(Article begins on next page)

# Performance Improvements Through Recommendations for a PLC network with Collaborative Caching in Remote Areas

Zunera Umar and Michela Meo
*Department of Electronics and Telecommunications*
*Politecnico di Torino*
Turin, Italy
{firstname.lastname}@polito.it

***Abstract*—The emergence of Power Line Communication (PLC) technology has facilitated the expansion of broadband access networks in remote areas, by utilizing existing wired power infrastructure. However, the growing demand for data, driven by the popularity of communication services, presents a formidable challenge to the underlying PLC technology. Collaborative caching involves the sharing of cached content among neighboring nodes, thereby improving cache hit ratio (CHR), reducing network and backhaul congestion, and ultimately enhancing network performance. Our research proposes a recommendation system integrated into the collaborative caching mechanism on a PLC network that suggests relevant content to the users based on users' preferences and historical usage patterns leading to an increase in CHR and a reduction in network congestion. The results indicate that the proposed system significantly improves network performance by reducing download delay and saving precious backhaul link resources thus making PLC networks more effective for remote areas.**

***Index Terms*—PLC, 6G networks, Edge caching, Remote Areas**

## I. Introduction

### A. Motivation

Despite the widespread availability of basic ICT services, a significant portion of people living in remote, sparsely populated, and rural areas lack access to them, leading to a significant digital divide between individuals across the globe [1], [2]. This can limit economic opportunities, educational access, and healthcare services, among other things. 6G technology has taken into account the peculiarities of rural and remote areas and aimed to bridge this digital divide by providing faster and reliable connectivity to these areas [3], [4].

Power line communication (PLC) is viewed as a viable solution for delivering high-speed connectivity over high- and medium-voltage power lines, thereby enhancing the capacity of backhaul networks in remote areas. PLC offers an affordable and easily accessible means of expanding network reach [5]. The evolution of internet services has led to a transformation in traffic demand. Recent findings indicate that video content now represents a significant majority of total traffic, accounting for approximately 82% of all internet traffic by the year 2021 [6]. Albeit its cost-effectiveness, PLC is known to have certain limitations in terms of its bandwidth capacity. This can result in significant delays when downloading video content, which can be a major challenge in the context of modern internet traffic demands [7].

Video service providers are caching their content at the edge in order to meet the growing demand for high-bandwidth data-intensive video streaming. They aim to reduce the need for expensive resources in conventional content delivery networks (CDNs), thereby lowering costs. Edge caching enables effective data retrieving and results in shorter delays which also reduces the overall network energy consumption and contribute in green networking [8], [9]. For a PLC network with limited connectivity, this would boost the overall performance of front-end network service requirements and reduce the load on back-end network.

### B. Our Approach and Contributions

The video traffic tends to be concentrated on a small number of popular content items, it favors the caching of the high demand content at the network edge and made Content Distribution Networks (CDNs), a vital part of today's internet. Therefore, in our previous study we suggested an amalgamation of PLC and edge caching to provide popular content in remote areas alongside connectivity and we found that edge caching helps in reducing the load on backhaul bottleneck link and improves the user experience by reducing the delays [10]. Nevertheless, the number of popular content items often exceeds the caching capacity, resulting in a significant number of users' requests that still be redirected to the backhaul link. Moreover, to facilitate a rapid change in popularity of the content a frequent update in caching is required which imposes an additional burden on backhaul.

User requests are correlated with previous preferences and several studies and practices suggest that the requests can be influenced by recommendation systems [11]. Recommendation systems can help improve caching efficiency and network performance by suggesting popular content items that are cached at the edge. Therefore, in this paper we envision and study a recommendation system on top of a PLC network with a distribution network where an Edge Server (ES) is deployed and Edge Devices (EDs) are connected through a broadband powerline communication (BPL). A large Cache at ES and comparatively smaller caching facilities are installed at EDs which store the most popular contents and they mutually collaborate to fulfill the user requests in remote areas. At ES, a recommendation engine assists users by offering suggestions for similar content items that may be available within the local caching system, in cases where the requested item cannot be found locally. Our major contributions are:

- We develop a framework to find the probabilities of requests associated with ES and EDs under the content acquisition model.
- We observe the effectiveness of a recommendation system by increasing number of demands from a local caching system and its overall impact on the collaborative caching network.
- We study the usefulness of our approach in the realm of PLC network for remote areas, where the link between the cloud and ES is the bottleneck link and whose resources should be minimally utilized and should be preserved for high priority services.

The rest of the paper is organized as follows. Section II, a background and some related works are presented. In Section III, we define the architecture of collaborative caching over PLC network and the content recommendation system. Section IV presents the system model related to caching, user request and data acquisition and a derivation of the average download time under the proposed method. Section V discusses the performance metrics and framework. Simulation results are then discussed in Section VI and a conclusion is presented in Section VII.

## II. Background & Related work

6G highlights the digital divide and its design incorporates the requirement of connectivity solutions for remote areas which should be affordable and provide sufficient data rates [2]. PLC is a reliable solution for data transfer due to its simple installation and maintenance, and has advantages over wireless communication. Progressive advancements in the field have improved achievable data rates to 1Gbps, making it a potential solution for providing internet connectivity in underdeveloped regions [12]. Since large portion of internet traffic comprises of data demand, the utilization of edge computing and artificial intelligence is a viable approach for enabling content-aware caching to enhance the data services in rural areas [13].

The idea of *soft cache hits* is introduced where the user is suggested by the alternative content when the requested content is not available and this idea is applied on femto caching to achieve additional gains [14]. Moreover, recommendations and caching is considered as a joint problem in a way to increase the caching efficiency without staking the quality of recommendations [15]. In the context of PLC network, a cache-enabled multiple-input multiple-output (MIMO) PLC framework is proposed, which caches frequently accessed content in proximity to users to minimize average downloading price for each use [16]. Furthermore, from Internet of Things (IoT) perspective, edge computing has been explored as a means of reducing redundant data transmission between end-users and the cloud over the PLC network [17].

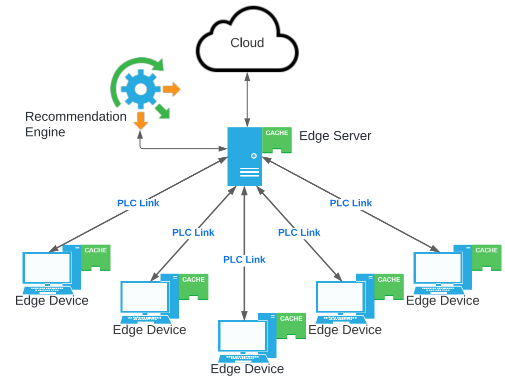


Fig. 1: The proposed collaborative caching network over PLC network.

In this study, we present recommendation system approach to enhance the performance of collaborative caching over the PLC network. The impact would be seen as reduction in load on bottleneck backhaul link and improvement in user satisfaction. We compared our approach with the previous study, [10], and presented the results in Section VI.

## III. Problem Setup

### A. Caching over PLC network

In PLC, electrical wires that are traditionally used for power transmission are used to transmit data and this provides a cost effective solution for bringing connectivity for unconnected part of the world, such as rural and remote areas.

In our scenario as depicted in Fig.1, the network consists of a PLC network with a Edge server (ES) and end devices (EDs), which are equipped with caching facility. Contents that are frequently requested (i.e., with high popularity index) are stored in the central node (ES), while less popular contents are cached in the user nodes (EDs) to support collaborative caching. A user request for a content can be fulfilled through following possible ways.

- Content is available in ED's own cache.
- Retrieving the content from ES, which is the central node that manages all the EDs and has most popular content cached.
- If the content is not available in ES cache, a database lookup is performed to check whether another ED has the content, subsequently the content is first downloaded by ES and then transferred to the destination node which initiated the request.
- In case content is not available in the local caching system, ES recommends the user with similar contents, which are also available in local caches i.e. ES and EDs; recommendation is related to the users past requests history.
- If the user refuses to accept the recommendation, the

content is retrieved from the cloud.

Caching capacity is proportional to its size which can become an additional bottleneck and cause performance issues. Therefore, the proposed recommendation system on top of caching over PLC network can help improve the user experience by providing personalized recommendations based on the user's interests and browsing history. This improves the overall performance of the system and can help reduce the load on the underlying PLC network [10].

### B. Content Recommendation System

In the proposed scenario, we incorporate a *content based* recommendation system that provides personalized suggestions to the users based on their past interactions with items and the content of those items. ES keeps track of previous requests of the user by maintening the database ($B$) of recent content requests. When a user requests a content that is not stored in local cache (ES and EDs), for each item stored in user's past history, ES computes the distance between the embeddings using cosine similarity. This way a pairwise similarity score $x_{ij}$ is computed for each of the past requested content $i \in B$ with every cached content $j \in M$. Using this carefully curated list of similar contents, our recommendation system selects items with the highest weighted average rating, i.e. $x_{ij} \to 1$ , based on user scores (including ratings and vote count), and presents these top $U$ recommendations to the user [18].

## IV. System Model

### A. Caching & Content Model

ES and EDs are equipped with cache facility with a limited capacity of $N$ contents. From $N$ contents those with lower popularity index are stored in EDs whereas most popular contents are saved in ES. The cached content is based on IMDB's weighted rating formula, which considers user vote count $v$ and average rating $R$.

$$\mathrm{W}eightedRating(WR) = \left(\frac{v}{v+m} \cdot R\right) + \left(\frac{m}{v+m} \cdot C\right)$$

Where $m$ is the minimum votes required to be considered for caching and $C$ is the mean vote across the complete catalog.

The cached content in ES and EDs belongs to a catalog $M$ and user can request a content which may be present in $M$ or otherwise recommendation system provides the user with similar suggestions to the user's past preferences.

### B. User Request Model

We assume that the content requests in each ED occur independently following a Poisson arrival process. The requests for the contents are associated with the probability $P$ defined through Zipf law with exponent parameter $\alpha$ [19], which controls the skewness of the popularity curve. Higher value of $\alpha$ defines a more skewed distribution and hence large probability of request gets associated with the most popular contents that are stored in ES and EDs. This leads to the definition of the two operational modes of the proposed system:

***1) Without Recommendation System:*** The user's request is satisfied by leveraging the cooperative caching resources of ES and EDs. If the requested content is not available in the local cache of ES and EDs, it may have to be retrieved from the cloud link, which incurs a higher retrieving cost and is only done if sufficient resources are available.

***2) With Recommendation System:*** For requested content which is not available in the local caching system (ES and EDs), ES suggests popular contents that are relevant to the user's past history and currently stored in the caching system. The probability of choosing an item from the list of recommended content is given by:

$$Prob_i = \frac{P_i}{\sum_{j=1}^{n} P_j}$$

Where $P_i$ is the popularity of $i$–th movie.

The user accepts the suggestion with probability $a$, whereas with probability $(1-a)$ it rejects the recommendation and requests for another content. The estimated value of $a$ on average is 0.8 on Netflix [20], but this may vary among users. Therefore, in our work we analyze the behaviour of the system with a small shift of value of $a$.

In this work the content popularity is assumed to remain constant over time and content catalog is only refreshed at low load periods. Hence, any impact of a change in content popularity is not taken into account. Also, the update of caches at both ES and EDs occurs when cloud resources are available and outside of peak hours in order to prevent congestion in the PLC network and provide users with the maximum available bandwidth.

### C. Content Acquisition Model

We use the user request model that is presented in previous study [10] whose definition is presented in IV-B. As shown in Fig. 2, the model draws a value from zipf distribution with an exponent parameter $\alpha$, the probability that the request of a content $m$ is served my ES or EDs can easily be calculated through the following rules:

***a) ES***: Up to $M_1$ most popular contents are cached at ES and probability of retrieving a content $m$ is given by:$P_\beta = P\{m \leq M_1\}$

***b) ED***: Those contents with popularity smaller then $M_1$ and larger then $M_2$ are randomly stored in EDs. Any content request whose popularity lies in this interval can be fetched from another ED with probability: $P_\gamma = P\{M_1 < m \leq M_2\}$. First the content is uploaded to the ES and then transferred to the requesting ED. The probability that ED $i$ requests a content that is stored in ED $i$–th own cache is: $P_{\theta_i} = P_\gamma / N$ ; where $N$ represents number of EDs in the system.

***c) Cloud resource or Recommendation system***: The probability that a content cannot be locally found and it should be retrieved from the cloud, if possible, is: $P_L = P\{m > M_2\} = 1 - P_\beta - P_\gamma$. However, at this point the user is given with the list of $U$ recommendations of similar and popular contents. Since the user accepts the recommendation with probability $a$ and rejects the recommendation with probability $(1-a)$ (Sec. III), then the new $P_L$ will be: $P_L^* = P_L(1-a)$ and consequently a gain will be seen in probability of finding a content in ES

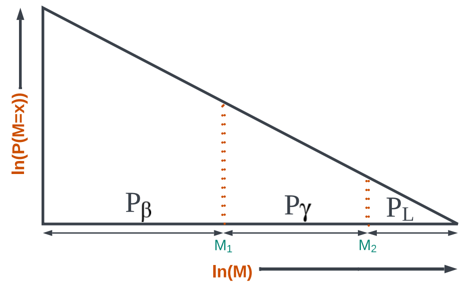


Fig. 2: Distribution of content with Zipf

and EDs equal to $(a.P_L)$. The probabilities $P_\beta$, $P_\gamma$, $P_L$ can be easily derived from cumulative density function of zipf distribution.

$$F(x) = P(M \leq x) = \frac{H_{x,\alpha}}{H_{M,\alpha}} \quad (1)$$

where $H_{M,\alpha}$ is the $M-$th generalized harmonic number considering $M$ number of contents cached in ES and ED.

### D. Average Download Time

All data requests from the ES to the ED $i$ are served using a processor sharing (PS) discipline over the link capacity $C_i$. Each of $n$ parallel services receives a capacity $C_i/n$ and the PLC link is modeled as $M/M/1$ queue with PS service discipline. $C_i$ The utilization of link $i$ is:

$$\rho_i = \frac{\lambda_i(1 - P_L^* - P_{\theta_i}) \cdot x}{C_i} \quad (2)$$

where $x$ is the size of the requested content in bytes, and is exponentially distributed. The term $(1 - P_L^* - P_{\theta_i})$ represents the probability of requests that are retrieved through the collaborative caching (ES & EDs).

Under the link utilization presented in (2), we compute the average download time for the content requested by ED $i$ from ES by following expression:

$$d_i(x) = \frac{x/C_i}{1 - \rho_i} \quad (3)$$

Consider the scenario in which the requested content is not available at ES cache. Then the average time it takes to download such a content is the combined time it takes to upload the content from the ED with the content to the ES, and to transfer the content to ED that made the request. Since the overall cache capacity of EDs is much less than ES and less popular contents are stored in EDs, the overall number of requests that are facilitated by EDs are fewer. This makes the load on uplink channel i.e. ED to ES small and we approximate the upload time of an average file size of $x$ bytes over a link capacity $C_i$ as $d_u = x/C_i$. Furthermore, the capacity of PLC link for uploading and downloading is considered the same.

Since the probability of attainment of content varies according to the distribution of content, we can calculate a total average download time $T_i$ using the content acquisition model defined in Sec. IV-C. In our previous work [10], the model does not include the gain achieved with the recommendation system. Since a fraction of the requests associated to cloud link is now downloaded through ES and EDs, the gain, $(a \cdot P_L)$, in $P_\beta$ and $P_\gamma$ can be computed by:

$$P_\beta^* = \frac{a \cdot P_L \cdot P_\beta}{P_\beta + (P_\gamma - P_{\theta_i})} + P_\beta \quad (4)$$

$$P_\gamma^* = \frac{a \cdot P_L \cdot (P_\gamma - P_{\theta_i})}{P_\beta + (P_\gamma - P_{\theta_i})} + P_\gamma \quad (5)$$

Where $P_\beta^*$ and $P_\gamma^*$ are the probabilities of requests with the proposed recommendation system. Note that gain is proportional to the original probability of requests (e.g. the contents with high popularity index are stored in ES) and the recommendation system is based on most popular contents similar to the user past history, albeit the chances of recommending the content stored in ES are higher as compared to EDs.

The average download time $T_i$ that is required by ED $i$ for downloading the content from ES and EDs under the collaborative caching with recommendation system can be computed as:

$$T_i = \frac{P_\beta^* \cdot d_i(x) + (P_\gamma^* - P_{\theta_i}) \cdot (d_i(x) + d_u)}{1 - P_L^* - P_{\theta_i}}$$

where $d_i(x)$ and $d_u$ are the average download time from the ES and upload time from ED to ES respectively.

Consider the scenario in which when a user rejects the recommendation and the content is downloaded through the bottleneck cloud link. The average download time in this case is the required total time for downloading the content through ES, EDs or cloud link and is given by:

$$T_{avg} = \frac{P_\beta^* \cdot d_i(x) + (P_\gamma^* - P_{\theta_i}) \cdot (d_i(x) + d_u) + P_L^* \cdot d_{cl}}{1 - P_{\theta_i}} \quad (6)$$

where $d_{cl}$ is the average download time from the cloud link which can be computed as a ratio of the average file size and the available capacity of the cloud link.

## V. Performance Metrics & Framework

In this section, we describe the framework and the metrics used to assess the performance of the proposed collaborative caching with recommendation system for improving the utilization of low capacity PLC network.

### A. Recommendation policy

Following scenarios are considered as a framework to analyze the proposed system along with the comparison to the previous system presented in [10]:

- *System with cache miss:* In this system, we analyze the impact of recommendation system for the scenarios where there is a cache miss due to the content unavailability in the local caching system or it happens due to the non availability of link between ES and cloud, leading to dropped content requests. However, the recommendation system offers users an alternative by suggesting locally available content to download instead.

- *System with partial cloud link capacity:* In this case, a part of the backhaul link's capacity towards the cloud is allocated for video on demand (VOD) services. The user experiences larger downloading delays however there are no cache misses in the system. Given the remote area circumstances, utilizing the link between ES and the cloud for VOD during peak hours, would be costly due to the limited capacity of the link. With the recommendation system, number of requests that would otherwise be driven toward this bottleneck link reduces. When the user rejects the recommended content, only then ES will download the content from bottleneck link between ES and cloud. This manifests that integrating a recommendation system would have a significant impact on optimizing the utilization of expensive resources.

### B. Performance evaluation metrics

We propose the following evaluation metrics to determine the performance of the overall system.

*a) Cache Hit Ratio:* When a content request is fulfilled from the locally cached content, which is stored in either the ES or the EDs, it is referred to as a cache hit. Or else, if the request is transferred to the cloud or either lost due to the unavailability of the cloud link, it is considered a cache miss. Moreover, if the user's initial request is not available locally and it undergoes a recommendation process and redeem an alternate content from ES and EDs, it will as well be a cache hit. We define cache hit ratio as:

$$CHR = H/T = 1 - N/T \tag{7}$$

Where $T$ is the total number of requests, and $H$ is the number of cache hit and $N$ is total cache misses for a given scenario.

*b) Average downloading cost:* This pertains to the proportion of the average download time for collaborative caching with recommendation system utilizing the local caches (ES and EDs) compared to the total average download time, $T_{avg}$, which can be derived from (6). We denote this cost with $D_{local}$ and it can be calculated as:

$$D_{local} = \frac{P^*_\beta \cdot d_i(x) + (P^*_\gamma - P_{\theta_i}) \cdot (d_i(x) + d_u)}{T_{avg}} \tag{8}$$

Similarly, When the user rejects a recommended content and the content request has to be fetched from the cloud, then the cost to retrieve that content is given by:

$$D_{cloud} = \frac{d_{cl} \cdot P^*_L}{T_{avg}} \tag{9}$$

where $d_{cl}$ is the average download time from partially available bottleneck cloud link capacity.

## VI. Performance Evaluation

### A. MovieLens dataset

We use metadata for 45,000 movies listed in the Full MovieLens Dataset [21]. It consists of 26 million ratings scaled from 1-5 obtained from 270,000 users for all 45,000 movies. The dataset consists of datapoints that features cast, crew, plot keywords, budget, etc., for implementing content based recommendation system based on similar features. To sort the popular content based on available ratings, we first derive the weighted average rating of the content using IMDB's formula on the basis of which we store the content in cache facility of ES and EDs. For generating the recommendations, we calculate the cosine distance between all pairs of content and identify the most similar items to the user's past history based on shared characteristics [18]. After identifying the similar items, they are arranged according to the popularity of the available contents and the most popular and similar contents are then suggested to the user.

Note that our main focus is to observe the impact of recommendation in order to boost the performance of collaborative caching network on a low capacity network. Therefore, we keep a simple recommendation system and we are not focusing on improving the recommendation system itself.

### B. Simulation setup

TABLE I: Simulation Parameters.

| Parameters | Symbols | Values |
|---|---|---|
| Video file size | $x$ | 420MB |
| PLC link capacity (ES to ED) | $C_i$ | 20Mbps |
| Cloud link capacity for VOD | $C_p$ | 10Mpbs |
| Packet length | $x_m$ | 1500B |
| Total number of video files | $M$ | 10000 |
| zipf parameter | $\alpha$ | $[0.2, 0.4, 0.6, 0.8, 1]$ |
| Video request rate | $\lambda$ | $[5] \cdot 10^{-3}$ s$^{-1}$ |
| Number of EDs | N | 5 |

To simulate the proposed model, we assume an ES and $N = 5$ EDs which are managed by ES. Each of them is facilitated by a cache to store the popular contents. ES and EDs are connected by a PLC link which has a capacity 20 Mbps and the requests generated by each ES are served on PS service discipline. MovieLens dataset is sorted based on vote count and vote average and then 1000 most popular contents are stored in ES and EDs. This classified content is further categorized into 70% high ranked contents which are cached in ES while the rest of 30% contents are distributed randomly in EDs.

The user request from each ED $i$ follows Poisson arrivals with exponentially distributed inter-arrival times whose parameter $\lambda_i$. Each user can request a content from a catalogue of $M = 10,000$ video files. The probability of requesting a content is based on Zipf($\alpha$) distribution, where $\alpha$ is the skewness parameter whose impact is presented thriough simulations. The request for the content which is not saved in ES and EDs goes through a recommendation process. The user may then choose to accepts or rejects the recommendation in case of which the content may be downloaded from the bottleneck link from ES to cloud. The default parameter $a = 0.8$ [20] is used for most of the simulations however we also observe the impact of change in $a$ in the range $a = [0.7 - 0.9]$ to realize the impact of user acceptance of recommendations on the overall system.

In order to verify the effectiveness of our proposed model, we developed a simulator using SimPy library in python. For

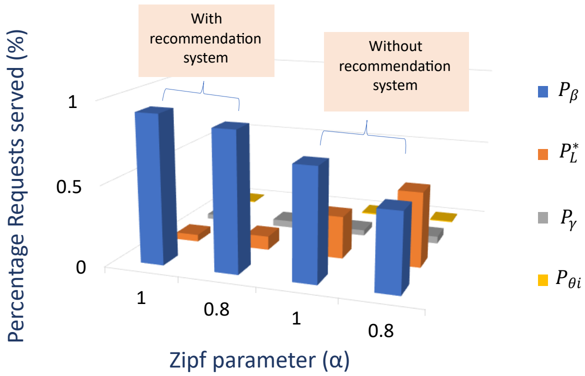


Fig. 3: Impact zipf distribution parameter $\alpha$ on probability of content requested through ES, EDs and lost with recommendation system and without recommendation system.

simulation purpose, the $M$ contents are supposed to be of fixed size of 420 MB with the packet length of 1500 B. All the useful simulation parameters are defined in Table I

### *C. Results*

#### *1) Varying zipf parameter $\alpha$*

In Fig. 3, we compare the performance of proposed system with recommendation and the system without recommendation for two different values of parameter $\alpha$ of the zipf distribution. In this analysis we keep the value of $a$ for the probability of accepting the recommendation constant to 0.8 and observe the change in percentage of the probability of requests served through ES and EDs and also the loss percentage when the cloud link is unavailable for content downloading. $P_L^*$ also represents the percentage of content retrieved from cloud if resources are available. When the value of $\alpha$ is large, a higher number of requests are associated to the highly popular content thereby the requests are directed towards ES. Comparing the two systems, it is evident from the figure that recommendations improve the percentage of requests that are served locally from ES and EDs to about 20% and 35% for $\alpha$ equals to 1 and 0.8 respectively. Furthermore, the gain we achieve in downloading the content locally also represent the reduction we get in the overall cache misses from the system.

Fig. 4 presents the cache hit ratio for different values of zipf parameter $\alpha$. As $\alpha$ decreases, the cache hit ratio becomes smaller since the distribution of requests is less skewed than before. Moreover, we present the comparative analysis of the proposed framework with the previous study and find that recommendations are able to boost cache hit ratio for up to 37% when $\alpha$ is 0.8. When the distribution of content is less skewed, the system faces a higher number of requests for the contents that are not locally stored and hence system suggests user with alternate contents. That is the working region where the beneficial impact of recommendation system is more visible.

Fig. 5 show the average download time acquired assuming a cloud connection with limited capacity, specifically 10 Mbps. While losses are decreased with the limited availability of the cloud link, the delay increases. The figure reveals the reduction in average downloading time while plugging in the recom-

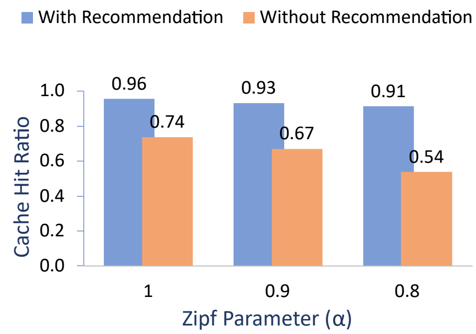


Fig. 4: Comparison of Cache Hit ratio with varying zipf distribution parameter $\alpha$ for system with and without recommendations.

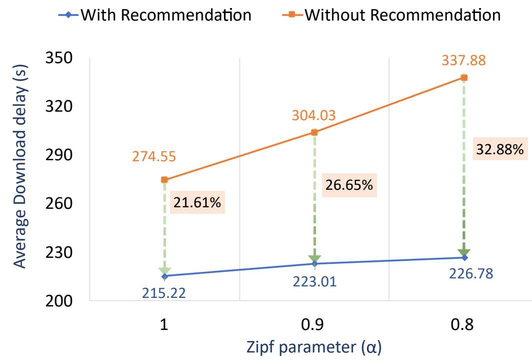


Fig. 5: Comparison analysis of average download delay associated with proposed system and system without recommendations under varying zipf distribution parameter $\alpha$.

mendation system in low bandwidth PLC network. In system lacking a recommendation system, the delay experienced by the users tends to increase linearly as the distribution becomes less skewed through a decrease in the parameter $\alpha$. In contrast, the presence of a recommendation system can potentially lower the delay by suggesting related content that is readily available locally, thus reducing the need for downloading from cloud resources.

#### *2) Varying Cache Size*

Fig. 6 compares the average downloading cost of proposed system with the baseline system with varying cache size. In the baseline case, the result shows that the cost is majorly coming from downloading the content from the precious cloud resources. Comparatively, in the proposed system, downloading cost from the cloud link is prominently reduced as more number of requests are served through local collaborative caching. Recommendations have positively effected by reducing the load on the backhaul link and increasing the efficiency of the PLC network through the reduction in downloading delays. The figure reveals an average of 52% reduction in the downloading cost of content from the cloud link with the proposed system. The findings also indicate that enlarging the cache size amplifies its influence on the average download cost from ES and EDs due to the increased availability of locally stored content, leading to reduced strain on the cloud link and a consequent decrease in the average download cost associated with the cloud link.

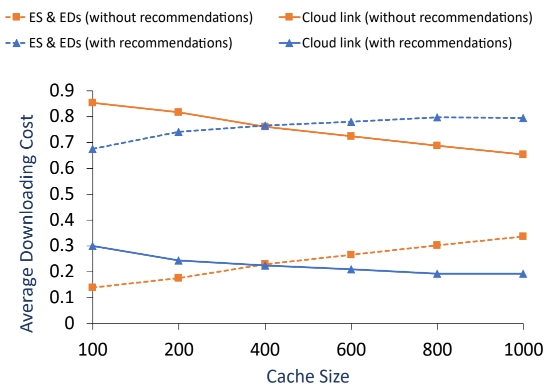


Fig. 6: Performance comparison with varying cache size; Average Downloading Cost for serving a request through collaborative caching and partially available cloud link

*3) Varying Recommendation accepting parameter (a)*

Fig. 7 presents the impact of user probability of choosing a content from the recommendations. On average, $a = 0.8$ on Netflix, but it may change on longer periods upon the behaviour of the user. We study the impact of variation in $a$ on cache hit ratio. We also consider different values of skewness parameter to find any correlation. The figure shows that a decrease in the parameter $a$ leads to a reduction in the cache hit ratio by 3%. Notably, this decrease is predominantly influenced by the variation in the parameter $\alpha$. Conversely, an increase in $a$ from its default value yields a 5% enhancement in the cache hit ratio.

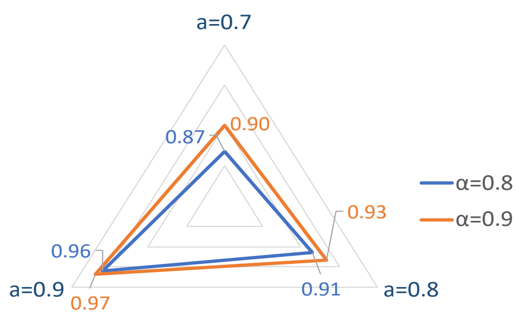


Fig. 7: Impact of changing recommendation accepting parameter $a$ on cache hit ratio for $(\alpha = 0.8, 0.9)$

## VII. Conclusion

Power Line Communication (PLC) has emerged as an efficient solution utilizing existing power infrastructure to provide connectivity for remote areas. However, PLC's limited capacity can impede the provisioning of video services requiring the download of large data files. Edge caching can alleviate this issue by pushing content to the network edge or user premises, thereby reducing delays in retrieving content and relieving the burden on PLC backhaul links. This paper investigates the improvement in performance of collaborative caching over the PLC network by incorporating a recommendation system. The results evidently reveal the effectiveness of the proposed approach by decreasing the downloading delays and reducing the burden on the backhaul link thereby making the PLC network more efficient.

## References


[1] L. Ye and H. Yang, "From digital divide to social inclusion: A tale of mobile platform empowerment in rural areas," *Sustainability*, vol. 12, no. 6, 2020.

[2] H. Oinas-Kukkonen, P. Karppinen, and M. Kekkonen, "5g and 6g broadband cellular network technologies as enablers of new avenues for behavioral influence with examples from reduced rural-urban digital divide," *Urban Science*, vol. 5, no. 3, 2021.

[3] C. Suraci, S. Pizzi, F. Montori, M. Di Felice, and G. Araniti, "6g to take the digital divide by storm: Key technologies and trends to bridge the gap," *Future Internet*, vol. 14, no. 6, 2022.

[4] C. Zhang, S. Dang, M.-S. Alouini, and B. Shihada, "Big communications: Connect the unconnected," *Frontiers in Communications and Networks*, vol. 3, 2022.

[5] D. Fink and R. J. Jeung, "Feasible connectivity solutions of plc for rural and remote areas," in *2008 IEEE International Symposium on Power Line Communications and Its Applications*, pp. 158–163, 2008.

[6] "Cisco annual internet report (2018–2023) white paper." Available at https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (2023/03/25).

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] H. Li, K. Ota, and M. Dong, "Eccn: Orchestration of edge-centric computing and content-centric networking in the 5g radio access network," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 88–93, 2018.

[9] M. A. Naeem, R. Ali, M. Alazab, Y. Meng, and Y. B. Zikria, "Enabling the content dissemination through caching in the state-of-the-art sustainable information and communication technologies," *Sustainable Cities and Society*, vol. 61, p. 102291, 2020.

[10] Z. Umar and M. Meo, "A collaborative caching over plc for remote areas," in *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 329–334, 2022.

[11] R. Zhou, S. Khemmarat, and L. Gao, "The impact of youtube recommendation system on video views," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, (New York, NY, USA), p. 404–410, Association for Computing Machinery, 2010.

[12] M. Sheikh-Hosseini, G. A. Hodtani, and M. Molavi–Kakhki, "Capacity analysis of power line communication point-to-point and relay channels," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 2, pp. 200–215, 2016.

[13] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.

[14] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1300–1313, 2018.

[15] T. Giannakas, P. Sermpezis, and T. Spyropoulos, "Show me the cache: Optimizing cache-friendly recommendations for sequential content access," 2018.

[16] Y. Qian, S. Li, L. Shi, J. Li, F. Shu, D. N. K. Jayakody, and J. Yuan, "Cache-enabled mimo power line communications with precoding design in smart grid," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 1, pp. 315–325, 2020.

[17] Y. Qian, L. Shi, J. Li, X. Zhou, F. Shu, and J. Wang, "An edge-computing paradigm for internet of things over power line communication networks," *IEEE Network*, vol. 34, no. 2, pp. 262–269, 2020.

[18] P. Lops, M. de Gemmis, and G. Semeraro, *Content-based Recommender Systems: State of the Art and Trends*, pp. 73–105. Boston, MA: Springer US, 2011.

[19] "Zipf distribution." Available at https://www.sciencedirect.com/topics/computer-science/zipf-distribution (2022/05/19).

[20] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, dec 2016.

[21] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, dec 2015.