Cache Placement Optimization for Layered Video Content

Estefanía Recayte
Institute of Communications and Navigation of DLR (German Aerospace Center),
Wessling, Germany. Email: {estefania.recayte}@dlr.de

***Abstract*—In this work, we investigate cache placement strategies for layered video content. We consider a library of video files that can be requested in different quality levels, according to a specific distribution. The study involves formulating and solving two distinct optimization problems to determine the most effective approach to cache placement. Our aim is to compare the following strategies: placement strategy which reduces congestion on the backhaul link by minimizing the number of transmissions necessary to meet user requests, and placement strategy which maximizes the probability of users being fully served from cached content. As shown in the solutions, these two performance metrics lead to different solutions for content that needs to be cached.**

I. Introduction

The increasing demand for multimedia video content with varying quality definitions, low latency constraints, and the need to conserve valuable backhaul resources has presented new design challenges to network operators. Edge caching is a key technology solution to relieve network core congestion, and is gaining momentum in both the research and industry communities. This technique involves caching intended content at the edge of the network [1]–[3], which significantly reduces backhaul traffic, latency, and power consumption. Two-step caching strategy is employed to achieve this goal, involving pre-fetching content at the edge (e.g. at small base stations, relays or helpers) during network off-peak periods (*placement phase*) to serve users without consuming backhaul capacity during network congestion (*delivery phase*).

Extensive research has been focused to developing methods for optimizing video content caching based on specific network metrics. Users often specify their desired video quality, such as a certain resolution for streaming services like Netflix or YouTube [4]. To address this demand, scalable video coding (SVC) is an encoding technique that offers multiple spatial resolutions, frame rates, and signal-to-noise ratios [5].

SVC divides a video into a sequence of ordered layers that define varying qualities, starting with a base layer (layer one) and followed by a number of enhancement layers. A user who requests the basic layer should only receive layer one, while those who request higher quality should receive all previously encoded lower levels. However, due to the limited capacity of caches and the vast amount of available content, it's necessary to develop suitable placement schemes to efficiently select which SVC-encoded video content should be cached.

E. Recayte acknowledges the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-RIC, project identification number: 16KISK022.

Several studies have investigated the optimization of video content placement for non-uniform demands in order to improve cache hit probability and network throughput. For instance, [6]–[8] examined the placement optimization of video content. In [9], the distribution of layered video over the Internet via caching was investigated with the aim of maximizing the revenue of a streaming service. Another optimization placement problem was studied in [10], which focused on minimizing the average user download time in a layered video streaming service over a heterogeneous wireless network. Layered video caching schemes were also explored in [11], where a popularity priority scheme was compared to a random caching scheme. The results of both schemes showed improvement in backhaul offloading and reduction of video transmission delay. Finally, [4] investigated layered video caching for distributed networks, proposing caching policies to optimize the average user delay while allowing cooperation between different operators.

While various solutions to basic caching problems for layered video content have been proposed in previous studies [4], [6]–[11], there is still a gap in the literature regarding the comparison and evaluation of cache policies aimed at minimizing backhaul transmissions versus those focused on maximizing cache hits.

This study aims to explore various optimization metric targets and examine their placement solutions in a cache-aided network with layered video content. Specifically, we formulate two optimization problems for SVC videos. The first problem aims to reduce the average transmissions over the backhaul link (i.e., transmissions from the server), while the second problem focuses on maximizing the probability that users are fully served by the content already present in the cache. The network under consideration is a two-tier system comprising a server, a cache-aided transmitter, and multiple users. Although the network appears simple, we demonstrate that the placement problem's solution is not straightforward. The main question we seek to answer is whether the cache should be filled with all layers of the most popular videos or the lower layers of all videos. We address this question by formulating two different optimization problems and presenting algorithms to solve them. In the results section, we

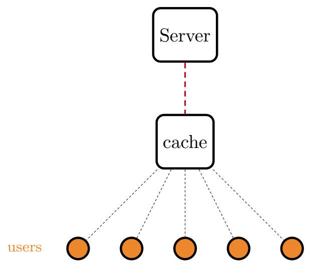

Fig. 1. System model: cache-aided transmitter is connected to the server through the backhaul link (dashed red) and multiple users are connected to the cache.

examine the behavior of each placement scheme and compare their performances. Additionally, we analyze the sensitivity of the proposed optimization problems to the distribution of video requests. Specifically, we investigate the impact of popularity skewness on the performance of each proposed solution.

II. System Model

We consider a two-tier network composed by a server and a cache-aided transmitter that provides video services to a number of users, as depicted in Fig 1. The link between the server and the cache-aided transmitter is referred to backhaul link, which is illustrated with a dashed red line in the figure. We assume that users has no direct connection to the server.

The server possesses a library of N videos denoted by $\mathcal{V}$. Each video v has up to Q different quality levels, with the l-th level of the i-th video represented as $v(i,l)$. The base layer with $l = 1$ contains essential information, while the higher layers correspond to increased video quality. The cache-aided transmitter has a finite capacity and can store up to $\mathsf{M} > 0$ base levels. The set of layers is denoted as $\mathcal{L} = \{1, 2, ..., Q\}$, while the video library containing all qualities of each video can be expressed as $\mathcal{V} = \{v(1,1), ..., v(1,Q), ..., v(N,1), ..., v(N,Q)\}$ with a cardinality of $|\mathcal{V}| = N \cdot Q$. The successful reception of video i of quality l requires the reception of all layers up to the l^{th}, i.e. content $v(i,1)$ to $v(i,l)$. We assume that all videos of the same quality level have the same size, denoted by $s(i,l)$[1], where the size of the basic level is normalized to one, i.e. $s(i,1) = 1\,\forall i$. The size of higher quality levels is a fraction of the base level and typically decreases with l, i.e. $s(i,1) \geq s(i,2) \geq ... \geq s(i,Q)$ [5]. When the l-th quality level is requested, the network must deliver a certain amount of content, which is equal to the sum of the sizes of all the layers up to and including the l-th layer, i.e. $\sum_{j=1}^{l} s(i,j)$.

The probability of the $i-th$ video being requested, independently from its quality, γ_i follows the Zipf distribution with parameter α leading to

$$\gamma_i = \frac{1/i^\alpha}{\sum_{j=1}^{N} 1/j^\alpha} \qquad i = 1, ..., N. \tag{1}$$

[1]Bytes are the standard unit of measure, but they can be abstracted within the context of our problem.

The probability of requesting the quality l of the i-th video $v(i,l)$ is denoted by $\beta(i,l)$ and for a given video i, implies that $\sum_{l=1}^{Q} = \beta(i,l) = 1$.

The caching strategy consists of two phases. Firstly, during the *placement phase*, the cache is filled with content based on the selected optimization strategy, and this is done offline. Secondly, during the *delivery phase*, user demands are served by the network. If the requested content is present in the cache, users are directly served from the transmitter. However, if the content is not available in the cache, it is sent from the server to the cache, which acts as relay to the users, using the backhaul link.

To formulate our optimization problems, we require a decision variable. We use the binary variable $x(i,l)$, which is defined as follows:

$$x(i,l) := \begin{cases} 1, & \text{if the } i\text{-th video of quality } l \text{ is placed in cache} \\ 0, & \text{otherwise.} \end{cases}$$

We represent the cache placement decisions using the matrix $\mathbf{X}$, where the value of $x(i,l)$ indicates whether the video $v(i,l)$ is present in the cache or not.

In this context, we study the placement optimization with the two different strategies. The first strategy focuses on minimizing the average amount of content to be transmitted via the backhaul link, and is referred as average transmission rate (ATR) scheme. The second strategy prioritizes maximizing the probability that users can be served entirely from cached content, referred as layered cache hit ratio (LCP) scheme. Let us illustrate the difference between the two strategies through the following example.

Example 1. *Consider a unique video with three different quality versions, $\mathcal{V} = \{v(1,1), v(1,2), v(1,3)\}$, with request probabilities given by*

$$\beta(1,1) = 0.3, \quad \beta(1,2) = 0.6, \quad \beta(1,3) = 0.1.$$

Assume that the size of each version is

$$s(1,1) = 1, \quad s(1,2) = 0.75, \quad s(1,3) = 0.25$$

while the memory size is $\mathsf{M} = 1$. Let us now analyze each placement strategy:
(i) ATR*: minimize the average transmission rate, the optimal solution is to place the content $v(1,2)$ and $v(1,3)$ in cache. However, none of the user requests is fully served by the cache, and hence the hit probability is*

$$\mathsf{P}_{\mathsf{hit}} = 0.$$

The average transmission rate $\mathsf{R}_{\mathsf{ATR}}$ can be evaluated as follows

$$\mathsf{R}_{\mathsf{ATR}} = s(1,1) \cdot \beta(1,1) = 0.3.$$

(ii) LCP*: the optimal solution to maximize the layered hit probability is to place the content $v(1,1)$ in cache. The probability that a user is fully served by the cache corresponds to the probability that $v(1,1)$ is requested and that is*

$$\mathsf{P}_{\mathsf{hit}} = 0.30.$$

Instead in this setup the average transmission rate $\mathsf{R}_{\mathsf{LCP}}$ *from the master node is*

$$\mathsf{P}_{\mathsf{hit}} = 0.30.$$

However, this strategy may lead to a higher average transmission rate $\mathsf{R}_{\mathsf{LCP}}$ *from the master node compared to the* ATR *scheme, indeed*

$$\begin{aligned}\mathsf{R}_{\mathsf{LCP}} &= s(1,2)\cdot\beta(1,2) + s(1,3)\cdot\beta(1,3)\\ &= 0.75\cdot 0.6 + 0.25\cdot 0.1 = 0.4735\end{aligned}$$

In practical scenarios, implementing the ATR scheme can potentially reduce congestion and improve network efficiency. On the other hand, the LCP scheme can enhance the user experience, especially when the backhaul link is not accessible.

This example illustrates how the choice of caching optimization strategy can significantly impact performance outcomes.

III. Average Transmission Rate Optimization

We define the average transmission rate R as the average amount of traffic that the server needs to transmit to fulfill users requests. If a requested content is not present in the cache, then the server transmits it through the backhaul link. We note that $x(i,l) = 1$ when the i-th video of quality l is in the cache, and $x(i,l) = 0$ otherwise. Therefore, the amount of traffic that the master has to send can be expressed as

$$[1 - x(i,l)]\, s(i,l).$$

If we consider the probability of each content being requested, we can calculate the expected amount of transmission that the server needs to support. Thus, we can express the average transmission rate as the sum of the product of the probability of a video of a particular quality being requested and the size of the content that needs to be transmitted if it is not present in the cache

$$\mathsf{R} = \sum_{i=1}^{N}\sum_{l=1}^{Q} \gamma_i\, \beta(i,l)\, [1 - x(i,l)] s(i,l). \tag{2}$$

In this scheme, the objective is to minimize the transmission from the server, which is represented by the objective function (2). The optimization problem for minimizing the average transmission rate over the backhaul link can be formulated as follows

$$\min \sum_{i=1}^{N}\sum_{l=1}^{Q} \gamma_i\, \beta(i,l)\, [1 - x(i,l)]\, s(i,l) \tag{3}$$

$$\sum_{i=1}^{N}\sum_{l=1}^{Q} s(i,l)\, x(i,l) \ \le \mathsf{M} \tag{4}$$

$$x(i,l) \in \{0,1\} \quad \forall i\, \forall l \tag{5}$$

where constraint (4) ensures that the total size of the cached content does not exceed the cache size M. Without loss of generality, in (4) is assumed that $\sum_{i=1}^{N}\sum_{l=1}^{Q} s(i,l) > \mathsf{M}$, otherwise, the problem is trivial. Constraint (5) accounts for the binary nature of the decision variable.

The problem described in (3)-(5) can be can be reformulated as the Knapsack problem (KP) [12]. The KP involves a set of items characterized by a profit p_i, a weight w_i, and a bin of capacity b. The goal is to select a subset of items that fit into the bin and maximize the overall profit. In our caching scenario, each item corresponds to a video quality level, the profit represents the probability of being requested, and the weight represents the video size. To transform our objective function into a maximization function like in the Knapsack problem, we can write it as follows

$$\begin{aligned}&\min \sum_{i=1}^{N}\sum_{l=1}^{Q} \gamma_i\, \beta(i,l)\, [1 - x(i,l)]\, s(i,l)\\ &= \max \sum_{i=1}^{N}\sum_{l=1}^{Q} \gamma_i\, \beta(i,l)\, x(i,l)\, s(i,l)\end{aligned}$$

The optimization problem presented in (3)-(5) can be solved using the algorithm proposed by Dantzig [12] as a simple solution to the Knapsack problem. The algorithm involves sorting the items in decreasing order according to their profit (probability) and gradually filling the available space until it is full. Algorithm 1 shows the version of the algorithm adopted for our caching scenario.

Algorithm 1: Average transmission rate (ATR) optimization

```
Input: $\gamma_i, \beta(i,l), s(i,l), M$
Output: X
1  for each $v(i,l)$ associate the following values:
2  $p_k = \gamma_i \cdot \beta(i,l)$  $n_k = i$ ,  $t_k = l$
3  sort items according to: $p_1 \ge p_2 \ge ... \ge p_{NQ}$
4  $\bar{\mathsf{M}} = \mathsf{M}$    residual cache capacity
5  $x(i,l) = 0, \quad i = 1,...,N, l = 1,...,Q$
6  for $i = 1,...,NQ$ do
7      if $s(n_i,t_i) \le \bar{\mathsf{M}}$ then
8          $x(i,l) = 1$
9          $\bar{\mathsf{M}} = \bar{\mathsf{M}} - s(n_i,t_i)$
10     else
11         $x(i,l) = \frac{\bar{\mathsf{M}}}{s(n_i,t_i)}$
12         return X
13     end
14 end
```

IV. Layered cache Hit Probability Optimization

We define the layered cache hit probability as the probability that a request for a video of quality l is fully served by cached content, denoted as $\mathsf{P}_{\mathsf{hit}}$. In a layered caching system, a layered cache hit occurs when a user receives from the cache all l layers related to the i-th content, i.e., $v(i,1), ..., v(i,l-1), v(i,l)$. The layered cache hit ratio can be calculated as follows

$$\mathsf{P}_{\mathsf{hit}} = \sum_{i=1}^{N}\sum_{l=1}^{Q} \gamma_i\, \beta(i,l)\, x(i,l). \tag{6}$$

Note that optimizing the layered cache hit schemes implies that if video $v(i,l)$ is present in cache, i.e $x(i,l) = 1$, then all lower qualities versions of the same video should be also present, i.e. $x(i,1) = 1, ..., x(i,l-1) = 1$. This scheme is justified by the fact that a service operator might choose this strategy, for example, when there is an expected significant increase in network traffic, and the backhaul link may not have the capacity to handle it. Additionally, this strategy can be suitable in cases where the backhaul link is unavailable, and the priority is to satisfy users with as much cached content as possible.

We aim to optimize the content placed in the cache in order to maximize the layered cache hit ratio. The objective function of our optimization problem is represented by Equation (6), which can be defined as follows

$$\max \sum_{i=1}^{N} \sum_{l=1}^{Q} \gamma_i \, \beta(i,l) \, x(i,l) \tag{7}$$

$$\sum_{i=1}^{N} \sum_{l=1}^{Q} s(i,l) \, x(i,l) \; \leq \; \mathsf{M} \tag{8}$$

$$x(i,l-1) \; \geq \; x(i,l) \quad \forall i \, \forall l \tag{9}$$

$$x(i,l) \; \in \; \{0,1\} \quad \forall i \, \forall l \tag{10}$$

Constraint (8) ensures that the total size of the content cached does not exceed the size of the cache M. It is assumed, without loss of generality, that the sum of the sizes of all cached content is greater than M, otherwise the problem is trivial. Constraint (9) ensures that if the l-th level of a video content is cached, then all previous levels of that content must also be cached. Lastly, constraint (10) accounts for the binary nature of the decision variable.

We present an algorithm to solve the optimization problem defined by (7)-(10). The first step is similar to the previous algorithm, where each video i of quality l is sorted in descending order according to the probability of being requested. The video with the highest probability that is not yet in cache is referred to as $v(i,l)$, and it is examined at each step of the algorithm while taking into account the remaining space in cache. The main point of the algorithm lies in two key checks before deciding to cache the video under examination. First, if the video has a quality level greater than one (i.e. $v(i, l > 1)$), all of the previous levels must also be cached. Second, if the video under consideration is of the base quality level (i.e. $v(i,1)$), the algorithm checks whether, for each video already placed in cache, the sum of probabilities of higher layers is greater than the probability of requesting $v(i,l)$. If the sum is greater, the upper levels of content already cached are also cached. Otherwise $v(i,1)$ is cached. Whenever a content is allocated to the cache, it is no longer taken under account in the algorithm. Algorithm 2 provides a pseudocode of how to implement the algorithm

TABLE I
SYSTEM MODEL PARAMETERS

l	$s(i,l)$	$\beta(i,l)$
1	1	0.25
2	0.7	0.10
3	0.15	0.15
4	0.15	0.50

V. Backhaul offloading index

To gain a more comprehensive understanding of the optimized caching schemes, we also consider the backhaul offloading index metric and investigate their performance when the distribution of requested videos changes.

The backhaul offloading index, denoted as O, quantifies the fraction of requested video traffic that can be transmitted directly from the cache without involving the backhaul link. We have that

$$\mathsf{O} = \frac{\sum_{i=1}^{N} \gamma_i \sum_{l=1}^{Q} \beta(i,l) \, s(i,l) \, x(i,l)}{\sum_{i=1}^{N} \gamma_i \sum_{l=1}^{Q} \beta(i,l) \, s(i,l)} \tag{3}$$

where the numerator represents the traffic offloaded from the backhaul link, while the denominator represents the total traffic requested, including both the content served by the cache and the content transmitted via the backhaul link. A higher value of the backhaul offloading index indicates better performance of the caching placement scheme.

We show in the results how the behavior of the cache placement solutions when the distribution of requested videos deviates from the distribution used for optimizing the LCP or the ATR. Such deviations can arise due to inaccuracies in estimating the distribution or infrequent updates of the cache leading to a mismatch between the distribution from the last placement period and the actual distribution of user demands. Recalling the Zipf distribution given in (1) and its skew parameter α then the new skew parameter ω which can change resulting in a new skew parameter

$$\omega > \alpha \text{ or } \omega < \alpha.$$

A change in α can cause requests to be concentrated in fewer popular files or, conversely, to be distributed more uniformly across more files.

VI. Numerical Results

We evaluate the average transmission load, the layered cache hit probability and the backhaul offload for both LCP and ATR schemes. In our scenarios, unless otherwise specified, it is assumed that each file is requested according the Zipf distribution with skew parameter $\alpha = 0.8$. We consider to have $N = 100$ different video content each with $Q = 4$ different quality level. For simplicity, we assume the l-th quality level, independently from the video, has the same size and the same probability to be requested. In table I are reported the values assumed for each quality level.

In Fig. 2 the average transmission rate R over the backhaul link as a function of the memory size M for different values of the skew parameter α of the Zipf is plotted. Blue curves

Algorithm 2: Layered cache hit prob optimization

```
Input: $\gamma_i, \beta(i,l), s(i,l), \mathsf{M}$
Output: X
for each $v(i,l)$ associate the following values:
$p_k = \gamma_i \cdot \beta(i,l);\ \ n_k = i\ ;\ \ t_k = l$
sort items according to: $p_1 \geq p_2 \geq ... \geq p_{NQ}$
$L \leftarrow n_1, n_2, ..., n_{NQ}$   add the index of sorted to a list
$\bar{\mathsf{M}} = \mathsf{M}$   residual cache capacity
$x(i,l) = 0;\quad i = 1, ..., N;\quad l = 1, ..., Q$
create a empty check list: $C$
while $\bar{\mathsf{M}} > 0$ do
    $n_k \leftarrow L.index(1)$
    if $s(n_k, t_k) \geq \bar{\mathsf{M}}$ then
        if $t_k > 1$ then
            for $q = 1, ..., t_k$ do
                if $s(n_k, q) \leq \bar{\mathsf{M}}$ then
                    $x(n_k, q) = 1$
                    $\bar{\mathsf{M}} = \bar{\mathsf{M}} - s(n_k, q)$
                else
                    $x(n_k, t_k) = \bar{\mathsf{M}}/s(n_k, t_k)$
                end
            end
            if $t_k < Q$ then
                $C.add(n_k)$
            end
        else
            for $\forall c \in C$ do
                $p_{min} = 1$
                $n_k \leftarrow C.index(1)$
                $\mathsf{M}_{check} = \bar{\mathsf{M}}$   $x_{check} = 0$   $P_{check} = 0$
                for $i = t_k + 1, ..., Q$ do
                    if $s(n_k, i) \leq M_{check}$ then
                        $P_{check} = P_{check} + p_i$
                        $\mathsf{M}_{check} = \mathsf{M}_{check} - s(n_k, i)$
                        $x_{check}(n_k, i) = 1$
                    end
                    if $P_{check} \leq p_{min}$ then
                        $p_{min} = P_{check}$,
                        $x_{min} = x_{check}$
                        $\mathsf{M}_{min} = \mathsf{M}_{check}$
                    end
                end
                if $p_{min} \leq p_k$ then
                    $x = x + x_{min}$,   $\mathsf{M} = \mathsf{M}_{min}$
                    delete from $C$ and $L$ cached content
                else
                    $x(n_k, t_k) = 1$,   $\mathsf{M} = \mathsf{M} - s(n_k, t_k)$
                    $C.add(n_k)$
                end
            end
        end
    else
        $x(n_k, t_k) = \bar{\mathsf{M}}/s(n_k, t_k)$
    end
end
```

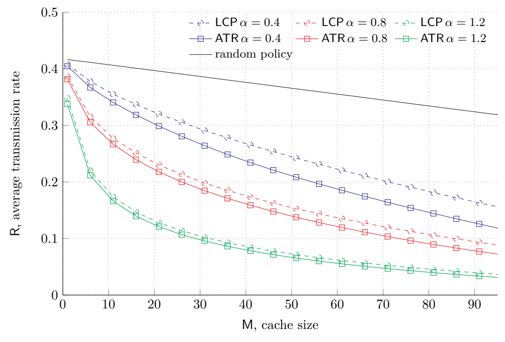

Fig. 2. Average transmission rate R over the backhaul link in function of the cache memory size M for LCP and ATR algorithm, and different values of skew distribution parameter α.

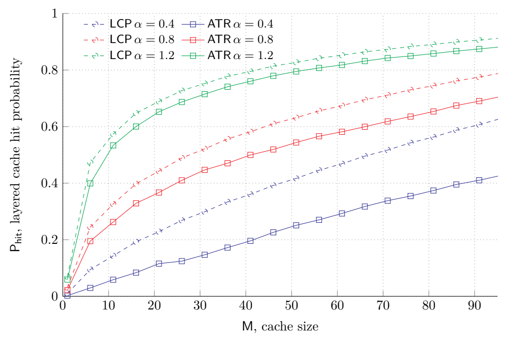

Fig. 3. Layered cache hit probability P_{hit} in function of the cache memory size M for LCP and ATR algorithm, and different values of skew distribution parameter α.

represent results for $\alpha = 0.4$, red for $\alpha = 0.8$ while green for $\alpha = 1.2$. The black curve represents a random policy which fills the cache without any optimization. Squared markers represent the caching scheme for optimizing R, i.e. ATR, while circle markers for optimizing P_{hit}, LCP. As expected, the average transmission from the server R decreases when the dimension of the transmitter cache M increases. We observe the effectiveness of the ATR scheme since, given a value of M and for each value of α, the ATR scheme requires less transmission from the backhaul than the LCP scheme. For fixed value of M and increasing values of α, the average transmission rate decreases. The reason is that the majority of request are concentrated into a small number of files. Note that, for the same reason when the skew parameter increases we have that the LCP tends to the results of the ATR scheme.

In Fig. 3 the layered cache hit probability as a function of the memory size M for different values of the skew parameter α of the Zipf distribution is plotted. Also in this case, we can observe that there is a gain in terms of hit probability by placing the content according the LCP instead of the ATR scheme. For a given cache size M, layered hit probability is higher when the values of α are higher. The trend of performance between the LCP and ATR when the skew parameter increases is also observed in this plot. Both

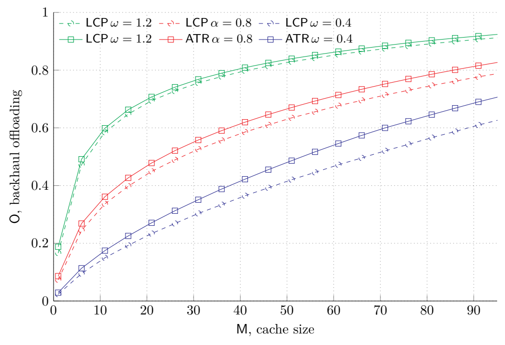

Fig. 4. Offloading traffic O over the backhaul link in function of the cache memory size M for LCP and ATR algorithm computed for $\alpha = 0.8$ and evaluated for a change of the parameter distribution ω.

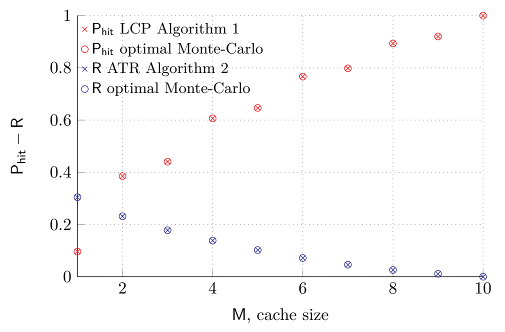

Fig. 5. P_{hit} and R in function of the cache memory size M for Monte-Carlo simulations and proposed algorithms.

Fig. 2 and Fig. 3 show the importance of an appropriate caching placement algorithm to obtain better performance in the network. Specially when the parameter α is low, the gain in terms of average transmission in the case of ATR and in layered cache probability in the case of LCP are notable.

In Fig. 4 the backhaul offloading traffic as a function of the memory size M is plotted. In all curves presented in this plot, we assume that the placement given by the LCP and the ATR schemes are fixed and were derived for a scenario of Zipf distribution $\alpha = 0.8$. Red curve represent the results obtained by the LCP and ATR schemes with $\alpha = 0.8$. Blue curves represent the performance when a change of distribution occurs, i.e. $\omega = 0.4$ while green curves represent a distribution with $\omega = 1.2$. We first observe that the ATR placement solution achieves higher values of the backhaul offload with respect to the LCP for each scenario depicted. Interestingly, the plot shows that if the Zipf distribution changes from $\alpha = 0.8$ to $\omega = 1.2$ then better results are obtained and the network amount of offloaded traffic in the backhaul is higher. This is always because of the effect of a smaller set of files are requested with higher probability. The opposite effect occurs for a change to $\omega = 0.4$ and in this case the amount of traffic in the backhaul decreases with respect to $\alpha = 0.8$.

Finally, we show the validity of the optimized results given by LCP and ATR in our last plot of Fig. 5. We also assumed in this case the system parameters given in Table I. We reduced the a library video content to $N = 10$ due to the expensive computation time that Monte-Carlo needs. The red markers indicates the result obtained for optimizing the layered cache hit probability while blue markers indicates the result obtained for optimizing the average transmission rate. Circle markers indicate results obtained with our algorithm proposed while crosses the Monte-Carlo simulations. In both cases, we can see the perfect match between the solution of the algorithm proposed and the simulation results.

VII. Conclusions

We have presented two caching optimization problems for layered video content. The first problem aims to minimize the average transmission through the backhaul link, while the second problem seeks to maximize the probability of completely serving users with cached content.

To solve them, we proposed efficient algorithms, whose effectiveness was cross-checked through Monte-Carlo simulations. Results showed that different caching placement strategies can lead to diverse outcomes concerning the average transmission, layered hit probability, and offloaded backhaul traffic. These findings offer valuable insights into achieving a balanced allocation of memory and backhaul resources when designing a cache-enabled network.

References

[1] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5g wireless networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[2] E. Recayte, “Coded caching at the edge of satellite networks,” in *ICC 2022 - IEEE International Conference on Communications*, 2022.

[3] ——, “Edge caching: On the perfomance of placement and delivery coding schemes,” *IEEE Transactions on Communications*, vol. 71, no. 6, pp. 3221–3232, 2023.

[4] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, “Caching and operator cooperation policies for layered video content delivery,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[5] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the h.264/avc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[6] U. Niesen, D. Shah, and G. W. Wornell, “Caching in wireless networks,” *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6524–6540, 2012.

[7] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed caching helpers,” *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[8] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, “Cache in the air: exploiting content caching and delivery techniques for 5g systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[9] J. Kangasharju, F. Hartanto, M. Reisslein, and K. Ross, “Distributing layered encoded video through caches,” *IEEE Transactions on Computers*, vol. 51, no. 6, pp. 622–636, 2002.

[10] C. Zhan and Z. Wen, “Content cache placement for scalable video in heterogeneous wireless network,” *IEEE Communications Letters*, vol. 21, no. 12, pp. 2714–2717, 2017.

[11] L. Wu and W. Zhang, “Caching-based scalable video transmission over cellular networks,” *IEEE Communications Letters*, vol. 20, no. 6, pp. 1156–1159, 2016.

[12] T. Dantzig and J. Mazur, *Number: The Language of Science*, ser. A Plume book. Penguin Publishing Group, 2007. [Online]. Available: https://books.google.de/books?id=YKJPEAAAQBAJ