Joint Transmission and Caching Policy Design for Energy Minimization in the Wireless Backhaul Link

Maria Gregori, Jesús Gómez-Vilardebò, Javier Matamoros
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
{maria.gregori, jesus.gomez, javier.matamoros}@cttc.cat

Deniz Gündüz
Imperial College of London, UK
d.gunduz@imperial.ac.uk

***Abstract*—Caching the most popular contents at Small Base Stations (SBSs) is envisioned as a promising solution to reduce both the load and energy consumption of the backhaul link connecting the SBSs to the core network. This paper considers a set of users whose demands are served by an SBS connected through a wireless backhaul link to a Macro Base Station (MBS). The SBS is capable of caching content in its limited cache memory. The transmission policy at the MBS and the caching policy at the SBS are jointly optimized in order to minimize the energy consumption in the backhaul link. The numerical results show significant improvements with respect to prior works.**

***Index Terms*—Proactive caching, wireless backhaul, energy-efficiency.**

I. Introduction

The fifth generation (5G) cellular communication system is aimed to cope with the wireless traffic explosion providing fast, reliable, and sustainable wireless connectivity. Small cell densification, i.e., the deployment of a large number of SBSs with different cell sizes (micro, pico, and femtocells), is widely accepted as a potential solution to achieve these goals. The traffic that can be served by an SBS is limited by the capacity of the backhaul link providing connection to the core network. This link is preferably wireless for various reasons such as, rapid deployment, self-configuration, and cost. Consequently, the backhaul link has limited capacity and, due to its relatively long range, consumes a significant amount of energy.

Caching the most popular contents (e.g., popular Youtube videos) in SBSs has been proposed both to alleviate the backhaul link congestion and to reduce its energy consumption [1]. As the storage capacity at the SBSs is limited (albeit large), efficient caching policies must be designed to meet the system requirements by taking into account, among others, the stochastic but predictable nature of users' file demand.

Caching policies can be classified into two groups according to the prior knowledge of the different system parameters (users' file demand, channel state information, etc.): (i) *offline* caching policies that assume non-causal and complete knowledge of these parameters, e.g., [2], [3]; and (ii) *online* caching policies that consider causal knowledge only [1], [4], [5]. The optimal offline policy is extremely useful because i) it serves as a theoretical bound on any online policy; and ii) it can be instrumental in designing low-complexity near-optimal online policies.

This work is partially supported by the EC-funded project NEWCOM# (n.318306), by the Spanish Government through the projects INTENSYV (TEC2013-44591-P) and E-CROPs (PCIN-2013-027) in the framework of ERA-NET CHIST-ERA, and by the Catalan Government (2014 SGR 1567).

In a popular approach to wireless caching, [5], [6], the system design is performed in two separated phases. First, in the *content placement* phase, each cache is filled with appropriate data, exploiting periods with low traffic. Then, in the *delivery phase*, the non-cached contents are transmitted when requested by users. In this setup, two types of caching gains have been identified, namely, the *local* and *global caching gains* [6]. The *local caching gain* is obtained when a requested file is locally available at the SBS cache. This reduces the traffic in the wireless backhaul link [5] and improves the quality of experience [6]. The *global caching gain* is obtained by multicasting network-coded information in the delivery phase [6]. However, this underlying separation between the caching (content placement) and transmission (delivery) phases has two limiting assumptions: i) the energy consumption of the content placement phase is not accounted for; and ii) cache content is never updated during the delivery phase. As a result, the benefits of proactive caching are inherently limited.

In this work, we consider a different approach to wireless caching by combining the content placement and delivery phases. Pre-downloading data in periods with low traffic is still allowed, but we now account for its energy consumption. As a result, an additional caching gain is obtained, which we call *pre-downloading caching gain*. This gain is obtained when the cache is used to pre-download data, which can be beneficial both to avoid non-favourable channel conditions, and to equalize the rate in the backhaul link, improving its energy efficiency. In this context, the authors of [2] and [3] derive joint caching and transmission policies that minimize the bandwidth and energy consumption, respectively. These works assume that the cache is solely used to pre-download content for a single user; thus, content is removed from the cache as soon as it is consumed by the user, ignoring any possible future requests. Consequently, the policies in [2] and [3] only exploit the pre-downloading caching gain. To the best of our knowledge, this is the first work that proposes *jointly optimal* transmission and caching policies by accounting for the local and pre-downloading caching gains.

II. System model and problem formulation

We consider U users served by an SBS in a Time Division Multiple Access (TDMA) fashion. As depicted in Fig. 1, the SBS has a finite cache memory of capacity C units of data, and is connected through a wireless backhaul channel to an MBS, which has access to the core network.

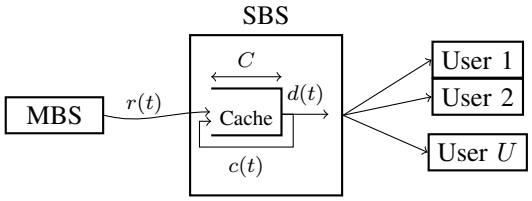

Fig. 1. System model.

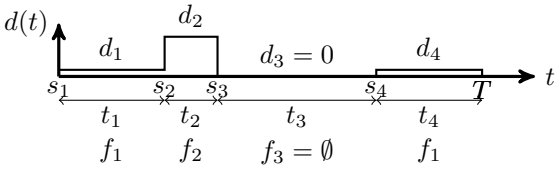

Fig. 2. The users request the files f_1, f_2 and f_1, while f_3 denotes a period of time without any requests. Since the file f_1 is requested in the first and fourth epochs, we have $d_1 = d_4$ and $t_1 = t_4$.

The instantaneous transmission power of the MBS, denoted by $p(t)$, is modeled by a generic power-rate function $p(t) = g(r(t))$, where $r(t)$ stands for the transmission rate of the MBS. We assume that the function $g(\cdot)$ is time invariant, strictly convex, increasing, continuously differentiable, and $g(0) = 0$. It is worth noting that these conditions are satisfied by common power-rate functions. For example, in the case of Gaussian signaling, we have $p(t) = \exp(r(t)) - 1$, which satisfies the above conditions.

We define T as the optimization time horizon, and $\mathbb{F} = \{f_1, \ldots, f_{|\mathbb{F}|}\}$ as the set of all possible file requests, where f_j denotes a specific file. During this time, the SBS must serve sequentially, in a TDMA fashion, a total of N files to the users. We assume a known user schedule. The n-th file request has a specific duration of t_n seconds; this period of time is denoted as the n-th *epoch*. Define s_n as the starting time of the n-th epoch, i.e., $s_n = \sum_{i=1}^{n-1} t_n$.

In the sequel, we define the variables needed to formulate the problem following the system model in Fig. 1.

Definition 1 (Instantaneous demand rate)**.** *The demand rate, $d(t) \geq 0$, $t \in [0, T]$, is the rate at which the SBS must serve data to the users to fulfill their demands.*

For simplicity and mathematical tractability, we consider that each file has associated a fixed duration and a constant demand rate, which depends on the rate at which data is consumed by upper layers in the protocol stack (e.g., a video file requires a higher rate than a news article). Accordingly, the demand rate can be written as $d(t) = \sum_{n=1}^{N} d_n \operatorname{rect}((t - (s_n + t_n/2))/t_n)$, where $\operatorname{rect}((t-a)/b)$ denotes the rectangular function centered at a with duration b. Thus, if a certain file is requested both in epochs n and n', these epochs must have $t_n = t_{n'}$ and $d_n = d_{n'}$. Fig. 2 shows an example of the instantaneous demand rate. Without loss of generality, we represent periods with no request as having a request for a file whose demand rate is equal to zero. As in [2], [3], we assume a known file demand profile (offline approach, see Section I).

As shown in Fig. 1, data is downloaded in the backhaul link at a rate $r(t)$ and stored at the SBS cache until it is served to the users at a rate $d(t)$ (which, without loss of generality, can be immediately). Finally, the served data is locally cached at the SBS at a rate $c(t)$, which is formally defined next.

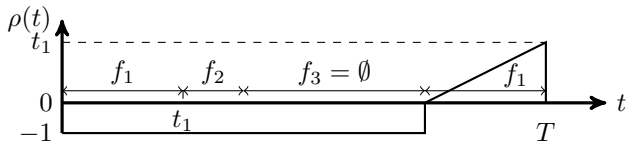

Fig. 3. Since the file f_1 is requested in the first and fourth epochs, the function $\rho(t)$ maps the instants in the fourth epoch to the corresponding time in the first epoch.

Definition 2 (Local caching rate)**.** *The local caching rate, $c(t)$, with $0 \leq c(t) \leq d(t)$, $t \in [0, T]$, is the rate at which user's consumed data is stored in the cache for future file requests.*

Note that we represent the locally cached data as a feedback link from the output to the input; however, in practice it is not necessary to remove the data from the cache to be stored again. Consequently, the SBS deletes the users' consumed data at rate $d(t) - c(t)$.

As argued in the introduction, the cache offers two different gains to save energy in the backhaul link, namely, *pre-downloading* and *local* caching gains. As shown in Fig. 1, the contents in the cache have two data sources: (i) pre-downloaded data, which is controlled by the transmission policy at the MBS, $r \triangleq r(t)|_{t=0}^{T}$, and contributes to the pre-downloading caching gain; and (ii) locally cached data, which is controlled by the local caching policy at the SBS, $c \triangleq c(t)|_{t=0}^{T}$, and contributes to the local caching gain. Note that given the tuple $(r, c, d(t)|_{t=0}^{T})$, the contents in the cache can be identified at any time t.

Definition 3 (Data departure curve)**.** *The data departure curve, $D(t, r)$, is the amount of data cumulatively served by the MBS by time $t \geq 0$ and can be obtained from the transmission policy as $D(t, r) = \int_0^t r(\tau) \mathrm{d}\tau$.*

Similarly, we describe the feasible region of the transmission policy by means of the following cumulative curves.

Definition 4 (Maximum data departure curve)**.** *The maximum data departure curve, $B(t, c)$, limits the maximum amount of data that can be cumulatively transmitted by time $t \geq 0$ such that no data overflow is caused at the cache memory. Thus, it is given by $B(t, c) = C + \int_0^t d(\tau) - c(\tau) \mathrm{d}\tau$ and depends on the caching policy c.*

The amount of data that must be downloaded from the MBS is given by the amount of data that is not locally available at the SBS. If the file had been previously served by the SBS, the net demand can be computed from the rate at which data was deleted during the previous request, $d(t) - c(\rho(t))$, where $\rho : [0, T] \to [0, T] \cup \{-1\}$ is a function that maps a certain time instant to the corresponding instant of the previous request of the same file, or takes the value -1 if the file has not been requested yet. We define $c(-1) \triangleq 0$. An example of $\rho(t)$ is depicted in Fig. 3 given the demand profile in Fig. 2. Since

non-causal knowledge of the user demands is assumed (offline approach), the function $\rho(t)$ is also known, $\forall t$. Now, we can introduce a lower bound on the data departure curve.

Definition 5 (Minimum data departure curve)**.** *The minimum data departure curve, $A(t,c)$, is the minimum amount of data that must be cumulatively transmitted by time $t \geq 0$ to satisfy the demand, and depends on the caching policy, c, i.e., $A(t,c) = \int_0^t d(\tau) - c(\rho(\tau))\mathrm{d}\tau$.*

Bearing all the above in mind, our aim is to *jointly* design the transmission policy at the MBS and the caching policy at the SBS to minimize the energy consumption at the MBS. In this work, we do not account for the energy consumption at the SBS because generally the distance between the users and the SBS is much shorter compared to the backhaul link (due to different cell radii); thus, the energy consumption at the MBS dominates the total consumption. Furthermore, if the SBS consumption were to be included in the optimization, then it would be required to estimate the channel between the users and the SBS, which is challenging due to user mobility. Thus, the problem is mathematically formulated as follows:

$$\min_{r,c} \quad \int_0^T g(r(\tau))\mathrm{d}\tau \tag{1a}$$

$$\text{s.t.} \quad D(t,r) \geq A(t,c), \qquad \forall t \in [0,T], \tag{1b}$$

$$D(t,r) \leq B(t,c), \qquad \forall t \in [0,T], \tag{1c}$$

$$r(t) \geq 0, \qquad \forall t \in [0,T], \tag{1d}$$

$$0 \leq c(t) \leq d(t), \qquad \forall t \in [0,T], \tag{1e}$$

where the constraint in (1b) imposes the fulfillment of the user demands, and (1c) prevents cache overflows. The constraints (1d) and (1e) restrict feasible transmission and caching rates. Note that a feasible caching policy, c, must satisfy $B(t,c) \geq A(t,c)\ \forall t \in [0,T]$, and any feasible data departure curve must lie within the tunnel between $B(t,c)$ and $A(t,c)$.

Remark 1. In the problem formulation, we have assumed that cached data can only be removed from the cache during the subsequent requests for the same data. As a result, by caching data, the demand from the MBS is reduced. Note that this assumption is without loss of optimality. Contrarily, if cached data were to be deleted prior to subsequent requests, the maximum data departure curve would be decreased while the minimum data departure curve remains the same, which would unnecessarily tighten the constraints.

This problem accepts a graphical representation once the local caching policy is fixed. For example, given the demand profile in Fig. 2 and a cache capacity $C = d_1 t_1$, the problem is represented in Fig. 4 for two different caching policies:

Policy 1: The policy c_1 in Fig. 4-a removes the data from the cache as soon as it is served, ignoring any possible future demands for the same file, i.e., $c_1(t) = 0$, $\forall t$. Thus, it exploits only the pre-downloading caching gain (this policy is proposed in [3]). Note that having $c_1(t) = 0$, $\forall t$, does not imply that the cache is always empty; indeed, it implies that there is a constant gap between the lower and upper bounds given by the cache capacity, C, which can be used by the optimal data

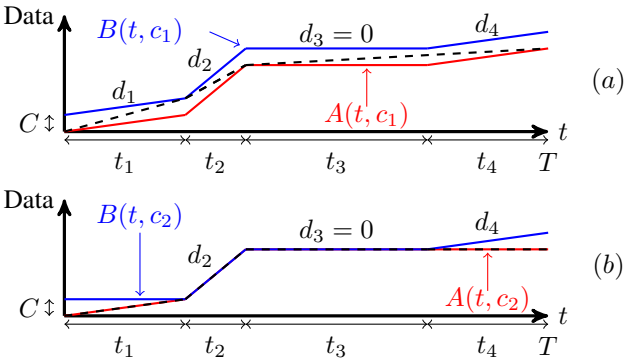

Fig. 4. Representation of the problem for two different caching policies.

departure curve to pre-download data.

Policy 2: The policy c_2 in Fig. 4-b caches the first file as it is later requested in the fourth epoch, i.e., $c_2(t) = d_1$ if $t \in [0, t_1]$ and $c_2(t) = 0$ otherwise. As a result, no data needs to be transmitted by the MBS in the fourth epoch. This policy exploits only the local caching gain.

Remark 2 (Constant rate transmission saves energy). Given $c(t)$ for all $t \in [0,T]$, the optimal data departure curve is given in [7] and can be visualized as the tightest string whose ends are tied to the origin and the point $(T, A(T,c))$, which is represented in Fig. 4 with the dashed lines. The conclusion is that, as far as the constraints allow, constant rate transmission saves energy due to the convexity of the power rate function.

Note that at any time instant, the free space in the cache can be obtained as $B(t,c) - D(t,r)$. Focusing on Policy 1 (see Fig. 4-a), the cache is full at $t = t_1$, and all the data in the cache belongs to f_2, which has been pre-downloaded to equalize the rates in the first and second epochs. As for Policy 2 (see Fig. 4-b), the cache is full at $t \in [t_1, t_1+t_2+t_3]$, and exclusively contains f_1. Note that the local caching policy adopted for file f_1 in the first epoch (upper bound) determines the demand (lower bound) of file f_1 in the fourth epoch.

From the previous discussion, two questions arise: (i) "which of the two caching policies achieves the lowest energy consumption?"; and (ii) "is any of these policies the optimal one?". One might think that the caching policy c_2 consumes less energy since fewer data has to be transmitted; however, this is not always true as the caching policy c_1 might achieve a lower consumption by equalizing the rate across epochs. In practice, the jointly optimal transmission and local caching policies must be designed by solving (1), which is challenging as it is an infinite-dimensional optimization problem.

III. Jointly Optimal Strategy

To solve the infinite-dimensional problem in (1), we first derive some structural properties of the jointly optimal strategy. Then, thanks to these properties, we will formulate (1) as a finite-dimensional convex optimization problem. As shown next, the optimization variables of the resulting problem are the number of data units to be deleted by the local caching policy in each epoch, q_n, and the transmission rate in that epoch, r_n, $n = 1, \ldots, N$.

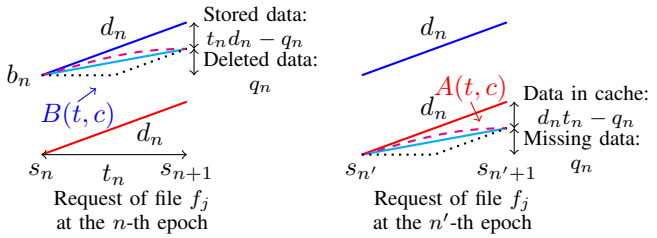

Fig. 5. Representation of $B(t,c)$ and $A(t,c)$ in Problem 1. The caching policy in the n-th epoch (left) determines the demand at epoch n' (right).

First, we study the optimal local caching rate within an epoch, in which a file of size $t_n d_n$ data units is served by assuming that q_n data units must be deleted from the cache ($0 \leq q_n \leq t_n d_n$), and the remaining $t_n d_n - q_n$ units will be locally cached for future demands. This problem is formally stated as follows:

Problem 1. *Consider that at the n-th epoch the file f_j is requested and the next request for this file appears at epoch $n' > n$. Assume that the optimal local caching policy, $c^\star$, is known $\forall t \in [0, s_n)$ and $t \in [s_{n+1}, T]$ (accordingly, we know $b_n \triangleq B(s_n, c^\star)$ and $A(s_{n'}, c^\star)$). Consider that the optimal policy stores $t_n d_n - q_n$ data units of file f_j at the n-th request to reduce the net demand at epoch n'. Which is the optimal local caching rate, $c^\star(t)$, in the interval $t \in [s_n, s_{n+1})$?*

This problem is represented in Fig. 5 for three possible solutions among infinite options: (i) the dotted strategy, which caches the $t_n d_n - q_n$ first data units of the file; (ii) the solid strategy, which caches data at a constant rate (for example, if the caching rate is $c(t) = 1/3$ one data unit out of every three is cached); and (iii) the dashed strategy, which has a continuous variation of the caching rate, storing at a lower rate at the beginning than at the end. Note that the local caching rate $c(t)$ for $t \in [s_n, s_{n+1})$ determines the shape of the maximum data departure curve at epoch n and the shape of the minimum data departure curve at epoch n', both being the same but properly shifted. Next lemma shows that the strategy (ii) is optimal.

Lemma 1. *Given q_n, the optimal local caching policy (not necessarily the unique one) in Problem 1 is to cache data at a constant rate, i.e., $c^\star(t) = (t_n d_n - q_n)/t_n$, $\forall t \in [s_n, s_{n+1})$.*

Proof: Let $b_n = B(s_n, c^\star)$, which is known since we know $c^\star(t)$, $\forall t < s_n$. From the problem statement, a valid caching policy, c, must satisfy $B(s_{n+1}, c) = b_n + q_n$. Thus, a feasible data departure curve must satisfy (among others): (i) $D(s_n, r) \leq b_n$; and (ii) $D(s_{n+1}, r) \leq b_n + q_n$.

To show that constant rate caching, $c^\star$, is optimal, we need to show that (a) $B(t, c^\star)$ does not constrain from above the optimal data departure curve for all $t \in (s_n, s_{n+1})$, and (b) $A(t, c^\star)$ does not constrain it from below for $t \in (s_{n'}, s_{n'+1})$. We prove (a) by contradiction. Assume that $B(t, c^\star)$ limits from above, at some $t_x \in (s_n, s_{n+1})$, the optimal data departure curve under the optimal caching policy (not necessarily $c^\star$), namely, $D(t, r^\dagger)$, i.e., $D(t_x, r^\dagger) > B(t_x, c^\star)$. Then, to satisfy (i) and (ii) above, there exist t_{x1}, t_{x2} such that $s_n \leq t_{x1} < t_x < t_{x2} \leq s_{n+1}$ with $D(t_{x1}, r^\dagger) = B(t_{x1}, c^\star)$ and $D(t_{x2}, r^\dagger) = B(t_{x2}, c^\star)$. Then, this implies that $D(t, r^\dagger)$ necessarily changes the rate for some $t \in (t_{x1}, t_{x2})$, which contradicts the optimality assumption of $D(t, r^\dagger)$ since we can construct a data departure curve that consumes less than $D(t, r^\dagger)$ by following $B(t, c^\star)$ in the interval $t \in [t_{x1}, t_{x2}]$. The proof of (b) follows similarly. ■

Corollary 1. *The optimal local caching rate (not necessarily the unique one) can be written as $c^\star(t) = \sum_{n=1}^{N} (t_n d_n - q_n^\star)/t_n \operatorname{rect}((t - (s_n + t_n/2))/t_n)$, where $q_n^\star$ denotes the optimal number of deleted data units at epoch n.*

Since $d(t)$ and $c(t)$ are step-wise functions whose value changes only at some epoch transition, we know that $A(t,c)$ and $B(t,c)$ are piece-wise linear functions whose slope changes only at some epoch transition. Consequently, we can obtain the following properties of the optimal transmission strategy, whose proof follows similarly to [8, Lemmas 5-6].

Lemma 2. *The optimal data departure curve, $D(t, r^\star)$, is a piece-wise linear function, whose slope can only change at time instants s_n, i.e., the optimal transmission rate of the MBS is $r^\star(t) = \sum_{n=1}^{N} r_n^\star \operatorname{rect}(t - (s_n + t_n/2)/t_n)$, where $r_n^\star$ denotes the transmission rate at the n-th epoch. Additionally, if the rate increases at time instant s_n ($r_n^\star < r_{n+1}^\star$), then the cache is full, $D(s_n, r^\star) = B(s_n, c^\star)$; and if if the rate decreases at time instant s_n ($r_n^\star > r_{n+1}^\star$), then the demand is met with equality, $D(s_n, r^\star) = A(s_n, c^\star)$.*

Thanks to Corollary 1 and Lemma 2, we can equivalently write the original problem in (1) as a function of the rates, r_n, and deleted data units, q_n:

$$\min_{\{q_n, r_n\}_{n=1}^N} \quad \sum_{n=1}^{N} t_n g(r_n) \tag{2}$$

$$\text{s. t.} \quad \sum_{i=1}^{n} t_i r_i \leq C + \sum_{i=1}^{n} q_i, \qquad n = 1, \ldots, N,$$

$$\sum_{i=1}^{n} t_i r_i \geq \sum_{i \in \mathbb{A}_n} t_i d_i + \sum_{i \in \mathbb{B}_n} q_{\bar{\rho}(i)}, \quad n = 1, \ldots, N,$$

$$r_n \geq 0, \quad 0 \leq q_n \leq t_n d_n, \qquad n = 1, \ldots, N,$$

where $\mathbb{A}_n \triangleq \{1, \ldots, n\} \setminus \mathbb{P}$, $\mathbb{B}_n \triangleq \{1, \ldots, n\} \cap \mathbb{P}$, and the set $\mathbb{P}$ contains the epoch indexes in which the associated files have been previously requested, and $\bar{\rho}(i)$, $i \in \mathbb{P}$, is a function that returns the epoch index of the previous request. For example, given the demand profile in Fig. 2, we have $\mathbb{P} = \{4\}$ and $\bar{\rho}(4) = 1$.

Remark 3. In the problem in (2), we have not included a constraint to force the number of deleted data units, q_n, to be integer. If we introduce an integer constraint, then (2) becomes an integer programming problem with its inherent complexity. In practice, as the data unit granularity (e.g., bit) is sufficiently small in comparison to the files sizes (of several mega bits), the integer constraint can be relaxed without affecting the performance. Consequently, (2) is a convex program (since

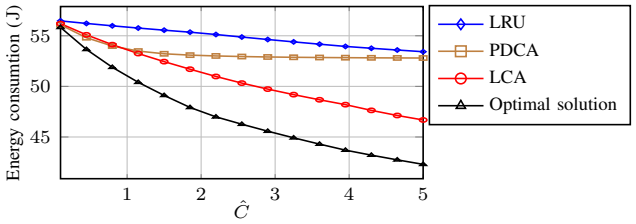

Fig. 6. Energy consumption at the MBS for different percentages, $\hat{C}$, of cache capacity over total transmitted data for $\gamma = 0.67$.

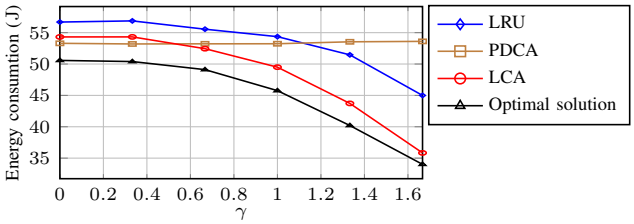

Fig. 7. Energy consumption at the MBS for different values of the Zipf parameter γ for $\hat{C} = 1.5$.

the objective function is convex and the constraints are linear), and can be readily solved.

IV. Results

In this section, we assess the performance of the proposed caching and transmission policies. We consider $N = 50$ epochs in which different files of size 1 Mnat are requested from a set of 100 possible files $|\mathbb{F}| = 100$. The file duration is distributed uniformly over $[1/3, 20]$ seconds, which implies demand rates in the interval $[0.05, 3]$ Mnat/s. We consider that the probability of requesting file f_j, θ_j, is independent and identically distributed across epochs and follows the Zipf distribution, i.e., $\theta_j = j^{-\gamma}/(\sum_{q=1}^{|\mathbb{F}|} q^{-\gamma})$, where γ models the skewness of the file popularity; when $\gamma = 0$, the popularity is uniform and the popularity becomes skewed when γ grows [4]. We consider the Shannon power-rate function $g(r(t)) = \exp(r(t)/W) - 1$, where $W = 1$ MHz is the bandwidth.

We compare the proposed jointly optimal transmission and caching solution, obtained by solving (2), with three sub-optimal solutions: the *Least Recently Used (LRU)* caching algorithm that always keeps in the cache the most recently requested files [9]; the *Pre-Downloading Caching Algorithm (PDCA)*, c_1, proposed in [3] that only uses the cache to pre-download data (see Policy 1 in Section II and Fig. 4-a); and the *Local Caching Algorithm (LCA)* that caches the most popular files the first time they are served and keeps these files in the cache for the whole transmission duration.

In this setup, Fig. 6 evaluates the energy consumption at the MBS for different sizes of the cache capacity, which is depicted as the percentage over the total requested data 50 Mnat, i.e., $\hat{C} = (100C)/(50 \text{ Mnat})$, for a Zipf parameter $\gamma = 0.67$. It is observed that, if only the pre-downloading caching gain is exploited (PDCA), the curve saturates and an increase in the cache capacity does not lead to a decrease in the total energy consumption. This saturation point occurs when the optimal data departure curve under caching policy c_1 does not touch the maximum data departure curve. On the contrary, LRU, LCA, and the optimal caching policy can further decrease the energy consumption. For small values of the cache capacity, PDCA outperforms LRU and LCA. The proposed optimal policy significantly outperforms all the other policies.

Fig. 7 evaluates the impact of the file popularity distribution over the energy consumption at the MBS. Thus, we vary the parameter γ in the x-axis and set the cache capacity to $\hat{C} = 1.5$. It is clearly observed that, while variations on the popularity does not have an impact on PDCA, they dramatically affect the consumption of the other policies, which exploit the local caching gain. Under uniform popularity distribution ($\gamma = 0$), PDCA outperforms LRU and LCA as file repetitions are unlikely. Finally, we observe that the energy consumption is dramatically reduced when the popularity distribution becomes skewed as more file repetitions are encountered.

V. Conclusions

This paper has investigated the problem of minimizing the energy consumption in the backhaul link that connects an MBS to an SBS, which has a cache memory. It has been argued that the cache offers two possible energy saving gains, namely, the *pre-downloading* and *local* caching gains. The jointly optimal transmission strategy at the MBS and caching policy at the SBS have been obtained by demonstrating that constant rate caching within an epoch is optimal, which allows a reformulation of the energy consumption minimization problem as a convex program. The *jointly* optimal solution correctly balances the tradeoff between *pre-downloading* and *local* caching gains, which has been verified by the conducted numerical simulations. To conclude, this offline policy will serve as a benchmark to evaluate online policies with only causal knowledge of the users' file demand, which is left as an open problem that will be addressed in our future work.

References

[1] E. Bastug, J.-L. Guenego, and M. Debbah, "Proactive small cell networks," in *Proceedings of the IEEE Int'l Conf on Telecom*, Casablanca, Morocco, May 2013, pp. 1–5.

[2] S. Sadr and S. Valentin, "Anticipatory buffer control and resource allocation for wireless video streaming," *arXiv: 1304.3056*, 2013.

[3] A. Güngör and D. Gündüz, "Proactive wireless caching at mobile user devices for energy efficiency," in *Proceedings of the IEEE Int'l Symp. on Wireless Comm. Systems (ISWCS)*, Brussels, Belgium, Aug. 2015.

[4] P. Blasco and D. Gündüz, "Multi-armed bandit optimization of cache content in wireless infostation networks," in *Proceedings of the IEEE Int'l Symp. on Inf. Theory*, Honolulu, HI, USA, Jun. 2014, pp. 51–55.

[5] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Multicast-aware caching for small cell networks," *arXiv: 1402.7314*, 2014.

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

[7] M. Zafer and E. Modiano, "A calculus approach to energy-efficient data transmission with quality-of-service constraints," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 898–911, Jun. 2009.

[8] M. Gregori and M. Payaró, "Energy-efficient transmission for wireless energy harvesting nodes," *IEEE Trans. Wireless Commun.*, vol. 12, no. 3, pp. 1244–1254, Mar. 2013.

[9] L. Rizzo and L. Vicisano, "Replacement policies for a proxy cache," *IEEE/ACM Trans. Netw.*, vol. 8, no. 2, pp. 158–170, Apr. 2000.