Journal of High Speed Networks 19 (2013) 215–235
DOI 10.3233/JHS-130474
IOS Press

Energy efficient data caching for content dissemination networks

Satoshi Imai [a,b,*], Kenji Leibnitz [c] and Masayuki Murata [b]
[a] *Fujitsu Laboratories Ltd., Kanagawa, Japan*
[b] *Graduate School of Information Science and Technology, Osaka University, Osaka, Japan*
[c] *Center for Information and Neural Networks, National Institute of Information and Communications Technology, and Osaka University, Osaka, Japan*

Abstract. As various multimedia services are being provided on networks, broadband traffic is growing as well. Reducing traffic is important because power consumption in networks has been increasing year by year. Meanwhile, in-network caching for CCN (*Content-Centric Networking*) is expected to achieve an adaptive content delivery in the network and to reduce traffic by storing content data on each router. Therefore, in-network caching is beneficial in view of energy efficiency. However, in order to improve energy efficiency of the network, it is necessary to cache content appropriately in consideration of both power consumption of content caching and transmission of traffic. In this paper, we propose a design method which derives the most energy efficient cache locations of content and a distributed cache mechanism for CCN to search for energy efficient cache locations. Furthermore, we demonstrate the effectiveness of the proposed method.

Keywords: Energy efficiency, content dissemination, CCN, in-network caching, content-centric networking

1. Introduction

The growing variety of multimedia services provided on networks is leading to an increase in network traffic. As a result, also the power consumption of network systems is increasing year by year. In the reports of the Ministry of Economy, Trade, and Industry [10], broadband traffic in Japan is growing at an annual rate of 30% and network power consumption is expected to occupy 20% of the total ICT power consumption by 2025.

Recently, power-saving mechanisms of network devices [8] have been studied in order to realize *Energy Proportional Networks* [9,18] in which power consumption of each device is proportional to its usage. In addition, energy efficiency can be improved by reducing the traffic flow in the network, because the traffic decrease can improve the effect of the power-saving mechanism in each device or can prevent that frequent incremental deployments of the network devices are required. As technologies for reducing network traffic, *Content Delivery Network* (CDN) architectures in metro/access networks, such as Akamai-CDN and Web-Proxy, are well known. The CDNs can manage the content delivery at the edge of the networks by allocating content replicas in cache servers which are in geographical proximity to users.

On the other hand, a content dissemination architecture called *Named Data Networking* (NDN) [13], utilizing caching functionality on routers, has recently been proposed. Content-Centric Networks (CCNs) for NDN can cache content data on many locations in networks and reduce traffic flow. However, network traffic is influenced by the cache locations because each content generates a different amount of traffic depending on its popularity. Moreover, many storages such as DRAM or SSD are required for content caching. Therefore, in order to realize energy efficiency in CCNs, the appropriate cache locations should be managed in consideration of the balance of traffic volume for delivering each content and memory used for storing content replicas.

*Corresponding author: Satoshi Imai, Fujitsu Laboratories Ltd., 4-1-1 Kamikodanaka, Kawasaki, Kanagawa, 211-8588 Japan. E-mail: imai.satoshi@jp.fujitsu.com.

Our research aims to realize energy efficient cache locations in CCN and we previously proposed a design method to minimize the sum of *power consumption of storage devices for content caching* and *power consumption of network devices for content delivery* [11]. Furthermore, we proposed a distributed cache mechanism to locally search for energy efficient cache locations attempting to be close to the optimal cache locations [12]. In the distributed cache mechanism, each router autonomously pre-designs a threshold of request rates of content before cache operation and judges whether to cache content data by comparing content request rates with the designed threshold. In this paper, we summarize the proposed methods and combine both approaches into the same framework to compare both. Additionally we evaluate the energy efficiency of the proposed methods for some scenarios.

The remainder of this paper is organized as follows. Section 2 discusses general issues of energy efficient caching in CCN and our approaches. This is followed by a summary of related work in Section 3. We first describe the design method to solve the cache location problem in Section 4 and the distributed cache mechanism to search for energy efficient cache locations in Section 5. Section 6 demonstrates simulation results and we conclude the paper in Section 7.

2. Issues and approaches for energy efficient data caching in CCN

CCN is a receiver-driven protocol where *Data* is only sent in response to a user's request. Each content is divided into chunks of fixed size and these chunks are cached on content routers along the content delivery route.

The content dissemination mechanism in CCN has the following features.

(1) *Content advertisement*: Content publishers advertise newly released content from the origin site of the content along predefined routes.
(2) *Content discovery/delivery*: A content request (*Interest*) is forwarded on each content router (*CR*) based on the *Forwarding Information Base* (FIB) until the requested content can be found. An *Interest* forwarded on a *CR* is added to the *Pending Interest Table* (PIT) in order to remember the interface on which to send back the replies (*Data*). When the requested content is found on a *CR*, *Data* of the content are transmitted based on the PIT.
(3) *Content storage*: *Data* are cached on all *CR*s along the transmission route based on the specific replacement policy such as Least Recently Used (LRU) or Least Frequently Used (LFU).

In-network caching for CCN can reduce network traffic by storing content data in many locations. However, this can result in inefficient storage of content replicas when the number of requests for content is small. Therefore, in order to realize energy efficiency of the network under the condition that network devices and memory can be deployed in proportion to their usage, the appropriate cache allocation should be executed in consideration of the *cache allocation power*, i.e., the memory power consumed by storing the content and the *traffic transmission power*, i.e., the total power consumed by network devices when data are transmitted.

Moreover, the caching structure in CCN tends to have a logical hierarchy (cf. Fig. 1), which is constructed by caching content in some *CR*s in a tree rooted at an origin site of the content. The caching hierarchy is constructed by routes between an origin site, caching nodes and users, such that

- less popular content is cached on nodes near to the origin site, and
- more popular content is cached on nodes near to users.

In order to realize energy efficient locations of content which can reduce the sum of *cache allocation power* and *traffic transmission power*, we should consider constraints of multiplexed caching hierarchies of content (cf. Fig. 1). Furthermore in CCN, a request should be autonomously executed by some *CR*s without needing to know the content locations.

Therefore, we first aim to provide reference locations to realize energy efficiency for cache strategies and introduce a new design method which can derive energy efficient locations of content on constraints of the caching

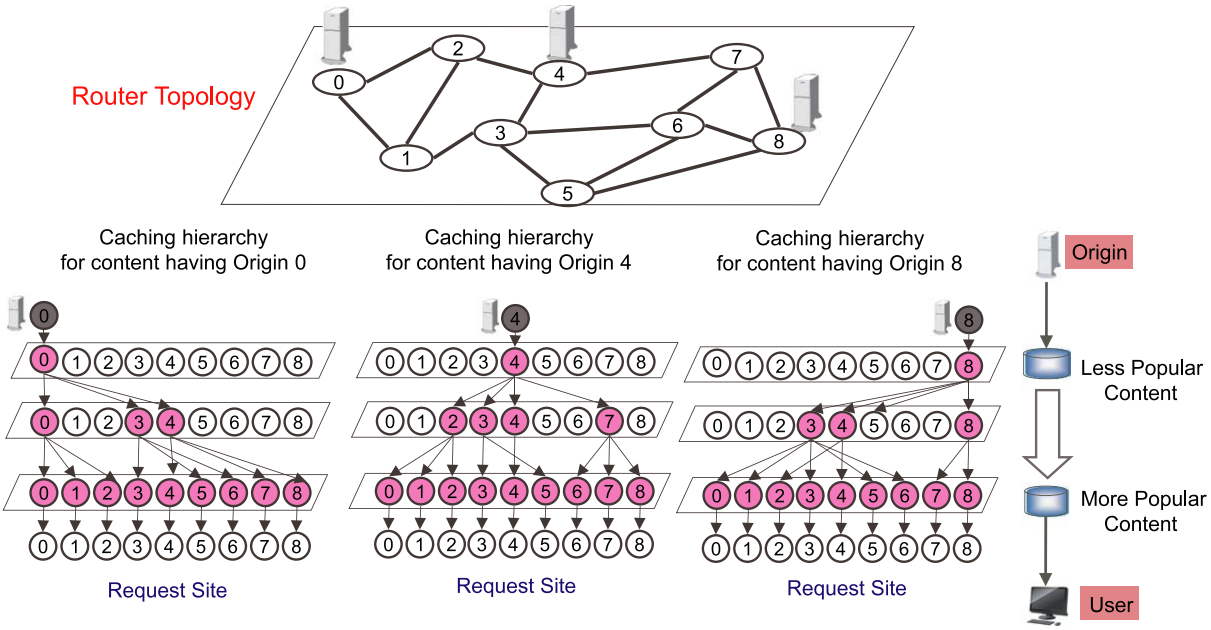


Fig. 1. Caching hierarchies for each origin site. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

hierarchy rooted at the content's origin site so as to minimize the sum of *cache allocation power* and *traffic transmission power*. Secondly, we show a distributed cache mechanism to search for energy efficient cache locations of content based on the caching hierarchies attempting to be close to the optimal solution.

3. Related work

Recently, energy efficiency in CCN has been attracting a lot of attention [7,14,15]. Lee et al. [14,15] survey the energy efficiency of various network devices deployed in access/metro/core networks. Furthermore, they evaluate the power-saving effect in the entire network when the deployment ratio of CCN-enabled edge/core routers is changed. As a result, they show that CCN is able to improve the energy efficiency of current CDNs.

Furthermore, Guan et al. [7] build energy models of traffic transmission power and caching power for content delivery architectures such as "Conventional and decentralized server-based CDN", "Centralized server-based CDN using dynamic optical bypass", and CCN. Using their energy models, they analyze the energy efficiency of each of these architectures. Those energy models are approximations based on the relations between the topological structure and the average hop-length from all sites to the nearest cache location. The appropriate number of cache locations for content can be estimated in order to save the network power consumption when the target topology, the required average hop-length, and the number of requests for content are given. In those papers, energy efficiency is represented differently depending on the cache locations for content.

Meanwhile, traditionally there are content placement algorithms [1,2,16,19] as a solution for *File Allocation Problems* [4] which minimize the cost imposed for content storage and queries (requests), or maximize the performance such as distance to content. Baev et al. [1] propose an Integer Linear Programming (ILP) model which minimizes content placement cost and an approximation solution using a linear relaxation. Furthermore, Qui et al. [19] propose some replica placement algorithms to solve the K-median problem for CDNs.

In contrast to the above-mentioned content placement problems, Borst et al. [2] formulate an ILP model based on a hierarchical structure for content locations to minimize bandwidth costs and propose a distributed solution of the problem. Moreover, they evaluate the cost-saving effect for a hierarchical topology which has symmetric

bandwidth cost for a parent node and some leaf nodes. However, these works do not discuss cache management to design or search for energy efficient content locations considering that caching hierarchies, which have different origin sites for content and asymmetric routes, are multiplexed in a network.

Therefore, we previously proposed a new ILP model to design the most energy efficient cache locations in consideration of the multiplexed caching hierarchies [11] and a distributed cache mechanism to locally search for energy efficient cache locations attempting to be close to the optimal cache locations [12]. In this paper, we summarize the details of the proposed methods and combine both approaches into the same framework which makes it easy to compare both. Additionally we show new evaluation results.

4. Optimal cache locations

In order to provide reference locations to evaluate energy efficiency for cache strategies, we show a 0–1 Integer Linear Programming Problem (0–1 ILP) model to design energy efficient cache locations of chunks of a target content satisfying the constraints of multiplexing caching hierarchies which have a different origin site for content [11].

The model designs cache locations of chunks of content in order of descending popularity to minimize total power consumption based on *Energy Proportional Networks*.

4.1. Design flow

The cache location design for each content is executed in the following steps.

Step 1: Get content popularity.
Step 2: Select a target content (kth popular content) in the order of content popularity.
Step 3: Extract route candidates to deliver the target content from a cache location to a requesting user on the shortest-path tree rooted at the origin site of the target content and define the route candidates as the design variables, cf. Fig. 2.
Step 4: Design the optimal cache locations of the target content based on the route candidates in consideration of the pre-designed routes for more popular content which have the same origin site as that of the target content.
Step 5: Return to Step 2 and execute the design of the next popular content ($[k+1]$th popular content).

4.2. Optimization model

The objective function is defined as follows. For a network composed of *CR*s and Wavelength Division Multiplexing (WDM) nodes in Fig. 2, this model designs cache locations and data (chunks) transmission routes for the delivery tree, which consists of a set of vertices (sites) V and route candidates R_c. The objective is to minimize

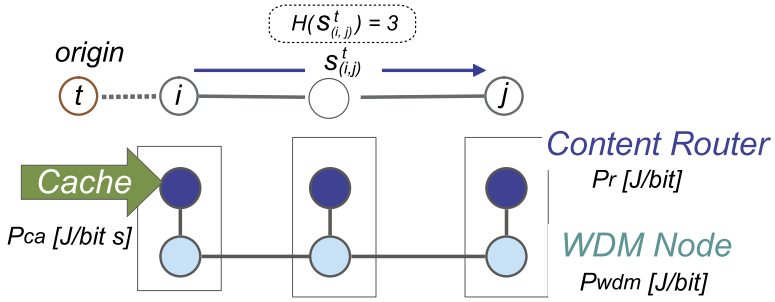


Fig. 2. Network model. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

the sum of cache allocation power Ca_i, i.e., the power for storing a chunk of the target content in *CR* on site i, and *traffic transmission power* $Tr_{k,(i,j)}$, i.e., the total power of *CR*s and WDM nodes when a chunk of content k is delivered on the route from site i to site j.

$$\textit{Minimize} \quad \sum_{i \in V} \{Ca_i \cdot u_i\} + \sum_{(i,j) \in R_c} \left\{Tr_{k,(i,j)} \cdot s^t_{(i,j)}\right\}. \tag{1}$$

The design variables are defined as follows:

- $u_i \in \{0, 1\}$ indicates whether or not to store a chunk of the target content on CR_i;
- $s^t_{(i,j)} \in \{0, 1\}$ describes whether or not to select the route from site i to site j defined on the shortest-path tree rooted at the origin site t of a chunk of the (kth popular) target content.

All variables in the optimization model are summarized in Table 1.

Here, the *cache allocation power* Ca_i (J) in 1 s when a chunk of the target content is cached in CR_i ($u_i = 1$) is defined as

$$Ca_i = 1 \cdot D \cdot P_{ca}, \tag{2}$$

where P_{ca} is the memory power density (J/(bit · s)) and D is the chunk size (bits) of the target content.

Furthermore, the *traffic transmission power* $Tr_{k,(i,j)}$ (J) when a chunk of (kth popular) target content is delivered on the candidate route $s^t_{(i,j)}$ from CR_i to CR_j on the shortest-path tree rooted at origin site t, is defined as

$$Tr_{k,(i,j)} = D \cdot R_{k,j} \cdot (P_r + P_{wdm}) \cdot H\left(s^t_{(i,j)}\right), \tag{3}$$

where P_r and P_{wdm} are the power densities (J/bit) of a *CR* and of a WDM node, respectively, and $R_{k,j}$ is the request rate from site j for the target content. Furthermore, we define $H(s^t_{(i,j)})$ as the hop-length of route $s^t_{(i,j)}$.

We next define the constraints for the proposed 0–1 ILP model. The transmission routes to site j, which requests the target content having origin root t, should be created

$$\sum_{i \in V} s^t_{(i,j)} = 1 \quad \forall j \in V. \tag{4}$$

Moreover, the starting site of the transmission route should be the cache location of the target content.

$$s^t_{(i,j)} \leqslant u_i \quad \forall i \in V,\ \forall (i,j) \in R_c. \tag{5}$$

Table 1
Variables in the optimal design of cache locations

Variable	Type of variable	Definition
$R_{k,j}$	Given	The request rate for target content k from a destination site j
P_{ca}	Given	Power density for storage (memory) (J/(bit · s))
P_r	Given	Power density of a *CR* (J/bit)
P_{wdm}	Given	Power density of a WDM node (J/bit)
D	Given	Data size of a chunk (bits)
u_i	Design	Binary variable for whether to store a chunk of target content k in CR_i or not
$s^t_{(i,j)}$	Design	Binary variable for whether to select the route from CR_i to CR_j defined on a shortest-path tree rooted at origin site t of a chunk of target content k or not

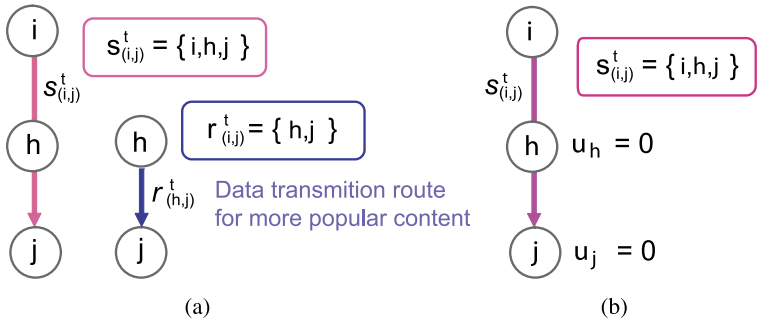


Fig. 3. Constraint conditions. (a) Hierarchical route. (b) Cache location. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

For the hierarchical route constraint, the target content should be cached on the shortest-path tree rooted at its origin site t. Furthermore, there should be more popular content, having the same origin site t as the target content, on the designed transmission route and caching hierarchy should be constructed as shown in Fig. 1.

$$s^t_{(i,j)} = 0 \quad \text{if } \mathbf{r}^t_{(h,j)} \notin \mathbf{s}^t_{(i,j)} \vee \mathbf{s}^t_{(i,j)} \in \{\mathbf{r}^t_{(h,j)} - h\}, \quad \forall (i,j) \in R_c. \tag{6}$$

In Fig. 3(a), $\mathbf{s}^t_{(i,j)}$ is the route sequence along the route $s^t_{(i,j)}$ of the target content (kth popular content). Meanwhile, $\mathbf{r}^t_{(h,j)}$ is the route sequence along the route of more popular content (mth popular content: $m < k$) having the same origin site t as the target content. Furthermore, site h represents the starting site of the route sequence $\mathbf{r}^t_{(h,j)}$ and $\{\mathbf{r}^t_{(h,j)} - h\}$ represents the subsequence excluding the starting site h from the route sequence $\mathbf{r}^t_{(h,j)}$.

Furthermore, the same replicas should not be cached on the designed transmission route of the target content (kth popular content). This constraint is illustrated in Fig. 3(b).

$$s^t_{(i,j)} + u_k \leqslant 1 \quad \forall (i,j) \in R_c, k \in \{\mathbf{s}^t_{(i,j)} - i\}. \tag{7}$$

5. Distributed cache mechanism

In a distributed caching framework such as CCN, it is often difficult to get the request distribution for all content items. Moreover, the optimization model lacks scalability for large-scale networks because it belongs to the class of *NP-complete* problems. Therefore, we presented a distributed mechanism in [12] to locally search for energy efficient cache locations of content (chunks) for the same network model as Fig. 2.

In the proposed mechanism, every *CR* automatically pre-designs a threshold of request rates of content using local information on each caching hierarchy before cache operation. Meanwhile during the cache operation, each *CR* measures request rates of content (initial *Interests* of content) and judges whether or not to cache chunks of the content using the pre-designed threshold as follows.

- When the request rate of content measured by CR_i is higher than the threshold set in CR_i, chunks of the content are stored on CR_i.
- When the request rate of content measured by CR_i is lower than the threshold set in CR_i, chunks of the content are not cached in CR_i.

5.1. Process flow

The threshold design is autonomously executed by each *CR* based on the following process. Here we define a parent *CR*, children *CR*s and branch *CR*s as the router directly over a target *CR*, routers directly under a target *CR* and all routers below a target *CR* on a sub-tree in a caching hierarchy, respectively, as shown in Fig. 4.

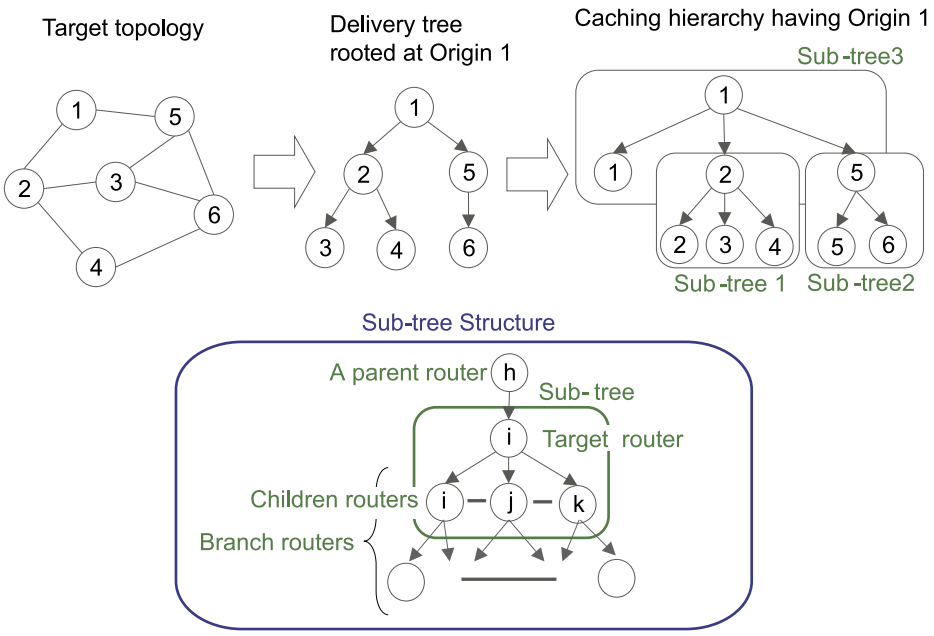


Fig. 4. Example of creating a caching hierarchy and the sub-tree structure. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

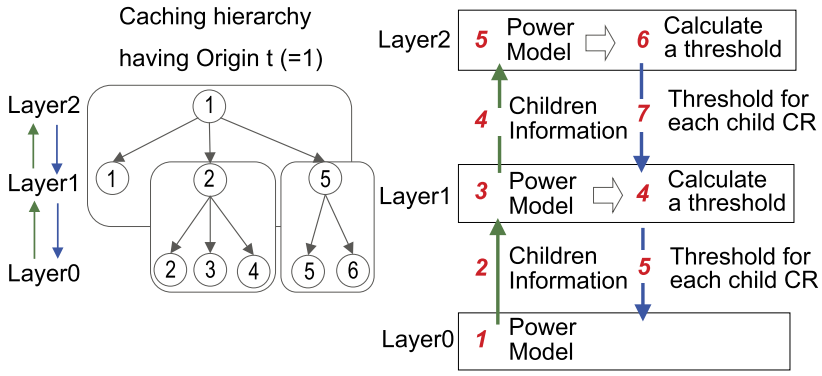


Fig. 5. Bottom-up process. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

For a caching hierarchy,

repeat

Target CR_i

step 1 model power consumption when a chunk is cached on CR_i and transmitted to all branch *CR*s.

step 2 design a threshold for each child *CR* using the power model and sets it to each child *CR* not including CR_i itself.

step 3 send the power model to parent CR_h

($\Rightarrow$ **step 1** for next target $CR_i \leftarrow CR_h$)

until CR_i is the root of the caching hierarchy.

Next, we show a distributed design algorithm of the threshold in Fig. 5.

5.2. Distributed algorithm for threshold design

We assume CR_i can know the following information from children *CR*s.

- Cr_i^t: the total number of children *CR*s for target CR_i on caching hierarchy t;
- Br_i^t: the total number of branch *CR*s for target CR_i on caching hierarchy t;
- H_i^t: the sum of hop-lengths from target CR_i to all branch *CR*s on caching hierarchy t.

For content requested by each site at the rate of λ (requests/s), the total power consumption when its chunk is cached on CR_i on caching hierarchy t and transmitted to all branch *CR*s is defined as

$$Power_i^t(\lambda) = Ca_i + Tr_i\big(\lambda, H_i^t\big), \quad \forall i, \tag{8}$$

Where the *cache allocation power* Ca_i (J) in 1 s when a chunk is cached on CR_i is shown in Eq. (2) and the *traffic transmission power* $Tr_i(\lambda, H_i^t)$ (J) when the chunk is transmitted from CR_i to all branch *CR*s is

$$Tr_i\big(\lambda, H_i^t\big) = D \cdot \lambda \cdot (P_r + P_{wdm}) \cdot H_i^t, \quad \forall i. \tag{9}$$

Next, CR_i designs a threshold using the power model. Under the assumption that content requests are symmetrically generated at all sites, CR_i designs a threshold by the following rule for a sub-tree (cf. Fig. 4) using children information: $Power_j^t(\lambda)$ ($\forall j \in \mathbf{C}_i^t$: a set of children *CR*s).

if "$Power_i^t(\lambda)$: *the power consumption when a chunk having request rate* λ *from each site is cached only on* CR_i *delivering it to all branch CRs*" is greater than "$\sum_{j\in\mathbf{C}_i^t} Power_j^t(\lambda)$: *the power consumption when a chunk having request rate* λ *from each site is cached on* CR_j ($\forall j \in \mathbf{C}_i^t$) *delivering it to all branch CRs*" **then**
the chunk is cached on child CR_j ($\forall j \in \mathbf{C}_i^t$)
else
the chunk is not cached on child CR_j ($\forall j \in \mathbf{C}_i^t$)
end if

CR_i calculates the boundary condition of the above-mentioned rule as follows.

$$Power_i^t\big(\lambda_b^t\big) = \sum_{j\in\mathbf{C}_i^t} Power_j^t\big(\lambda_b^t\big), \tag{10}$$

which leads to

$$\lambda_b^t = \frac{(Cr_i^t - 1) \cdot D \cdot P_{ca}}{D \cdot (P_r + P_{wdm}) \cdot (H_i^t - \sum_{j\in\mathbf{C}_i^t} H_j^t)}. \tag{11}$$

Therefore, the threshold Th_j^t of CR_j ($j \in \mathbf{C}_i^t$) in caching hierarchy t can be derived as

$$Th_j^t = Br_j^t \lambda_b^t. \tag{12}$$

After designing it, CR_i sets the threshold Th_j^t to child CR_j not including CR_i itself and the cache management in each *CR* is executed as follows.

For chunks of content having origin t,

- when the request rate of a content (initial *Interest*) measured by CR_j is higher than Th_j^t, chunks of the content are cached on the CR_j;

Table 2
Variables in the distributed cache mechanism

Variable	Type of variable	Definition
P_{ca}, P_r, P_{wdm}, D	Given	cf. Table 1
λ_b^t	Design	The boundary condition of the request rate of content generated from each site to calculate the threshold Th_j^t
Th_j^t	Design	Threshold of the request rate for CR_j in caching hierarchy t
Cr_j^t	Measure	The total number of children *CR*s of CR_i on caching hierarchy t
Br_j^t	Measure	The total number of branch *CR*s in a delivery tree rooted at CR_i on caching hierarchy t
H_j^t	Measure	The sum of hop-length from CR_i to all branch *CR*s.

- when the request rate of a content (initial *Interest*) measured by CR_j is lower than Th_j^t, chunks of the content are not cached on the CR_j.

Furthermore after calculating the power model such as Eq. (8), each *CR* sends it to its parent *CR*. Here, variables used in the threshold design are summarized in Table 2.

6. Evaluation

We first verified the tradeoff between cache allocation power and traffic transmission power for a chunk of content using the optimization model and evaluated the effectiveness of the distributed cache mechanism.

The simulation conditions are set to the following.

- *Test networks*: NSF topology with 14 *CR*s (Topology A), cf. Fig. 6(a)/US-backbone topology with 24 *CR*s (Topology B), cf. Fig. 6(b).
- *Content information*: Zipf-distributed requests from each site j for $K = 10{,}000$ content items are defined as $R_{k,j} = \lambda = rk^{-\alpha}/c$, $c = \sum_{k=1}^{K} k^{-\alpha}$, cf. Fig. 7(a). We set α to 0.8 for UGC (User Generated Content) and 1.2 for VoD [5] and r to 100. Furthermore, the origin site t of content ID k is set randomly based on a uniform distribution, cf. Fig. 7(b) and (c). The chunk size D and the average content size are set to 10 kB and 10 MB [6,20]. The number of chunks n_k of content ID k follows a geometric distribution $\frac{1}{\sigma}(1 - (\frac{1}{\sigma}))^{n_k}$ with the average number of chunks $\sigma = 1000 = 10\,\text{MB}/10\,\text{kB}$.
- *Power density of each device*: The power density of a memory device (J/(bit · s)) and a *CR* or WDM node (J/bit) are set to the values given in Table 3.

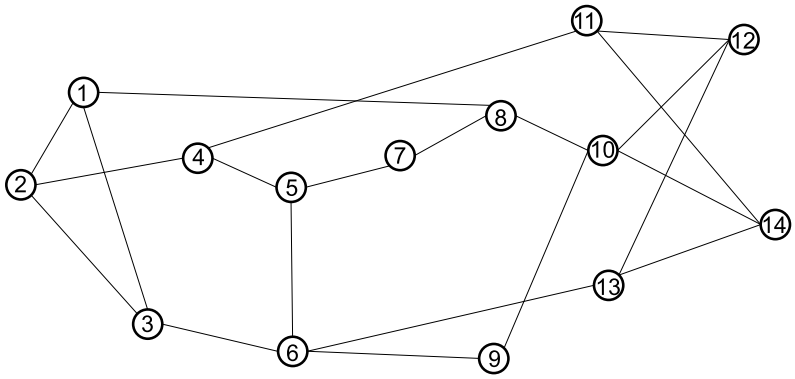
6.1. Tradeoff of power consumption

We now compare power consumption for a chunk of content from a Zipf distribution with $\alpha = 1.2$ when the chunk is allocated using the following cache allocation policies.

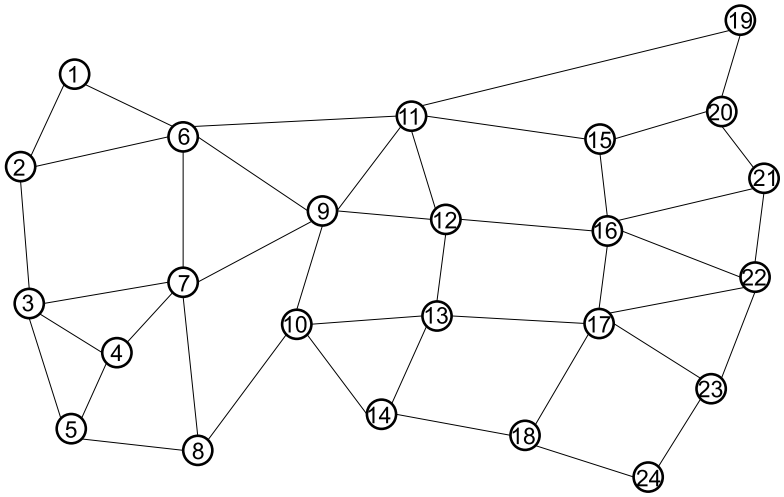
- *Cache allocation on 1 CR*: The replica of a chunk of content is stored in a *CR* on its origin site.
- *Cache allocation on all CRs*: The replicas of a chunk of content are stored in all *CR*s.
- *Optimal cache allocation*: The replicas of a chunk of content are stored in *CR*s designed by our optimization model.

Figures 8 and 10 show the power consumption for a chunk of content in each topology. Furthermore, Figures 9 and 11 present the snapshots of each caching policy for content ID:398 having the origin site 7 in Topology A and content ID:395 having the origin site 12 in Topology B.

In these results, the cache locations for more popular content and less popular content in optimal caching are same as those in caching on all *CR*s and caching on 1 *CR*, respectively. However for content having intermediate popularity, the number of cache locations for a chunk in optimal caching is different from that in the other policies

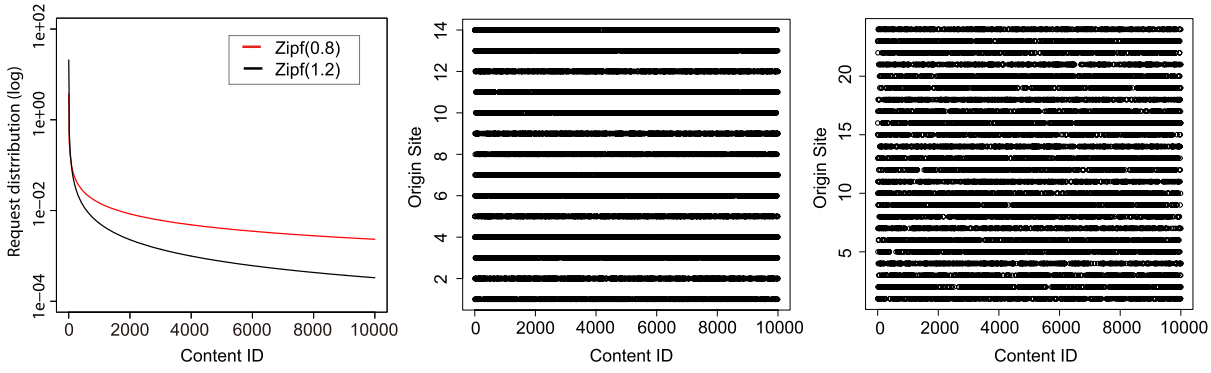


(a)



(b)

Fig. 6. Test topologies. (a) Topology A. (b) Topology B.



(a) (b) (c)

Fig. 7. Request distribution of the content and origin sites of content. (a) Request distribution of the content $R_{k,*}$. (b) Origin sites of content in Topology A. (c) Origin sites of content in Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

Table 3
Power density parameters

Device (Product)	Power	Spec	Power density
DRAM	10 W	4 GB	$P_{ca} = 3.125 \times 10^{-10}$ J/(bit · s)
Content router (CRS-1)	4185 W	320 Gbps	$P_r = 1.3 \times 10^{-8}$ J/bit
WDM (FLASHWAVE9500)	800 W	480 Gbps	$P_{wdm} = 1.67 \times 10^{-9}$ J/bit

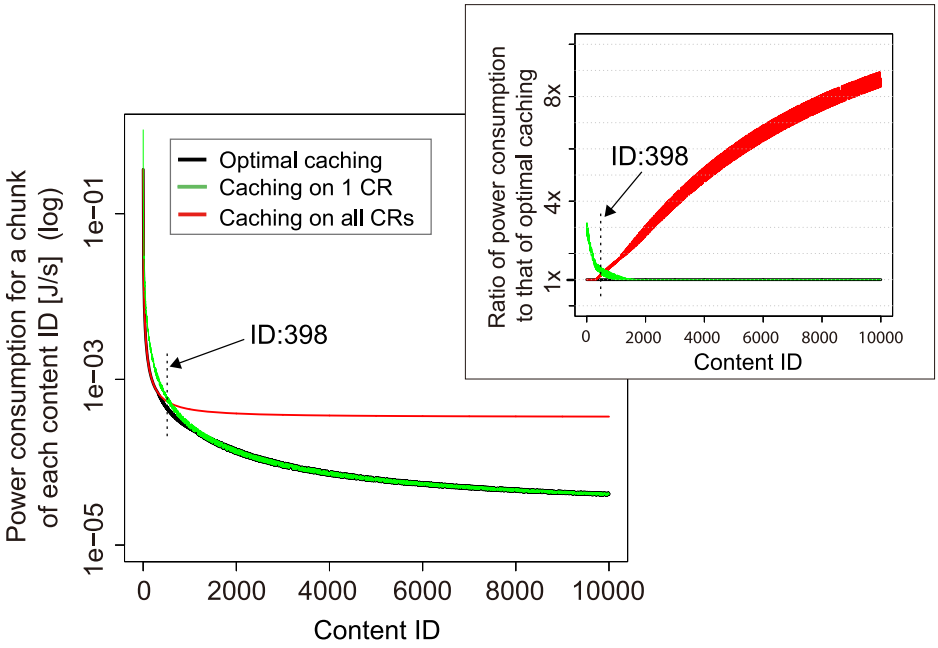


Fig. 8. Power consumption for a chunk of each content ID and the ratio of power consumption to that of optimal caching (Zipf: $\alpha = 1.2$, Topology A). (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

according to its request rates. Moreover, Figs 9 and 11 show the tradeoff relationship between the cache allocation power and the traffic transmission power and different content delivery topologies are derived from each cache allocation strategy. As a result, we see that the optimal cache allocation can derive the most energy efficient locations for a chunk of content by balancing between the cache allocation power and the traffic transmission power according to the content popularity.

Therefore, distributed cache mechanisms for CCN should consider the tradeoff between the cache allocation power and the traffic transmission power and search for efficient cache locations of chunks of content by balancing the tradeoff on the caching hierarchies to be close to the optimal cache locations. Next, we demonstrate the effectiveness of our distributed cache mechanism.

6.2. *Effectiveness of the distributed cache mechanism*

We evaluated the effectiveness of the distributed cache mechanism in chunk-level simulation and compared energy efficiency of three caching policies for a *CR* with different memory sizes (2, 4, 8, 12, 16, 20, 24, 32, 64, ∞ GB).

- *Optimal caching + LFU*: A chunk of content is cached on locations designed by the optimization model. When memory usage in a *CR* is above 100%, the chunk having the lowest request rate is discarded from memory of the *CR* according to LFU.

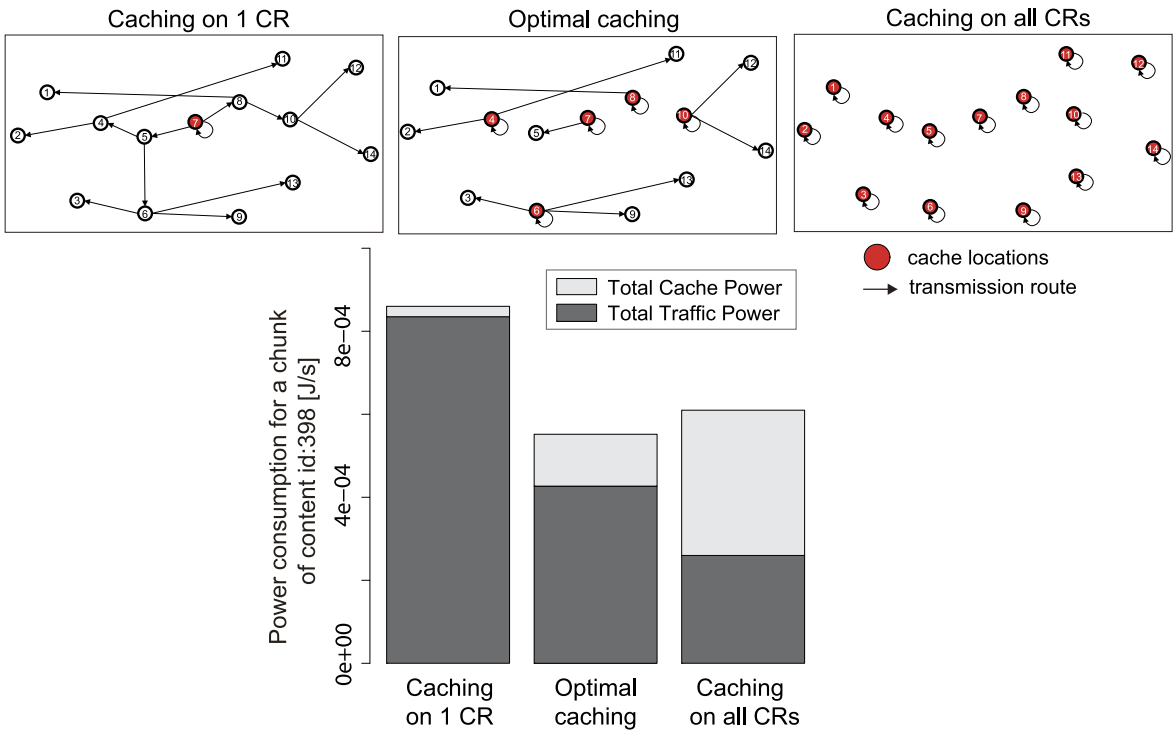


Fig. 9. Power consumption for a chunk of content ID:398 (Zipf: $\alpha = 1.2$, Topology A). (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

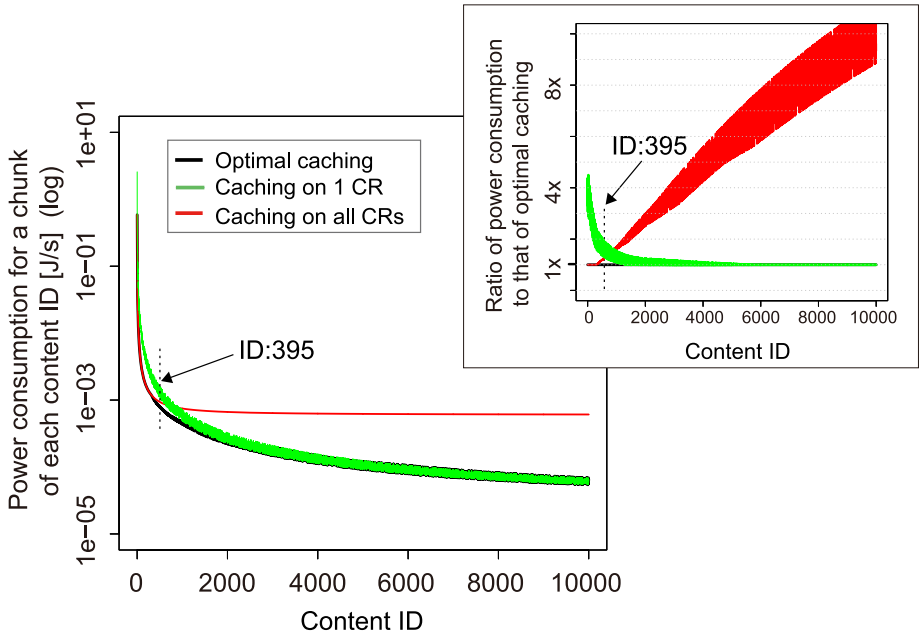


Fig. 10. Power consumption for a chunk of each content ID and the ratio of power consumption to that of optimal caching (Zipf: $\alpha = 1.2$, Topology B). (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

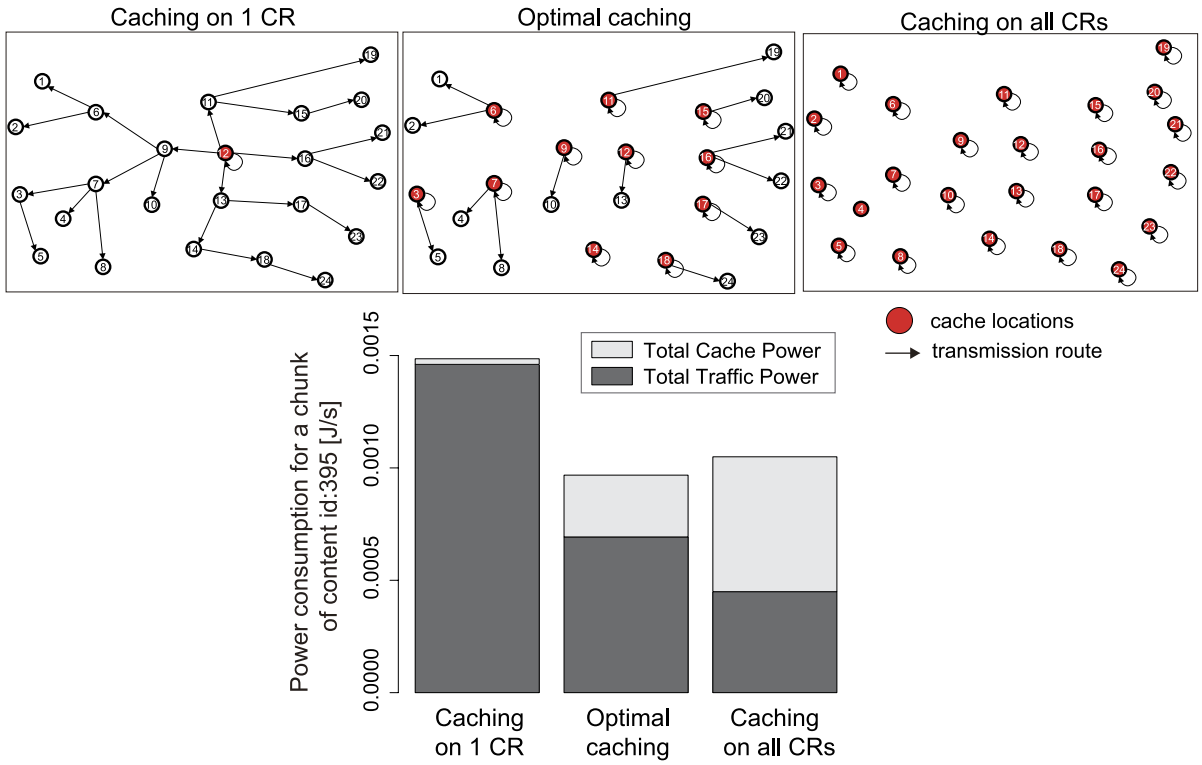


Fig. 11. Power consumption for a chunk of content ID:395 (Zipf: $\alpha = 1.2$, Topology B). (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

- *Threshold-based caching + LFU*: A chunk of content is cached when the request rate of the content is above a pre-designed threshold/is not cached when the request rate is below the pre-designed threshold. Moreover, when memory usage in a *CR* is above 100%, the chunk having the lowest request rate is discarded from memory of the *CR* according to LFU.
- *Pure LFU*: A chunk of content is cached in all *CR*s thorough which it passes. Only when memory usage in a *CR* is above 100%, the chunk having the lowest request rate is discarded from memory of the *CR* according to LFU.

In the simulations, the generation of each request follows an exponential distribution and all *CR*s measure request rates of initial *Interests* of content for Pure LFU and threshold-based caching. The simulation time is set to 7200 s.

We now compare the three caching policies in view of the total power consumption for two topologies. As examples, Fig. 12 shows the time change of total power consumption (J/s), which is the sum of traffic transmission power and cache allocation power, when the memory size of each *CR* is set to 32 and 64 GB. Additionally, Fig. 13 presents differences of the average power consumption, which is the average of total power consumption in the steady state of the simulation when the memory size is changed.

These results shown in Fig. 13(a)–(d) demonstrate that the power consumption in threshold-based caching is only slightly larger than that in optimal caching and lower than that in Pure-LFU. On the other hand, Fig. 13(c) shows larger differences of power consumption between threshold-based caching and optimal caching compared with the other results. In Fig. 14, we additionally show the total power consumption cumulated in the order from more popular content to less popular content for each topology which is composed of *CR*s having infinite memory. As shown in Fig. 14(c), the cumulative differences of power consumption between threshold-based caching and optimal caching are small for more popular content with IDs below 2000, but the cumulative power consumption

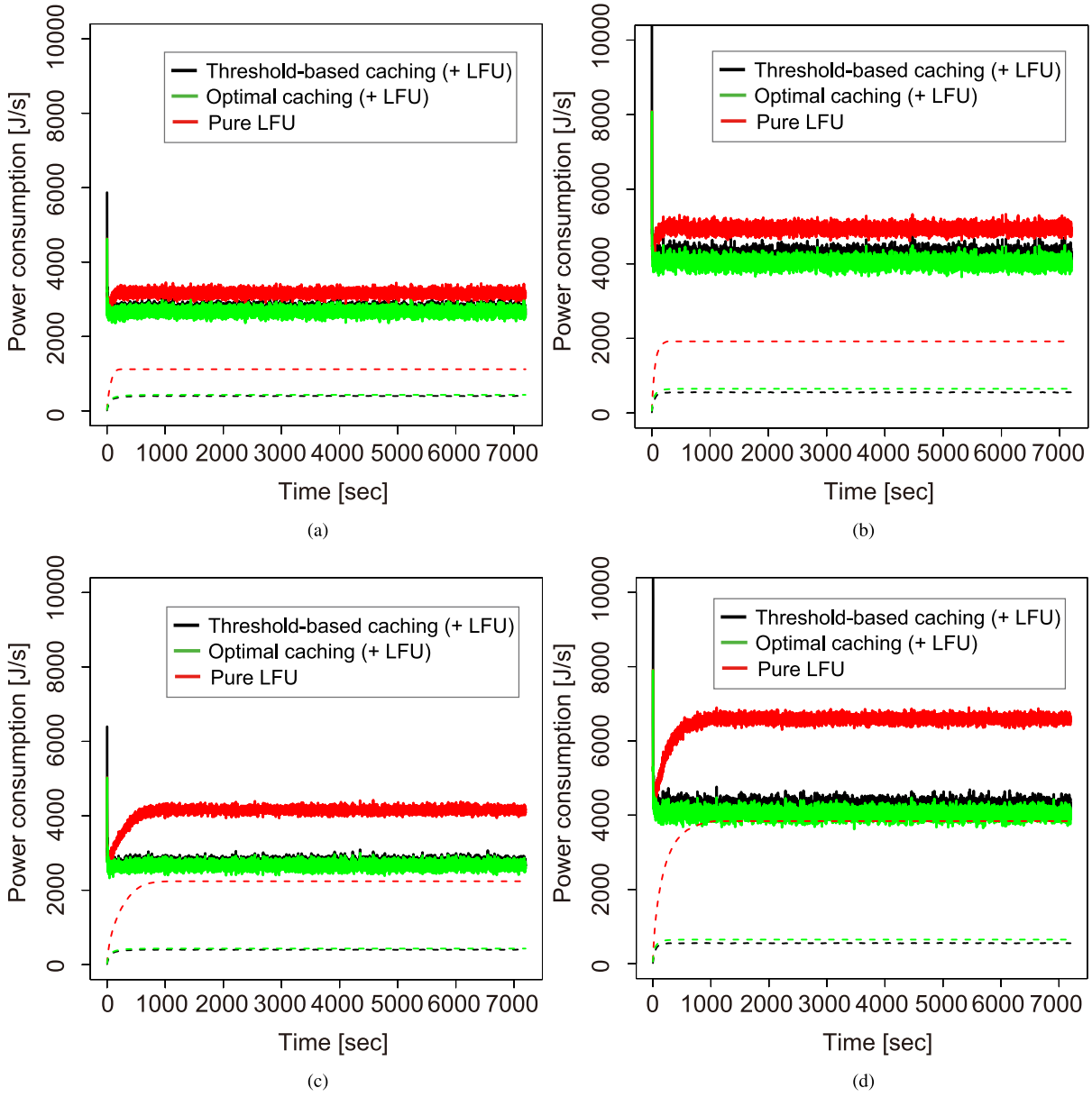


Fig. 12. Example of time change of total power consumption for content following a Zipf distribution with $\alpha = 1.2$ (*solid line*: total power consumption; *dashed line*: cache allocation power in the total power consumption). (a) 32 GB memory, Topology A. (b) 32 GB memory, Topology B. (c) 64 GB memory, Topology A. (d) 64 GB memory, Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

shows larger differences for less popular content with IDs over 2000, in the long tailed distribution. That is why less popular content with $\alpha = 0.8$ has higher request rates and larger difference of power consumption than less popular content with $\alpha = 1.2$.

Furthermore in Fig. 15, we demonstrate the total power consumption for Topology B when the memory size of each *CR* is infinite and the request distribution is changed. For the Zipf-distributed requests from each site as $R_{k,j} = rk^{-\alpha}/c$, we changed α to $\{0.5, 0.8, 1.2, 1.5\}$ and r to $\{50, 100, 150\}$, respectively, as shown in Fig. 16. As

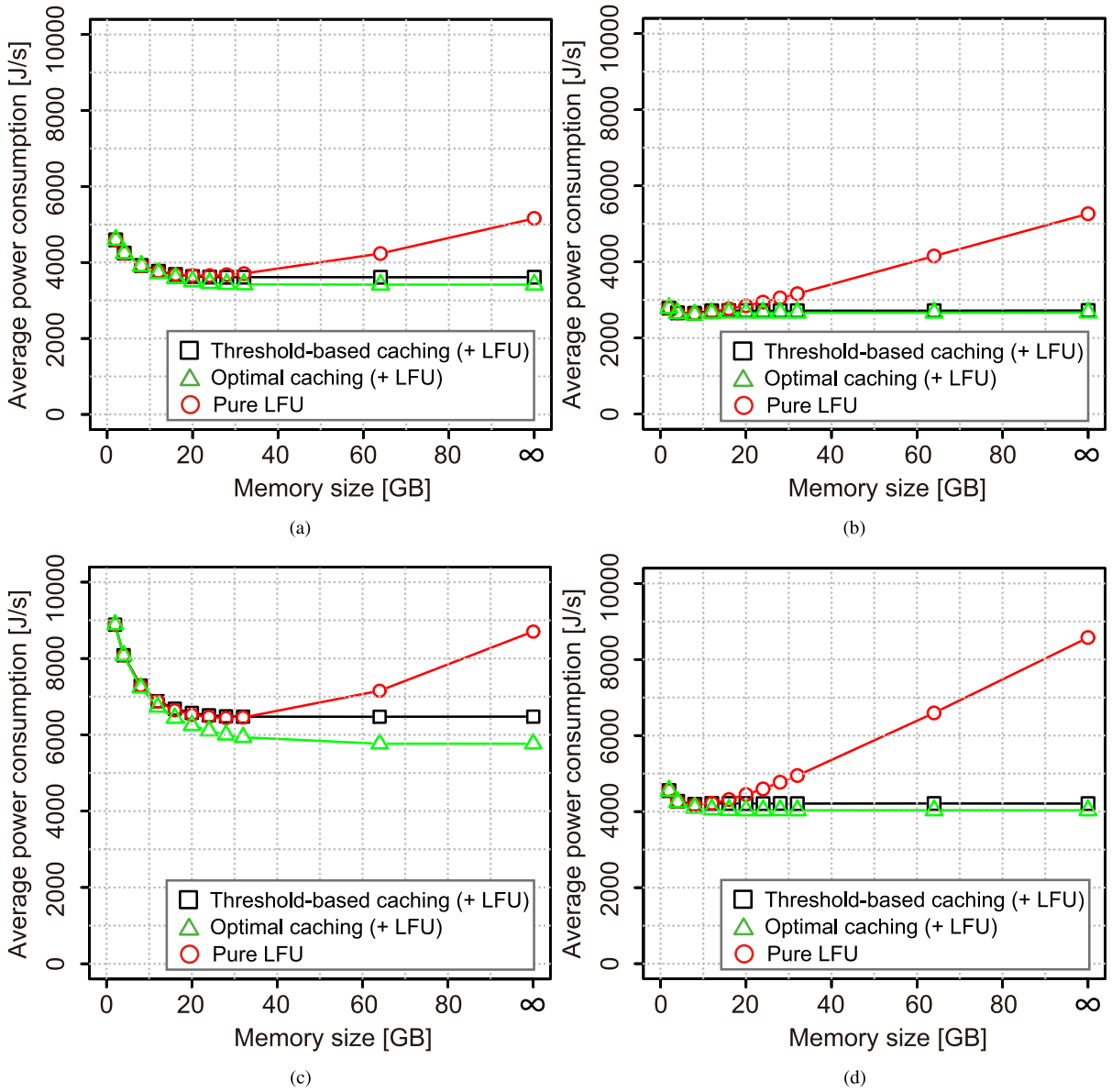


Fig. 13. Average power consumption when the memory size is changed. (a) Zipf: $\alpha = 0.8$, Topology A. (b) Zipf: $\alpha = 1.2$, Topology A. (c) Zipf: $\alpha = 0.8$, Topology B. (d) Zipf: $\alpha = 1.2$, Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

a result, we see that the differences between threshold-based caching and optimal caching are small when α is large or r is small, that is, less popular content has lower request rates. Meanwhile, energy efficiency in threshold-based caching is close to that in Pure LFU as α becomes smaller or r becomes larger.

Moreover in these results, the power consumption in threshold-based caching for content with $\alpha = 1.2$ is smaller than that for content with $\alpha = 0.8$ because there are many contents having the request rates over the threshold for content with $\alpha = 0.8$, which discloses that the memory usage for content with $\alpha = 0.8$ is larger than that for content with $\alpha = 1.2$ in Fig. 17.

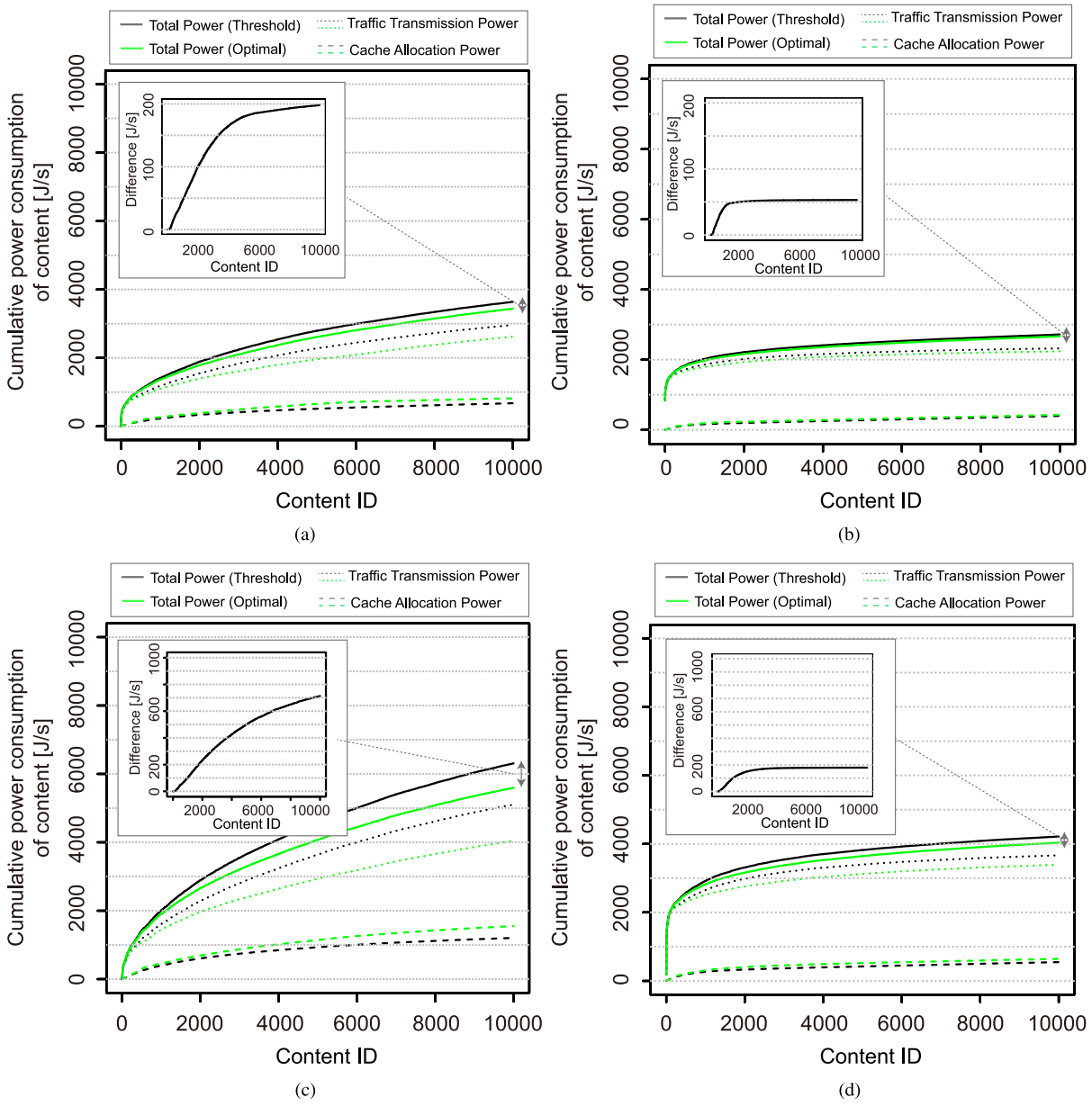


Fig. 14. The cumulative power consumption of chunks of content for threshold-based caching and optimal caching when the memory size is infinite. (a) Zipf: $\alpha = 0.8$, Topology A. (b) Zipf: $\alpha = 1.2$, Topology A. (c) Zipf: $\alpha = 0.8$, Topology B. (d) Zipf: $\alpha = 1.2$, Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

In Fig. 17, the total power consumption in threshold-based caching and optimal caching does not change even when the memory size is large, because the memory usage converges on the appropriate usage in view of energy efficiency. Meanwhile, the total power consumption in Pure LFU increases as the memory size is larger. This is why the cache allocation power becomes dominant in the total power consumption according to increasing memory size.

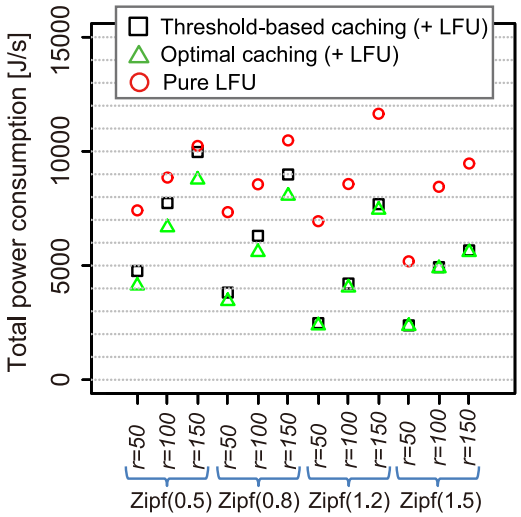


Fig. 15. The total power consumption when the memory size is infinite and the request distribution is changed in Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

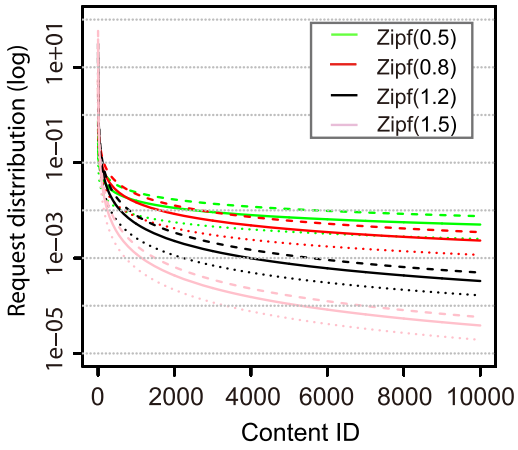


Fig. 16. Request distribution of the content (*dashed line*: $r = 150$, *solid line*: $r = 100$, *dotted line*: $r = 50$). (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

As a result, we can see that the threshold-based caching using local searching realizes more energy efficient caching than Pure LFU and is near to the optimal solution using the whole network information. Meanwhile, these results show that the energy efficiency highly depends on the distribution of content popularity.

We next analyze the cache hit ratio defined as the ratio of chunks cached in the network for each request. Figure 18 shows the average cache hit ratio for chunks of all content items when the memory size of each *CR* is changed. For content with $\alpha = 0.8$ in Fig. 18(a) and (c), the cache hit ratio in threshold-based caching is near to that in Pure LFU when the memory size is small and that in optimal caching only when the memory size is larger. For content with $\alpha = 1.2$ in Fig. 18(b) and (d), the cache hit ratio in threshold-based caching is almost same as that in optimal caching even when the memory size is small. Meanwhile in all cases, the cache hit ratio in Pure LFU is lower than that in the other policies.

Furthermore, the cache hit ratio in threshold-based caching for content with $\alpha = 0.8$ is higher than that for content with $\alpha = 1.2$ because less popular content with $\alpha = 0.8$ has higher request rates and is more easily cached in each *CR* than content with $\alpha = 1.2$. Although the cache hit ratio also depends on the distribution of content popularity, we see that the threshold-based caching can achieve good performance near to the optimal caching and improve not only the power consumption but also the cache hit ratio by effectively using network resources such as memory usage and bandwidth.

Additionally, Fig. 19 shows the average hop length for chunks of all content items when the memory size of each *CR* is changed. For evaluation, we add a penalty of $+5$ to the hop length of content which is not cached on any *CR*s in the network and for which a request (*Interest*) reaches its origin server. Figure 19(b) and (d) illustrate that the average hop length in threshold-based caching is near to that in optimal caching. Furthermore the average hop-length in Pure LFU is longer than the others when the memory size of each *CR* is small, because many chunks of less popular content are not cached in the network due to insufficient memory and many requests of content reach the origin server. Meanwhile, Fig. 19(a) and (c) show the average hop-length in optimal caching and threshold-based caching is close to that in Pure LFU when the memory size is small. This is why chunks of many contents are allocated on many *CR*s as with Pure LFU. Moreover, as the memory size is larger, the average hop length in Pure

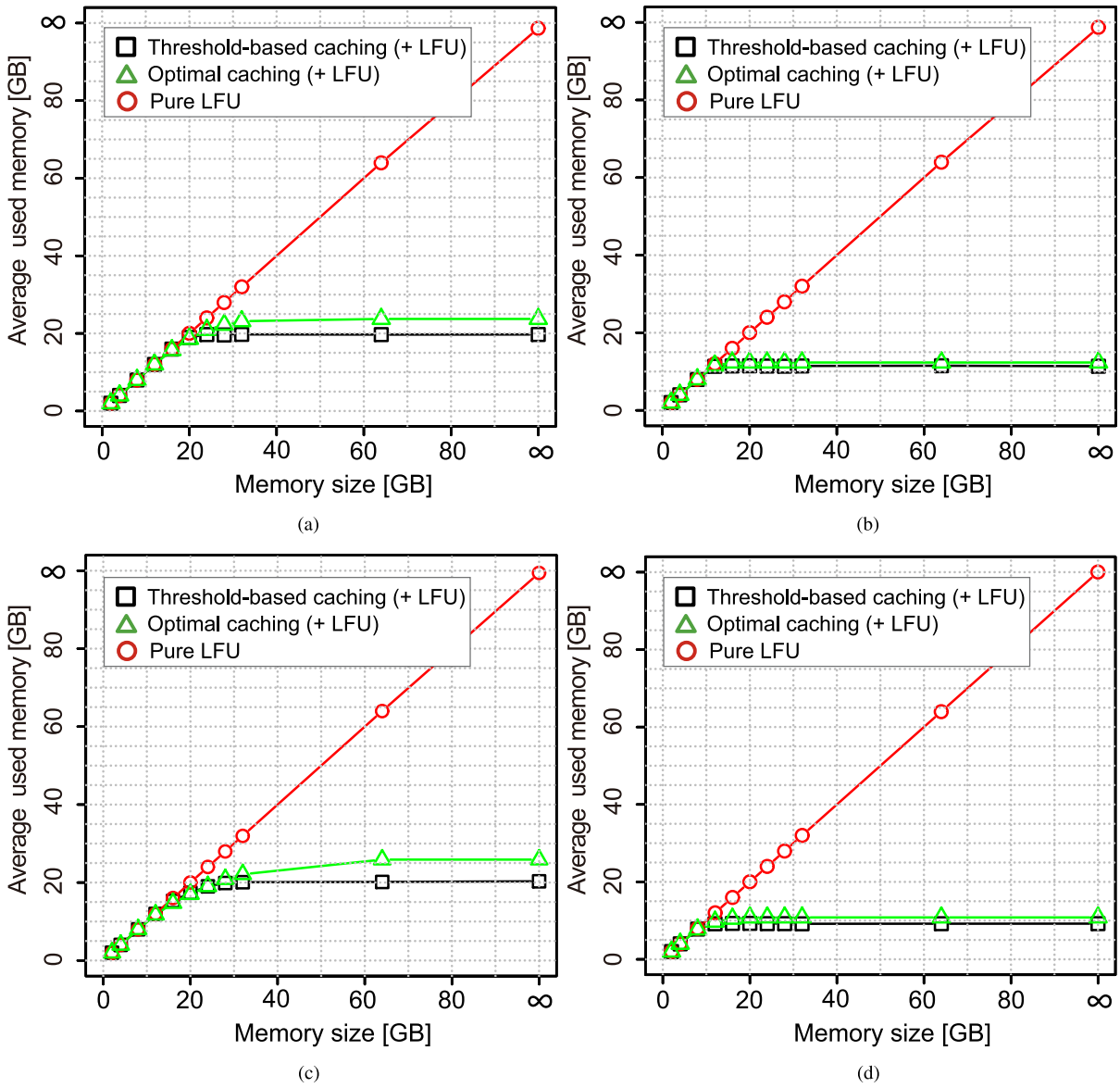


Fig. 17. Average used memory when the memory size is changed. (a) Zipf: $\alpha = 0.8$, Topology A. (b) Zipf: $\alpha = 1.2$, Topology A. (c) Zipf: $\alpha = 0.8$, Topology B. (d) Zipf: $\alpha = 1.2$, Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

LFU becomes shorter than the others. Furthermore, the average hop length in threshold-based caching is longer than that in optimal caching because threshold-based caching discards more data than optimal caching as shown in Fig. 17(a) and (c).

Although the hop length also depends on the distribution of content popularity, we can see that the threshold-based caching can control content locations to be close to the optimal caching in consideration of the tradeoff between hop length i.e., response performance, and power consumption of the network.

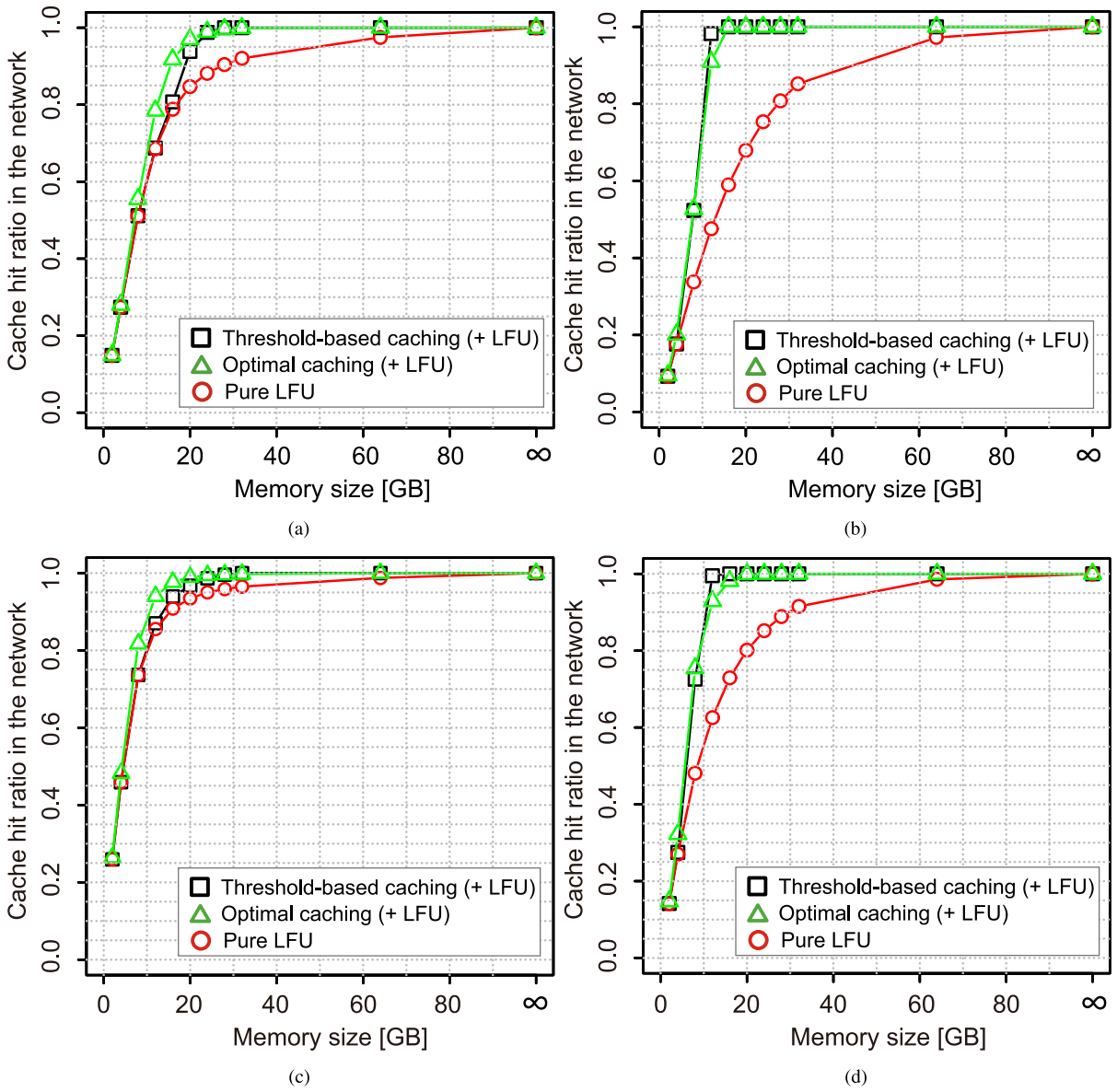


Fig. 18. Cache hit ratio in the network when the memory size is changed. (a) Zipf: $\alpha = 0.8$, Topology A. (b) Zipf: $\alpha = 1.2$, Topology A. (c) Zipf: $\alpha = 0.8$, Topology B. (d) Zipf: $\alpha = 1.2$, Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

7. Conclusions

We introduced an energy efficient design method to derive the optimal cache locations of chunks of content in order to provide reference locations to evaluate energy efficiency for cache strategies, which can consider the tradeoff between the cache allocation power and traffic transmission power under the constraints of the caching hierarchy. Furthermore, we proposed a distributed cache mechanism to locally search for energy efficient cache locations of chunks of content. In the mechanism, each *CR* pre-designs a threshold of request rates of chunks for

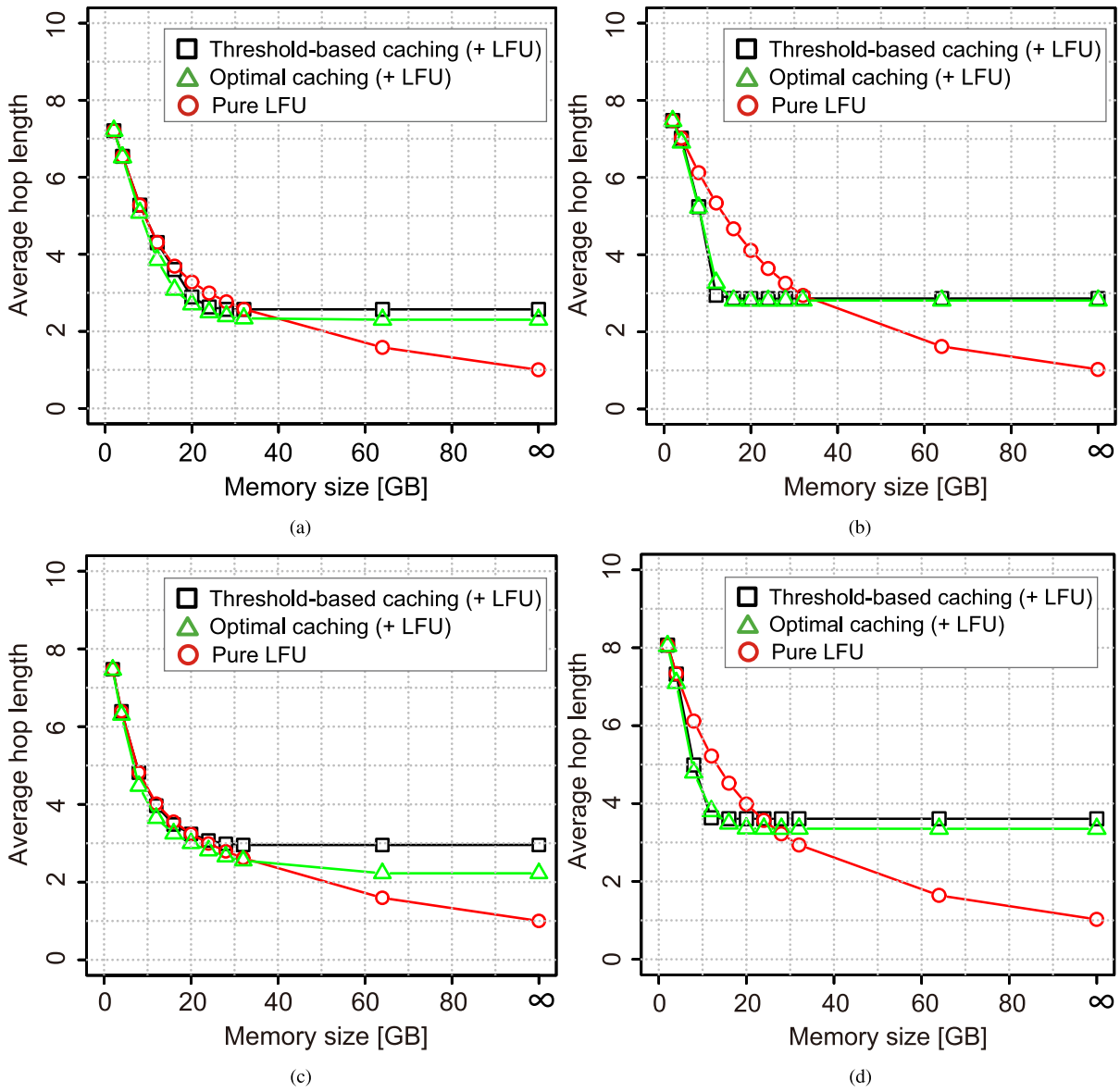


Fig. 19. Average hop length when the memory size is changed. (a) Zipf: $\alpha = 0.8$, Topology A. (b) Zipf: $\alpha = 1.2$, Topology A. (c) Zipf: $\alpha = 0.8$, Topology B. (d) Zipf: $\alpha = 1.2$, Topology B. (Colors are visible in the online version of the article; http://dx.doi.org/10.3233/JHS-130474.)

each caching hierarchy and judges whether or not to cache the chunks by comparing measured request rates with the threshold.

In the simulation, we revealed the tradeoff between the cache allocation power and the traffic transmission power for a chunk of content having different request rates and demonstrated that the proposed distributed caching is near to the optimal solution derived by the proposed optimization model and can improve the total power consumption and the cache hit rate in the target network compared with Pure LFU. Furthermore, we showed that the energy efficiency of the proposed method depends on the distribution of content popularity.

As future work, we plan on enhancing the energy efficient cache mechanism to adapt to heterogeneous traffic conditions in consideration of required response performance.

References

[1] I. Baev, R. Rajaraman and C. Swamy, Approximation algorithms for data placement in arbitrary networks, in: *Proc. of the 12th ACM–SIAM Symposium on Discrete Algorithms (SODA)*, Washington, DC, USA, 2001.

[2] S. Borst, V. Gupta and A. Walid, Distributed caching algorithms for content distribution networks, in: *Proc. of INFOCOM'10*, San Diego, CA, USA, 2010.

[3] G. Carofiglio, V. Gehlen and D. Perino, Experimental evaluation of memory management in content-centric networking, in: *Proc. of IEEE International Conference on Communications (ICC)*, Kyoto, Japan, 2011.

[4] Z. Drezner, *Facility Location: A Survey of Applications and Methods*, Springer, Berlin, 1995.

[5] C. Fricker, P. Robert, J. Roberts and N. Sbihi, Impact of traffic mix on caching performance in a content-centric network, in: *IEEE NOMEN'12, Workshop on Emerging Design Choices in Name-Oriented Networking*, Orlando, FL, USA, 2012.

[6] P. Gill, M. Arlitt, Z. Li and A. Mahanti, Youtube traffic characterization: a view from the edge, in: *Proc. of IMC'07*, San Diego, CA, USA, 2007, pp. 15–28.

[7] K. Guan, G. Atkinson and D.C. Kilper, On the energy efficiency of content delivery architectures, in: *Proc. of the 4th IEEE International Conference on Communications (ICC) Workshop on Green Communications*, Kyoto, Japan, 2011.

[8] M. Gupta and S. Singh, Greening of internet, in: *Proc. of ACM SIGCOMM'03*, Karlsruhe, Germany, 2003, pp. 19–26.

[9] T. Harder, V. Hudlet, Y. Ou and D. Schall, Energy efficiency is not enough, energy proportionality is needed!, in: *Proc. of DASFAA'11*, Hong Kong, China, 2011, pp. 226–239.

[10] T. Hoshino, Expectations on innovative energy-saving technologies of information and communication equipment, in: *Green IT Symposium*, Ministry of Economy, Trade and Industry, 2007.

[11] S. Imai, K. Leibnitz and M. Murata, Energy efficient content locations for in-network caching, in: *Proc. of APCC'12*, Jeju, Korea, 2012.

[12] S. Imai, K. Leibnitz and M. Murata, Energy-aware cache management for content-centric networking, in: *Proc. of First International WorkShop on Energy-Aware Systems*, Communications and Security, Barcelona, Spain, 2013.

[13] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs and R.L. Braynard, Networking named content, in: *Proc. of CoNEXT'09*, Rome, Italy, 2009.

[14] U. Lee, I. Rimac and V. Hilt, Greening the internet with content-centric networking, in: *Proc. of e-Energy*, Passau, Germany, 2010, pp. 179–182.

[15] U. Lee, I. Rimac, D.C. Kilper and V. Hilt, Toward energy-efficient content dissemination, *IEEE Network* **25**(2) (2011), 14–19.

[16] A. Leff, J.L. Wolf and P.S. Yu, Replication algorithms in a remote caching architecture, *IEEE Transactions on Parallel and Distributed Systems* **4**(11) (1993), 1185–1204.

[17] Z. Li and G. Simon, Time-shifted TV in content centric networks: The case for cooperative in-network caching, in: *Proc. of the 4th IEEE International Conference on Communications (ICC) Workshop on Green Communications*, Kyoto, Japan, 2011.

[18] P. Mahadevan, P. Sharma, S. Banerjee and P. Ranganathan, A power benchmarking framework for network devices, in: *Proc. of NETWORKING'09*, Vol. 5550, Aachen, Germany, 2009, pp. 795–808.

[19] L. Qiu, V.N. Padmanabhan and G.M. Voelker, On the placement of web server replicas, in: *Proc. of INFOCOM*, Anchorage, AK, USA, 2001.

[20] D. Rossi and G. Rossini, Caching performance of content centric networks under multi-path routing, Technical report, Telecom Paris Tech., 2011.