\}}$) and energy cost per hop ($e_{\{br,cr,er\}}$) through the network segments resulting in e^{ik}_{tr}. This is a prudent assumption since we do not distinguish among the chunks in the system according to their location in the network or their respective stores. For an energy-aware/-efficient cache management scheme, the network nodes should be aware of transport cost among themselves. This is feasible via information sharing due to relatively static nature of the infrastructure. For that purpose, the energy cost of signaling between two network nodes, node i and node k, is also present, which is usually dependent on their distance: $e^{ik}_{req} = f(d_{ik})$. The consumption figures e^{ik}_{tr}, e^{ik}_{req} and e^{i}_{pr} (energy cost for cache replacement) contribute to the energy cost of fetching from another node's (*node k*) cache to AP_i, namely e^{ik}_{g} which is related to processing and transmission cost for transport of chunks. If o_j is already available (cached) at AP_i, e_g value is 0. Moreover, the energy cost of caching per unit chunk in AP_i (e^{i}_{c}) and the energy cost of transmission per unit chunk between AP_i and UE_l (e^{il}_{f}) are accumulated into the total energy consumption for content o_j access at AP_i, denoted as E_{ij}. At the expense of this energy dissipation, the cache management strategy yields a cache hit ratio h_i for AP_i.

For quasi-stationary or fixed access probabilities for content, a near-optimum strategy policy for EE is to estimate the probabilities via profiling and to keep the most popular items in the cache considering the energy cost as well [26]. This is more evident for dense user populations with correlated access characters such as enterprise users in business districts. These users follow a relatively packed access profile in time, space and content dimensions. In individual user-centric caching, this is inefficient since each node (user equipment) has to cache the similar content. However, this correlation can lead to higher efficiency for network-based local caches adapted to user communities [4]. In this work, we assume a fixed/slowly varying

Table 1: System parameters.

Parameter	Explanation
o_j	A distinct chunk, $o_j \in I$, $j = 1, ..., I$.
s_{oj}	The size of chunk o_j, $s_{oj} \in \mathbb{S}$, set of chunk sizes
N	The number of access points (AP_i) operating in the zone
M_i	The cache size of AP_i
C_i	The set of cached chunks in AP_i, $\sum s_{(o_j \in C_i)} \leq M_i$
I	The set of chunks (distinct objects) in the system
L	The number of user equipments (UEs) in the area
p_{ij}	The request probability of item (chunk) j through AP_i
d_{ij}	The distance between AP_i and o_j. If o_j is already available (cached) at AP_i, this quantity is 0.
e^{ik}_{g}	Energy cost of fetching per unit chunk from another node's (*node k*) cache to AP_i. This quantity is related to processing and transmission cost for transport of chunks among network nodes. If o_j is already available (cached) at AP_i, its value is 0.
e^{i}_{c}	Energy cost of caching per unit chunk in AP_i
e^{il}_{f}	Energy cost of transmission per unit chunk between AP_i and UE_l
e^{ik}_{req}	Energy cost of signalling between AP_i and *node k* for cache coordination and information sharing
e^{ik}_{tr}	Energy cost of transmission per unit chunk between AP_i and *node k*
e^{i}_{pr}	Energy cost of processing per unit chunk for cache replacement at AP_i
$\widehat{C_i}$	The set of decached (removed from cache) chunk(s) due to incoming request in AP_i
h_i	The cache hit ratio for AP_i
$N_{\{br,cr,er\}}$	Number of {*backbone, core, edge*} routers on the path for fetching an object
$e_{\{br,cr,er\}}$	Energy consumption per unit chunk on a {*backbone, core, edge*} router on the path for fetching an object

access pattern for the network users. The chunk request probabilities, p_{ij}'s, are estimated through a sliding window mechanism. For a specific AP_i, the sliding window scheme records the time gap ΔT between the current time and Kth most recent reference for a chunk j and calculates $p_j = K/\Delta T$ as the estimated values utilized by the relevant algorithms.

4.1. General Assumptions

The key assumption is that we consider a receiver-driven chunk-based ICN where the content is stored and identified as uniquely identifiable chunks (segments). These chunks are transported at chunk level with built-in network storage for caching. We do not make any assumptions on specific naming or content-based routing mechanisms employed in ICN [10]. We assume necessary mechanisms such as FIB caching and hierarchical deployment are already employed in the routing nodes. Moreover, the Zipf-nature of the Web content consumption implies that for a relatively small period of time, a limited set of ICN is utilized by an FIB to forward flows of interest messages [27], which helps the scalability issue.

As noted in Section 2, we do not consider the caching at the wireless user nodes, rather focus on the network-centric optimization in the infrastructure-based wireless network, especially next-generation wireless networks such as LTE Advanced. The main reason is the emerging capability due to the standardization of wireless acccess points acting as IP routers for these systems. Figure 2 provides LTE naming of nodes in addition to general node types such as core router for exemplifying the mapping for a next-generation wireless system.

We assume that the cache operates in a *weakly-consistent* manner, i.e. the applications can be served *stale* data from the cache occasionally [28, 29]. The consistency mechanism employs Time-To-Live (TTL) monitoring in order to invalidate stale chunks in the cache.

4.2. Energy Consumption Model and Reward Structure

We focus on a single of AP, namely AP_0, in a so-called *cache zone* for optimization purposes. Therefore, we can drop the indice i identifying the specific AP. The energy consumption for content o_j access at AP_i, E_{ij}, is constituted of three main components, namely e_c, e_f and e_g:

$$E_{0j} = E_j = \begin{cases} e_c(o_j) + e_f(o_j) & \text{if } d_{ij} = 0 \\ e_f(o_j) + e_g(o_j) & \text{otherwise} \end{cases} \tag{1}$$

where e_c is the cost of caching and locally serving the chunk, e_f is the energy cost for wireless transmission of the chunk from AP_0 to the requester user equipment and e_g is the energy cost of fetching the chunk from other caches or the source. The quantity $e_c(o_j)$ is equal to $s_{o_j} \cdot e_c$ while $e_f(o_j)$ is a function of wireless transmission power, transceiver circuitry consumption and channel conditions between AP and UE. The last quantity $e_g(o_j)$ in (1) is written as

$$\begin{aligned} e_g(o_j) &= e_{req}(o_j) + e_{tr}(o_j) + e_{pr}(o_j) & (2)\\ &= e_{req} + e_{tr}(o_j) + e_{pr} \cdot s_{o_j} & (3) \end{aligned}$$

Table 2: Energy consumption figures for the wired segment [3].

Node Type	Absolute Consumption (W/Gbps)	Normalized Consumption (unit energy per unit data)	Hop count
Backbone router	15	1	5
Core router	28.6	1.9	6
Edge router	80	5.3	1

For $e_g(o_j)$, $e_{tr}(o_j)$ component is based on the availability of the chunk in the network as a cached item or from the original publisher. The chunk is assumed to be located by the ICN infrastructure and transmitted over the wired network to the serving AP. For that case, we employ the trace-based analysis of chunk propagation in [3] for hop-count estimation and related power consumption through the transmission over the wired core and edge networks. The energy consumption figures for various network nodes given in [3] are shown in Table 2. Then

$$e_{tr}(o_j) = s_{o_j} \cdot [N_{br} \cdot e_{br} + N_{cr} \cdot e_{cr} + N_{er} \cdot e_{er}] \tag{4}$$

which is basically dependent on the distance between AP and the chunk's location, i.e. the hop count and the hop types over the route.

In the above analysis, we also dropped ik identification since we assume $e_x^{ik} = e_x$ for all i, k values. Then the expected total energy consumption at AP_0 is simply

$$E_{tot} = \sum_{j=1}^{|I|} p_j \cdot E_j \tag{5}$$

representing p_{0j} as p_j since we only focus on AP_0.

The expected caching benefit for a chunk is simply the aggregate saving due to the avoidance of accessing that item in other nodes, i.e.,

$$reward(o_x) = p_{o_x} \cdot [E_{pull}(o_x) - E_{localaccess}(o_x)] \tag{6}$$

where

$$E_{localaccess} = s_{o_x} \cdot e_c \tag{7}$$

and

$$E_{pull} = e_g(o_x) \tag{8}$$

In other words, the expected reward of caching an item in AP_0 is the expected avoidance of energy consumption for fetching the item from any of the network nodes other than AP_0 minus the cost of having and fetching that chunk locally in AP_0. C is the set of items in the cache at the time when the caching replacement decision is started. $\widehat{C}$ denotes the set of decached (removed from cache) chunk(s) due to incoming request. The caching benefit of newly coming o_k due to access request is not controllable since it is determined exogenously by the incoming request. However, the cache replacement algorithm controls the victim chunks $o_j \in \widehat{C}$ and therefore which chunks to keep under the constraint of vacant space for o_k. Then the optimization problem for cache replacement becomes

$$\begin{aligned} &\text{maximize} && \sum_{o_j \in C} \text{reward}(o_j) \cdot \mathbb{1}_{[o_j \notin \widehat{C}^{+}]} && \text{(9a)} \\ &\text{subject to} && \sum_{o_j \in C} s_{o_j} \cdot \mathbb{1}_{[o_j \notin \widehat{C}^{+}]} \le M - s_{o_k}, && \text{(9b)} \\ & && j \in \{1, 2, ..., I\}. && \end{aligned}$$

where $\mathbb{1}_{[x]}$ is the indicator function which is equal to 1 if x is true and 0 otherwise, and o_k is the newly cached chunk. $\widehat{C}^{+}$ refers to the decached set $\widehat{C}$ after the decision is finalized. The constraint (9b) states that after decaching, the total size of items remaining in the cache should be such that there is vacant space (at least with the size of o_k) for the incoming chunk o_k.

This problem can be mapped to 0-1 knapsack problem which is known to be NP-hard [30, 29]. Given a knapsack with maximum capacity W, and a set S consisting of n items with weight w_i and benefit value b_i ($w_i, b_i, W \in \mathbb{Z}$), the problem is how to pack the knapsack (to select the subset of items) to achieve maximum total benefit of packed items. A possible brute force solution is to try all 2^n subsets of S, which is not scalable due to complexity. In this work, we propose a greedy heuristic, namely ENACI, which incorporates energy reward, popularity, TTL and delay (i.e. chunk loss due to delay sensitivity) on the energy consumption of the investigated system and is tailored for wireless access networks.

The optimal solution for 0-1 knapsack problem is available as a dynamic programming solution [30]. The pseudocode is shown in Algorithm 1. *V[n,W]* is a two-dimensional array of size (n,W) that is updated to keep temporary values and to contain the the final solution at the completion of the algorithm run. The time complexity for this algorithm is $O(I \cdot M)$, where I is the number of cacheable chunks and M is the cache size. This $O(I \cdot M)$ times operation is compromised of the following steps: $O(I \cdot M)$ times to fill the V-table, which has $(I + 1) \cdot (M + 1)$ entries, each requiring $O(1)$ time to compute. $O(I)$ time to trace the solution, because the tracing process starts in row I of the table and moves up 1 row at each step. Therefore, lower complexity heuristic algorithms are important for this problem. The complexity of our algorithm described in Section 4.3 is $O(I)$ since it runs over cached items and performs $O(1)$ operations for each item in the cache.

We compare our algorithm ENACI to the Least Recently Used (LRU) algorithm and to the baseline case BASE solved using dynamic programming algorithm. The baseline controller does not utilize TTL or delay and thus a less ”intelligent” decision maker solving the cache management based on (1) via dynamic programming. In addition to *BASE* case, the packet drops due to delay violations and TTL evictions are integrated into ENACI model as described below. The third algorithm, LRU, replaces the chunk(s) least recently accessed until the space is sufficient for o_k. LRU is a very common algorithm employed extensively in hardware and software-based caches. Please refer to [31] for a more detailed and analytical treatment of LRU.

Algorithm 1 Optimal solution algorithm for 0-1 knapsack problem.

```
procedure KnapsackSolver($v, w, n, W$)
    for $w \leftarrow 0, W$ do
        $V[0, w] \leftarrow 0$
    end for
    for $i \leftarrow 1, n$ do
        for $w \leftarrow 0, W$ do
            if $(w[i] \le w) \wedge (v[i] + V[i-1, w - w[i]] > V[i-1, w])$ then
                $V[i, w] \leftarrow v[i] + V[i-1, w - w[i]]$
                $keep[i, w] \leftarrow 1$
            else
                $V[i, w] \leftarrow V[i-1, w]$
                $keep[i, w] \leftarrow 0$
            end if
        end for
    end for
    $K \leftarrow W$
    for $i \leftarrow n, 1$ do
        if $keep[i, K] == 1$ then
            output $i$
            $K \leftarrow K - w[i]$
        end if
    end for
    return $V[n, W]$
end procedure
```

4.3. *Energy Aware Caching for Wireless ICN (ENACI)*

We layout the design space for ENACI scheme followed by its detailed description in this section. In addition to the primary objective of conveying ENACI, we render the rationale behind its structure and design.

4.3.1. *Design Space for an Energy-Aware Caching Heuristic f_x*

For an energy-aware caching scheme, we need a simple yet effective heuristic regarding two aspects:

- *Simple computation*: The main premise behind a heuristic is to have reduced complexity compared to optimal solutions or to come up with a scheme when algorithmic approaches are not attainable. An effective caching scheme should exhibit computational advantage in that regard.

- *Limited data support requirements for decision logic*: The more data support needed for decision logic (i.e. parameter values used in computations), the lower feasibility and the higher overhead. Since caching infrastructure is a distributed system, the availability of information such as metric values can render a caching management scheme infeasible due to lack of access to those data. Thus, data requirements should be minimized.

Considering these factors, we employ the most salient factors for chunk selection or eviction in our ENACI heuristic. These parameters and their interactions are shown in Figure 3. They can be listed as:

- *Energy reward*: This is the main factor, for which f_x aims to opt for chunks with higher values.
- *Popularity*: Keeping the more popular chunks facilitates the exploitation of statistical nature of chunk requests.
- *Delay sensitivity*: Preferring delay sensitive chunks alleviates energy consumption with retransmissions due to latency. For caching related literature focusing solely on the "edge", that factor is not rather apparent. However, when an infrastructure-based wireless network with an end-to-end chunk transmission is considered, it becomes more important. This quantity is more application-oriented compared to TTL, which is typically applied in a comprehensive setting in the caching framework.
- *TTL*: Cached chunks with larger TTL values improve EE since cache replacements are less common. However, TTL value is typically chunk-content independent and adopts a single value or a value from a set of very limited size.
- *Chunk size*: For chunk diversity and count, smaller chunks are preferable. Although this tendency may increase the complexity of a cache management scheme due to larger number and diversity of chunks, such a drawback is negligible compared to the overall gain.

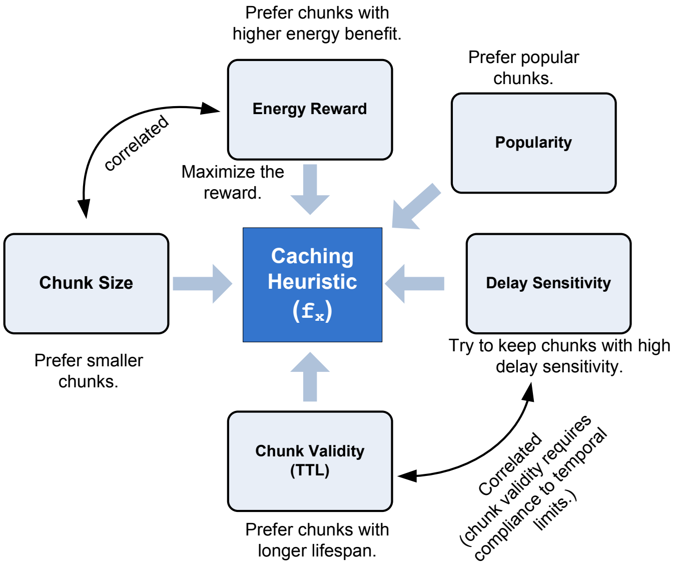
Data support requirements for our scheme (the practicality of the heuristic considering information input) can be analyzed regarding these parameters. While the content popularity can be estimated using access statistics as explained in Section 4, we assume availability of energy consumption quantities e_x through information sharing among nodes. Although this capability may appear as an emerging overhead, it is not significant since the mean statistics of these figures are sufficient. These data are not very dynamic and simple to calculate using simple averaging methods. Moreover, another required parameter value, the chunk size(s), is already available in the cache. TTL can be a system-wide parameter configured into the caching framework or can be embedded in the chunk headers for more flexible operation. Finally, the delay sensitivity is application-dependent and can be preconfigured in caches according to general traffic classes or embedded in chunk headers as an additional data field.

4.3.2. ENACI Scheme

ENACI is a greedy algorithm which eliminates the least benefit chunk(s) from the cache in each step, until the available space is sufficient for caching the new chunk o_k. This benefit is a function of popularity, TTL, size, delay sensitivity (σ) and energy benefit of the replaced chunk(s). Specifically,

$$benefit(o_j) = f(reward_j, s_j, TTL_j, \sigma_j) \tag{10}$$

where the delay sensitivity, σ_j, is modeled as a metric representing the delay factor on the object drop probability for o_j due to delay threshold violation.



Prefer chunks with higher energy benefit.
Energy Reward
Prefer popular chunks.
Popularity
correlated
Maximize the reward.
Chunk Size
Caching Heuristic (f_x)
Delay Sensitivity
Prefer smaller chunks.
Try to keep chunks with high delay sensitivity.
Chunk Validity (TTL)
Correlated (chunk validity requires compliance to temporal limits.)
Prefer chunks with longer lifespan.

Figure 3: Design space and imperatives for heuristic design.

The benefit is directly proportional to energy reward and inversely proportional to size, since having more chunks is favorable to increase hit ratio. Moreover, TTL is directly proportional since the cached elements are valid for a longer period of time for larger TTL values. From the delay perspective, the retransmissions caused by packet drops due to latency also incur energy wastage and therefore needs to be considered for caching decisions. In that regard, it is more favorable to keep chunks for delay sensitive traffic in the caches, for instance streaming video content. To address the delay sensitivity of objects, the delays caused by fetching items not found in the cache should be considered. The delay sensitivity parameter σ_j incorporates this factor into the benefit calculation. In that regard, the objects that take longer to fetch should be preferentially cached and retained in the cache [29]. Considering all these factors, we use a composite metric for the benefit of having o_j in the cache:

$$benefit(o_j) = \frac{reward_j}{s_j} \cdot \frac{TTL_j}{TTL_{max}} \cdot (\sigma_j)^{\gamma} \tag{11}$$

where σ_j is the calculated drop probabillity p_j^d according to the delay threshold t_j for o_j and the delay distribution for the chunks in the network. This delay for chunk retrieval to APs is modeled with an exponential distribution having a cut-off delay equal to 200 unit energy. The exponent γ is a tuning parameter for controlling the effect of σ_j on the benefit computation and used as 2 in the performance evaluation. Each chunk belongs to one of T delay threshold classes, i.e. $t_j \in \mathbb{T}$, and are uniformly distributed to these classes in a random assignment at the beginning of the performance evaluation.

For the cache replacement operation, at each query, if the requested chunk is not already cached, the chunks starting from the least benefiting according to (11) are decached till the space

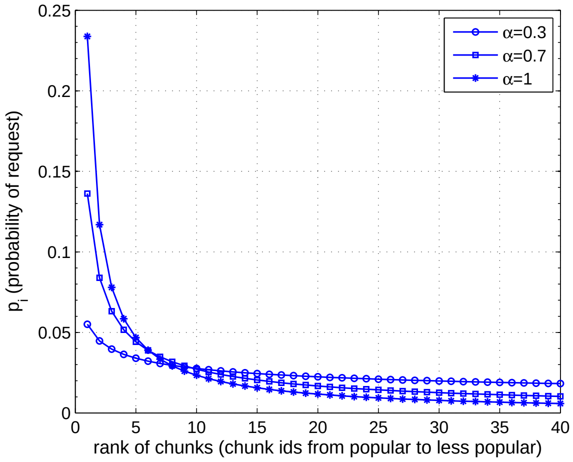


Figure 4: Probability distribution p_i for chunk i for different α values ($I = 40$).

is opened for the incoming chunk.

5. Numerical Evaluation

For performance evaluation, the chunk popularity is modeled as a cut-off Zipf distribution [32]. In other words, we assume that the relative frequency with which the chunks are accessed follows a generalization of Zipf's law which is widely employed in cache management and CCN works [31, 33, 34, 35, 36, 37]. Although there are other more-tailored popularity models for specific kinds of application content, Zipf distribution provides a more general and widely-applicable model and allows an application-agnostic analysis for performance investigation of cache management frameworks. Specifically, let the N chunks be ranked in order of their popularity where o_j is the *jth* most popular object. Let p_{ij} be the probability that an incoming access is for o_j through AP_i. Focusing on a single AP_0 and p_j with a cut-off Zipf-like distribution, for $1 \leq j \leq N$ and $\alpha \geq 0$ [31]

$$p_j = \frac{\Omega}{j^\alpha} \tag{12}$$

where

$$\Omega = \left(\sum_{j=1}^{N} \frac{1}{j^\alpha} \right)^{-1}. \tag{13}$$

Zipf's law implies that the top-ranked flow rates are exceptionally large but rare and the lower-ranked rates are smaller but more common [38]. Thus, a small number of the most popular contents account for a large portion of user requests [39]. Figure 4 plots p_j as a function of α. Parameter α is the shape parameter that describes the relative popularity of objects in the distribution [40]. It is clear that the larger the α value, the better the compactness of the p_j distribution. We consider the range of α observed in [32] using network traces from a variety of sources. When $\alpha = 0$, (12) is simplified as $p_j = 1/N$ for all i, and the data access rates to all chunks are the same (equally-likely case). As noted in [31], this corresponds to the worst case since larger α value is expected to improve the cache performance.

Table 3: Simulation parameters.

Parameter	Value
M	42
$\lvert I \rvert$	700
TTL	500
K	2
$\mathbb{S}$	{6}, {4, 6, 8}, {3, 6, 9}, {1, 2, 6, 10, 11}, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}
α	{0.1, 0.2, 0.3, 0.5, 0.7, 1, 1.2, 1.4}
$\mathbb{T}$	{35, 45, 50, 55, 60}
γ	2
e_c	4
e_f	1
e_{req}	10
e_{pr}	9
$N_{\{br,cr,er\}}$	{5, 6, 1}
$e_{\{br,cr,er\}}$	{16, 32, 85}

The baseline simulation parameters are listed and summarized in Table 3. The explanations for these parameters are given in Table 1. The units for the energy, size, and time quantities are unit energy, unit size, and unit time, respectively. During the performance evaluation, the relevant parameter values are altered according to the case, which is explained in the corresponding subsection.

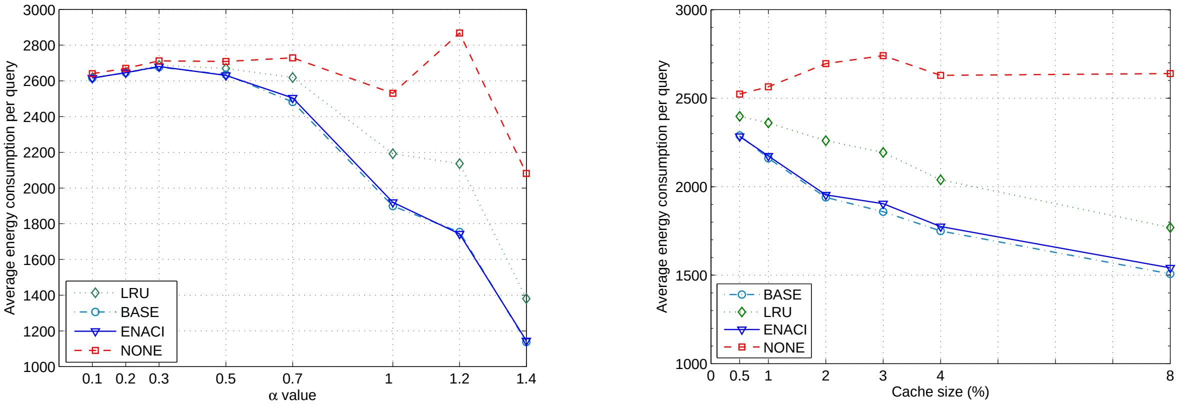
5.1. Effect of Zipf Parameter α

Larger α values imply more skewed data access probability distribution as noted in Section 5. Typically, this kind of popularity pattern favors caching and improves caching performance. As seen in Figure 5(a), since it is in exponential relation, the effect of α starts to manifest itself more apparently especially after it is above 0.5. For smaller values, the gain due to caching is almost negligible. As α gets larger, LRU provides significant savings compared to no caching. Moreover, ENACI provides almost the same performance compared to BASE. For $\alpha = 1.2$, it is slightly better [1750 unit energy per chunk query (denoted as the unit *ue/cq*) for ENACI vs. 1760 ue/cq for BASE].

In general, all caching algorithms behave as expected and the apparent profile is the increasing energy EE with increasing α value. For instance, the energy consumption of ENACI decreases from 2616 ue/cq to 1145 ue/cq as α ranges from 0.1 to 1.4. This outcome is due to increasing cache hit ratio leading to energy savings while serving he requested chunks to the consumers.

5.2. Effect of Cache Size M

Increasing cache size is expected to benefit EE due to two factors. First, a larger cache is able to store more objects leading



(a) Effect of α on the average energy consumption per object query.

(b) Effect of cache size M on the average energy consumption per object query ($\alpha = 1$).

Figure 5: Performance comparison of caching mechanisms for different α values and cache sizes.

to higher hit ratio. Second, storing and serving a chunk from the cache typically outweighs fetching it remotely in terms of EE. Figure 5(b) depicts the performance of evaluated schemes for increasing cache size dimensioned as percentage of aggregate chunk size, which is simply equal to $|I| \cdot \overline{s_o}$. Similar to Figure 5(a), the performance of ENACI is very close to BASE. As the cache size increases from 0.5% to 8%, the energy consumption for ENACI improves from 2284 ue/cq to 1542 ue/cq, which corresponds to 32.4% decrease. It manages to keep an advantageous performance gap with LRU ranging from 5% to 14%. The minimum gap is for the smallest cache size which corresponds to the convergence of all caching algorithms. The average energy consumption for no caching (NONE case) is 2632 ue/cq, compared to 2169 ue/cq, 1938 ue/cq, and 1916 ue/cq for LRU, ENACI, and BASE, respectively.

5.3. *Effect of Object Size Composition*

The object size composition may have a significant effect for cache replacement algorithms, especially for heuristics parameterized with s_{o_j}. Therefore, we also investigate the performance for different object size compositions. The object size composition $S_i \in \mathbb{S}$ has a symmetric distribution around the mean object size ($\overline{s_o}$) equal to 6 units and is given in Table 3. The composition ID i in the graph corresponds to the order of elements in $\mathbb{S}$ in the table, i.e. 1 for {6} while 5 for {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}. The objects are assigned uniformly-distributed to object sizes for a selected S_i. Figure 6 shows that the sensitivity of our scheme to object size configuration is low and thus it maintains a similar performance improvement under varying size compositions. This robustness is beneficial for a cache management scheme since the incoming object traffic can be composed of diverse object sizes. Moreover, the performance of ENACI is very close to BASE in all cases. The performance gap between ENACI and BASE never exceeds 2.2% of BASE results which occurs for the last case (ID = 5) with $E_{BASE} = 2155$ ue/cq and $E_{ENACI} = 2204$ ue/cq.

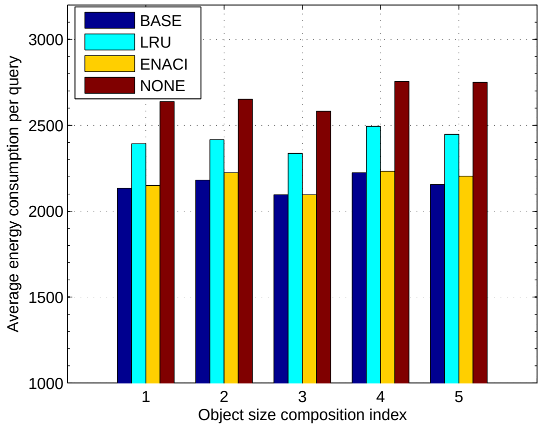


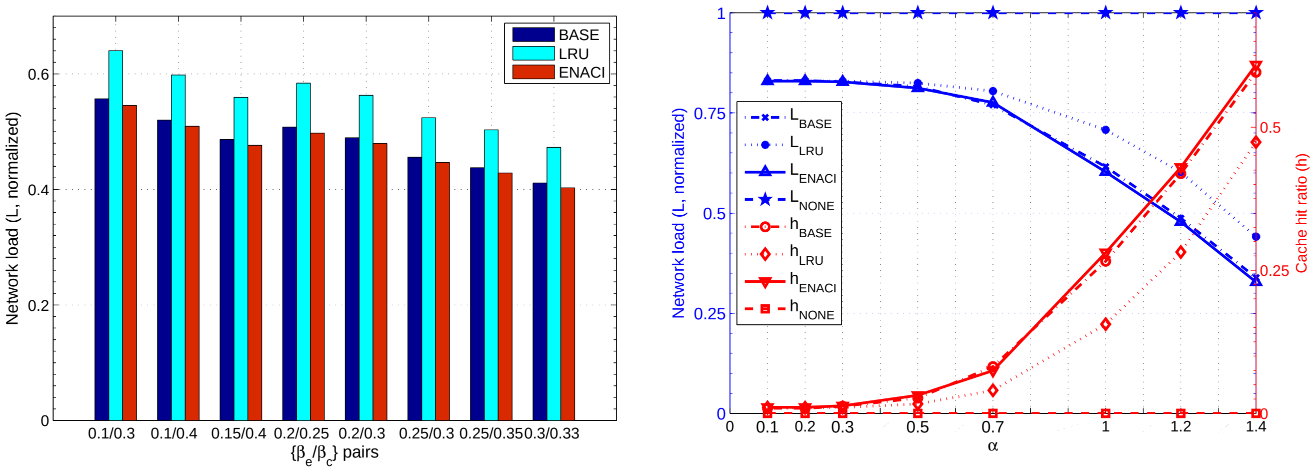
Figure 6: Effect of size composition on the average energy consumption per object query ($\alpha = 0.9$, $M = 42$, $\overline{s_o} = 6$).

5.4. *Impact of Caching on Network Traffic*

Caching closer to the content consumer drastically reduces network load and thus energy consumption caused by the transportation of data to users. We also evaluate the impact of caching on network traffic in our network layout. The network load in our system is measured as the amount of data times the number of hops it travels [41]. Specifically,

$$\begin{aligned} L &= \rho\overline{s}_o[N_e\beta_e + (N_c + N_e)\beta_c \\ &\quad + (N_b + N_c + N_e)(1 - \beta_e - \beta_c)] \qquad (14) \\ &= (N_b + N_c)(1 - \beta_e) - N_b\beta_c + N_e \qquad (15) \end{aligned}$$

where $\rho = 1 - h$ is the probability that a request is not served from the AP cache, i.e. cache miss ratio. The probabilities β_e and β_c represent the probability that a missed object in the AP cache is found in the edge or core network, respectively. These two values implicitly determine the probability β_b that the object has to be fetched from the universal source since $\beta_e + \beta_c + \beta_b = 1$. The β composition can be considered as an indicator for the assumed performance of the caching in other network segments. For instance, $\{\beta_e, \beta_c\} = \{0.25, 0.30\}$, i.e. $\beta_b = 0.45$,



(a) For different β values ($\alpha = 0.9$)

(b) For different α values ($\beta_{\{e,c\}} = \{0.1, 0.2\}$)

Figure 7: Network traffic volume and cache hit ratio for varying α and β.

corresponds to a very effective in-network caching system since 55% of all requests not served in APs are served in the edge or core network without leaping to backbone network for the universal source.

Figure 7 shows the network traffic load and cache hit ratio for BASE, LRU, ENACI and NONE with varying α and β values. For NONE case, we assume caching is absent not just in AP_i but in any network segment, thus implying a baseline case. In Figure 7(a), normalized network load for $\alpha = 0.9$ with varying β_e/β_c pairs is shown. For the “cache-less” NONE mode, the network traffic volume is much larger compared to the modes with caching. This quantity is used for normalizing the performance results of other schemes. Since we assume that the other nodes in the network do not employ caching, the pull operation implies the transfer of a chunk from the universal source in that mode. For the best case with $\beta_e = 0.3$ and $\beta_c = 0.33$, the network load reduction for BASE, LRU, and ENACI are 59%, 53%, and 60%, respectively. These results support the envisaged benefit of traffic localization (increasing proximity between the data and the requester) for network and server load reduction. Furthermore, the average network loads for BASE, LRU, and ENACI over the entire range are 48.3%, 55.5%, and 47.3%, respectively.

In the last experiment, we investigate the effect of α on the network load figures. Figure 7(b) shows the simulation results for network load L as well as the cache hit ratio h for increasing α. As α gets larger, the performance of caching improves in general. Therefore, the resulting energy consumption exhibits the same behavior. This trend is in accordance with Figure 5(a). The load reduction is substantial even for moderate α values. For instance, when $\alpha = 0.7$, ENACI provides 22% reduction for network traffic. The results for BASE and LRU are similar in that case. For $\alpha = 1.1$, the advantage of ENACI over LRU becomes significant: 60% vs. 71% normalized network load. Furthermore, ENACI outperforms BASE but with a minor gap in this case.

6. Conclusions

In this work, we have discussed and evaluated cache replacement policies for information-centric operation at the edge of infrastructure based wireless networks. We have focused on the objective of EE. ICN provides new degrees of freedom for CSPs to meet mobile broadband requirements. Moreover, the increasing content-centric access over wireless networks poses ICN-based approaches more attractive. In that regard, caching and in-network storage is a crucial constituent of this paradigm. In our study, we investigate the utilization of caches in the wireless access nodes for increasing performance in terms of EE. The flexible setup of edge caching to facilitate multimode networking (conventional or content-centric) is beneficial for EE. The proposed greedy heuristic ENACI algorithm provides an energy-efficient cache replacement scheme with relatively low complexity for ICN based wireless access networks. As future work, we plan to include a more elaborate energy consumption model for remote retrieval of content from the providers in the Internet. Another potential topic is the analysis for cooperative AP caching embedded in a more realistic incumbent caching infrastructure.

References

1. Carzaniga, A., Papalini, M., Wolf, A.L.. Content-based publish/subscribe networking and information-centric networking. In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*; ICN ’11. ISBN 978-1-4503-0801-4; 2011, p. 56–61.
2. Gür, G., Alagöz, F.. Green wireless communications via cognitive dimension: an overview. *Network, IEEE* 2011;**25**(2):50–56. doi: 10.1109/MNET.2011.5730528.
3. Lee, U., Rimac, I., Kilper, D., Hilt, V.. Toward energy-efficient content dissemination. *Network, IEEE* 2011;**25**(2):14–19.
4. Hasslinger, G., Hohlfeld, O.. Efficiency of caches for content distribution on the Internet. In: *Teletraffic Congress (ITC), 2010 22nd International*. 2010, p. 1–8. doi:10.1109/ITC.2010.5608730.

5. Pallis, G., Vakali, A.. Insight and perspectives for content delivery networks. *Commun ACM* 2006;**49**(1):101–106. doi: 10.1145/1107458.1107462.
6. Muscariello, L., Carofiglio, G., Gallo, M.. Bandwidth and storage sharing performance in information centric networking. In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*; ICN '11. New York, NY, USA: ACM. ISBN 978-1-4503-0801-4; 2011, p. 26–31. doi:10.1145/2018584.2018593.
7. Alzahrani, B.A., Reed, M.J., Riihirvi, J., Vassilakis, V.G.. Scalability of information centric networking using mediated topology management. *Journal of Network and Computer Applications* 2014;(0):–.
8. Psaras, I., Clegg, R.G., Landa, R., Chai, W.K., Pavlou, G.. Modelling and evaluation of CCN-caching trees. In: *Proceedings of the 10th international IFIP TC 6 conference on networking - Volume Part I*; NETWORKING'11. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-642-20756-3; 2011, p. 78–91.
9. Choi, J., Han, J., Cho, E., Kwon, T., Choi, Y.. A survey on content-oriented networking for efficient content delivery. *Communications Magazine, IEEE* 2011;**49**(3):121–127. doi:10.1109/MCOM.2011.5723809.
10. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.. Networking named content. In: *Proceedings of the 5th international conference on emerging networking experiments and technologies*; CoNEXT '09. New York, NY, USA: ACM. ISBN 978-1-60558-636-6; 2009, p. 1–12. doi:10.1145/1658939.1658941.
11. Xu, Y., Li, Y., Lin, T., Wang, Z., Niu, W., Tang, H., et al. A novel cache size optimization scheme based on manifold learning in content centric networking. *Journal of Network and Computer Applications* 2014; **37**(0):273 – 281.
12. Valancius, V., Laoutaris, N., Massoulié, L., Diot, C., Rodriguez, P.. Greening the Internet with nano data centers. In: *Proceedings of the 5th international conference on Emerging networking experiments and technologies*; CoNEXT '09. New York, NY, USA: ACM. ISBN 978-1-60558-636-6; 2009, p. 37–48. doi:10.1145/1658939.1658944.
13. Li, Q., Niu, H., Papathanassiou, A., Wu, G.. 5g network capacity: Key elements and technologies. *Vehicular Technology Magazine, IEEE* 2014; **9**(1):71–78. doi:10.1109/MVT.2013.2295070.
14. Caire, G.. The role of caching in 5G wireless networks; 2013. Invited Talk at *IEEE ICC 2013*.
15. Yin, L., Cao, G.. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing* 2006;**5**(1):77–89. doi: http://doi.ieeecomputersociety.org/10.1109/TMC.2006.15.
16. Ma, Y., Jamalipour, A.. A cooperative cache-based content delivery framework for intermittently connected mobile ad hoc networks. *Wireless Communications, IEEE Transactions on* 2010;**9**(1):366–373.
17. Fiore, M., Casetti, C., Chiasserini, C.. Caching strategies based on information density estimation in wireless ad hoc networks. *Vehicular Technology, IEEE Transactions on* 2011;**60**(5):2194–2208.
18. Dimokas, N., Katsaros, D., Manolopoulos, Y.. Cooperative caching in wireless multimedia sensor networks. *Mobile Networks and Applications* 2008;**13**(3-4):337–356. doi:10.1007/s11036-008-0063-3.
19. Hoeller, N.. Dynamic approximative data caching in wireless sensor networks. *2013 IEEE 14th International Conference on Mobile Data Management* 2010;:291–292.
20. Stamos, K., Pallis, G., Vakali, A.. Caching techniques on cdn simulated frameworks. In: Buyya, R., Pathan, M., Vakali, A., editors. *Content Delivery Networks*; vol. 9 of *Lecture Notes Electrical Engineering*. Springer Berlin Heidelberg. ISBN 978-3-540-77886-8; 2008, p. 127–153. doi: 10.1007/978-3-540-77887-5-5.
21. Chen, Y., Katz, R., Kubiatowicz, J.. Dynamic replica placement for scalable content delivery. In: Druschel, P., Kaashoek, F., Rowstron, A., editors. *Peer-to-Peer Systems*; vol. 2429 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. ISBN 978-3-540-44179-3; 2002, p. 306–318.
22. Lo Presti, F., Petrioli, C., Vicari, C.. Dynamic replica placement in content delivery networks. In: *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on*. 2005, p. 351–360.
23. Bartolini, N., Lo Presti, F., Petrioli, C.. Optimal dynamic replica placement in content delivery networks. In: *Networks, 2003. ICON2003. The 11th IEEE International Conference on*. 2003, p. 125–130.
24. Famaey, J., Iterbeke, F., Wauters, T., Turck, F.D.. Towards a predictive cache replacement strategy for multimedia content. *Journal of Network and Computer Applications* 2013;**36**(1):219 – 227.
25. Silvestre, G., Monnet, S., Krishnaswamy, R., Sens, P.. Caju: A content distribution system for edge networks. In: *Euro-Par 2012: Parallel Processing Workshops*; vol. 7640 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. ISBN 978-3-642-36948-3; 2013, p. 13–23.
26. Xu, J., Hu, Q., Lee, W.C., Lee, D.L.. Performance evaluation of an optimal cache replacement policy for wireless data dissemination. *Knowledge and Data Engineering, IEEE Transactions on* 2004;**16**(1):125–139. doi:10.1109/TKDE.2004.1264827.
27. Blefari-Melazzi, N., Detti, A., Morabito, G., Salsano, S., Veltri, L.. Supporting information-centric functionality in software defined networks. In: *Proceedings of the IEEE ICC - workshop on SDN*. 2012, .
28. Lin, Y., Chang, Y., et al. Modeling frequently accessed wireless data with weak consistency. *Journal of Information Science and Engineering* 2002;**18**(4):581–600.
29. Shim, J., Scheuermann, P., Vingralek, R.. Proxy cache algorithms: design, implementation, and performance. *Knowledge and Data Engineering, IEEE Transactions on* 1999;**11**(4):549–562. doi:10.1109/69.790804.
30. Li, W., Chan, E., Chen, D.. Energy-efficient cache replacement policies for cooperative caching in mobile ad hoc network. In: *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*. 2007, p. 3347–3352.
31. Lin, Y.B., Lai, W.R., Chen, J.J.. Effects of cache mechanism on wireless data access. *Wireless Communications, IEEE Transactions on* 2003; **2**(6):1247–1258. doi:10.1109/TWC.2003.819019.
32. Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.. Web caching and zipf-like distributions: evidence and implications. In: *Proceedings of IEEE INFOCOM '99*; vol. 1. 1999, p. 126–134. doi: 10.1109/INFCOM.1999.749260.
33. Li, X., Guang, T., Veeravalli, B.. Design and implementation of a multimedia personalized service over large scale networks. In: *Multimedia and Expo, 2006 IEEE International Conference on*. 2006, p. 77–80. doi:10.1109/ICME.2006.262554.
34. Hu, X., Papadopoulos, C., Gong, J., Massey, D.. Not so cooperative caching in named data networking. In: *Global Communications Conference (GLOBECOM), 2013 IEEE*. 2013, p. 2263–2268. doi: 10.1109/GLOCOM.2013.6831411.
35. Byan, S., Lentini, J., Madan, A., Pabon, L., Condict, M., Kimmel, J., et al. Mercury: Host-side flash caching for the data center. In: *Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on*. 2012, p. 1–12. doi:10.1109/MSST.2012.6232368.
36. Gomaa, H., Messier, G., Davies, R., Williamson, C.. Media caching support for mobile transit clients. In: *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on*. 2009, p. 79–84. doi:10.1109/WiMob.2009.23.
37. Serpanos, D., Karakostas, G., Wolf, W.. Effective caching of web objects using zipf's law. In: *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*; vol. 2. 2000, p. 727–730 vol.2. doi:10.1109/ICME.2000.871464.
38. Wallerich, J., Feldmann, A.. Capturing the variability of internet flows across time. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. 2006, p. 1–6. doi: 10.1109/INFOCOM.2006.37.
39. Ni, J., Tsang, D.H.K.. Large-scale cooperative caching and application-level multicast in multimedia content delivery networks. *Communications Magazine, IEEE* 2005;**43**(5):98–105. doi: 10.1109/MCOM.2005.1453429.
40. Borst, S., Gupta, V., Walid, A.. Distributed caching algorithms for content distribution networks. In: *INFOCOM, 2010 Proceedings IEEE*. 2010, p. 1–9. doi:10.1109/INFCOM.2010.5461964.
41. Krishnan, P., Raz, D., Shavitt, Y.. The cache location problem. *Networking, IEEE/ACM Transactions on* 2000;**8**(5):568–582. doi: 10.1109/90.879344.