Cache-Enabled Power Line Communication Networks: Caching Node Selection and Backhaul Energy Optimization

Yuwen Qian, Liuqiang Shi, Long Shi, *Member, IEEE*, Kui Cai, *Senior Member, IEEE*, Jun Li, and Feng Shu

***Abstract*—Power line communication (PLC) advances smart grid technology by offering a convenient and efficient data transmission service. However, due to the severe PLC channel condition and ever-growing traffic load in the PLC backhaul, reliable communications among widely distributed in-home PLC users is highly energy-consuming. In this paper, we put forth a cache-enabled PLC network comprising a single master base station (MBS) and mutiple slave base stations (SBSs) each bound with an in-home network to alleviate the backhaul traffic load and reduce the backhaul energy consumption. In the proposed scheme, the SBSs with the local caches pre-download the popular contents from the MBS through the PLC backhaul, and the in-home users request the contents from nearby caching nodes (CNs) or the MBS. Apart from the cache-enabled PLC design, our contributions also lie in the selection of CNs and optimal content placement algorithm. First, we propose a selection algorithm to activate the SBSs with high influence as the CNs, according to a directed graph that models the requesting relations among all SBSs. Second, we design an optimal content placement algorithm to minimize the backhaul energy consumption for the contents with different sizes and user preferences. Simulation results show that our cache-enabled PLC scheme with the caching node selection and optimal content placement can dramatically reduce the backhaul energy consumption, compared with the existing caching strategies.**

***Index Terms*—Smart grid, power line communications, content placement, backhaul energy, caching node selection.**

I. Introduction

SMART grid makes integrated use of energy, information, and communication technologies to provide sensing, measurement, and management for power grids. The distributed sensing and measurement for electrical systems have generated massive data, which demands a reliable transmission among electrical devices connected with the power lines. As the natural candidate communication for smart grid, power line communication (PLC) has been widely adopted.

Similar to modern wireless networks, the PLC networks also confront with two nail-biting challenges. On one hand, in practice, PLC encounters a more hostile communication environment compared with conventional wired and wireless communications due to severe signal fading and impulsive noises. In particular, the presence of impulsive noises in the PLC channels severely degrades the signal reception. In addition, PLC channels vary fast with the locations of transmitters, network topologies and connected loads [1]. Thus, it is a great challenge for PLC to provide reliable and fast communications for the smart grid users. On the other hand, in the near future, the PLC network will be struggling with the huge backhaul burden, caused by the unprecedented growth of data traffic attributing to the widespread use of smart equipments and applications in smart grid.

To address the first challenge, much effort has been devoted to research in the wireless domain on the small base networks integrated with advanced signal processing and coding technologies to enhance communication reliability and network coverage [2], [3]. These small cell deployments are mainly in the form of home small cells, known as femtocells and picocells. In regard to the second challenge, recent research outcome makes plain that duplicate downloads of a few popular contents occupy a significant portion of total data traffic [4]. Driven by the finding, the concept of caching revives in the field of wireless networks to bring popular contents closer to users, which alleviates the traffic load over backhaul [5]. The enticing benefit has elevated the wireless caching to a central position in the small cell networks, which has been attracted significant research interest in various cache-enabled small cell networks [6]–[8].

However, it should be stressed that wireless caching cannot be easily plugged into the PLC systems for the following reasons. First, the fading and noise effects in the PLC channels are different from those in the wireless channels, making the data delivery even more challenging [9]. Second, wireless caching cannot offer the data storage service for those electrical devices without any wireless interfaces. Therefore, how to design a cach-enabled PLC networks with SBSs is a great challenge and remains open. In addition, long distance PLC

Manuscript received March 8, 2019; revised October 13, 2019; accepted March 30, 2020. Date of publication April 3, 2020; date of current version May 19, 2020. The work of Long Shi and Kui Cai was supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 under Grant MOE2016-T2-2-054, and in part by SUTD-ZJU under Grant ZJURP1500102. The work of Jun Li was supported by the National Natural Science Foundation of China under Grant 61727802 and Grant 61872184. The associate editor coordinating the review of this article and approving it for publication was B. Kantarci. *(Corresponding authors: Long Shi; Jun Li.)*

Yuwen Qian, Liuqiang Shi, Jun Li, and Feng Shu are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: admon@njust.edu.cn; 116104000620@njust.edu.cn; jun.li@njust.edu.cn; shufeng@njust.edu.cn).

Long Shi and Kui Cai are with the Science and Math Cluster, Singapore University of Technology and Design, Singapore (e-mail: slong1007@gmail.com; cai_kui@sutd.edu.sg).

Digital Object Identifier 10.1109/TGCN.2020.2985378

transmissions over the PLC backhaul links are energy consuming. How to minimize the backhaul energy cost is also a crucial issue for the cache-enabled PLC networks.

As the first major concern in the cache-enable network, installing the cache devices for all SBSs may result in not only an unaffordable cost but also a waste of caching resource. To reduce the cost, prior works in the field of wireless networks has made some effort on the caching node selection. For example, [10] proposed a caching node selection strategy based on the influence factor to reduce the system overheads in the device-to-device (D2D) networks. In addition, [11] combined social-aware networks via D2D with the caching mechanism, in which the crucial caching nodes are selected according to social relationship. As follow-up, [12] proposed the matching theory based algorithm with the aid of social utility function to select the important users in cellular networks as the caching nodes. Therefore, the research on selecting the caching nodes in the cache-enabled PLC networks remains open.

Second, to reduce backhaul energy consumption, the efficient content placement has captured intensive attention in various wireless caching networks. In [13], the authors investigate the cache placement problem in fog radio access networks, by taking into account flexible physical-layer transmission schemes and diverse content preferences of different users. In [14], the authors put forth an energy-efficient content placement approach to maximize the total system energy. In addition, [15] developed an optimal content placement method to minimize the backhaul energy consumption in a densely deployed wireless access network.

However, the content placement methods in [14], [15] focused on the files with equal size, which is an ideal assumption for practical systems. Considering the files with arbitrary sizes, the authors of [16] proposed a cooperative coded caching scheme to improve the energy efficiency in heterogeneous small cell networks. In [17], the authors investigated an optimal content placement scheme to minimize the energy consumption for the heterogeneous wireless networks by encoding the files with maximum-distance separable codes. In these coded prefetching systems, coded files are first parted into blocks and stored in different SBSs and then and then cooperatively delivered to users [18]. Nonetheless, when adopting the coded prefetching, it will cause the PLC network to be overwhelmed due to the frequent cooperations among SBSs that connect with each other via the narrow-band PLC channel.

To tackle the first concern, we propose a small-cell cache-enabled PLC network comprising a single master base station and multiple slaver BSs each providing Internet access for in-home users (IUs). All SBSs connect to the backbone networks through an MBS installed at the transformer unit and connect to each other via power lines. Building upon the concept of wireless caching, we install cache devices in some SBSs as the caching nodes to predownload the popular contents requested by the IUs. As such, the backhaul traffic in the PLC network is offloaded and the burden on the PLC backhaul can be alleviated. In particular, we illustrate the requesting relations among all SBS through a directed grapha and propose a selection algorithm to activate the SBSs with high IoS as the caching nodes. Regarding the second concern, we develop an optimal content placement strategy to minimize the backhaul energy consumption for the proposed caching network. Overall, the contributions of this paper are summarized as follows:

- We design a small-cell cache-enabled PLC network, where the cache devices are installed in the SBSs at the PLC access nodes to prefetch the popular contents in the off-peak time and serve the IUs in the peak time. As such, the proposed system can alleviate backhaul traffic of the PLC networks (see Section II).
- We propose a selection strategy to activate a group of SBSs with high IoS as the caching nodes, aiming at an efficient use of limited caching resources. The core of the selection strategy lies in the influence of SBS (IoS) characterization according to the requesting relations among all SBSs (see Section III).
- We develop an optimal content placement to minimize the backhaul energy consumption. Considering the requesting files with arbitrary sizes, we put forth a heuristic algorithm to obtain the optimal solution (see Section IV).

The rest of this paper is organized as follows. Section II describes the system model. Section III presents the cache node selection algorithm. Section IV-C formulated the optimal content placement problem and numerical results are presented in Section V. Finally, Section VI concludes this paper.

II. System Model

A. Cache-Enabled PLC Network

As shown in Fig. 1, we propose a cache-enabled PLC network with a single *master* base station (MBS) and K *slaver* base stations (SBSs). The MBS is deployed in the transformer to connect the local network with the backbone network. The SBSs are installed in the indoor power meters to provide network access for in-home users (IUs) [9]. We refer to the unique SBS bound with each in-home network as the *serving* SBS of IUs in this network.

Before delving into the specific designs, we clarify the PLC links among the IUs, SBSs, and MBS, as illustrated in Fig. 2. First, there is a direct link of low voltage (LV) power line between the IU and its serving SBS in the same in-home network. Second, there is no direct link between the IU and the SBSs in different homes. Third, all SBSs are connected with each other via LV power lines. Fourth, IUs in different in-home networks communicate with each other only via their serving SBSs. Finally, the MBS and SBSs are connected with the LV power lines, referred to as the PLC backhaul. In the proposed cache-enabled network, cache devices are installed in some selected SBSs. For simplicity, we refer to the SBSs with cache devices as *caching nodes* (CNs). Notably, the proposed scheme selects a portion of SBSs as the CNs. The motivations behind the CN selection are as follows: First, from PLC providers' point of view, the CN selection reduces the hardware cost and makes an efficient use of limited caching resources. Second, in practice, some IUs are traditional electricity customers that only require basic services without any multimedia contents. As such, it is not cost-effective to install the cache devices in

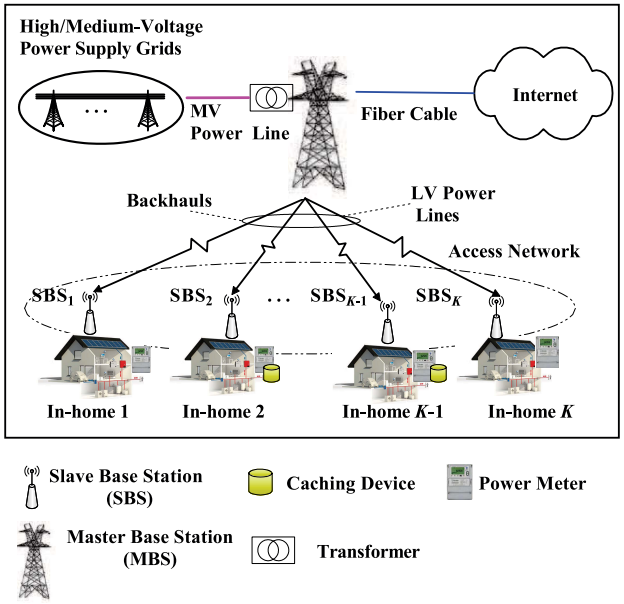

Fig. 1. The cache-enabled PLC network with a single MBS and K SBSs, where some SBSs are selected as the CNs to buffer popular contents and each SBS can obtain its requested contents from nearby CNs or the MBS.

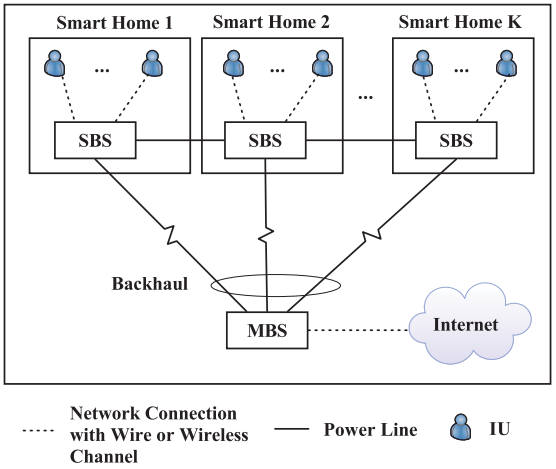

Fig. 2. The connections among the IUs, SBSs, and MBS.

the serving SBSs of these IUs. We will specify how to select the CNs in Section III.

The role of the CNs is to pre-download the popular contents requested by IUs from the MBS in the off-peak time. Then each IU can obtain its requested contents from nearby CNs in the peak time. As such, the proposed network can alleviate the backhaul traffic load caused by duplicated transmissions of popular contents from the MBS to IUs.

According to the connections in Fig. 2, we emphasize that each IU cannot directly contact the CN in different homes unless the CN is the serving SBS in the same home. In this context, we require that the user requests in the same in-home network can only be forwarded by its serving SBS to the MBS or CNs. From now on, we treat the request of IU equivalently as that of its serving SBS.

This paper considers two cases for the IUs to download contents. First, the content requested by the SBSs are buffered in the CNs. In this case, each SBS directly fetches the contents from the CNs and forwards them to its IUs. Second, the content requested by the SBSs are missed in the CNs. In this case, each SBS fetches the contents from the MBS via backaul.

B. PLC Channel Model

In this paper, the backhaul transmissions from the MBS to SBSs are over the typical PLC channels. The received signal at the kth SBS is

$$y_k = h_k x + z_k, \tag{1}$$

where $k = 1, 2, \ldots, K$. In (1), x is the signal transmitted from the MBS, y_k and z_k are the received signal and noise at the kth SBS respectively, h_k is the channel transfer function from the MBS to the kth SBS. Since the PLC channel can be modeled as the multi-path propagation channel, we have [19]

$$h_k = \sum_{i=1}^{I} g_{i,k} A_{i,k}\left(f, d_{i,k}\right) e^{-j2\pi f \tau_{i,k}}, \tag{2}$$

where I is the number of paths in the PLC channel, $g_{i,k}$ and $d_{i,k}$ are the weight factor and distance of the ith path respectively, $\tau_{i,k}$ is the time delay of the signal via the ith path from the MBS to the kth SBS, f is the frequency, and the attenuation of PLC channel is given by

$$A_{i,k}\left(f, d_{i,k}\right) = e^{-(a_0 + a_1 \cdot f^{a_3}) d_{i,k}}, \tag{3}$$

where a_0, a_1, and a_3 are the attenuation parameters.

In addition, the noise of PLC channels can be modeled with the Bernoulli-Gaussian noise by [19]

$$z_k = n_G + n_B n_I, \tag{4}$$

where n_G and n_I are zero-mean Gaussian random variables with the variances σ_G^2 and σ_I^2 respectively, and n_B is a Bernoulli random variable with parameter q. From (4), the variance of noise z_k is

$$\sigma_k^2 = \mathbf{E}\left(n_G^2\right) + \mathbf{E}\left(n_B^2\right)\mathbf{E}\left(n_I^2\right) = \sigma_G^2 + q\sigma_I^2, \tag{5}$$

where $\mathbf{E}(\cdot)$ is the statistical expectation operator.

C. Requesting Probability Distribution

Let $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$ be a set that collects all N contents with f_n being the nth content. In general, different SBSs request different types of contents, leading to different content popularity. The paper assumes the requests for different contents follow the Zipf distribution. Then, the requesting probability of the kth SBS for the nth content is [20]

$$Q_{k,n} = \phi_{k,n}^{-\beta} \Big/ \sum_{\zeta=1}^{N} \phi_{k,\zeta}^{-\beta}, \tag{6}$$

where $\phi_{k,n}$ denotes the preference degree of the kth SBS for the nth content, $\phi_{k,n} = 1, 2, \ldots, N$, and the Zipf parameter β is determined by the popularity of contents. According to (6), a smaller $\phi_{k,n}$ corresponds to a higher probability that the kth SBS requests the nth content.

In this paper, we focus on the backhaul energy consumption from the MBS to the SBSs rather than that from each SBS to its IUs, as the energy consumption in the in-home network is

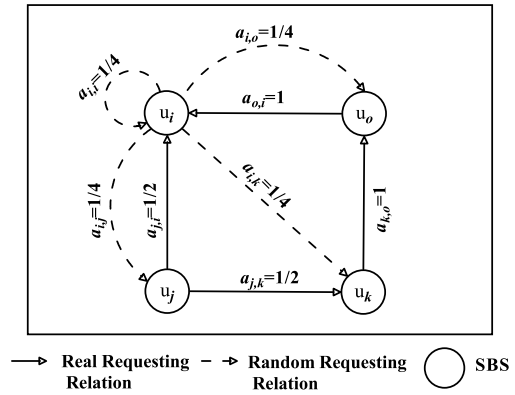

Fig. 3. The directed graph $\mathcal{G}$ according to the requesting relation among four SBSs.

much less that in the backhaul. As such, from power suppliers' perspective, it is of particular interest to minimize the backhaul energy consumption by a proper CN selection.

III. Proposed CN Selection

In this section, we elaborate the selection strategy of CNs. First, we formulate the directed graph based on the requesting relations among the SBSs. Second, we select the SBSs that are frequently requested as the CNs.

A. Graph Model of Requesting Relations

Recall that each SBS in the in-home network can forward the IU requests within the network, and all these SBSs are fully connected with each other. We establish the *requesting relation* between any two SBSs, if one SBS has received the content request from the other SBS before. We call the SBS that have received the requests before as the *target SBS*. We remark that some SBS may have multiple target SBSs.

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_K\}$ be the set of SBSs with u_k being the kth SBS. To select CNs, we propose to use the directed graph to model the requesting relations among SBSs. Define a graph as $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{E}$ is the set of directed edges and $\mathcal{U}$ is the set of nodes (i.e., SBSs). In the graph, each edge in $\mathcal{E}$ represents the requesting relation between two related nodes in $\mathcal{U}$. As an illustrating example, Fig. 3 shows a directed graph that models the requesting relations among four nodes.

Generally, the nodes in $\mathcal{U}$ fall into two categories. The first one is the nodes that have target SBSs, such as u_j, u_k, and u_o in Fig. 3. For example, u_j has two target SBSs u_k and u_i. Define $\mathcal{U}_1$ as the set that collects the nodes in the first category, and $\mathcal{U}_1 = \{u_j, u_k, u_o\}$ in this figure. We use the solid arrow lines to illustrate the directed edges induced by the *real* requesting relations among the nodes in the first category. The second one is the nodes that are not related to any target SBS, such as u_i in the figure. In this scenario, we randomly select one out of the four SBSs as the target SBS of u_i. Put differently, all SBSs can serve as the contents provider for u_i. We use the dash arrow lines to illustrate the *random* requesting relations induced by u_i. Define $\mathcal{U}_2$ as the set that collects the nodes in the second category and $\mathcal{U}_2 = \{u_i\}$ in this figure.

In addition, the set $\mathcal{E} = \{(u_j, u_i), (u_j, u_k), (u_k, u_o), (u_o, u_i), (u_i, u_*)\}$, where u_* denote an randomly selected target SBS for u_i and (u_j, u_k) represents the directed edge from u_j to u_k. Given (u_j, u_k), u_k is an adjacent node of u_j, and u_k is a target SBS of u_j.

Then, we use the *adjacency matrix* to collectively express the relations induced by adjacent nodes in the graph $\mathcal{G}$. Define $\mathbf{A} = \{a_{j,k}\}_{K\times K}$ as the adjacency matrix, where $a_{j,k}$ is the weight of the directed edge (u_j, u_k). Moreover, $a_{j,k}$ represents the probability that u_k is selected as the target node of u_j. Fig. 3 indicates

$$\mathbf{A} = \begin{bmatrix} a_{i,i} & a_{i,j} & a_{i,k} & a_{i,o} \\ a_{j,i} & a_{j,j} & a_{j,k} & a_{j,o} \\ a_{k,i} & a_{k,j} & a_{k,k} & a_{k,o} \\ a_{o,i} & a_{o,j} & a_{o,k} & a_{o,o} \end{bmatrix}. \tag{7}$$

The entries of $\mathbf{A}$ are identified as follows: First, $a_{j,j} = a_{k,k} = a_{o,o} = 0$, since there is no relations between the u_j and itself in Fig. 3 (u_k and u_o follow similarly). Second, if u_j has multiple adjacent nodes (i.e., target nodes), the adjacent nodes are selected as the target SBSs of u_j with equal probability. Let Φ_j denote the number of target SBSs of u_j. In Fig. 3, $\Phi_j = 2$ and the probabilities that u_k and u_i are selected as the target SBSs of u_j are both $\frac{1}{\Phi_j} = 1/2$. That is, $a_{j,i} = a_{j,k} = 1/2$. Furthermore, given that u_i falls into the second category, all SBSs can be the target node of u_i with equal probability. Thus, $a_{i,i} = a_{i,j} = a_{i,k} = a_{i,o} = 1/4$. Overall, we have

$$a_{j,k^*} = \begin{cases} \frac{1}{\Phi_j}, & \text{if } j \in \mathcal{U}_1, \\ \frac{1}{K}, & \text{if } j \in \mathcal{U}_2, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where u_{k^*} denotes any target node of u_j. From (8), we can rewrite (7) as

$$\mathbf{A} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{9}$$

B. Calculation of IoS

We refer to the target SBS as the *influential SBS* if this target SBS more frequently receives the requests than others. To evaluate the frequency, we define the *influence of SBS* (IoS) as the probability that an SBS receives requests from other SBSs. Generally, we select the SBSs with IoS larger than a specific threshold as the CNs.[1]

The spirit of the IoS computation builds upon the simplified algorithm of PageRank that was proposed by Google Search to measure the importance of website pages [21]. The underlying assumption of PageRank is that more important websites are likely to receive more links from other websites. Similar to PageRank, we use IoS to measure the importance of a typical SBS. The typical SBSs with higher IoS implies that the SBSs are more frequently requested and the targets of the typical SBS are more likely to be selected as CNs. It is stressed that

[1]The threshold of IoS is preset as shown in Algorithm 1 of Section III-C.

the IoS of each SBS will be divided equally among all outbound links and passed to its target nodes. Let $p(k)$ be IoS of the kth SBS. In the general case, the IoS for any SBS can be expressed as

$$p(k) = \sum_{a_{j,k}=1/\Phi_j, u_j \in \mathcal{U}_1} \frac{p(j)}{\Phi_j} + \sum_{u_i \in \mathcal{U}_2} \frac{p(i)}{K}, \quad 0 < p(k) < 1, \tag{10}$$

where the IoS of u_k attributes to that of nodes in two different categories. To be specific, the first term in the right-hand side of (10) comes from the weighted IoSs of adjacent nodes of u_k in the first category, while the second term reflects the weighted IoSs of nodes in the second category. In this paper, we compute the IoS of each SBS iteratively. The key idea of the iteration lie in the following three steps.

The first part is the IoS from each SBS u_j in the SBS set containing all SBSs linking to u_k, divided by the number Φ_j of targets from u_j. The second part is the IoS from each SBS u_i without target, which is an element in $\mathcal{U}_2$ and randomly selects u_k as the target with the probability 1/K.

Step 1: Iteration of the IoS vector. With (8) and (10), we have

$$\begin{pmatrix} p_{n+1}(1) \\ \vdots \\ p_{n+1}(K) \end{pmatrix} = \begin{pmatrix} a_{1,1} & \cdots & a_{K,1} \\ \vdots & \ddots & \vdots \\ a_{1,K} & \cdots & a_{K,K} \end{pmatrix} \begin{pmatrix} p_n(1) \\ \vdots \\ p_n(K) \end{pmatrix}, \tag{11}$$

where n is the number of iterations. Then, we rewrite (11) as a matrix form:

$$\mathbf{p}_{n+1} = \mathbf{A}^T \mathbf{p}_n, \tag{12}$$

where $\mathbf{p}_n = (p_n(1), p_n(2), \ldots, p_n(K))^T$ is the IoS vector in the nth iteration and T is the transposition operator.

Step 2: Adjustment of the adjacency matrix. This step is used to adjust $\mathbf{A}$ to deal with the cases where the targets of some SBSs are not consistent with the relations given in $\mathbf{A}$. Notably, $\mathbf{A}$ indicates the requesting relations in the past time slots, which can be used to predict the targets in the future. Thus, each SBS selects its targets according to $\mathbf{A}$ with large probability in future. However, there exists the SBSs that select their targets randomly without following the relations in $\mathbf{A}$. For example, from $\mathbf{A}$ in Fig. 3, u_o is the target of u_k for the time being. Thus, it is more likely that u_k will request from u_o with a large probability in the next time slot. Nevertheless, it is possible that u_k randomly selects the targets from all SBSs.

Let ρ be the probability that the SBSs select the target SBSs according to $\mathbf{A}$. Otherwise, the SBSs randomly select their targets with probability $1-\rho$. Thus, we can adjust the adjacency matrix as

$$\widehat{\mathbf{A}} = \frac{1-\rho}{K} \mathbf{e}\mathbf{e}^T + \rho \mathbf{A}, \tag{13}$$

where $\mathbf{e}$ is a K-dimensional unit column vector.

Step 3: Computation of IoS vector. Assuming that the MBS is aware of $\widehat{\mathbf{A}}$, the IoS vector is updated as

$$\mathbf{p}_{n+1} = \widehat{\mathbf{A}}^T \mathbf{p}_n. \tag{14}$$

Theorem 1: The iteration in (14) is convergent.

Algorithm 1 CN Selection

Input: $\mathbf{A}$, ρ;
Output: $\mathcal{C}$;
1: Initialize $\mathbf{p}_0$ and set $n = 0$;
2: **repeat**
3: Update $\mathbf{p}_n$ with (14);
4: Set $n = n + 1$;
5: **until** $||\mathbf{p}_{n+1} - \mathbf{p}_n||_1 < \tau$;
6: Set $\mathbf{p}^* = \mathbf{p}_n$, $M = 1$, and Γ;
7: **for** $k = 1$ *to* K **do**
8: Select $p(k)$ from $\mathbf{p}^*$;
9: **if** $p(k) \geq \Gamma$ **then**
10: Select the kth SBS as the Mth CN;
11: Set $c_M = u_k$;
12: Set $M = M + 1$;
13: **end if**
14: **end for**
15: **return** $\mathcal{C} = \{c_1, c_2, \cdots, c_M\}$.

Proof: See the Appendix. ■

Finally, from (14), the IoS vector is given by

$$\mathbf{p}^* = \lim_{n \to \infty} \mathbf{p}_n, \tag{15}$$

where the solution $\mathbf{p}^*$ keeps almost invariant as n goes large. In practice, we can find such $\mathbf{p}^*$ when n>40.

C. Algorithm of CN Selection

Complying with the IoS iteration in (14), we develop the CN selection algorithm as Algorithm 1.

From lines 1 to 6 of Algorithm 1, we calculate the $\mathbf{p}^*$ in (15) with reference to the power iteration method in [22]. In line 1, we initialize the IoS vector as

$$\mathbf{p}_0 = (p(1), \ldots, p(K))^T = \left(\frac{1}{K}, \ldots, \frac{1}{K}\right)^T. \tag{16}$$

In line 5, τ is an arbitrarily small positive value. From lines 6 to 13, we select the SBSs as the CNs according to $\mathbf{p}^*$. In line 6, we choose an IoS threshold $\Gamma (0 \leq \Gamma \leq 1)$ to select the CNs. As lines 9-13 indicated, the SBSs whose IoSs are larger than Γ are selected as the CNs. In line 15, the algorithm returns the CN set $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ that contains M CNs selected from the K SBSs. From line 11, $\mathcal{C} \subset \mathcal{U}$.

IV. Minimization of Backhaul Energy Consumption

This section formulates the optimization problem of the backhaul energy consumption in the proposed caching network. Then, we convert the NP-hard problem to be tractable. Finally, we present an algorithm to solve the problem.

A. Problem Formulation

To characterize the backhaul energy consumption, we compute the transmission rate of the backhaul link. Let R_k be

the backhaul transmission rate from the MBS to the *k*th SBS. First, we have [19]

$$R_k = B\sum_{i=0}^{1}\xi_i \log_2(1+\alpha_i\gamma_k), \tag{17}$$

where B is the PLC channel bandwidth, $\xi_i = q^i(1-q)^{1-i}$ with q defined in (5), and

$$\alpha_0 = 1+q\eta,\ \ \alpha_1 = \frac{1+q\eta}{1+\eta}, \tag{18}$$

with $\eta = \frac{\sigma_I^2}{\sigma_G^2}$. In addition, the received signal to interference plus noise ratio (SINR) at the *k*th SBS is

$$\gamma_k = \frac{P_{M\to k}|h_k|^2}{\sum\limits_{j=1,j\neq k}^{K} P_{M\to j}\left|h_j\right|^2+\sigma_k^2}, \tag{19}$$

where $P_{M\to k}$ is the transmit power of MBS targeting at the *k*th SBS, and the channel transfer function h and the noise power σ_k are defined in (2) and (5) respectively.

Second, let $E_{k,n}$ be the energy consumed by the transmission of the *n*th content from the MBS to the *k*th SBS. Then, we have

$$E_{k,n} = P_{M\to k}\frac{l_n}{R_k}, \tag{20}$$

where l_n is the size of the *n*th content. Define the set of content sizes corresponding to $\mathcal{F}$ as $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$.

Finally, we derive the energy consumption in our proposed caching network. Let $S_{m,n}$ be the total backhaul energy consumption of all SBSs when f_n is cached by the *m*th CN. According to (6) and (20), we have

$$S_{m,n} = \sum_{k=1}^{K}\left(1-I_{k,m}\right)Q_{k,n}E_{k,n}, \tag{21}$$

where $E_{n,k}$ is the backhaul energy consumption for the *k*th SBS downloading f_n from MBS. In (21), $I_{k,m}$ is an indicator, defined as

$$I_{k,m} = \begin{cases} 1, & \text{if } c_m \text{ is a target node of } u_k, \\ 0, & \text{otherwise,} \end{cases} \tag{22}$$

where $u_k \in \mathcal{U}$ is the *k*th SBS and $c_m \in \mathcal{C}$ is the mth CN.

To minimize the backhaul energy consumption, we formulate the optimal content placement problem as

$$\begin{aligned} \boldsymbol{\mathcal{P}_0}\text{:}\ \min_{x_{m,n}}\ & \sum_{m=1}^{M}\sum_{n=1}^{N} S_{m,n}x_{m,n} && \text{(23a)}\\ \text{s.t.}\ & \sum_{n=1}^{N} l_n x_{m,n} \le C_m,\ \ \forall m && \text{(23b)}\\ & \sum_{m=1}^{M} x_{m,n} \le 1,\ \ \forall n && \text{(23c)}\\ & x_{m,n}\in\{0,1\},\ \ \forall m,n, && \text{(23d)} \end{aligned}$$

where C_m is the cache size of the mth CN, and we define

$$x_{m,n} = \begin{cases} 1, & \text{if } f_n \text{ is cached in } c_m, \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

where $f_n \in \mathcal{F}$ and $c_m \in \mathcal{C}$.

It is notable that $\boldsymbol{\mathcal{P}_0}$ is a NP hard problem. However, we can equivalently transform this intractable problem into a knapsack assignment problem [23]. To be consistent, we map the commonly used notations in this paper to that in [23]. First, we treat $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ as the knapsack set and C_m as the size of the *m*th knapsack. Second, we regard $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$ as the item set. Therefore, the problem that the N items are assigned with the M knapsacks is equivalent to $\boldsymbol{\mathcal{P}_0}$.

B. Problem Conversion

According to [24], the NP-hard problem can be solved with the Lagrangian relaxation.

First, we relax (23b) with a Lagrange multiplier vector defined as $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_M)$ and $\boldsymbol{\mathcal{P}_0}$ can be converted as

$$\begin{aligned} \boldsymbol{\mathcal{P}_1}:\ \min_{x_{m,n}}\ & \sum_{m=1}^{M}\sum_{n=1}^{N} S_{m,n}x_{m,n} \\ & + \sum_{m=1}^{M}\lambda_m\left(\sum_{n=1}^{N} l_n x_{m,n} - C_m\right) && \text{(25a)}\\ \text{s.t.}\ & \sum_{m=1}^{M} x_{m,n} \le 1,\ \ \forall n && \text{(25b)}\\ & x_{m,n}\in\{0,1\},\ \ \forall m,n && \text{(25c)}\\ & \lambda_m \ge 0,\ \ \forall m, && \text{(25d)} \end{aligned}$$

where λ_m is the *m*th Lagrange multiplier.

For simplicity, we rewrite (25) as

$$\begin{aligned} \boldsymbol{\mathcal{P}_2}:\ \min_{x_{m,n}}\ & \sum_{m=1}^{M}\sum_{n=1}^{N}\psi_{m,n}x_{m,n} - \sum_{m=1}^{M}\lambda_m C_m \\ \text{s.t.}\ & \text{(25b), (25c), (25d),} && \text{(26)} \end{aligned}$$

where $\psi_{m,n} = \lambda_m l_n + S_{m,n}$.

Given $\boldsymbol{\lambda}$, we equivalently transform $\boldsymbol{\mathcal{P}_2}$ into

$$\begin{aligned} \boldsymbol{\mathcal{P}_2^*}:\ \min_{x_{m,n}}\ & \sum_{m=1}^{M}\sum_{n=1}^{N}\psi_{m,n}x_{m,n} \\ \text{s.t.}\ & \text{(25b), (25c).} && \text{(27)} \end{aligned}$$

Obviously, $\boldsymbol{\mathcal{P}_2}$ is tractable by decomposing $\boldsymbol{\mathcal{P}_2^*}$ into N sub-problems, and the *n*th sub-problem that deals with the placement of the *n*th content f_n is formulated as

$$\begin{aligned} \boldsymbol{\mathcal{P}_{2,n}}:\ \min_{x_{m,n}}\ & \sum_{m=1}^{M}\psi_{m,n}x_{m,n} \\ \text{s.t.}\ & \text{(25b), (25c).} && \text{(28)} \end{aligned}$$

Then, we can obtain the optimal solution vector to $\boldsymbol{\mathcal{P}_2}$ by solving the N sub-problems given $\boldsymbol{\lambda}$.

Finally, due to the Lagrange relaxation, $\boldsymbol{\mathcal{P}_2}$ is the lower bound to $\boldsymbol{\mathcal{P}}_0$ [24]. In this case, we use the Lagrangian dual problem to maximize $\boldsymbol{\mathcal{P}_2}$, given by

$$\begin{aligned} \boldsymbol{\mathcal{P}_3}:\ \max_{\boldsymbol{\lambda}}\ & \boldsymbol{\mathcal{P}_2} \\ \text{s.t.}\ & \text{(25b), (25c), (25d).} && \text{(29)} \end{aligned}$$

Algorithm 2 Optimal Content Placing Algorithm

1: $t = 0$, $\boldsymbol{\lambda} = \mathbf{0}$, $\left|\left|\nabla\left(\boldsymbol{\lambda}^{(t)}\right)\right|\right|^2 = \infty$;
2: Set $S_{m,n}$, T;
3: **while** $t \leq T$ or $\left|\left|\nabla\left(\boldsymbol{\lambda}^{(t)}\right)\right|\right|^2 \leq \varepsilon$ **do**
4: **for** $n = 1$ *to* N **do**
5: Solve $\boldsymbol{\mathcal{P}(n)}$ according to (28) as m^*;
6: Set the m^*th CN to place f_n;
7: **end for**
8: Update $\boldsymbol{\lambda}^{(t)}$;
9: $t = t+1$;
10: **end while**
11: Obtain $\mathbf{X}$;
12: Calculate $\boldsymbol{\Theta}$;
13: $\mathbf{w} = \text{Sort}(\mathcal{F}, \boldsymbol{\Theta})$;
14: **for** $j = N$ *to* 1 **do**
15: Set $m^* = \text{Find\_CN}(\mathbf{w}[j], \mathbf{X})$;
16: **if** $\text{Length}(\mathbf{w}[j]) \leq C_m$ **then**
17: Set $m' = m^*$
18: **else**
19: $\mathcal{M} = \text{Available\_CN\_Set}(\mathbf{w}[j])$;
20: $m' = \text{Find\_optimal\_CN}(\mathbf{w}[j], \mathcal{M})$;
21: **end if**
22: $C_{m'} = C_{m'} - \text{Length}(\mathbf{w}[j])$;
23: **end for**
24: **return** $\mathbf{X}'$.

To solve $\boldsymbol{\mathcal{P}_3}$, we use the sub-gradient approach in [24]. In this way, the optimal solution to $\boldsymbol{\mathcal{P}_3}$ is also optimal for $\boldsymbol{\mathcal{P}_0}$.

C. Solution Algorithm

In the part, we develop the optimal content placement algorithm (OCPA) to solve $\boldsymbol{\mathcal{P}_0}$, as shown in Algorithm 2.

Generally, the algorithm consists of two modules. From lines 1 to 11, the first module adopts a sub-gradient approach to solve $\boldsymbol{\mathcal{P}_3}$. However, the Lagrange relaxation used in this module may lead to a infeasible solution to $\boldsymbol{\mathcal{P}_3}$. As lines 12 to 24 shown, the second module adjusts the solution from the first module to be feasible by a heuristic approach. In the following, we elaborate Algorithm 2.

Lines 1 and 2 initialize the parameters. For example, we set T as the maximal iterations, obtain $S_{m,n}$ from (21), and compute the sub-gradient as follows:

$$\nabla\left(\lambda_m^{(t)}\right) = \sum_{n=1}^{N} l_n x_{m,n}^{(t)} - C_m, \tag{30}$$

where $\lambda_m^{(t)} \in \boldsymbol{\lambda}^{(t)}$ denotes the mth Lagrange multiplier in the tth iteration.

In line 5, we identify the optimal solution of the nth sub-problem in (28) as

$$m^* = \underset{m \in \{1,\ldots M\}}{\arg\min} \ \psi_{m,n}, \tag{31}$$

where m^* is the optimal CN to place the nth content. Thus, according to (24), $x_{m^*,n} = 1$. If $m \neq m^*$, $x_{m,n} = 0$. Then, we solve $\boldsymbol{\mathcal{P}_2}$ by the optimal solutions to all N sub-problems.

In line 8, $\boldsymbol{\lambda}^{(t)}$ is updated by

$$\boldsymbol{\lambda}^{(t+1)} = \max\left\{\boldsymbol{\lambda}^{(t)} + \vartheta_t \nabla\left(\boldsymbol{\lambda}^{(t)}\right),\ \mathbf{0}\right\}, \tag{32}$$

where ϑ_t is given by

$$\vartheta_t = \frac{V_{ub}(t) - V_{lb}(t)}{\left|\left|\nabla\left(\boldsymbol{\lambda}^{t}\right)\right|\right|^2}\delta_t, \tag{33}$$

where δ_t is the step size and $V_{ub}(t)$ and $V_{lb}(t)$ are the upper and lower bounds of $\boldsymbol{\mathcal{P}_0}$, respectively.

We note that ε is a small positive number in line 3. When the loop terminates, $\mathbf{X}$ is the solution to $\boldsymbol{\mathcal{P}_3}$. However, due to the relaxation with (23b), it is possible that the placement size for some CN induced by $\mathbf{X}$ is larger than its cache size. In this case, $\mathbf{X}$ is an infeasible solution. To cope with the problem, we design a heuristic approach to adjust $\mathbf{X}$ to be feasible in the second module. In this approach, we first devise a placing order for the contents according to $\mathbf{X}$, and then select the CNs whose remaining cache size is enough to place the contents. Thus, the key of the heuristic approach lies in the design of the placing order.

In lines 12 and 13, we sort the contents such that the content with the smaller placement energy is placed with higher priority. Let $\boldsymbol{\Theta}$ denote the placement order vector with $\Theta(n)$ being the placement order of f_n, given by

$$\Theta(n) = \frac{\overline{\phi}(n)}{l_n}, \quad \forall f_n \in \mathcal{F}, \tag{34}$$

where the size of the nth content l_n is defined in (20) and $\overline{\phi}(n)$ is the average preference degree of all SBSs on f_n, given by

$$\overline{\phi}(n) = \frac{1}{K}\sum_{k=1}^{K} \phi_{k,n}, \quad \forall f_n \in \mathcal{F}. \tag{35}$$

Let us elaborate (34). The larger l_n costs a larger energy consumption for downloading f_n, while the larger $\overline{\phi}(n)$ corresponds to the smaller requesting probability of f_n from (6). As a result, the smaller Θ can save more energy. Thus, the content with smaller Θ has the higher priority to be placed.

As shown in line 13, we sort the contents according to $\boldsymbol{\Theta}$ in ascending order and the sorted contents are stored in an array, denoted as $\mathbf{w}$. From lines 14 to 24, we adjust $\mathbf{X}$ to be feasible. In line 15, we find the m^*th CN to place the content $\mathbf{w}[j]$ according to $\mathbf{X}$. If the CN has enough memory to cache the content, we adopt $\mathbf{X}$ as the placement solution, as depicted in lines 16 and 17. Otherwise, we adjust the placement solution. In line 19, we find the available CN set for the content $\mathbf{w}[j]$, defined as

$$\mathcal{M} = \{m \in \{1, \ldots, M\} | \text{Length}(\mathbf{w}[j]) \leq C_m\}. \tag{36}$$

where $\text{Length}(\mathbf{w}[j])$ is the operator used to obtain the size of $\mathbf{w}[j]$. In line 20, we identify the optimal CN according to (28) as

$$m' = \underset{m \in \mathcal{M}}{\arg\min}\, \psi_{m,n}, \tag{37}$$

where the m'th CN is the adjusted CN selected from $\mathcal{M}$. Note that a content will not be downloaded if there is no available CN in $\mathcal{M}$ for the content. Finally, in line 24, the algorithm returns the feasible optimal content placement $\mathbf{X}'$.

TABLE I
PARAMETERS SETTING FOR PLC NETWORKS

Parameters	Symbol	Value
PLC Path Number	I	5
PLC Path Weight	g	-0.15
PLC Path Distance	d	244.8m
PLC Channel Attenuation	a_1	7.8×10^{-10}
PLC Channel Bandwidth	B	10MHz
MBS Transmission Power	$P_{M\to k}$	30dB
Noise Power	σ_k^2	$-15dB$

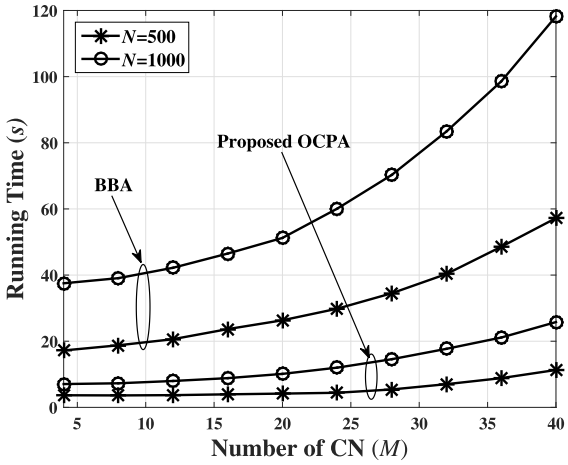

Fig. 4. Running time versus the different numbers of CN.

V. NUMERICAL RESULTS

We simulate a cache-enabled PLC network with an MBS and *K* SBSs in Fig. 1. The content set has *N* contents with different sizes and the sizes are randomly generated with the uniform distribution, denoted as U(1, 5). The requesting probabilities of the SBSs follow the Zipf distribution with $\beta = 0.85$ and a randomly generated $\phi_{k,n}$ in (6). Moreover, we use the parameters of the PLC networks according to the HomePlug AV2 standard [25], as listed in Table I.

To evaluate the performance of proposed optimal placement algorithm, three typical contents placement algorithms are used as the benchmark, described as follows.

- *Random Algorithm (RA):* Contents are randomly assigned with CNs.
- *Popularity Based Algorithm (PBA):* Contents with higher popularity are more likely to be downloaded by the CNs.
- *Branch and Bound Algorithm (BBA):* Contents are placed with the integer programming algorithm [26].

Fig. 4 compares the average running time between the proposed OCPA and BBA versus the number of CNs. The number of SBSs is set as 100 and the numbers of contents (i.e., *N*) are 500 and 1000. First, we observe that the running time of OCPA is much less than that of BBA. The average running time for OCPA is mainly affected by the number of contents rather than the number of CNs. Additionally, the running time of OCPA is less than 30 seconds even though the number of CNs is up to 40. Second, we observe that the running time of BBA increases slightly when the number of CNs (i.e., *M*) is less than 15. However, when *M*>15, the running time increases in an exponential way. The running time reaches 120 seconds when $M = 40$. The reason is that the complexity of the integer programming algorithm dramatically increases with the numbers of CNs and contents.

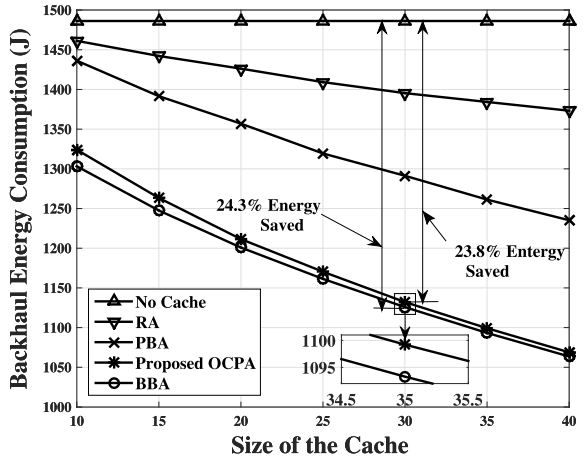

Fig. 5. Backhaul energy consumption comparison among different caching algorithms.

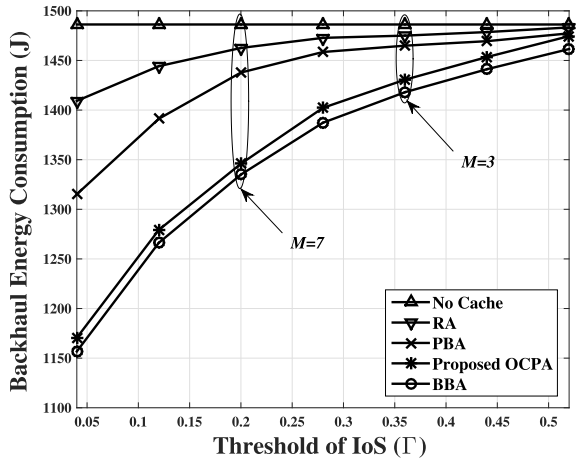

Fig. 6. Backhaul energy consumption comparison under different Ios thresholds.

Fig. 5 dipects the backhaul energy consumption versus the size of the cache sizes in the CNs under different caching algorithms. The number of CNs, SBSs, and contents are set as 17, 50, and 1000, respectively. First, we observe that the backhaul energy consumption decreases as the cache size goes up. Second, BBA slightly outperforms OCPA due to a smaller energy consumption. The reason is that the OCPA is the approximate solution algorithm while the BBA is an exact one. Third, we can also find that the backhaul energy consumption of OCPA is much smaller than RA and PBA.

Fig. 6 shows the backhaul energy consumption with different algorithms versus the threshold of IoS. The cache size is set as 20. We observe that all caching algorithms consume more backhaul energy under a larger Γ. The reason is that a larger Γ implies that a smaller number of SBSs can be selected as the CNs. For example, when $\Gamma = 0.36$, 3 out of 50 SBSs are selected as the CNs since their associated IoS are larger than Γ. When $\Gamma = 0.2$, 7 out of 50 SBSs are selected as the CNs.

Since the cache size of each CN is fixed, a smaller number of CNs implies that the total cache memory shrinks, resulting in less cached contents. Consequently, this will cause more backhaul energy consumption due to higher backhaul traffic.

VI. Conclusion

In this paper, we have designed a cache-enabled PLC network to reduce the backhaul energy consumption. Toward this end, we first deployed the cache devices at some SBSs to serve the IUs by storing the popular contents from the MBS. Second, we modeled the requesting relations among the SBSs as a directed graph. In particular, we proposed an algorithm to select a portion of SBSs as the CNs by computing the IoS according to the directed graph. Third, we formulated the optimal content placement problem to minimize the backhaul energy consumption. Going forward, several follow-up directions deserve further investigation. First, wireless caching can be used to further extend the coverage of the PLC networks. The hybrid design of PLC and wireless caching networks remains open. Second, it is of interest to select the SBSs as CNs in PLC networks according with the social networks, where each CN is selected from the SBSs clustered by friendship relation with the same content preference.

Appendix A
Proof of Theorem 1

We model the iteration in (14) with a Markov process, where $\mathbf{p}_n$ corresponds to the nth state of the Markov process and $\widehat{\mathbf{A}}$ is the state transfer matrix. In this way, the convergence of (14) is equivalent to the convergence of the Markov process. From [27], the corresponding Markov process is convergent under the condition that the state transfer matrix is stochastic, irreducible and non-periodic. Therefore, to validate the convergence of (14), we need to prove $\widehat{\mathbf{A}}$ is stochastic, irreducible and non-periodic.

According to (13), we denote the sum of jth row of $\widehat{\mathbf{A}}$ as

$$r_j = \sum_{k=1}^{K} r_{j,k} = (1-\rho)\sum_{k=1}^{K}\frac{1}{K} + \rho\sum_{k=1}^{K} a_{j,k}, \quad j = 1, 2, \ldots, K. \tag{38}$$

From (8), we have

$$a_j = \sum_{k=1}^{K} a_{j,k} = \begin{cases} \sum_{k=1, u_j \in \mathcal{U}_1}^{K} \frac{1}{\Phi_j}, & \Phi_j \geq 1 \\ \sum_{k=1, k\neq j}^{K} \frac{1}{K}, & \Phi_j = 0 \end{cases} \tag{39}$$

With (39) and (38), $\widehat{\mathbf{A}}$ is a stochastic matrix, since $r_j = 1,\ \forall j$.

Moreover, Let $\mathbf{J} = \{\pi_{j,k}\}_{K\times K}$ be the *accessibility matrix* of $\widehat{\mathbf{A}}$, where

$$\pi_{j,k} = \begin{cases} 1, & r_{j,k} > 0 \\ 0, & r_{j,k} = 0. \end{cases} \tag{40}$$

In (13), $\frac{1-\rho}{K}\mathbf{e}\mathbf{e}^T$ is composed of K^2 positive value $\frac{1-\rho}{K}$, which ensures that $r_{j,k}$ is positive. Then, $\pi_{i,j} \equiv 1$ from (40). Therefore, $\widehat{\mathbf{A}}$ is irreducible, since $\mathbf{J}$ is irreducible.

Finally, given a positive integer i, the ith power of $\widehat{\mathbf{A}}$, denoted as $\widehat{\mathbf{A}}^i$, is also positive, since $\widehat{\mathbf{A}}$ is positive. Thus, $\widehat{\mathbf{A}}$ is non-periodic.

Overall, the Markov process is convergent, since $\widehat{\mathbf{A}}$ is a stochastic, irreducible, and non-periodic. As such, the iteration in (14) is convergent.

References

[1] Y. Qian, J. Li, Y. Zhang, and D. K. Jayakody, "Performance analysis of an opportunistic relaying power line communication systems," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3865–3868, Dec. 2018.

[2] D. López-Pérez, M. Ding, H. Claussen, and A. H. Jafari, "Towards 1 Gbps/UE in cellular systems:understanding ultra-dense small cell deployments," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 2078–2101, 4th Quart., 2015.

[3] H. Zhang, Y. Dong, J. Cheng, M. Hossain, and V. Leung, "Fronthauling for 5G LT E-U ultra dense cloud small cell networks," *IEEE Commun. Mag.*, vol. 23, no. 6, pp. 48–53, Dec. 2017.

[4] Y. Sheng, "Scalable intelligence-enabled networking with traffic engineering in 5G scenarios for future audio-visual-tactile Internet," *IEEE Access*, vol. 6, pp. 30378–30391, 2018.

[5] W. Han, A. Liu, and V. Lau, "PHY-Caching in 5G wireless networks: Design and analysis," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 30–36, Aug. 2016.

[6] B. N. Bharath, K. G. Nagananda, and H. V. Poor, "A learning-based approach to caching in heterogenous small cell networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 1674–1686, Apr. 2016.

[7] M. Gregori, J. Gómez-Vilardebó, J. Matamoros, and D. Gündüz, "Wireless content caching for small cell and D2D networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1222–1234, May. 2016.

[8] J. Liao, K. Wong, Y. Zhang, Z. Zheng, and K. Yang, "Coding, multicast and cooperation for cache-enabled heterogeneous small cell networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6838–6853, Oct. 2017.

[9] Y. Qian *et al.*, "Cache-enabled MIMO power line communications with precoding design in smart grid," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 1, pp. 315–325, Mar. 2020.

[10] T. Fang, H. Tian, Y. Yang, X. Liu, D. Wu, and X. Chen, "An influence factor based caching node selection algorithm in D2D networks," in *Proc. IEEE ICCT*, Oct. 2017, pp. 805–809.

[11] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[12] J. Li *et al.*, "On social-aware content caching for D2D-enabled cellular networks with matching theory," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 297–310, Feb. 2018.

[13] J. Liu, B. Bai, J. Zhang, and K. B. Letaief, "Cache placement in Fog-RANs: From centralized to distributed algorithms," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7039–7051, Nov. 2017.

[14] S. Wang, X. Huang, Y. Liu, and R. Yu, "Cachinmobile: An energy-efficient users caching scheme for fog computing," in *Proc. IEEE ICCC*, Jul. 2016, pp. 1–6.

[15] B. Dai and W. Yu, "Joint user association and content placement for cache-enabled wireless access networks," in *Proc. IEEE ICASSP*, Mar. 2016, pp. 3521–3525.

[16] Q. Jia, R. Xie, T. Huang, J. Liu, and Y. Liu, "Energy-efficient cooperative coded caching for heterogeneous small cell networks," in *Proc. IEEE INFOCOM*, Atlanta, GA, USA, Oct. 2017, pp. 468–473.

[17] F. Gabry, V. Bioglio, and I. Land, "On energy-efficient edge caching in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3288–3298, Sep. 2016.

[18] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[19] A. Dubey, R. K. Mallik, and R. Schober, "Performance analysis of a power line communication system employing selection combining in correlated log-normal channels and impulsive noise," *IET Commun.*, vol. 8, no. 7, pp. 1072–1082, May 2014.

[20] Z. Chang, Y. Gu, Z. Han, X. Chen, and T. Ristaniemi, "Context-aware data caching for 5G heterogeneous small cells networks," in *Proc. IEEE ICC*, May 2016, pp. 1–6.

[21] Wikimedia Inc. (2018). *Page Rank*. [Online]. Available: https://en.wikipedia.org/wiki/PageRank

[22] H. Huang, S. Yoo, D. Yu, and H. Qin, "Diverse power iteration embeddings: Theory and practice," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2606–2620, Oct. 2016.

[23] Y. Chen and J. Hao, "Memetic search for the generalized quadratic multiple knapsack problem," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 908–923, Dec. 2016.

[24] C. Lemaréchal, "Lagrangian relaxation," *Comput. Combinatorial Optim.*, vol. 2241, pp. 112–156, Nov. 2001. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-45586-8_4#citeas
[25] A. Smith, *Homeplug AV White Paper*, Home Plug Powerline, Alliance, Dunwoody, GA, USA, 2005.
[26] P. Avella, M. Boccia, and I. Vasilyev, "A branch-and-cut algorithm for the multilevel generalized assignment problem," *IEEE Access*, vol. 1, pp. 475–479, 2013.
[27] R. Tweedie and S. Meyn, *Markov Chains and Stochastic Stability*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

Yuwen Qian received the Ph.D. degree in automatic engineering from Nanjing University of Science and Technology, Nanjing, China, in 2011, where he was a Lecturer from July 2002 to April 2019. Since 2019, he has been an Associate Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology. His main research interests include information security, smart grid, and power line communications.

Liuqiang Shi received the B.E. degree in electronic information science and technology from Anqing Normal University, Anqing, China, in 2015. He is currently pursuing the M.E. degree in communication and information system with Nanjing University of Science and Technology. His current research interests include cooperative modulation recognition in PLC and channel allocation in hybrid network of PLC and wireless.

Long Shi (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of New South Wales, Sydney, Australia, in 2012. He was a visiting student with the Chinese University of Hong Kong, Hong Kong, in 2010 and the University of Delaware in 2011. From 2013 to 2016, he was a Postdoctoral Fellow with the Institute of Network Coding, Chinese University of Hong Kong. From 2014 to 2017, he was a Lecturer with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is currently a Research Fellow with the Singapore University of Technology and Design. His current research interests include mobile-edge computing, wireless caching, and deep reinforcement learning.

Kui Cai (Senior Member, IEEE) received the B.E. degree in information and control engineering from Shanghai Jiao Tong University, Shanghai, China, and the joint Ph.D. degree in electrical engineering from the Technical University of Eindhoven, The Netherlands, and the National University of Singapore. She is currently an Associate Professor with Singapore University of Technology and Design. Her main research interests are in the areas of coding theory, information theory, and signal processing for various data storage systems and digital communications. She received the 2008 IEEE Communications Society Best Paper Award in coding and signal processing for data storage. She served as the Vice-Chair (Academia) of the IEEE Communications Society and Data Storage Technical Committee from 2015 to 2016.

Jun Li received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. From January 2009 to June 2009, he worked with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow with the School of Electrical Engineering and Telecommunications, University of New South Wales, Australia. From April 2012 to June 2015, he was a Research Fellow with the School of Electrical Engineering, University of Sydney, Australia. Since June 2015, he has been a Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include network information theory, channel coding theory, wireless network coding, and resource allocations in cellular networks.

Feng Shu received the B.S. degree from Fuyang Teaching College, Fuyang, China, in 1994, the M.S. degree from Xidian University, Xi'an, China, in 1997, and the Ph.D. degree from Southeast University, Nanjing, China, in 2002. From October 2003 to October 2005, he was a Postdoctoral Researcher with the National Key Mobile Communication Laboratory, Southeast University. From September 2009 to September 2010, he was a Visiting Postdoctoral Researcher with the University of Texas at Dallas. In October 2005, he joined the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, where he is currently a Professor and a Supervisor of Ph.D. and graduate students. His research interests include wireless networks, wireless location, and array signal processing.