IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 27, 2018, accepted March 30, 2018, date of publication April 6, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2823722

An In-Network Caching Scheme Based on Energy Efficiency for Content-Centric Networks

YING AN[1] **AND XI LUO**[2]
[1]Institute of Information Security and Big Data, Central South University, Changsha 410083, China
[2]Department of Information Technology, Hunan Police Academy, Changsha 410000, China

Corresponding author: Ying An (anying@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402541 and in part by the Major Science and Technology Special Project of Hunan Province under Grant 2017SK1040.

ABSTRACT Content-centric networking (CCN) has emerged as a promising architecture for future Internet due to its in-network caching capability and the receiver-driven content retrieval paradigm. Recently, the growing energy consumption driven by explosive increase of network traffic has become a key issue in CCN and caused widespread academic concern. In this paper, we construct a model to analyze the energy consumption of content distribution in CCN, and propose an energy efficiency based in-network caching scheme. In this scheme, a judging condition is designed to reduce the total energy consumption of content dissemination, and then in combination with content popularity and node importance, a cache placement strategy is proposed to optimize the selection of caching nodes. Furthermore, a neighbor cooperation-based cache replacement strategy is also proposed, which uses the cache resource of neighbor nodes to increase the chances of content being cached and improve the quality of caching service and resource utilization. Simulation results demonstrate that our scheme can outperform the existing schemes in terms of the high cache hit rate, the low average response hops, and the low whole energy consumption.

INDEX TERMS Content-centric networking, energy efficiency, in-network caching, neighbor cooperation.

I. INTRODUCTION

Today, with the rapid development of Internet technology, new network applications emerge endlessly, the content-oriented and personalized information service has become the prevailing trend of network development [1], [2]. However, the traditional host-based Internet architecture cannot satisfy the growing content access requirements due to its lack of the native support for the content-oriented data distribution service [3]. To better address the Internet usage shift from a sender-driven end-to-end communication paradigm to a receiver-driven content retrieval one, many innovative information-centric networking (ICN) architectures have been proposed [4]. Content-centric networking (CCN) [5] is one instance of ICN, which has emerged as a promising architecture for future Internet and attracted vast attention. On the basis of CCN, a similar architecture called Named Data Networking (NDN) [6] follows the basic CCN functionality with some modifications and has developed into the fastest moving project in the NSF Future Internet Architecture program. In CCN, content identification, routing and retrieval are all realized by using content names independently of where the content host is located. Furthermore, ubiquitous in-network caching [7]–[9] is generally adopted to relieve the severe pressure on network bandwidth driven by dramatic growing network traffic. It requires every node has the content caching capability so that the network-wide cache makes the network not only a content transmission carrier but also a content storage carrier. Although ubiquitous caching can improve the performance of content delivery, it may also bring a large amount of redundant caching which worsens the resource utilization and energy efficiency.

Several works have been done recently to use in-network storage and content-oriented routing to improve the efficiency of content distribution in CCN [10]–[14]. Unfortunately, these researches on in-network caching mainly focus on improving network resource utilization, however, the energy efficiency issues have not been fully studied yet. In fact, Internet has become one of the leading players in energy consumption as the dramatic increase of network users and traffic. It is reported that the energy consumption of Internet is estimated to account for up to 10% worldwide energy consumption and keeps constantly increasing [15]. Therefore, the energy consumption has been an unnegligible problem in the research of CCN caching technology.

In this paper, we propose an Energy-Efficiency Based In-network Caching Scheme for CCNs. Compared to the existing works, the contribution of this paper is as follows.

(1) We establish an energy consumption model for CCN content delivery and find that the energy efficiency of content caching depends largely on the request rate of content and the caching node's distance from the content server. On this basis, a judging condition for energy efficiency optimization is given to reduce the overall network energy consumption.

(2) A probabilistic cache placement strategy based on energy efficiency is proposed. On the premise of meeting the energy efficiency judging condition, each node uses the frequency and recency of content access to estimate the content popularity, and then the content is cached with some probability which is determined by the content popularity and the betweenness centrality of node in order to achieve the tradeoff between energy efficiency and cache performance.

(3) A neighbor cooperation based cache replacement strategy is also proposed. In this strategy, when a node runs out of cache space, it will transfer the low popularity contents to the neighbors with enough available resources to reduce cache replacements. If there is no free space in the neighbors, a cooperative replacement based on content popularity is realized to increase the time that the content is cached in network.

The remainder of the paper is organized as follows. In Section 2, the related work is introduced. In Section 3, we present the system model. Section 4 presents the design details of the proposed energy efficiency based in-network caching scheme. In Section 5, we evaluate our caching scheme by extensive simulations on NS-3. Finally, we conclude our work in Section 6.

II. RELATED WORK

Although it is still under debate whether the ubiquitous caching can improve the performance in CCN [16], a lot of research about in-network caching has been done in recent years. Some researchers design cache strategies according to the characteristics of content, such as popularity, size, etc. For example, Ming et al. propose a cooperative caching scheme based on content location and popularity, in which, the more popular the content is and the closer the content is to users, the longer the content will be cached by the routers [17]. This scheme is based on the hypothesis that network topology and content popularity are both preliminarily known, however, these information are very difficult to acquire in actual networks. Kim *et al.* [18] propose a producer-driven differentiated caching scheme. In this scheme, the contents are marked with different service classes by the corresponding producers according to the content popularity and request rate. And then the core routers can easily make caching decisions depending on the service class type of content and the current network conditions. In [19], the Round-Trip Time (RTT) is considered as a key indicator of the user-centric performance, and then a RTT-based caching scheme is proposed from the perspective of the reduction of the RTT. In this scheme, the chunk with large router computed RTT will be cached with a high probability, meanwhile the chunk with the smallest value of the product of the router computed RTT and the request frequency will be preferentially replaced. Moreover, other similar caching schemes of this type also include RBC-CC [20], RCBP [21], PPC [22], OFAM-CCN [23], etc.

Other scholars believe that some information about the node's properties is vital to the cache decision-making in CCN. In [24], a centrality based selective caching algorithm is proposed. In this algorithm, betweenness centrality is adopted as a metric of node importance and contents are only cached at the nodes having the highest betweenness centrality along the delivery path. Bernardini et al. argue that the nodes with more social relationships (which are called influential nodes) tend to receive more attention from others so that the contents produced by them are also more likely to be consumed by others. On this basis, a socially-aware caching scheme is proposed, which evaluates the influence of users by using some centrality measures, and then the content published by influential users will be given priority to cache proactively [25].

As mentioned above, most existing works aim to optimize cache performance from the perspective of network resource utilization by considering the properties of node and/or content. However caching of content in network inevitably raises issues related to energy consumption. As the growth of network traffic, energy efficiency of networking systems has become an urgent issue due to increasing energy costs. Through a deep analysis about energy-efficient caching techniques in ICN from the placement, content placement and request-to-cache routing perspectives, Fang et al. point out that energy efficient caching is still a challenging research area. There are still a lot of issues that need to be addressed, although some existing caching technologies can improve the energy efficiency of ICN to some extend [29]. Braun et al. evaluate the impacts of different caching strategies on network performance from the aspect of energy consumption, and moreover, some opening issues are also discussed further [30]. Llorca *et al.* [31] provide an information-centric optimization framework for the energy efficient dynamic caching problem by comprehensively considering the temporal and spatial dynamics of content popularity and the heterogeneity of network. In this framework, an offline cache placement strategy based on integer linear program is proposed firstly, which uses the globe knowledge of user requests and network resources to obtain the maximum efficiency gains. And then a distributed online caching algorithm is also presented, in which each node makes caching decision independently according to its estimate of the global energy benefit. Li *et al.* [32] translate the energy optimization of in-network caching into minimizing the average response hops of content dissemination and then develop an aging popularity based caching scheme for CCN. Fang *et al.* [33] formulate the energy-efficient distributed in-network caching problem as a non-cooperative game and then propose an energy-efficient distributed caching scheme. In this scheme, each node synthesizes both transport energy consumption and caching energy consumption to make its caching decision locally. Additionally, some other caching schemes are also proposed to reduce the energy costs in different ways in [34]–[38]. Although these works optimize the energy efficiency to a certain extent by limiting the energy consumption of content caching and delivery, the impacts of the features of users and contents on network performance are not fully considered.

In terms of caching decision in CCN, the closer the cached location of content is to the user, the faster the content can be acquired. On the other hand, the closer the cached location of content is to the content server, the less the content is replicated and also the less the energy consumption for caching of the content is. Therefore, it is vitally important for the design of CCN caching strategy to find a tradeoff between quality of service and energy consumption. This paper proposes a novel cooperative caching scheme, in which caching nodes are selected based on energy efficiency, content popularity and node importance. Meanwhile, when the cache space of node is insufficient, some contents are picked out and transferred to the neighbors instead of being replaced directly.

III. SYSTEM MODEL

In this section, we first briefly present an overview of CCN. Then a simple network model and an energy consumption model for CCN content distribution is formulated. To make the presentation easier to follow, we briefly summarize the notations of the key parameters in Table 1.

TABLE 1. **Notations of The key parameters.**

Symbols	Notations
v_i	The i-th node in the network
O_k	The k-th content object in the network
t	Investigated time interval
s_k	The size of the content object O_k
q_i^k	The request rate for O_k at node v_i
ω_c	Power density of caching in content router(W/bit)
ω_r	Energy density of a router(J/bit)
ω_l	Energy density of a link(J/bit)
h_{ij}	Distance in hops between v_i and v_j
E_t	The transport energy consumption
E_c	The caching energy consumption
E_{tot}	The total energy consumption
f_i^k	The request frequency for O_k at v_i
P_k	The popularity of content O_k
P_{max}	The maximum popularity of the contents cached at node v_k
t_{now}	The current time
t_j	The past arrival time of the j-th request for content O_k
λ	Adjusting parameter
$p_c\left(k\right)$	The probability of content O_k being cached by node v_i
θ	A weight coefficient which is used to regulate the impacts of content popularity and node betweenness centrality on caching probability
B_i	The betweenness centrality of node v_i
B_{max}	The highest betweenness centrality value among all the nodes in the delivery path of content O_k
$L\left(v_i\right)$	The total size of accumulated cache contents in sampling process
$C\left(v_i\right)$	The capacity of the cache
$R_{ccl}\left(v_i\right)$	The accumulated cache load ratio

A. OVERVIEW OF CCN

Content-Centric Networking (CCN) is a new Internet architecture with routing and caching centered on contents. A complete content can be a combination of multiple fragments. Communication in CCN is driven by receivers i.e., data consumers, through the exchange of two types of packets: Interest packet and Data packet. Both types of packets carry a name which is independent of the location information and can uniquely identify a complete content or any fragment(s) of a content that can be transmitted in one Data packet. Along with these two packets, three data structures are maintained at each node to properly carry out the packet forwarding functions: a Pending Interest Table(PIT), a Forwarding Information Base (FIB), and a Content Store (CS). The PIT stores the information about unsatisfied Interests including content name, incoming and outgoing interface(s). The FIB is analogous to the IP table maintained on Internet routers. Every entry in the FIB is the tuple of name prefix and outgoing face(s) which is used to forward the Interest packets. The CS is a cache used to temporarily store the received Data packets.

The content retrieval procedure in CCN can be briefly described as follows: When a content is required, the consumer will put the name of the desired content into an Interest packet and send it to the network. Routers use the name to forward the Interest packet to the content source. Each CCN router that receive the Interest packet first search in its CS for matching content; if there is a hit, the router returns the matched Data packet on the interface from which the Interest packet came. Otherwise, the router looks up the name in its PIT. If the specified name is found, the incoming interface of this Interest will be added in the PIT entry for later returning the matched content through the reverse path to the consumers. In the absence of a matching PIT entry, the router will forward the Interest packet toward the content source(s) based on information in the FIB as well as the corresponding forwarding strategy. When the desired content is sent in reply to the corresponding Interest packets, it always takes the reverse path of the Interest packets. Each CCN router on the path will remove the corresponding entry in its PIT whenever the Interest is satisfied and determines whether to replicate the content according to the caching strategy.

B. A NETWORK MODEL

We assume that an arbitrary topology of CCN can be modeled as an connected graph $G = (V, E)$, in which the set of routers is denoted by $V = (v_1, v_2, \ldots, v_n)$, $U \subseteq V$ stands for the set of end nodes and $E \subseteq V \times V$ is the set of network bidirectional links. Let $O = (O_1, O_2, , O_m)$ represent the set of available contents in the network. In the initial state, all the contents are distributed in the Original Content Servers (OCS) which are located near the edge of network and connect to the edge routers. The end nodes are mainly responsible for collecting the interest packets of users and transmitting them to the network. For any router v_i, each is equipped a cache that can store up to M_i contents.

C. AN ENERGY CONSUMPTION MODEL

This subsection provides a model to analyze the energy consumption of CCN. In CCN, the energy is mainly consumed by content caching and content transmission. Therefore, the total energy consumption E_{tot} is composed of two major parts: the caching energy consumption E_c and the transport energy consumption E_t, namely, $E_{tot} = E_c + E_t$.

For any router v_i and any content O_k, let s_k be the size of O_k and q_i^k represent the request rate for O_k at node v_i, in terms of request counts within the investigated time interval t. Moreover, we suppose that ω_c is the power density of caching in content router, ω_r is the energy density of a router, and ω_l is the energy density of a link. According to the energy consumption model proposed in [32], the caching energy consumed at v_i for the storage of O_k during t time interval can be expressed by:

$$E_c = \omega_c s_k t \tag{1}$$

The transmission energy consumption mainly consists of the energy consumption at routers and energy consumption along the links. The former is the energy consumed by the router for handling the data in its forwarding process, and the latter is mainly contributed by Wavelength Division Multiflexing (WDM) or the optical repeaters in the long-haul transmission. For any requesting node v_j, let h_{ij} be the distance in hops between v_i and v_j, then the number of routers along the path from v_i to v_j is $h_{ij} + 1$. Thus, the energy consumption at the router can be expressed as $q_i^k s_k \omega_r \left(h_{ij} + 1\right)$, and the energy consumption along the links is $q_i^k s_k \omega_l h_{ij}$. Therefore, the transmission energy required for transmitting O_k from v_i to v_j can be expressed by:

$$E_t = q_i^k s_k \left[h_{ij}\left(\omega_r + \omega_l\right) + \omega_r\right] \tag{2}$$

Thus, the total energy consumption can be expressed by:

$$E_{tot} = \omega_c s_k t + q_i^k s_k \left[h_{ij}\left(\omega_r + \omega_l\right) + \omega_r\right] \tag{3}$$

Due to the effect of in-network caching, the user requests are generally served by the caching nodes while the OCSs are infrequently accessed. Thus, the energy consumption of the OCSs is ignored in this paper so as to simplify the analysis. Therefore, if node v_j obtains content O_k from the OCS directly, the total energy consumption E'_{tot} mainly consists of the transmission energy consumption at routers and the transmission energy along the links. Assuming that the distance between the OCS of O_k and the requesting node is h_{sj}, then E'_{tot} can be expressed by:

$$E'_{tot} = q_i^k s_k \left[h_{sj}\left(\omega_r + \omega_l\right) + \omega_r\right] \tag{4}$$

IV. AN ENERGY EFFICIENCY BASED IN-NETWORK CACHING SCHEME

The major aim of in-network caching is to speed up the response for potential data access of users and improve the overall network performance by using the caching ability of intermediate nodes. It will inevitably increase the requirement for cache space and energy consumption. So, the tradeoff between energy efficiency and network performance is very important for the design of CCN caching strategies. In this subsection, we introduce an energy-efficient cooperative caching scheme, which mainly includes two parts: an Energy Efficient Cache Placement (EECP) strategy and a Neighbor Cooperation based Cache Replacement (NCCR) strategy. In EECP, an energy efficiency judging condition is determined based on the energy consumption model above so as to optimize the energy efficiency of caching decision and then contents are cached with some probability which is determined by the content popularity and node centrality when the judging condition is satisfied. If the cache space of caching node is insufficient, the NCCR strategy is executed to use the cache resources of neighbors to improve the quality of cache service and resource utilization as much as possible. The principle of the proposed caching scheme is described as follows.

A. ENERGY EFFICIENCY JUDGING CONDITION

In CCN, the content caching of intermediate nodes makes the contents close to the end nodes so that the contents can be obtained within a short distance in the future. From the aspect of energy consumption, it actually tries to reduce the transmission energy consumption and response delay for the future content delivery at the cost of certain caching energy consumption and storage resource. However, when some intermediate node v_i caches the content O_k, in order to reduce the total energy consumption required for content retrieval, it is necessary that the sum of the energy consumed at v_i for the storage of O_k and the transmission energy required for obtaining O_k from v_i must be less than the total energy required for getting O_k direct from the OCS. According to the model given in subsection 3.2, we have:

$$\omega_c s_k t + q_i^k s_k \left[h_{ij}\left(\omega_r + \omega_l\right) + \omega_r\right] < q_i^k s_k \left[h_{sj}\left(\omega_r + \omega_l\right) + \omega_r\right] \tag{5}$$

The above inequality can be simplified as follows:

$$\frac{q_i^k}{t} > \frac{\omega_c}{\left(\omega_r + \omega_l\right)\left(h_{sj} - h_{ij}\right)} \tag{6}$$

According to the physical meaning of each variable, the left side of Function (6) is the request frequency for O_k at v_i, which is denoted by f_i^k; in the right side, $h_{sj} - h_{ij}$ is the hop distance between the OCS and v_i, which is replaced by h_{si}. ω_c, ω_r and ω_l are the given device parameters, which are all with fixed values. For simplicity, we assume that the cache device equipped in each node is dynamic random access memory (DRAM), and the communication is conducted via Wavelength Division Multiplexing (WDM) optical fiber cables. The values of the three device parameters are referenced to [32] as shown in Table 1. Therefore, Function (6) becomes to

$$f_i^k > \frac{\omega_c}{(\omega_r + \omega_l)\, h_{si}} \tag{7}$$

In our proposed caching scheme, a hop count field *hops* is added to the header of each content packet. When the requested content is found at some node, the node will create a response content packet and the value of *hops* is initialized to zero. On the way that the response content packet returns to the requesting user, the value of *hops* increases hop by hop to record the distance from the content responder to the current node. Thus, we choose Function (7) as the energy judging condition. In the caching decision-making, the request frequency of the specified content and the hop distance to the content source are substituted into Function (7) for calculating, and the content will not be cached by the current node if the energy judging condition cannot be met.

B. ENERGY EFFICIENT CACHE PLACEMENT STRATEGY

When the energy judging condition mentioned above is satisfied, it means that choosing the current node as the new caching node can decrease the total energy consumption required for obtaining the content next time. Then the content will be cached with some probability by synthetically considering the node-related and content-related attributes. From the angle of content, the popularity reflects the demand of users for some content to some extent. That is to say, the more popular the content is, the more users interested in it there are, and the more likely the content is requested in the future. Thus, the popular content should be cached preferentially by the node. In the existing research works, the popularity of content is mostly assumed as a predetermined attribute. However, the interest of users in some content often varies with time in practice, which results in corresponding changes of the content popularity. Therefore, in EECP scheme, we borrow the method proposed in [39], in which each node estimates the content popularity by considering both the frequency and recency of content access. The calculating method is detailed as follows.

$$P_k = \sum_{j=1}^{n} \left(\frac{1}{2}\right)^{\lambda (t_{now} - t_j)} \tag{8}$$

Here, P_k is the popularity of content O_k estimated by the current node. t_{now} represents the current time and t_j represents the past arrival time of the j-th request for content O_k. λ is an adjusting parameter and $\lambda \in [0, 1]$, it is used to allow a trade-off between frequency and recency. Obviously, the smaller the value of λ is, the more the frequency contributes to the content popularity. Conversely, the larger the value of λ is, the greater the influence of recency on the content popularity is. According to the analysis result in [39], the value of λ is set to e^{-4} in this paper. The higher popularity the content has, the more likely the content is cached by the current node.

From the angle of node, the nodes with relatively high centrality usually have much stronger connection capability than the others, and they have more opportunities to receive the requests from the other nodes. It can obtain higher cache hits and faster response time by caching the contents at the high centrality nodes. Therefore, we choose the ego network betweenness centrality as another metric to determine the caching probability of content. The higher betweenness centrality the node has, the more likely it is selected as the caching node.

Thus, when content O_k arrives at node v_i, the probability of O_k being cached by v_i, denoted by $p_c(k)$, will be calculated according to the popularity of O_k and the betweenness centrality of v_i, namely,

$$p_c(k) = \theta \frac{P_k}{P_{max}} + (1-\theta) \frac{B_i}{B_{max}} \tag{9}$$

where, θ is a weight coefficient which is used to regulate the impacts of content popularity and node betweenness centrality on caching probability. In this paper, we set $\theta = 0.5$. P_{max} is the maximum popularity of the contents cached at node v_k, B_i is the betweenness centrality of node v_i, and B_{max} is the highest betweenness centrality value among all the nodes in the delivery path of O_k.

Generally speaking, the betweenness centrality need be calculated based on the full network topology. However, it is not practical for the nodes to efficiently obtain the knowledge of delivery paths between all pairs of nodes in CCN. So we borrow the method used in [24] and choose the ego network betweenness centrality to approximate the betweenness centrality in our paper. The ego network can be defined as a special network consisting of a node together with all of its immediate neighbors and all the links among those nodes. In order to build the ego networks, each node only need to broadcast the list of its one-hop neighbors in one hop range, and then according to the received neighbor lists, the ego network can be constructed by adding links that connect to itself or its own neighbors. Due to the simple structure of ego networks, for any node v_k, its ego network betweenness can be easily calculated by $\sum \sigma_{v_i,v_j}(v_k)/\sigma_{v_i,v_j}$. Here, σ_{v_i,v_j} is the number of content delivery paths from v_i to v_j and $\sigma_{v_i,v_j}(v_k)$ is the number of content delivery paths from v_i to v_j that pass through node v_k.

In CCN, content packet is returned to its requester through the reverse path of the corresponding interest packet. Thus, we use the interest packet to record the highest betweenness value among all the nodes it traverses. More concretely, each interest packet has a field to record the highest betweenness value. When the interest packet arrives at some node, it will compare the betweenness value of the node with the current record value of this field. If the betweenness value of the node is larger than the current record value of this field, then the larger value will be written in the corresponding field. At last, the highest betweenness value among the nodes along the transmission path of the interest packet will be recorded. Then the value will be copied onto the data packets when the requested content is found at some node. By doing so, each node in the delivery path can obtain the value of B_{max}

Algorithm 1 Decision-Making Process of EECP on Content O_k for Node v_i

```
if the data packet containing content $O_k$ arrives at node $v_i$ then
    get the values of $h_{si}$ and $B_{max}$ from the data packet containing $O_k$;
    if $f_i^k > \frac{\omega_c}{(\omega_r+\omega_l)h_{si}}$ then
        $p_c(k) = \theta\frac{P_k}{P_{max}} + (1-\theta)\frac{B_i}{B_{max}}$;
        cache $O_k$ with probability $p_c(k)$;
        if $v_i$ is selected as the caching node of $O_k$ then
            $h_{si} = 0, B_{max} = 0$;
            replicate the data packet containing $O_k$ and cache the copy;
        end if
    end if
    forward the data packet containing $O_k$ to the next hop;
end if
```

from the data packet. The pseudo-code for EECP is shown in Algorithm 1.

C. NEIGHBOR COOPERATION BASED CACHE REPLACEMENT STRATEGY

In most of the existing caching schemes, when the cache resources of node are insufficient, the nodes often release cache space by discarding some contents directly to satisfy the caching requirements of subsequent contents. For the important nodes under heavy caching pressure, the frequent cache replacement will reduce the average time of content cached in network. It results in a significant decrease in cache performance because the cached contents may have been replaced early before they have a chance to response to the user requests. However, there may still be some available cache space in the neighbor nodes. Therefore, we propose a neighbor cooperation based cache replacement (NCCR) strategy, in which the caching node can transfer some cached contents to its neighbor node that has enough cache resources. Thus, the contents can be stored in the network for a relatively long time by making full use of the cache resources of neighbors. And finally, the whole resource utilization and cache performance can be improved.

In order to select the appropriate neighbor node for content migration, the node needs to obtain and compare the cache status of its neighbor nodes. Assume that $L(v_i)$ denotes the total size of accumulated cache contents(the current contents in cache plus the contents replaced) in sampling process, and $C(v_i)$ denotes the capacity of the cache. For any node v_i, let $R_{ccl}(v_i)$ be the accumulated cache load ratio, which is defined as the ratio of $L(v_i)$ with $C(v_i)$, namely,

$$R_{ccl}(v_i) = \frac{L(v_i)}{C(v_i)} \tag{10}$$

Algorithm 2 Decision-Making Process of NNCR on Content O_k for Node v_i

```
if $O_k$ arrives $v_i$ && $v_i$ is selected as the caching node then
    if $freeCacheSize(v_i) < size(O_k)$ then
        $P_{min}(v_i) = \infty, rccl = \infty$;
        for each content $O_n$ cached at $v_i$ do
            if $popularity(O_n) < P_{min}(v_i)$ then
                $P_{min}(v_i) = popularity(O_n)$;
                $O_{min}(v_i) = O_n$;
            end if
        end for
        for each one-hop neighbor $v_j$ of $v_i$ do
            if $R_{ccl}(v_j) < rccl$ then
                $rccl = R_{ccl}(v_j)$;
                $N_t = v_j$;
            end if
        end for
        if $freeCacheSize(N_t) < size(O_{min}(v_i))$ then
            if $P_{min}(v_i) > P_{min}(N_t)$ then
                transfer $O_{min}(v_i)$ to $N_t$ and replace $O_{min}(N_t)$;
            else
                discard $O_{min}(v_i)$ directly;
            end if
        else
            transfer $O_{min}(v_i)$ to $N_t$;
        end if
    end if
end if
```

In this paper, the accumulated cache load ratio is used to measure the cache status of node. In NCCR, each node periodically calculates its accumulated cache load ratio and the popularity of its cached contents, and exchanges the information of the content with the lowest popularity (such as the content name and popularity) and the accumulated cache load ratio with all its one-hop neighbors. When any content O_k arrives at node v_i, if v_i is chosen as the caching node but does not have enough available space to cache O_k, then v_i will pick out the content with lowest popularity from its cached contents as the migrating content, denoted by O_t. And the neighbor node which has the lowest accumulated cache load ratio will be chosen as the migrating destination node, denoted by N_t. If there is enough cache space in N_t, then v_i will transfer O_t to N_t and record the corresponding information about the name of O_t and the migrating path. If the available cache space of N_t is insufficient too, v_i will compare the popularity of the contents cached in v_i and N_t. Specifically, let $O_{min}(v_i)$ and $O_{min}(N_t)$ be the content with the lowest popularity in node v_i and N_t, and their popularities are expressed as $P_{min}(v_i)$ and $P_{min}(N_t)$, respectively. If $P_{min}(v_i)$ is larger than $P_{min}(N_t)$, $O_{min}(v_i)$ will be selected as the migrating content O_t, and then v_i will migrate O_t to N_t to replace $O_{min}(N_t)$. Otherwise, $O_{min}(v_i)$ will be discarded by v_i directly. The pseudo-code for NCCR is shown in Algorithm 2.

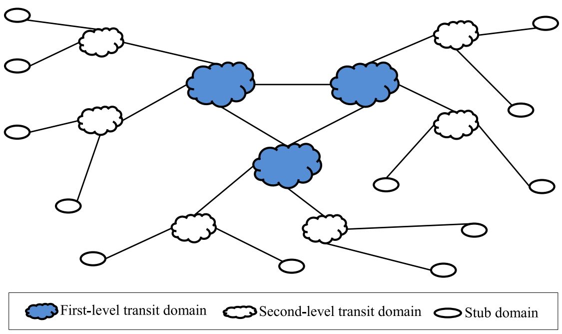

FIGURE 1. Simulation topological diagram.

V. PERFORMANCE EVALUATIONS

In this section, the performance evaluations are provided in detail to validate our proposed caching scheme. We first state some assumption and parameter settings used for experiments. And then, we present extensive simulation results to evaluate the cache performances with different cache size, different content number and different user request patterns respectively.

A. THE SIMULATION ENVIRONMENTS SETTING

To evaluate our proposed caching scheme, we conducted simulations using the ndnSIM [40], a NS-3 based simulator which is specially designed for NDN implementation. Since the proposed scheme concludes cache placement strategy (EECP) and cache replacement strategy (NCCR), we compare our caching scheme (EECP+NCCR) with the following three schemes: (1) Leave Copy Everywhere (LCE) [6] + LRU; (2) the aging popularity based energy-efficient caching scheme (APC) [32] + LRU; (3) EECP + LRU.

In our simulations, the physical network topology is constructed by using the Transit-Stub model of GT-ITM. This topology uses a 3-level hierarchy of routing domains in which there are 3 first-level transit domains, each interconnects 2 second-level transit domains, and each second-level transit domain interconnects 2 lower level stub domains. There are 10 routers in a transit domain and a stub domain. The network topological diagram is shown in Figure 1. Meanwhile, we assume that the content popularity follows a Zipf distribution with parameter α and each user generates content requests according to a Poisson process of intensity 50 requests per second. Each router is equipped with a cache of the same size and the initial cache state of each node is empty. Considering the edge routers may cover different number of end users, we assume the average request rate λ at edge routers follows the uniform distribution. The Interest packet is transmitted by flooding. The remaining major experimental parameters are set as given in Table 2 except special declaration.

The performances of the four caching schemes are evaluated with different cache sizes, different content numbers and different values of Zipf parameter α, respectively.

TABLE 2. Experimental parameter settings.

Major parameters	Default	Range
Number of content	2000	$100 \sim 5000$
Cache size of node(MB)	100	$5 \sim 1000$
User request pattern	Zipf: $\alpha = 0.8$	$0.6 \sim 1.2$
Simulation time(sec)	200	
$\omega_c\ (W/bit)$	1×10^{-9}	
$\omega_r\ (J/bit)$	2×10^{-8}	
$\omega_l\ (J/bit)$	1.5×10^{-9}	

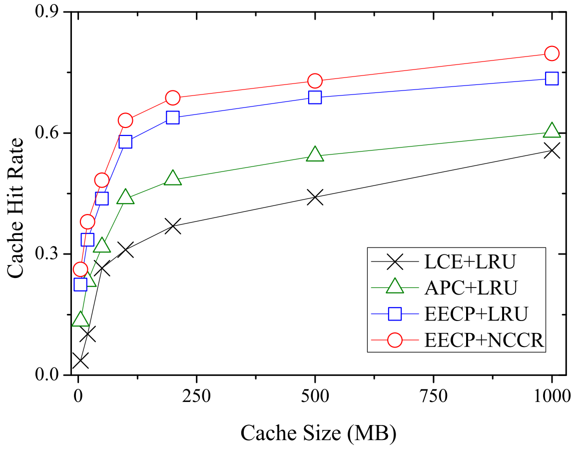

FIGURE 2. Cache hit rate vs. Cache size.

The major metrics used in our performance evaluation are given as follows.

(1) Cache hit rate: It is defined as the probability that a request is responded by the caching nodes instead of the original content servers.

(2) Average response hops: It is referred to as the average number of the routers traversed by the response packets from the originator or cache to the requesting router.

(3) Energy saving rate: It is the ratio of the saved energy by in-network caching to the total energy consumption incurred without caching.

B. PERFORMANCE ANALYSIS

1) IMPACT OF CACHE SIZE

In this subsection, we first investigate the impact of cache size on the cache performance of the above listed caching schemes. Figure 2 illustrates the cache hit rate with varying cache size. With the increase of cache size, the caching capacity of each node is enhanced and the cache replacement is reduced. Thus, the cache hit rate of each caching scheme increases as the increase of cache size. Due to lack of reasonable selection for caching location and caching content, LCE+LRU obtains the least cache hits among the four schemes. APC+LRU takes the aging popularity of content into account to improve the effectiveness of caching decision, thus its cache hit rate is higher than that of LEC+LRU. By comprehensively considering various factors related to content and node, EECP effectively optimizes the caching decision. It can be seen from the figure, the cache hit rates of the two combined strategies in which EECP is adopted are

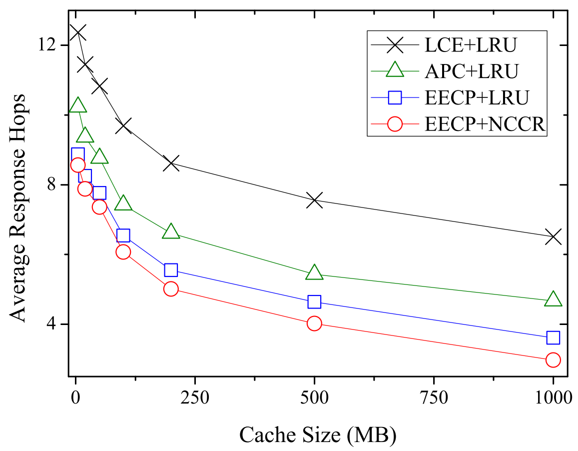

FIGURE 3. **Average response hops vs. Cache size.**

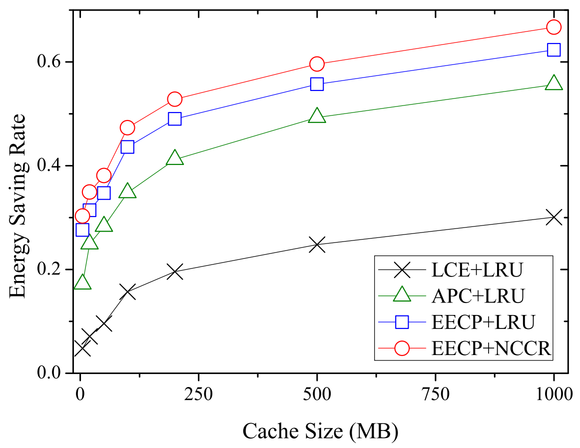

FIGURE 4. **Energy saving rate vs. Cache size.**

significantly higher than that of the other two. Furthermore, by making full use of the resources of neighbor nodes, NCCR greatly increases the chance of responding the user requests through content caching and improves the network resource utilization. Therefore, EECP+NCCR achieves the best performance on cache hit rate among the four schemes.

The impact of cache size on average response hops is shown in Figure 3. As the increase of cache size, user requests are more likely served by the corresponding content caching nodes. Therefore, the average response hops of each scheme decreases with the increase of cache size. Owing to the consideration of aging popularity, APC improves the accuracy of caching decision. Thus, the average response hops of APC+LRU is significantly lower than that of LCE+LRU. It can be seen from Figure 3 that, when the cache size of node is 100MB, the average response hops of APC+LRU is 7.42, which is about 23.4% lower than that of LCE+LRU. In EECP+NCCP, according to the energy efficiency judging condition, the caching locations of contents are restricted within the locations in the delivery path away from the OCS, and the content replicas can approach to the users gradually by using the probabilistic caching based on content popularity and node centrality. Therefore, it achieves the lowest average response hops among the four schemes. As can be seen in Figure 3, when the node cache changes from 5MB to 1000MB, the average response hops of EECP+NCCR is about 8.74% lower than that of EECP+LRU on average, which fully demonstrates the advantage of NCCR.

At last, Figure 4 shows the energy saving rate of each scheme with different values of cache size. We observe that the energy saving rate of each scheme increases as the cache size of node increases. Due to the blindness and radicalness, LCE+LRU brings about a large amount of cache replacement and redundancy. Consequently, it consumes great amount of energy and obtains the lowest energy saving rate. By contrast, EECP+NCCR decreases the total energy consumption of content access by using the energy efficiency judging condition. In additional, when there is no sufficient available cache space, the node transfers the appropriate contents to its neighbors instead of discarding them directly. It effectively increases the time that the content cached in the network. Thus, the energy saving rate of EECP+NCCR is significantly higher than that of the others. As can be seen from Figure 4, when the cache size is 1000MB, the energy saving rate of EECP+NCCR achieves 66.7%. Compared with EECP+LRU (62.3%), APC+LRU (55.6%) and LCE+LRU (30.1%), it obtains about 7.1%, 19.9% and 121.6% improvement, respectively.

2) THE IMPACT OF CONTENT NUMBER

The impact of content number on cache performance is discussed here, and the results are shown in Figures 5, 6 and 7. Since when the cache size is fixed, cache resources will become much scarcer as content number increases. It will inevitably lead to an increase of cache replacements and badly weaken the positive effects of content caching on providing fast response to subsequent requests. From Figures 5 - 7, we can see that when the content number is increased from 100 to 5000, the cache hit rate and energy saving rate of each scheme decline significantly, but their average response hops are distinctly on the increase. In the three schemes which use LRU as the cache replacement strategy, EECP+LRU obtains relatively higher cache hit rate, less average response hops and higher energy saving rate. The performance advantage is attributed to the fact that EECP can accurately estimate the content popularity and cache the content at appropriate nodes which users can easily access. In addition, EECP+NCCR outperforms EECP+LRU in terms of cache hit rate, average response hops and energy saving rate. The reason is that, by way of the neighbor cooperation based cache replacement, there is a great possibility that the contents can live much longer through properly using the free cache space of neighbor nodes. Thus, users can obtain the desired contents at the near caching nodes and avoid consuming too much energy and time to access the distant caching nodes. This further proves the superiority of NCCR strategy. As shown

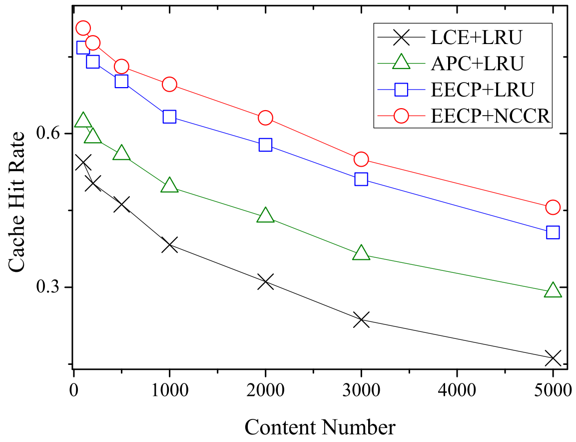

FIGURE 5. Cache hit rate vs. Content number.

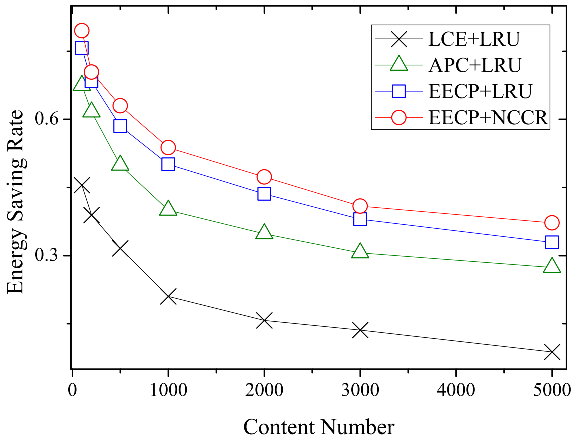

FIGURE 7. Energy saving rate vs. Content number.

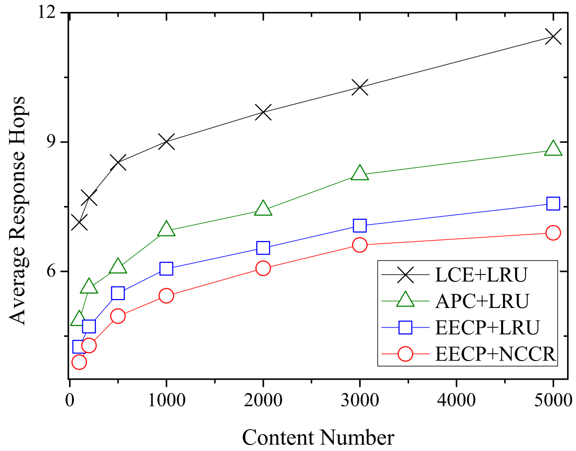

FIGURE 6. Average response hops vs. Content number.

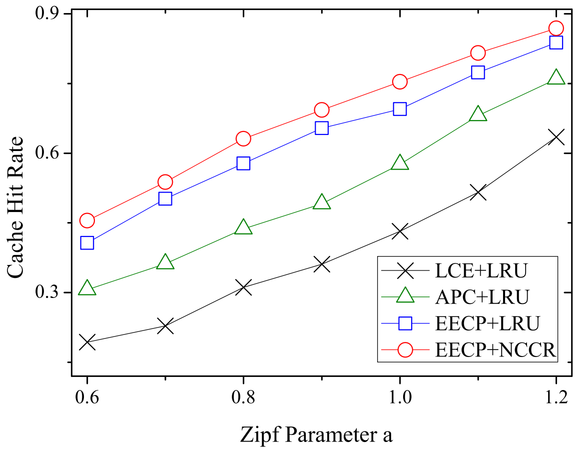

FIGURE 8. Cache hit rate vs. Zipf parameter α.

in Figures 5 - 7, when the content number is 2000, the cache hit rate of EECP+NCCR reaches 63.1% and the average response hops is only 6.07. Compared with EECP+LRU, it achieves nearly 9.2% increase in cache hit rate and 7.2% reduction in average response hops, respectively. Meanwhile, EECP+NCCR saves up to 47.3% energy by its appropriate in-network caching strategy and achieves 8.5%, 35.9% and 201.3% improvement than EECP+LRU, APC+LRU and LCE+LRU, respectively.

3) THE IMPACT OF ZIPF PARAMETER α

It is generally agreed that the content access preferences of users follow a Zipf distribution with parameter α. α is the skewness factor indicating the concentration degree of content access and its value differs greatly from the applications. The larger the value of α is, the more user requests are issued for the popular contents, and vice versa. In this subsection, we change the value of α to observe the performance of the four caching schemes in different applications.

When the value of α is small, the concentration degree of content access is relatively low. The diversity of content requests increases the diversity of caching contents. Due to the limitation of node cache size, it inevitably results in frequent cache replacements and eventually reduces the benefits arising from content caching. Thus the performance of each

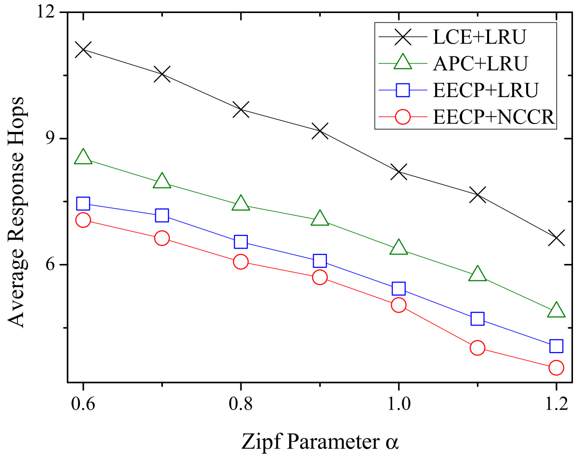

FIGURE 9. Average response hops vs. Zipf parameter α.

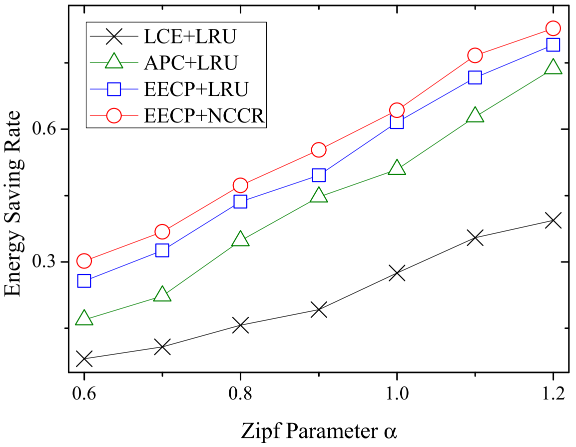

FIGURE 10. **Energy saving rate vs. Zipf parameter α.**

scheme is relatively poor under a smaller α, as shown in Figures 8, 9 and 10. As parameter α increases, however, the content diversity decreases and the number of popular contents increases. The popular contents can be cached for a relatively long time to provide more chances to respond to user requests. Therefore, the curves of cache hit rate and energy saving rate are steadily increasing, while the curves of average response hops are monotonously decreasing. Remarkably, EECP+NCCR can significantly outperform the other schemes in terms of cache hit rate and average response hops. It can be seen from Figure 8 and 9 that, when the value of α increases to 1, the cache hit rate of EECP+NCCR reaches 75.4%. Compared with EECP+LRU (69.5%), APC+LRU (57.6%) and LCE+LRU (43.2%), EECP+NCCR achieves 8.5%, 30.9% and 74.5% improvement of cache hit respectively. While the average response hops of EECP+NCCR is only 5.04, which is nearly 7.2%, 20.9% and 38.6% lower than of EECP+LRU, APC+LRU and LCE+LRU. The main reason is that our scheme tends to cache the popular contents and push the contents to the place which is relatively closer to the users. In addition, EECP+NCCR also obtains the highest energy saving rate among the four caching schemes. As shown in Figure 10, EECP+NCCR can save up to 64.3% energy when α equals to 1. Compared with EECP+LRU, APC+LRU and LCE+LRU, the improvement of EECP+NCCR in energy saving rate approaches to 4.4%, 26.3% and 133.8%, respectively. It is due to that our caching scheme can effectively control the total energy consumption according to the energy efficiency judging condition.

VI. CONCLUSION

In order to guarantee high cache performance and improve the energy efficiency of CCN caching system, an energy-efficiency based in-network caching scheme is proposed in this paper. According to the energy consumption analysis of content distribution, we first design an energy efficiency judging condition to reduce the total energy consumption of content access. On this basis and in combination with content popularity and node importance, an energy efficient cache placement strategy is proposed to optimize the selection of caching nodes. Furthermore, a neighbor cooperation based cache replacement strategy is also proposed, which utilizes the cache resource of neighbor nodes to increase the chances of content being cached and improve the quality of caching service and resource utilization. The simulation results show that the proposed caching scheme can effectively improve the whole cache performance and benefit energy efficiency in content dissemination.

As a future work, we plan to investigate the effects of other factors on cache performance, such as network topology, link bandwidth and nodal sociality. Meanwhile, we will extend our algorithm to the mobile network and other complex network environments to further evaluate its effectiveness.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the anonymous referees for their comments and suggestions. In addition, they do would also like to appreciate Yao Liu for his wise advice on the caching scheme design and experiments.

REFERENCES

[1] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, May 2014.
[2] U. Lee, I. Rimac, and V. Hilt, "Greening the Internet with content-centric networking," in *Proc. e-Energy*, Apr. 2010, pp. 201–220.
[3] U. Lee, I. Rimac, and V. Hilt, "Toward energy-efficient content dissemination," *IEEE Netw.*, vol. 25, no. 2, pp. 14–19, Mar. 2011.
[4] A. P. Bianzino *et al.*, "A survey of green networking research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, 1st Quart., 2012.
[5] J. Kurose, "Challenges in the design and analysis of content-centric networks," in *Proc. ACM SIGCOMM*, Aug. 2013, pp. 1–2.
[6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012.
[7] S. Zhang *et al.*, "MobiCache: Cellular traffic offloading leveraging cooperative caching in mobile social networks," *Comput. Netw.*, vol. 83, pp. 184–198, Jun. 2015.
[8] Y. B. Sun, Y. Zhang, and H. L. Zhang, "Survey of research on information-centric networking architecture," *Acta Electron. Sinica*, vol. 44, no. 8, pp. 2009–2017, Aug. 2016.
[9] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.
[10] Y. Li, H. Xie, Y. Wen, Z.-L. Zhang, and C.-Y. Chow, "How much to coordinate? Optimizing in-network caching in content-centric networks," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 3, pp. 420–434, Sep. 2015.
[11] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, "CPHR: In-network caching for information-centric networking with partitioning and hash-routing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2742–2755, Oct. 2016.
[12] O. B. Rhaiem, L. C. Fourati, and W. Ajib, "New hierarchical parent-child caching strategy (H-CS) for CCN-based video streaming," in *Proc. IEEE AINA*, May 2016, pp. 109–115.
[13] Y. Xu *et al.*, "A dominating-set-based collaborative caching with request routing in content centric networking," in *Proc. ICC*, Nov. 2013, pp. 3624–3628.
[14] Y. Liu, D. Zhu, and W. Ma, "A novel cooperative caching scheme for Content Centric Mobile Ad Hoc Networks," in *Proc. ISCC*, Jun. 2016, pp. 824–829.
[15] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP network energy cost: Formulation and solutions," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 463–476, Apr. 2016.

[16] A. Ghodsi *et al.*, "Information-centric networking: Seeing the forest for the trees," in *Proc. HotNets*, Nov. 2011, pp. 1–6.

[17] Z. Ming, M. Xu, and D. Wang, "Age-based cooperative caching in information-centric networking," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–8.

[18] Y. Kim *et al.*, "Differentiated forwarding and caching in named-data networking," *J. Netw. Comput. Appl.*, vol. 60, pp. 155–169, Jan. 2016.

[19] K. Yokota, K. Sugiyama, J. Kurihara, and A. Tagami, "RTT-based caching policies to improve user-centric performance in CCN," in *Proc. AINA*, Mar. 2016, pp. 124–131.

[20] J. Li *et al.*, "RBC-CC: RBC-based cascade caching scheme for content-centric networking," *J. Netw. Syst. Manage.*, vol. 25, no. 2, pp. 375–396, Apr. 2017.

[21] G. Domingues *et al.*, "Enabling opportunistic search and placement in cache networks," *Comput. Netw.*, vol. 119, pp. 17–34, Jun. 2017.

[22] W. Sirichotedumrong *et al.*, "Prioritized probabilistic caching algorithm in content centric networks," in *Proc. ICCIT*, Jul. 2016, pp. 255–265.

[23] R. Jmal and L. C. Fourati, "An OpenFlow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy," *Comput. Stand. Inter.*, vol. 51, pp. 22–29, Mar. 2017.

[24] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks (extended version)," *Comput. Commun.*, vol. 36, no. 7, pp. 758–770, Apr. 2013.

[25] C. Bernardini, T. Siverston, and O. Festor, "Socially-aware caching strategy for content centric networking," in *Proc. IFIP Netw.*, Jun. 2014, pp. 1–9.

[26] Y. Cheng, M. Wu, M. Zhao, and K. Wang, "Socially-aware NodeRank-based caching strategy for Content-Centric Networking," in *Proc. ISWCS*, Sep. 2016, pp. 297–303.

[27] J. Guan, W. Quan, C. Xu, and H. Zhang, "The location selection for CCN router based on the network centrality," in *Proc. IEEE CCIS*, Nov. 2012, pp. 568–582.

[28] J. A. Khan and Y. Ghamri-Doudane, "SAVING: Socially aware vehicular information-centric networking," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 100–107, Aug. 2016.

[29] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of energy-efficient caching in information-centric networking," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 122–129, Nov. 2014.

[30] T. Braun and T. A. Trinh, "Energy efficiency issues in information-centric networking," in *Proc. EE-LSDS*, Apr. 2013, pp. 271–278.

[31] J. Llorca *et al.*, "Dynamic in-network caching for energy efficient content delivery," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 245–249.

[32] J. Li, B. Liu, and H. Wu, "Energy-efficient in-network caching for content-centric networking," *IEEE Commun. Lett.*, vol. 17, no. 4, pp. 797–800, Apr. 2013.

[33] C. Fang *et al.*, "An energy-efficient distributed in-network caching scheme for green content-centric networks," *Comput. Netw.*, vol. 78, pp. 119–129, Feb. 2015.

[34] Y. Zhang, F. Li, X. Ma, K. Wang, and X. Liu, "Cooperative energy-efficient content dissemination using coalition formation game over device-to-device communications," *Can. J. Elect. Comput. Eng.*, vol. 39, no. 1, pp. 2–10, 2016.

[35] M. A. Hail *et al.*, "Caching in named data networking for the wireless Internet of Things," in *Proc. Int. Conf. Recent Adv. Internet Things*, Apr. 2015, pp. 1–6.

[36] S. Imai, K. Leibnitz, and M. Murata, "Energy efficient content locations for in-network caching," in *Proc. 18th Asia–Pacific Conf. Commun. (APCC)*, Oct. 2012, pp. 554–559.

[37] H. Xu, H. Huang, and G. Wang, "An energy-efficient caching strategy based on coordinated caching for green content-centric network," in *Proc. Int. Conf. Security, Privacy Anonymity Comput., Commun. Storage*, Nov. 2016, pp. 168–175.

[38] H. Wu, B. Liu, and Y. Li, "Towards energy-efficient caching in content-centric networking," *J. Commun.*, vol. 11, no. 12, pp. 1036–1047, Dec. 2016.

[39] T. Le, Y. Lu, and M. Gerla, "Social caching and content retrieval in disruption tolerant networks (DTNs)," in *Proc. ICNC*, Feb. 2015, pp. 905–910.

[40] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2: An updated NDN simulator for NS-3," Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0028, Nov. 2016.

YING AN received the B.S. degree in automatic control, the M.S. degree in communications engineering, and the Ph.D. degree in computer science from Central South University, China. He is currently an Associate Professor with the Institute of Information Security and Big Data, Central South University. His research interests include information centric networks, delay tolerant networks, and optimization of future Internet.

XI LUO received the B.S. and M.S. degrees in communications engineering and the Ph.D. degree in computer science from Central South University, China. She is currently a Lecturer with the Department of Information Technology, Hunan Police Academy, China. Her general research interests include content-centric networks, in-network caching technology, and content distribution.

• • •