Wireless Content Caching for Small Cell and D2D Networks

Maria Gregori, Jesús Gómez-Vilardebó, Javier Matamoros and Deniz Gündüz

***Abstract*—The fifth generation wireless networks must provide fast and reliable connectivity while coping with the ongoing traffic growth. It is of paramount importance that the required resources, such as energy and bandwidth, do not scale with traffic. While the aggregate network traffic is growing at an unprecedented rate, users tend to request the same popular contents at different time instants. Therefore, caching the most popular contents at the network edge is a promising solution to reduce the traffic and the energy consumption over the backhaul links. In this paper, two scenarios are considered, where caching is performed either at a small base station, or directly at the user terminals, which communicate using Device-to-Device (D2D) communications. In both scenarios, joint design of the transmission and caching policies is studied when the user demands are known in advance. This joint design offers two different caching gains, namely, the *pre-downloading* and *local caching gains*. It is shown that the finite cache capacity limits the attainable gains, and creates an inherent tradeoff between the two types of gains. In this context, a continuous time optimization problem is formulated to determine the optimal transmission and caching policies that minimize a generic cost function, such as energy, bandwidth, or throughput. The jointly optimal solution is obtained by demonstrating that caching files at a constant rate is optimal, which allows to reformulate the problem as a finite-dimensional convex program. The numerical results show that the proposed joint transmission and caching policy dramatically reduces the total cost, which is particularised to the total energy consumption at the Macro Base Station (MBS), as well as to the total economical cost for the service provider, when users demand economical incentives for delivering content to other users over the D2D links.**

***Index Terms*—Proactive caching, 5G, wireless backhaul, small cells, energy-efficiency, device-to-device.**

I. Introduction

Wireless traffic has experienced a tremendous growth in the last years due to the wide spread use of hand-held devices connected to the Internet, e.g., mobile phones, tablets, etc. This traffic increase is expected to continue steadily in the coming years; for example, more than 127 exabytes of worldwide mobile traffic is forecasted for the year 2020 [1]. Video traffic is the major data source due to the growing success of on-demand video streaming services [1]. Traffic resulting from video on-demand services exhibits the *asynchronous content reuse* property [2], according to which a few popular files, requested by users at different times (as opposed to television broadcasting services), account for most of the data traffic.

Maria Gregori, Jesús Gómez-Vilardebó, and Javier Matamoros are with the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), 08860 Barcelona, Spain (e-mails: {maria.gregori, jesus.gomez, javier.matamoros}@cttc.cat). Deniz Gündüz is with the Imperial College of London, UK (e-mail: d.gunduz@imperial.ac.uk).

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 the ERA-NET CHIST-ERA program, and by the Catalan Government (2014 SGR 1567).

To cope with this growing traffic requirements, lots of efforts have been devoted towards the definition of the fifth generation of cellular communication systems (5G), which is expected to be operative by 2020. The 5G system must provide fast, flexible, reliable, and sustainable wireless connectivity, while supporting the growing mobile traffic. Device-to-Device (D2D) communications, small cell densification, millimeter wave, and massive MIMO are currently investigated as main enabling technologies for its success.

Small cell densification refers to the deployment of a large number of Small Base Stations (SBSs) with different cell sizes (micro, pico, and femtocells) allowing a larger spatial reuse of the resources. The major drawback of cell densification is that the traffic that can be served by an SBS is limited by the capacity of the backhaul link, which provides connection to the core network. This link is preferably wireless for various reasons such as, rapid deployment, self-configuration, and cost. However, wireless backhaul connections entail limited capacity and significant energy consumption (due to its relatively long range).

Caching the most popular contents at the network edge has been proposed in [3] to increase connectivity, in [4] to reduce the delay, and in [5] to alleviate the backhaul link congestion and to reduce its energy consumption. Video traffic (e.g., popular Youtube videos) is especially suitable to be cached since it requires high data rates and exhibits the aforementioned asynchronous content reuse property. The contents can be cached either at SBSs equipped with a cache memory (also coined as "femtocaching") [5]–[11], or directly at the users' devices [2], [12], [13]. The users can exchange the cached content through D2D communications [14], which allows direct communication between nearby mobile users. In practice, due to limited cache and energy resources, users are unwilling to serve data over the D2D links unless they obtain incentives (e.g., economical) from the operator [15].

In a popular approach to wireless caching, [7], [11], [16], the system design is performed in two separated phases. First, in the *content placement* phase, each cache is filled with appropriate data, exploiting periods of time in which the network is not congested. 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* [11]. On the one hand, the *local caching gain* is obtained when a requested file is locally available in the cache (either at the SBS or at the users) by serving this file from the cache without connecting to the Macro Base Station (MBS). This reduces the traffic in the wireless backhaul link [7] and improves the quality of experience [11]. On the other hand, the *global caching gain* is obtained by multicasting network-coded information in the delivery phase [11], [16]. However, this underlying separation

between the caching (content placement) and transmission (delivery) phases has two limiting assumptions: i) the content placement phase is cost-free (e.g., in terms of energy or bandwidth); 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. In particular, we consider that the cache is initially empty, and it is dynamically filled with contents, i.e., we combine the content placement and delivery phases. This approach still allows to pre-download data over low-traffic periods; however, we now account for the cost of downloading these contents. As a result, an additional caching gain is obtained, which we call *pre-downloading gain*. Essentially, the *pre-downloading caching gain* is obtained when the cache is used to pre-download data, which can be beneficial to avoid non-favourable channel conditions, and to equalize the rate in the backhaul link, improving its energy efficiency, and reducing its peak load. In this context, the authors of [17] and [18] derive 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 [17] and [18] 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 strategies by accounting for both the local and pre-downloading caching gains.

To fully exploit the aforementioned gains, efficient cache management policies must be designed by taking into account, among others, the stochastic but predictable nature of users' demands. The cache management policies can be classified into two groups according to the prior knowledge of the different system parameters (users' requests, channel state information, etc.): (i) *offline* caching policies that assume non-causal and complete knowledge of these parameters, e.g., [17]–[19]; and (ii) *online* caching policies that consider only causal or probabilistic knowledge of these parameters, e.g., [6]–[10], [16], [20]. The characterization of the optimal offline policy is extremely useful because: i) it serves as a theoretical bound on the performance achievable by any online policy; and ii) it can be instrumental in designing low-complexity near-optimal online policies. Finding the optimal online policy is extremely challenging since the cache management problem is usually a hard combinatorial problem. As a result several works have resorted to heuristic algorithms [6], [7].

In contrast to previous literature, the aim of this paper is to study the *jointly optimal* transmission and caching policies by taking into account both the local and pre-downloading caching gains under two different scenarios. The first scenario considers a caching SBS that serves the demands from multiple users. When the users' demand is not locally available at the SBS, the SBS downloads the content from an MBS through a wireless backhaul link. We addressed this scenario in [21] assuming that the SBS serves the users in a time division fashion; in this paper, we allow the SBS to serve multiple users simultaneously. In the second scenario, we consider that the MBS directly serves demands from users, which can cache the received data proactively, and later cooperate with other users through D2D communications. The key difference between the two scenarios is that, in the former, the cache is centralized at the SBS, whereas in the later, it is distributed across users. The main contributions of the paper are summarized next:

- For the two scenarios mentioned above, we study the joint design of the optimal transmission and caching policies by formulating a continuous time optimization problem aimed at minimizing a generic cost function (e.g., the energy, throughput, or bandwidth requirement).
- For the first scenario, where caching is performed at the SBS: (i) we show that, within each time slot, it is optimal to cache data at a constant rate, which permits reformulating the problem as a convex program; (ii) we solve this convex problem by means of dual decomposition and propose a subgradient algorithm to obtain the optimal dual variables; and (iii) we derive the structure of the optimal transmission power at the MBS and the caching policy at the SBS.
- For the second scenario, where information is cached at the users and shared through D2D communications: (i) we show that, within each time slot, each user should cache data at a constant rate; (ii) we show that each user should transmit the files in the D2D links at a constant rate; and (iii) we reformulate the problem as a convex optimization problem.
- Finally, the two scenarios are compared through numerical simulations. Fist, we compare the performance of a centralized cache with a distributed one and assess the impacts of pre-downloading and local caching gains in each scenario. Second, we evaluate how the cost of the MBS increases with the economical incentives requested by users for transmitting data over the D2D links.

The remainder of the paper is structured as follows. Section II focuses on the first scenario, where caching is performed at an SBS. In particular, the system model is presented in Section II-A; the optimal transmission strategy is derived for a fixed caching policy in Section II-B; and the problem is solved in Section II-C. Section III is devoted to the second scenario where caching is performed at the user terminals. The system model for this scenario is introduced in Section III-A and the resulting problem is solved in Section III-B. Section IV presents the numerical results. Finally, the paper is concluded in Section V.

Notation: Vectors and vector valued functions are denoted by lower case boldface letters, i.e., $\mathbf{v}$ and $\boldsymbol{\rho}(\mathbf{v})$, respectively. $(\mathbf{v}_u)_{u=1}^U$ defines a column vector obtained by stacking the column vectors $\mathbf{v}_1, \ldots, \mathbf{v}_U$ and $[\mathbf{v}]_k$ returns the k-th element of the vector $\mathbf{v}$. Symbol $\preceq$ denotes the component-wise "smaller than or equal to" inequality. Finally, $[x]^+ \triangleq \max\{0, x\}$.

II. SBS CACHING FOR 5G NETWORKS

A. System model and problem formulation

As depicted in Fig. 1, we consider U users served by an SBS. The SBS has a finite cache memory of capacity C units, and is connected through a wireless backhaul channel to an

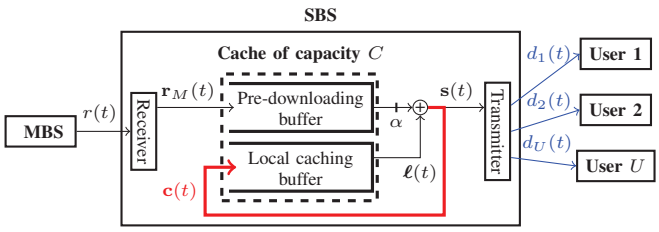


Fig. 1. System model when caching is performed at an SBS that serves U users.

MBS, which has access to the core network. We assume that the MBS and the SBS operate in different frequency bands; thus, no interference is produced between the two. We define T as the optimization time horizon consisting of N time slots of duration T_s each. In each time slot, each user, u, $u = 1, \ldots, U$, requests one file from the set of all possible files $\mathbb{F} = \{f_0, \ldots, f_F\}$. We define l_j as the length (in data units) of file f_j, $j = 0, \ldots, F$. File f_0 has length $l_0 = 0$ and represents slots without requests. Similarly to [19], we fix the duration of each file in the set $\mathbb{F}$ to the duration of one time slot, T_s.[1] File f_j is consumed at a constant rate l_j/T_s by the users.

As shown in Fig. 1, data is transmitted by the MBS at a rate $r(t)$. The SBS receives this information and separates the streams associated to different user requests, obtaining the rate vector $\mathbf{r}_M(t) = (r_u(t))_{u=1}^U$ with $r_u(t)$ being the rate associated to user u. The downloaded data, $r(t) = \sum_{u=1}^U r_u(t)$, is then stored at the SBS cache until it is served to the users (which, without loss of generality, can happen immediately). The SBS has a demand rate denoted by $\mathbf{s}(t)$ to satisfy the users' demand rates, $d_u(t)$, $\forall u$. As it will be explained later, the SBS demand rate is obtained as the sum of the users' demand rates, $d_u(t)$, after removing multiple demands for the same file within the same slot. When serving a content to a user, the SBS either deletes or locally caches it. This is dictated by the local caching rate $\mathbf{c}(t)$. Notice that the cache is represented with two different virtual buffers, namely, the pre-downloading and local caching buffers. This representation with virtual buffers allows us to distinguish between the cached data that is downloaded in advance from the MBS from the locally cached data that is used to reduce future requests from the MBS. In this context, we define the vector $\boldsymbol{\ell}(t) \triangleq (\ell_u(t))_{u=1}^U$ whose u-th component $\ell_u(t)$ denotes the rate at which data is removed from the local caching buffer to reduce the demand at time t from the MBS associated to u-th user request[2]. In the sequel, we provide formal definitions for $d_u(t)$, $\mathbf{s}(t)$, and $\mathbf{c}(t)$. As in [17], [18], we assume a known demand profile (i.e, offline approach, see Section I); accordingly, we assume that the demand variables $d_u(t)$ and $\mathbf{s}(t)$ are known for the period $[0, T]$.

We define $\delta_u(j, n)$ as the *user request indicator variable* that takes value 1 when user u requests file f_j in slot n, $n = 1, \ldots, N$, and 0, otherwise. Since each user requests one file from $\mathbb{F}$ per slot, we have $\sum_{j=0}^F \delta_u(j, n) = 1$, $\forall u, n$.

[1]Note that any generic file can be partitioned into smaller files to meet the requirement of having the same duration T_s.

[2]In Fig. 1 we represent the locally cached data as a feedback link from the output to the input. Note that the data removed from the local caching buffer, $\boldsymbol{\ell}(t)$, can be instantaneously cached again if dictated by the local caching rate $\mathbf{c}(t)$ (implying that in practice the content is not removed from the cache).

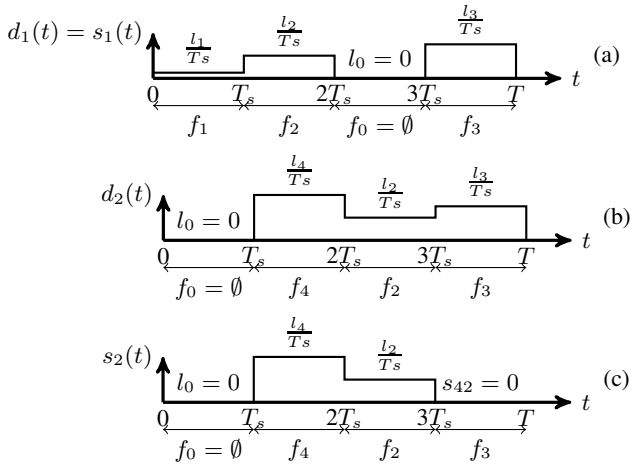


Fig. 2. (a) and (b) denote the demand rates of users one and two, respectively. The values above the curves represent the value of d_{nu}. User 1 requests the files f_1, f_2, f_0, and f_3, and user 2 requests f_0, f_4, f_2, and f_3 in this order. The user request indicator variable for user 1 takes values $\delta_1(1,1) = \delta_1(2,2) = \delta_1(0,3) = \delta_1(3,4) = 1$, and 0, otherwise; similarly, for user 2, $\delta_2(0,1) = \delta_2(4,2) = \delta_2(2,3) = \delta_2(3,4) = 1$, and it is 0, otherwise. (c) shows the required SBS demand rate for user 2 (for user 1, we have $s_1(t) = d_1(t)$ as shown in (a)). The values above the curves represent the value of s_{n2}. Note that $s_{n2} = d_{n2}$ for all the slots except the fourth one, where we have $s_{42} = 0$, because the two users request file f_3 in the fourth slot; thus, we have $\sigma_2(3,4) = 0$.

Definition 1 (User demand rate)**.** *The demand rate of user u, $d_u(t) \geq 0$, $t \in [0, T]$, is the rate at which user u requests data from the SBS, i.e., $d_u(t) \triangleq \sum_{n=1}^N d_{nu}\, \mathrm{rect}((t - (n - 1/2)T_s)/T_s)$, where d_{nu} denotes the demand rate of the u-th user at the n-th time slot, i.e., $d_{nu} = \sum_{j=0}^F \delta_u(j,n) l_j/T_s$; and $\mathrm{rect}((t-a)/b)$ stands for the rectangular function centered at a with duration b.*

Figs. 2(a)-(b) depict users' demand rates when two users are served from the SBS. Note that if a certain file is requested by multiple users at the same time slot (as in the fourth time slot in Fig. 2), these requests can be simultaneously handled by the SBS without the need of downloading the same file multiple times from the MBS. Therefore, in order to determine the minimal demand rate of the SBS, we must account only once for simultaneous requests of the same file within one slot. Without loss of generality, we account for the request of the user with the smallest index u. Accordingly, we define the *SBS request indicator variable* $\sigma_u(j,n)$ that takes value 1 for the user with the smallest index u requesting file f_j in time slot n (i.e., $\sigma_u(j,n) = 1$ if $\delta_u(j,n) = 1$ and $u < u', \forall u' \neq u : \delta_{u'}(j,n) = 1$), and 0, otherwise.

Definition 2 (SBS demand rate)**.** *The demand rate of the SBS is denoted by the vector $\mathbf{s}(t) = (s_u(t))_{u=1}^U$, $t \in [0, T]$. The u-th component of this vector, $s_u(t) \geq 0$, identifies the rate at which the data corresponding to the u-th user request must be available at the SBS to fulfill this request. Thus, we have $s_u(t) \triangleq \sum_{n=1}^N s_{nu}\, \mathrm{rect}((t - (n - 1/2)T_s)/T_s)$, where s_{nu} denotes the SBS demand rate at the n-th slot for the u-th user*

request, $s_{nu} = \sum_{j=0}^{F} \sigma_u(j,n) l_j / T_s$.

Given the users' demands in Figs. 2(a)-(b), the associated SBS demand rates are shown in Figs. 2(a) and 2(c), respectively. Note that if the SBS demand, $\mathbf{s}(t)$, is satisfied, then the SBS can serve all the user requests, $d_u(t), \forall u$. In the remainder of this section, unless it is stated otherwise, by a user request we refer to the request seen by the SBS, $s_u(t)$, instead of the request on the user side, $d_u(t)$.

Definition 3 (Local caching rate)**.** *The local caching rate is represented by the vector* $\mathbf{c}(t) \triangleq (c(t,u))_{u=1}^{U}$ *whose* u*-th component,* $c(t,u)$*, denotes the rate at which the SBS caches the content associated to user* u *at time* t*. Thus, we have* $0 \leq c(t,u) \leq s_u(t)$*,* $t \in [0,T]$.

Remark 1. The variables in the system model must be able to indicate which portions of each file are cached at time t. Given a certain user request u and time instant t, we can identify the file being requested (through the SBS indicator variables). Then, the cached portions of a file can be identified with the tuple $\{\mathbf{r}_M(t), \mathbf{c}(t), \mathbf{s}(t)\}$. Note that we could have defined the vectors $\mathbf{r}_M(t), \mathbf{c}(t)$, and $\mathbf{s}(t)$ in terms of the files instead of user requests (i.e., with dimensions $F \times 1$ instead of $U \times 1$), which would simplify the identification of the cached files; however, this would dramatically increase the computational complexity of the algorithms proposed in the remainder of the paper as, in general, the number of available files, F, is several orders of magnitude larger than the number of users connected to the SBS, U.

Our aim is to *jointly* design the transmission policy at the MBS, $r \triangleq \{r(t)\}_{t=0}^{T}$, and the local caching policy at the SBS, $\mathbf{c} \triangleq \{\mathbf{c}(t)\}_{t=0}^{T}$, to minimize a generic cost function in the backhaul link, $\int_0^T g(r(\tau))d\tau$, where $g(r(t))$ denotes the instantaneous cost, which depends on the instantaneous transmission rate at the MBS. As in [22], we assume that the instantaneous cost function $g(\cdot)$ is time invariant, convex, increasing, continuously differentiable, and $g(0) = 0$. In the following, we give four examples of cost functions that satisfy these conditions:

1) **Energy consumption minimization:** If the objective is to minimize the total network energy consumption, then the instantaneous cost is given by the instantaneous total power consumption, $p(t)$. In the case of Gaussian signaling, we have $p(t) = g(r(t)) = (\exp(r(t)) - 1)/h + P_c + p_S(t)$, where h denotes the channel gain, P_c stands for the static circuitry consumption at the MBS, and $p_S(t)$ is the SBS power consumption, which is known as it can be computed from the power-rate function at the SBS and the users' demands.

2) **Energy cost minimization:** The instantaneous power consumption above has to be multiplied by the energy cost, ξ_{MBS}, paid by the network operator to the electricity utility. Thus, the instantaneous cost function is $g(r(t)) = \xi_{MBS} \cdot ((\exp(r(t)) - 1)/h + P_c + p_S(t))$.

3) **Bandwidth minimization:** In this case, the cost function is given by the bandwidth-rate function, $w(t) = g(r(t)) = f^{-1}(r(t))$, obtained as the inverse of the rate-bandwidth function, $r(t) = f(w(t))$. Again, in the case of Gaussian signaling, we have $r(t) = f(w(t)) = w(t)\mathsf{log}(1 + Ph/w(t))$, where P denotes the constant transmission power and h stands for the channel gain.

4) **Traffic minimization:** To minimize the data transmitted by the MBS, we obtain $g(r(t)) = r(t)$.

As argued in the introduction, the cache offers two different gains to reduce the cost in the backhaul link, namely, *pre-downloading* and *local* caching gains. As shown in Fig. 1, the cache has two inputs: (i) the pre-downloaded data from the MBS, which is controlled by the transmission policy at the MBS, r, and contributes to the pre-downloading caching gain; and (ii) the locally cached data, which is controlled by the local caching policy at the SBS, $\mathbf{c}$, and contributes to the local caching gain. The design of r and $\mathbf{c}$ is constrained by the cache size and the required demand rate at the SBS. In the following, we define these constraints in terms of the cumulative transmitted data [22].

Definition 4 (Data departure curve)**.** *The data departure curve,* $D(t,r)$*, is the amount of total data served by the MBS by time* $t \geq 0$*, and can be obtained from the transmission policy,* r*, as* $D(t,r) \triangleq \int_0^t r(\tau)\mathrm{d}\tau$.

Due to the finite cache capacity, an upper bound on $D(t,r)$ must be imposed to avoid data overflows from the SBS cache. This upper bound is imposed by the maximum data departure curve that, as defined next, increases as data is removed from the SBS cache. The rate at which data is removed from the cache at time t is obtained as $\sum_{u=1}^{U} s_u(t) - c(t,u)$.

Definition 5 (Maximum data departure curve)**.** *The maximum data departure curve,* $B(t,\mathbf{c})$*, limits the maximum amount of total data that can be transmitted by the MBS by time* $t \geq 0$ *such that no data overflow at the cache memory is generated. Thus, it is given by* $B(t,\mathbf{c}) \triangleq C + \int_0^t \sum_{u=1}^{U} s_u(\tau) - c(\tau,u)\mathrm{d}\tau$ *and depends on the caching policy* $\mathbf{c}$*.*

The lower bound on the data departure curve is given by the minimum amount of total data that must be downloaded from the MBS to satisfy the SBS demand rate. The *net SBS demand rate* from the MBS (the demand rate at point α in Fig. 1) is the requested data that is not available in the local caching buffer. Consider that, at a certain time instant t, user u requests a file that had been previously requested by user u' at time t', $t' < t$. Then, the net SBS demand rate at time t of the u-th user request is given by $s_u(t) - \ell_u(t)$, where, as mentioned earlier, $\ell_u(t)$ denotes the rate at which data is removed from the local caching buffer to reduce the demand at time t from the MBS. Note that the rate $\ell_u(t)$ must be equal to the caching rate adopted during the previous request of the file requested by user u at time t, i.e., $\ell_u(t) = c(t',u')$ (as otherwise data is unnecessarily downloaded from the MBS). To compute the net SBS demand rate for any user and time instant, we define the vector function $[t',u'] = \boldsymbol{\rho}(t,u)$. This function returns the time instant t' and the index of the user, u', which last requested the file requested by user u at time t. If the file being requested at time t by user u has not been requested previously, we set $\boldsymbol{\rho}(t,u) = [-1,-1]$, and define $c(-1,-1) \triangleq 0$ (since the files that have not been requested are not yet available at the SBS). The function $\boldsymbol{\rho}(t,u)$ is depicted in Fig. 3 for the demand profile in Fig. 2. Using the function

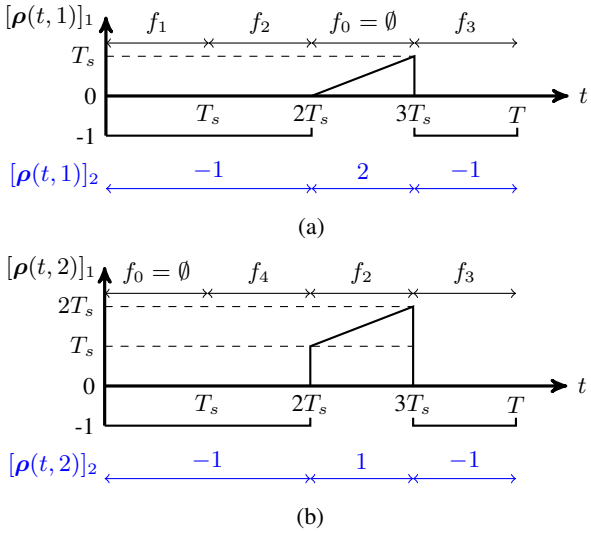


Fig. 3. Representation of the functions $\boldsymbol{\rho}(t,u)$ that maps a certain file request in the SBS to its previous occurrence in time ($t' = [\boldsymbol{\rho}(t,u)]_1$) and user index ($u' = [\boldsymbol{\rho}(t,u)]_2$). (a) corresponds to the first user and (b) to the second one.

$\boldsymbol{\rho}(t,u)$, the net SBS demand rate at time t associated with the u-th user request is given by $s_u(t) - c(\boldsymbol{\rho}(t,u))$. Since non-causal knowledge of the user demands is available (offline approach), the function $\boldsymbol{\rho}(t,u)$ is known, $\forall t,u$. Next we define the minimum data departure curve to satisfy the SBS demand.

Definition 6 (Minimum data departure curve)**.** *The minimum data departure curve, $A(t,\mathbf{c})$, is the minimum amount of total data that must be transmitted by the MBS by time $t \geq 0$ to satisfy the SBS demand, and depends on the caching policy, $\mathbf{c}$, i.e., $A(t,\mathbf{c}) \triangleq \sum_{u=1}^{U} \int_0^t s_u(\tau) - \ell_u(\tau)\mathrm{d}\tau = \sum_{u=1}^{U} \int_0^t s_u(\tau) - c(\boldsymbol{\rho}(\tau,u))\mathrm{d}\tau$.*

Bearing all the above in mind, the problem is mathematically formulated as follows:

$$\min_{\{r(t),\mathbf{c}(t)\}_{t\in[0,T]}} \quad \int_0^T g(r(\tau))\mathrm{d}\tau \tag{1a}$$

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

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

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

$$\mathbf{0} \preceq \mathbf{c}(t) \preceq \mathbf{s}(t), \qquad \forall t \in [0,T], \tag{1e}$$

where the constraint (1b) prevents cache overflows, and (1c) imposes the fulfillment of the users' demands. The constraints (1d) and (1e) guarantee feasible transmission and local caching rates. Note that a feasible caching policy, $\mathbf{c}$, must satisfy $B(t,\mathbf{c}) \geq A(t,\mathbf{c})$ for all $t \in [0,T]$, and any feasible data departure curve must lie within the tunnel between $B(t,\mathbf{c})$ and $A(t,\mathbf{c})$.

Remark 2. In the problem formulation, we have assumed that cached data can only be removed from the cache during the subsequent requests of the same data. As a result, by caching data the net SBS demand rate will be reduced. We remark here that this assumption is without loss of optimality. Contrarily, consider a policy that caches a certain data content at time t_1, its subsequent request occurs at t_2, but the content is deleted at $t_3 \in (t_1,t_2)$. As this content has to be downloaded again at t_2, this policy is unnecessarily using cache space in (t_1,t_3).

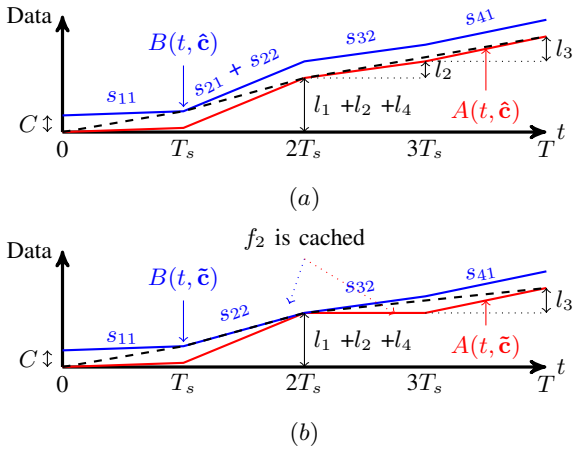


Fig. 4. Representation of the problem for two different caching policies. In this example, we have set $C = l_2$.

Remark 3. In realistic 5G scenarios, several SBSs will be served by the same MBS. This work considers that the MBS assigns orthogonal resources to each SBS and that the SBSs have non overlapping coverage areas. As a result, a problem of the form of (1) is obtained for each SBS. Further gains can be achieved by multicasting information to different SBSs, or by cooperation among SBSs with overlapping coverage areas [5], [23]. However, this will inherently couple the design of the SBSs' caching policies, and is out of the scope of this work.

B. Optimal transmission strategy for a fixed caching policy

In this section, we derive the optimal transmission strategy for a fixed caching policy. Interestingly, when the local caching policy, $\mathbf{c}$, is given, the problem in (1) accepts an intuitive graphical representation. For example, under the SBS demand rate in Figs. 2(a) and 2(c), the problem is represented in Fig. 4 for two different caching policies:

Policy 1: The policy $\hat{\mathbf{c}}$ shown in Fig. 4(a) removes the data from the cache as soon as it is served to a user, ignoring any possible future requests for the same file, i.e., $\hat{\mathbf{c}}(t) = \mathbf{0}$, $\forall t$. Consequently, it only exploits the pre-downloading caching gain. This caching policy was proposed in [18]. Observe that if $\hat{\mathbf{c}}(t) = \mathbf{0}$, $\forall t$, then there is a constant gap of C between the lower and upper bounds, i.e., $B(t,\hat{\mathbf{c}}) = C + A(t,\hat{\mathbf{c}})$ (c.f. Definitions 5 and 6). The optimal data departure curve exploits this gap by pre-downloading data.

Policy 2: The policy $\tilde{\mathbf{c}}$ shown in Fig. 4(b) caches the file f_2, when requested by user 1 in the second time slot, thus anticipating the next request in the third slot by user 2, i.e., $\tilde{c}(t,1) = s_{21}$ if $t \in [T_s, 2T_s]$ and $\tilde{c}(t,u) = 0$, otherwise. As a result, no data needs to be transmitted by the MBS in the third slot.

Lemma 1 (Constant rate transmission is optimal [22])**.** *Given a feasible caching policy $\mathbf{c}$, the optimal data departure curve can be obtained as the tightest string whose ends are tied to the origin and the point $(T, A(T,\mathbf{c}))$, which is represented in Fig. 4 with the dashed lines. In particular, if the instantaneous cost,*

$g(\cdot)$, is strictly convex, then this is the unique optimal data departure curve; contrarily, if $g(\cdot)$ is linear multiple optimal departure curves exist.

The free memory space in the cache can be obtained as $B(t,\mathbf{c})-D(t,r)$, $\forall t$. Focusing on Policy 1 (see Fig. 4(a)), the cache is full at $t=T_s$, and all the data in the cache belongs to f_2 and/or f_4, which have been pre-downloaded to equalize the rates in the first and second time slots. As for Policy 2 (see Fig. 4(b)), the cache is full at $t=2T_s$, and exclusively contains f_2. Note that by caching f_2 in the second slot the upper bound is tightened (the net cache capacity is reduced) while the lower bound is relaxed (the demand at the third slot is reduced).

From the previous discussion, two questions arise: i) "which of the two caching policies achieves the lowest MBS cost?", and ii) "is any of these policies the optimal one?". One might be tempted to think that the caching policy $\tilde{\mathbf{c}}$ has a lower cost since fewer data has to be transmitted; however, this does not necessarily hold true since the caching policy $\hat{\mathbf{c}}$ might achieve a lower cost by equalizing the rate across time slots. In practice, the jointly optimal transmission and local caching policies must be obtained by solving (1), which turns out to be challenging since this problem belongs to the class of infinite-dimensional optimization problems [24].

C. Jointly optimal caching and transmission policies

To solve the infinite-dimensional problem in (1), we first derive some structural properties of the optimal strategy. Then, leveraging on these properties, we will formulate (1) as a finite-dimensional convex program of affordable complexity. As shown next, the optimization variables of the resulting problem are the amount of data to be cached in each slot for each request, q_{nu}, $\forall n,u$, and the transmission rate of the MBS at each slot, r_n, $\forall n$.

As illustrated in Fig. 4, the caching policy changes the shape of the upper and lower bounds on the data departure curve. For example, in Fig. 4(b), we have observed that the data locally cached in the second slot reduces the demand in the third slot. Since the caching rate can have continuous variations over time, we can potentially have arbitrary non-decreasing curves as the upper and lower bounds, $B(t,\mathbf{c})$ and $A(t,\mathbf{c})$. However, these curves are coupled through the caching policy $\mathbf{c}$. In other words, the caching rate of a certain request determines the reduction in the demand rate of the subsequent request. The following lemma shows that (within a time slot) caching data at a constant rate turns out to be optimal.

Lemma 2 (Constant rate caching is optimal)**.** *The (not necessarily unique) optimal local caching rate is a step-wise function that can be written as $\mathbf{c}^\star(t)=(c^\star(t,u))_{u=1}^U$, where $c^\star(t,u)=\sum_{n=1}^N (q_{nu}^\star/T_s)\,\mathrm{rect}((t-(n-1/2)T_s)/T_s)$, and $q_{nu}^\star$ denotes the optimal amount of cached data for the request of the u-th user at slot n.*

Proof: See the Appendix. ∎

Since $s_u(t)$ and $c^\star(t,u)$ are step-wise functions whose value can only change at slot transitions, we know that $A(t,\mathbf{c}^\star)$ and $B(t,\mathbf{c}^\star)$ are piece-wise linear functions (c.f. Definitions 5 and 6) whose slopes can only change at slot transitions. Consequently, we can obtain the following properties of the optimal transmission strategy.

Lemma 3. *The (not necessarily unique) optimal data departure curve, $D^\star(t,r^\star)$, can be written as a piece-wise linear function, whose rate (or, equivalently, the slope of $D^\star(t,r^\star)$) may only change at time instants $n\cdot T_s$, $n=1,\dots,N-1$, i.e., $r^\star(t)=\sum_{n=1}^N r_n^\star\,\mathrm{rect}((t-(n-1/2)T_s)/T_s)$, where $r_n^\star$ denotes the optimal transmission rate of the MBS at the n-th slot. Additionally, if the rate increases at the n-th slot transition ($r_n^\star<r_{n+1}^\star$), then $D^\star(nT_s,r^\star)=B(nT_s,\mathbf{c}^\star)$; and if the rate decreases at the n-th slot transition ($r_n^\star>r_{n+1}^\star$), then $D^\star(nT_s,r^\star)=A(nT_s,\mathbf{c}^\star)$.*

Proof: The proof follows similarly to [25, Lemmas 5 and 6] by identifying nT_s as ℓ_m, $B(nT_s,\mathbf{c}^\star)$ as $D_{max}^{(m)}(\ell_m)$, and $A(nT_s,\mathbf{c}^\star)$ as $D_{min}^{(m)}(\ell_m)$. ∎

From Lemmas 2 and 3, we can equivalently rewrite the original problem in (1) as a function of the MBS rates at each slot, $\mathbf{r}\triangleq(r_n)_{n=1}^N$, and cached data units at the SBS for each user request and time slot, $\mathbf{q}\triangleq((q_{nu})_{u=1}^U)_{n=1}^N$:

$$\min_{\mathbf{r},\mathbf{q}} \quad \sum_{n=1}^{N} T_s g(r_n) \tag{2a}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{n} T_s r_\ell \le C+\sum_{\ell=1}^{n}\sum_{u=1}^{U} T_s s_{\ell u}-q_{\ell u}, \qquad \forall n, \tag{2b}$$

$$\sum_{\ell=1}^{n} T_s r_\ell \ge \sum_{\ell=1}^{n}\sum_{u=1}^{U} T_s s_{\ell u}-q_{\bar{\boldsymbol{\rho}}(\ell,u)}, \qquad \forall n, \tag{2c}$$

$$r_n\ge 0, \qquad \forall n, \tag{2d}$$

$$0\le q_{nu}\le T_s s_{nu}, \qquad \forall n,u, \tag{2e}$$

where the constraints (2b)-(2e) correspond to the discrete versions of the constraints in (1b)-(1e), respectively. The function $(n',u')=\bar{\boldsymbol{\rho}}(n,u)$ returns the slot, n', and user, u', of the previous request of the file associated to (n,u), or returns $(-1,-1)$ if it is the first request of the file. Note that $\bar{\boldsymbol{\rho}}$ is the discrete version of the function $\boldsymbol{\rho}$; and as before, we define $q_{-1-1}\triangleq 0$.

Remark 4. In (2), we have considered that the amount of cached data, q_{nu}, is a nonnegative real number. Note that if we introduce an integer constraint to enforce data unit granularity (e.g., bit), then the problem in (2) becomes an integer program with its inherent complexity. In practice, as the data unit granularity (bit) is sufficiently small in comparison to the files sizes (of several Mbits) and cache capacity (of several Gbits), the integer constraint can be relaxed without jeopardizing the performance. Consequently, (2) is a convex program (since the objective function is convex and the constraints are affine), and, thus, can be solved efficiently.

By studying the Karush Kuhn Tucker conditions of the primal problem in (2), it is difficult to derive the structure of the optimal solution $\{\mathbf{r}^\star,\mathbf{q}^\star\}$ due to the constraints in (2b) and (2c) that couple the optimization variables. However, the structure of the optimal primal variables can be obtained by resorting to dual decomposition. From convex optimization theory [26], the solution of the dual problem, $\max_{\boldsymbol{\lambda},\boldsymbol{\mu}}\delta(\boldsymbol{\lambda},\boldsymbol{\mu})$,

Algorithm 1 Projected subgradient

Initialization:
Set $k := 0$ and initialize $\boldsymbol{\lambda}^{(0)}$ and $\boldsymbol{\mu}^{(0)}$ to any value such that $\boldsymbol{\lambda}^{(0)} \succeq \mathbf{0}$, $\boldsymbol{\mu}^{(0)} \succeq \mathbf{0}$.
Step 1: If a termination condition is met, the algorithm stops.
Step 2: Compute $\mathbf{r}^{(k)}, \mathbf{q}^{(k)}$ as the solution to the problem in (3) given the current multipliers, $\boldsymbol{\lambda}^{(k)}$ and $\boldsymbol{\mu}^{(k)}$.
Step 3: Update the dual variables following the subgradient, i.e., $[\boldsymbol{\lambda}^{(k+1)}]_n = \lambda_n^{(k+1)}$ and $[\boldsymbol{\mu}^{(k+1)}]_n = \mu_n^{(k+1)}$, $\forall n$, with

$$\lambda_n^{(k+1)} = \left[\lambda_n^{(k)} + \epsilon^{(k)} \left(-C + \sum_{\ell=1}^{n} T_s r_\ell^{(k)} - \sum_{u=1}^{U} T_s s_{\ell u} - q_{\ell u}^{(k)}\right)\right]^+$$

$$\mu_n^{(k+1)} = \left[\mu_n^{(k)} - \epsilon^{(k)} \left(\sum_{\ell=1}^{n} T_s r_\ell^{(k)} - \sum_{u=1}^{U} T_s s_{\ell u} - q_{\bar{\rho}(\ell,u)}^{(k)}\right)\right]^+.$$

Step 4: Set $k := k + 1$ and go to Step 1.

provides a lower bound on the primal problem in (2). We have defined $\boldsymbol{\lambda} = (\lambda_n)_{n=1}^N$ and $\boldsymbol{\mu} = (\mu_n)_{n=1}^N$, where λ_n and μ_n are the Lagrange multipliers associated to the n-th cache capacity and demand constraints, respectively. The function $\delta(\boldsymbol{\lambda}, \boldsymbol{\mu})$ stands for the dual function that is defined as follows [26]:

$$\delta(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{r},\mathbf{q}} \quad \mathcal{L}(\mathbf{r}, \mathbf{q}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad (3)$$
$$\text{s.t.} \qquad r_n \geq 0, \forall n, \quad 0 \leq q_{nu} \leq T_s s_{nu}, \forall n, u,$$

where $\mathcal{L}(\mathbf{r}, \mathbf{q}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ denotes the Lagrangian, i.e., $\mathcal{L}(\mathbf{r}, \mathbf{q}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{n=1}^N T_s g(r_n) + \lambda_n\big(-C + \sum_{\ell=1}^n T_s r_\ell - \sum_{u=1}^U T_s s_{\ell u} - q_{\ell u}\big) - \mu_n\big(\sum_{\ell=1}^n T_s r_\ell - \sum_{u=1}^U T_s s_{\ell u} - q_{\bar{\rho}(\ell,u)}\big)$.

Since the primal problem in (2) is convex and the Slater constraint qualification holds, the duality gap (difference between the optimal values of the primal and dual problems) is zero [26]. To solve the dual problem, we have implemented the projected subgradient method, presented in Algorithm 1, that guarantees convergence to the optimal dual variables, $\boldsymbol{\lambda}^\star$ and $\boldsymbol{\mu}^\star$, if the updating step size $\epsilon^{(k)}$ is correctly chosen [27].

Step 2 of Algorithm 1 requires to solve the problem in (3), where the optimization variables are $\mathbf{r}, \mathbf{q}$, and the Lagrange multipliers ($\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$) are fixed. To do so, we first rewrite the Lagrangian by reordering the sums over n and ℓ, which allows us to separate the terms associated to each r_n and q_{nu}, i.e., $\mathcal{L}(\mathbf{r}, \mathbf{q}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{n=1}^N T_s g(r_n) - r_n T_s\big(\sum_{\ell=n}^N \mu_\ell - \lambda_\ell\big) + \sum_{n=1}^N \sum_{u=1}^U q_{nu}\big(\sum_{\ell=n}^N \lambda_\ell - \sum_{\ell=\psi(n,u)}^N \mu_\ell\big) + \sum_{\ell=1}^N \mu_\ell(\sum_{n=1}^\ell \sum_{u=1}^U T_s s_{nu}) - \sum_{\ell=1}^N \lambda_\ell(C + \sum_{n=1}^\ell \sum_{u=1}^U T_s s_{nu})$. The function $\psi(n,u)$ returns the slot index of the subsequent request of the file being served at slot n to user u. Now, the problem in (3) is decoupled in the optimization variables ($\mathbf{r}$ and $\mathbf{q}$) and can be easily solved by decomposing it into the following simpler subproblems:

$$\min_{r_n \geq 0} \quad T_s g(r_n) - r_n T_s \left(\sum_{\ell=n}^{N} \mu_\ell - \lambda_\ell\right), \qquad \forall n, \quad (4)$$

$$\min_{0 \leq q_{nu} \leq T_s s_{nu}} \quad q_{nu} \left(\sum_{\ell=n}^{N} \lambda_\ell - \sum_{\ell=\psi(n,u)}^{N} \mu_\ell\right), \quad \forall n, u. \quad (5)$$

Let $\bar{r}_n$ be the solution to the equation $\mathrm{d}g(r_n)/\mathrm{d}r_n = \sum_{\ell=n}^N \mu_\ell - \lambda_\ell$. If $\bar{r}_n$ is real and positive, the optimal solution to (4) is $r_n^\star(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \bar{r}_n$; otherwise, it is $r_n^\star(\boldsymbol{\lambda}, \boldsymbol{\mu}) = 0$.

Corollary 1. *When the objective is the minimization of the energy consumption over the backhaul link ($g(r_n) = (\exp(r_n) - 1)/h$), the optimal solution to* (4) *is found as $r_n^\star(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \log(h(\sum_{\ell=n}^N \mu_\ell - \lambda_\ell))$ if $h(\sum_{\ell=n}^N \mu_\ell - \lambda_\ell) > 1$ and $r_n^\star(\boldsymbol{\lambda}, \boldsymbol{\mu}) = 0$, otherwise.*

The solution to (5) is

$$q_{nu}^\star(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \begin{cases} 0 & \text{if } W_{nu} > 0, \\ T_s s_{nu} & \text{if } W_{nu} < 0, \\ \bar{q}_{nu} \in [0, T_s s_{nu}] & \text{if } W_{nu} = 0, \end{cases} \qquad (6)$$

with $W_{nu} \triangleq \sum_{\ell=n}^N \lambda_\ell - \sum_{\ell=\psi(n,u)}^N \mu_\ell$. Accordingly, the primal variables at the q-th iteration of the subgradient, which are necessary in Step 2 of Algorithm 1, are given by $\mathbf{r}^{(k)} = \big(r_n^\star(\boldsymbol{\lambda}^{(k)}, \boldsymbol{\mu}^{(k)})\big)_{n=1}^N$ and $\mathbf{q}^{(k)} = \Big(\big(q_{nu}^\star(\boldsymbol{\lambda}^{(k)}, \boldsymbol{\mu}^{(k)})\big)_{u=1}^U\Big)_{n=1}^N$, where $\boldsymbol{\lambda}^{(k)}$ and $\boldsymbol{\mu}^{(k)}$ denote the Lagrange multipliers at the q-th iteration of the subgradient.

When the subgradient algorithm converges to the optimal Lagrange multipliers, $\{\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star\}$, the duality gap is zero, i.e., the optimal solution of the dual and primal problems are the same. Note that given the optimal dual variables, $\{\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star\}$, there might be multiple minimizers of the problem in (3). Precisely, $q_{nu}^\star(\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)$ can take multiple values when $W_{nu} = 0$ (see (6)). Then, the optimal primal variables $\{\mathbf{r}^\star, \mathbf{q}^\star\}$ are within the set of minimizers of $\delta(\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)$ in (3); in particular, $\{\mathbf{r}^\star, \mathbf{q}^\star\}$ are the minimizers that are feasible in the primal problem (2) and satisfy the slackness conditions [27]. In practice, to avoid waiting until the exact convergence to $\{\boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star\}$, the average across iterations of the primal iterates can be used as an approximate solution to the problem in (2) [28].

Interestingly, it turns out that the parameter W_{nu}, which only depends on the Lagrange multipliers, characterizes the caching policy: if W_{nu} is positive, the associated file is not cached; while, if W_{nu} is negative the file is completely cached; and, finally, if $W_{nu} = 0$ the SBS caches a portion of the file (the exact amount of cached data units must be obtained as mentioned in the previous paragraph). Additionally, from the expression of W_{nu}, we observe that the SBS caching policy prioritizes the files that are requested again in the near future.[3]

III. Caching at user devices

A. System model and problem formulation

In this section, as depicted in Fig. 5, we consider a region in space covered by an MBS that must serve the demand of U users. The MBS allocates an *orthogonal* channel to each user, whose transmission rate is denoted by $r_u(t)$, $u = 1, \ldots, U$. We assume that users cooperate with the network operator (possibly in exchange of incentives) by acting as an SBS for the rest of the users through dedicated D2D links. We consider

[3]From the KKT optimality conditions, we have $\mu_n > 0$ if the n-th demand constraint in (2c) is satisfied with equality (and zero otherwise). Consider that users u and u' request different files and that the subsequent request of these files appears first for user u, i.e., $\psi(n,u) < \psi(n,u')$. Then, from the expression of W_{nu} in (6), we have $W_{nu} \leq W_{nu'}$ and, as a result, the SBS prioritizes caching the file requested by user u.

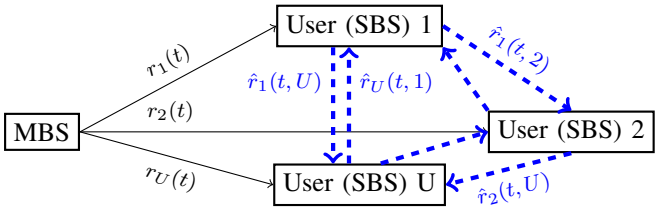


Fig. 5. System model when caching is performed at the user devices, which act as an SBS for the other users through D2D communications.

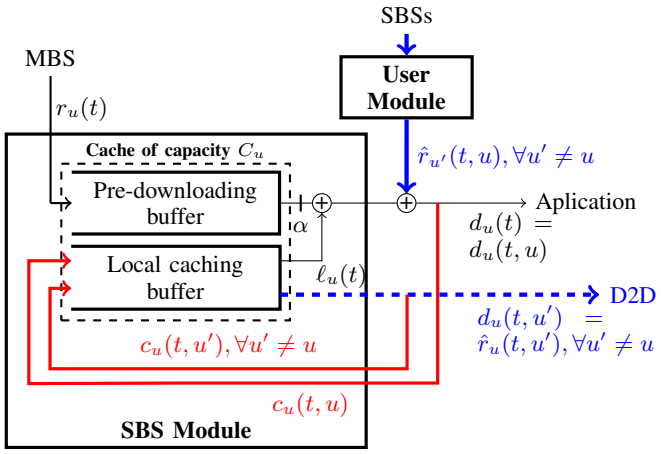


Fig. 6. Block diagram of the u-th user terminal. The solid lines correspond to data streams associated to traffic of the u-th user, and the dashed ones represent traffic served to other users, $u' \neq u$, that is transmitted through the D2D links.

that users are closely located; thus, a certain user u can act as an SBS for any other user $u' \neq u$. The rate in the D2D link from SBS (user) u to user u' at time t is denoted by $\hat{r}_u(t,u')$, $u' \neq u = 1,\dots,U$. We assume that the D2D links operate over different frequency resources than those used by the MBS. A user is allowed to download only its own traffic from the MBS; that is, users do not download content that is of no interest to them, solely to serve another user. However, downloaded files can be locally cached to later serve other users through the D2D links.

As represented in Fig. 6, the D2D user terminals are composed of two main modules: the user module and the SBS module. The *user module* acts exactly as a user terminal in the previous scenario, i.e., it receives data from the SBSs (now from the SBS modules of other users) and feeds it directly to the application layer. For fair comparison with the previous scenario, we do not allow data to be cached within the user module. As a result, if user u caches data at time t to reduce the demand from the MBS of another user u' at time t', $t' > t$, then user u must send this data over the D2D link at time t', and not earlier.

The *SBS module* at the user terminal essentially acts as the SBS terminal in the previous scenario; the main difference is that the SBS module in the u-th user terminal is allowed to download only contents corresponding to its own demand from the MBS, i.e., $d_u(t,u) \triangleq d_u(t)$. This module contains a cache memory of capacity C_u, represented with two different virtual buffers to ease interpretation. The first virtual buffer is used to represent pre-downloaded contents from the MBS associated to the u-th user's demand. The second virtual buffer represents locally cached data from previous demands. When the u-th user serves its demand, at a rate $d_u(t,u)$, to the application layer, data can be cached at a rate $c_u(t,u) \leq d_u(t,u)$. This cached data can be later requested by other users; in particular, $d_u(t,u')$ denotes the demand of user u from user u' at time t, which is served through the D2D links; accordingly, we have $d_u(t,u') = \hat{r}_u(t,u')$. The caching rate associated to demand $d_u(t,u')$ is denoted by $c_u(t,u')$. Finally, $\ell_u(t)$ denotes the rate at which data is removed from the local cache of user u to reduce its own demand, which is used if a file is requested twice by the user. To simplify the notation, in the remainder of the paper we refer to variable $\ell_u(t)$ as $\hat{r}_u(t,u)$. However, note that this data stream does not require D2D resources.

As before, the aim is to jointly design the transmission and caching rates, $\mathbf{r}_M(t) \triangleq (r_u(t))_{u=1}^U$, $\hat{\mathbf{r}}(t) \triangleq ((\hat{r}_u(t,u'))_{u'=1}^U)_{u=1}^U$, and $\mathbf{c}(t) \triangleq ((c_u(t,u'))_{u'=1}^U)_{u=1}^U$, that minimize a general cost function on the rates of the MBS, $\mathbf{r}_M(t)$, and D2D links, $\hat{\mathbf{r}}(t)$. This cost function accounts for: (i) the cost of transmissions from the MBS to different users, $\sum_{u=1}^U g_u(r_u(t))$, with $g_u(r_u(t))$ standing for the instantaneous cost of the u-th link; and (ii) the cost of transmissions over the D2D links, $\sum_{u=1}^U \sum_{u' \neq u} \hat{g}_{uu'}(\hat{r}_u(t,u'))$, where $\hat{g}_{uu'}(\hat{r}_u(t,u'))$ denotes the cost of the D2D link from user u to user u'. Again, we assume that the functions $g_u(\cdot)$, $\hat{g}_{uu'}(\cdot)$, $\forall u,u' \neq u$, are time invariant, convex, increasing, continuously differentiable, and $g_u(0) = 0$, $\hat{g}_{uu'}(0) = 0$, $\forall u,u' \neq u$.[4] As in Section II, different objective functions can be modeled by appropriately selecting the cost functions $g_u(\cdot)$ and $\hat{g}_{uu'}(\cdot)$ (e.g., energy consumption, energy cost, bandwidth, or traffic minimization). As mentioned earlier, the operator must give incentives to users that transmit over the D2D links [15]. In this context, $\hat{g}_{uu'}(\cdot)$ can represent the economical incentive, $\xi_U \geq 0$, paid by the operator to the users for the data transmitted over the D2D links, i.e., $\hat{g}_{uu'}(\hat{r}_u(t,u')) = \xi_U \hat{r}_u(t,u')$. The total incentive of user u is $\int_{t=0}^T \sum_{u' \neq u} \xi_U \hat{r}_u(t,u')$. The total economical cost of the operator is the sum of the cost of energy used by the MBS, $\int_0^T \xi_{MBS} \left(\sum_{u=1}^U W/U(\exp(r_u(\tau)U/W) - 1) \right) \mathrm{d}\tau$, and the incentives paid to the users, $\int_0^T \xi_U \sum_{u=1}^U \sum_{u' \neq u} \hat{r}_u(\tau,u') \mathrm{d}\tau$. Observe that, if this cost function is adopted, no channel state information of the D2D links is required at the MBS.

For problem tractability, we forbid users to simultaneously cache the same data by including the constraint $\sum_{u'=1}^U c_{u'}(t,u) \leq d_u(t), \quad \forall u$. The implications of this assumption are later discussed in Remark 5.

Since we have one dedicated MBS link per user, we need a data departure curve from the MBS to each user u, i.e., $D_u(t,r_u) = \int_0^t r_u(\tau)\mathrm{d}\tau$. The u-th data departure curve is constrained from above by the u-th user cache capacity, C_u (through the maximum data departure curve B_u), and from below by its net demand from the MBS (through the minimum data departure curve A_u).

Next, we derive the expression of the minimum data de-

[4]Notice that by considering a different cost function for each link, it is possible to model, among others, different channel gains.

parture curve for the transmission from the MBS to the u-th user, to fulfill the user's demand $d_u(t)$. Since from the previous assumption the users' cache contents are non-overlapping, the net demand of user u from the MBS (i.e., the demand in point α in Fig. 6) is obtained by subtracting from $d_u(t)$ the sum of the rates in the D2D links to user u and the locally cached data at user u, $\ell_u(t) = \hat{r}_u(\tau, u)$. Thus, the lower bound on $D_u(t, r_u)$ reads as $A_u(t, \hat{\mathbf{r}}) \triangleq \int_0^t d_u(\tau) - \sum_{\forall u'} \hat{r}_{u'}(\tau, u) \mathrm{d}\tau$. The maximum data departure curve at user u, $B_u(t, \mathbf{c}, \hat{\mathbf{r}})$, can be obtained by adding C_u to the minimum data departure curve, $A_u(t, \hat{\mathbf{r}})$, and subtracting the locally cached data, i.e., $B_u(t, \mathbf{c}, \hat{\mathbf{r}}) \triangleq C_u + A_u(t, \hat{\mathbf{r}}) - \int_0^t \sum_{\forall u'} (c_u(\tau, u') - \hat{r}_u(\tau, u')) \mathrm{d}\tau$, where the locally cached data is computed as the integral up to time t of the difference between the data rates entering and leaving the local caching buffer.

Bearing all the above in mind, the problem with optimization variables $\{\mathbf{r}_M(t), \hat{\mathbf{r}}(t), \mathbf{c}(t)\}_{t \in [0,T]}$ is mathematically formulated as follows:

$$\min \int_0^T \sum_{u=1}^{U} \left(g_u(r_u(\tau)) + \sum_{u' \neq u} \hat{g}_{uu'}(\hat{r}_u(\tau, u')) \right) \mathrm{d}\tau \quad (7a)$$

$$\text{s.t.} \quad D_u(t, r_u) \leq B_u(t, \mathbf{c}, \hat{\mathbf{r}}), \qquad \forall u, t \in [0, T], \ (7b)$$

$$D_u(t, r_u) \geq A_u(t, \hat{\mathbf{r}}), \qquad \forall u, t \in [0, T], \ (7c)$$

$$\sum_{u'=1}^{U} c_{u'}(t, u) \leq d_u(t), \qquad \forall u, t \in [0, T], \ (7d)$$

$$c_u(t, u') \leq c_u(\boldsymbol{\rho}(t, u')), \qquad \forall u, u' \neq u, t \in [0, T], \ (7e)$$

$$\hat{r}_u(t, u') \leq c_u(\boldsymbol{\rho}(t, u')), \qquad \forall u, u', t \in [0, T], \ (7f)$$

$$c_u(t, u') \geq 0, r_u(t) \geq 0, \hat{r}_u(t, u') \geq 0, \forall u, u', t \in [0, T], \ (7g)$$

where the constraints in (7b) prevent cache overflows at the users, and those in (7c) impose the fulfillment of the users' demands. The constraints in (7d) impose that the same data cannot be simultaneously cached by different users, and those in (7e) and (7f) restrict the maximum caching rate and D2D transmission rate at user u when serving the requests of other users $u' \neq u$ to the rate locally available in the cache of user u, $c_u(\boldsymbol{\rho}(t, u'))$, respectively. The function $\boldsymbol{\rho}(t, u)$ is defined in the previous section.[5] Finally, the constraints in (7g) impose nonnegative caching and transmission rates, respectively.

Remark 5. For problem tractability, we have included the constraint (7d) of caching non-overlapping data at the users. This assumption is without loss of optimality when the D2D link costs are equal and linear, i.e., $\hat{g}_{uu'}(x) = \xi_U x, \forall u, u' \neq u$, which, as mentioned earlier, characterizes the incentives paid by the operator to users. Then, there exists an optimal solution where users do not cache overlapping contents. Further gains can be achieved by caching overlapping contents in the general case of unequal or non-linear D2D cost functions. However, due to the corresponding combinatorial structure, the optimal solution considering overlapping caching at users is elusive, and is left as an open problem for future work.

[5]The function $\boldsymbol{\rho}(t, u)$ as defined in the previous section is inherently enforcing that if two users u and u' request the same file at a given time slot, then both users have to download the data that is not cached at other users from the MBS (i.e, they cannot help each other in the current slot for this file). However, we could redefine the function $\boldsymbol{\rho}(t, u)$ to allow instantaneous transmissions over the D2D links, by making the request of one user to point the other one, i.e., $\boldsymbol{\rho}(t, u') = [t, u]$; thus, only user u downloads the data from the MBS, which is instantaneously sent to user u' through the D2D link.

B. Jointly optimal strategy

As in Section II, we first derive some structural properties of the caching policy $\mathbf{c}$, the D2D transmission policy $\hat{\mathbf{r}}$, and the data departure curves, $D_u(t, r_u)$, which allow us to reformulate the problem with a finite number of optimization variables. In the following lemma we show that, within each slot, it is optimal that the users cache and transmit data at constant rate.

Lemma 4. *The (not necessarily unique) optimal caching rate and D2D transmission rates in problem* (7) *can be written as a piece-wise constant functions* $c^\star(t, u') = \sum_{n=1}^{N} (q_u^\star(n, u')/T_s) \operatorname{rect}((t - (n - 1/2)T_s)/T_s)$ *and* $\hat{r}_u(t, u') = \sum_{n=1}^{N} (b_u^\star(n, u')/T_s) \operatorname{rect}((t - (n - 1/2)T_s)/T_s)$, *where* $q_u^\star(n, u')$ *is the optimal amount of cached data at user* u *for the request of user* u' *at slot* n, *and* $b_u^\star(n, u')$ *is the optimal amount of transmitted data from user* u *to user* u' *within slot* n.

Proof: The proof follows similarly to the proof of Lemma 2 and is omitted for brevity. ∎

Since the optimal local caching rate and D2D transmission rates are piece-wise constant, we know that the constraints in (7b) and (7c) are piece-wise linear, and the slopes of the constraints can only change at some slot transition. From this, and similarly to Lemma 3, we can prove that the optimal data departure curve for each user can be written as a piece-wise linear function. Thus, the associated transmission rate from the MBS to each user reads as $r_u^\star(t) = \sum_{n=1}^{N} r_{nu}^\star \operatorname{rect}(t - (n - 1/2)T_s/T_s)$, where $r_{nu}^\star$ is the optimal transmission rate from the MBS to user u at the n-th slot.

The problem in (7) can be equivalently rewritten in terms of cached data $q_u(n, u')$, the transmitted data at the D2D links $b_u(n, u')$, and the transmission rates from the MBS to each user at each slot, r_{nu}, as

$$\min \sum_{u=1}^{U} \sum_{n=1}^{N} T_s \left(g_u(r_{nu}) + \sum_{u' \neq u} \hat{g}_{uu'} \left(\frac{b_u(n, u')}{T_s} \right) \right) \quad (8a)$$

$$\text{s.t.} \sum_{\ell=1}^{n} T_s r_{\ell u} \leq C_u + \sum_{\ell=1}^{n} \Big(d_{\ell u} T_s + \sum_{u'=1}^{U} b_u(\ell, u') - b_{u'}(\ell, u) - q_u(\ell, u') \Big), \quad \forall u, n, \ (8b)$$

$$\sum_{\ell=1}^{n} T_s r_{\ell u} \geq \sum_{\ell=1}^{n} d_{\ell u} T_s - \sum_{u'=1}^{U} b_{u'}(\ell, u), \qquad \forall u, n, \ (8c)$$

$$\sum_{u'=1}^{U} q_{u'}(n, u) \leq T_s d_{nu}, \qquad \forall n, u, \ (8d)$$

$$q_u(n, u') \leq q_u(\bar{\boldsymbol{\rho}}(n, u')), \qquad \forall n, u, u' \neq u, \ (8e)$$

$$b_u(n, u') \leq q_u(\bar{\boldsymbol{\rho}}(n, u')), \qquad \forall n, u, u', \ (8f)$$

$$q_u(n, u') \geq 0, b_u(n, u') \geq 0, r_n \geq 0, \qquad \forall n, u, u', \ (8g)$$

where the constraints in (8b)-(8g) stand for the discrete versions of the constraints in (7b)-(7g), respectively. The function $\bar{\boldsymbol{\rho}}$ is defined as in Section II-C.

The problem in (8) is a convex program because the objective function is convex and the constraints are linear. Accordingly, it can be efficiently solved by, e.g., interior point methods.

IV. Numerical results

In this section, we assess the performance of the proposed caching and transmission policies in both scenarios, namely, the *SBS scenario* in Section II and the *D2D scenario* in Section III. We consider $N = 20$ time slots of duration 10 seconds each, and $F = 2000$ video files. The file lengths are uniformly distributed in the interval $[0.3, 150]$ Mnats with mean file length $\mathbb{E}[\ell_j] = 75.15$ Mnats. We assume that the probability of requesting file f_j, θ_j, is independent and identically distributed across time slots and users, and follows the Zipf distribution, i.e., $\theta_j = j^{-\gamma}/(\sum_{q=1}^{|\mathbb{F}|} q^{-\gamma})$. Parameter γ models the skewness of the file popularity; when $\gamma = 0$, popularity is uniform and it becomes more skewed as γ grows [9]. We consider the Shannon power-rate function $g(r(t)) = W(\exp(r(t)/W) - 1)$, where W is the channel bandwidth. In the SBS scenario, the MBS allocates the whole bandwidth $W = 10$ MHz to communicate with the SBS; while in the D2D scenario, the MBS splits evenly the total bandwidth across the U users; as a result the bandwidth in each subchannel is $W = 10/U$ MHz. The cache capacity C is alternatively expressed by means of the percentage over the average requested data per user, i.e., $\hat{C} = (100C)/(N\mathbb{E}\{\ell_j\})$. For the D2D scenario, the total cache capacity is evenly distributed across users, $C_u = C/U$. Unless otherwise stated, we set the number of users to $U = 3$, the Zipf distribution parameter to $\gamma = 1$, and the total cache capacity to $\hat{C} = 10$ ($C = 15.03$Mnats).

We compare the proposed jointly optimal transmission and caching strategies, obtained by solving (2) and (8), with four sub-optimal strategies: the *No caching* strategy that serves as a benchmark for comparison with traditional systems without cache memory; the *Least Recently Used (LRU)* caching algorithm that always keeps in the cache the most recently requested files [29]; the *Pre-Downloading Caching Algorithm (PDCA)* that uses the cache only to pre-download data (see Policy 1 in Section II-A and Fig. 4(a)) [18]; and the *Local Caching Algorithm (LCA)* that exploits only the local caching gain (i.e, the MBS transmits at the net demand rate without allowing pre-downloading). The optimal policy, the PDCA, and the LCA are offline policies as non-causal knowledge of the file demand is required; while the *No caching* and *LRU* strategies are online policies that may depend only on the previous file requests.

First, in Figs. 7-8, we focus on the energy consumption at the MBS considering that the users cooperate altruistically with the MBS (i.e., $\hat{g}_{uu'} = 0, \forall u, u \neq u$). This allows us to fairly compare the SBS and D2D scenarios, and to evaluate the effect of having a centralized or distributed cache. Afterwards, in Fig. 9, we consider the cost of the

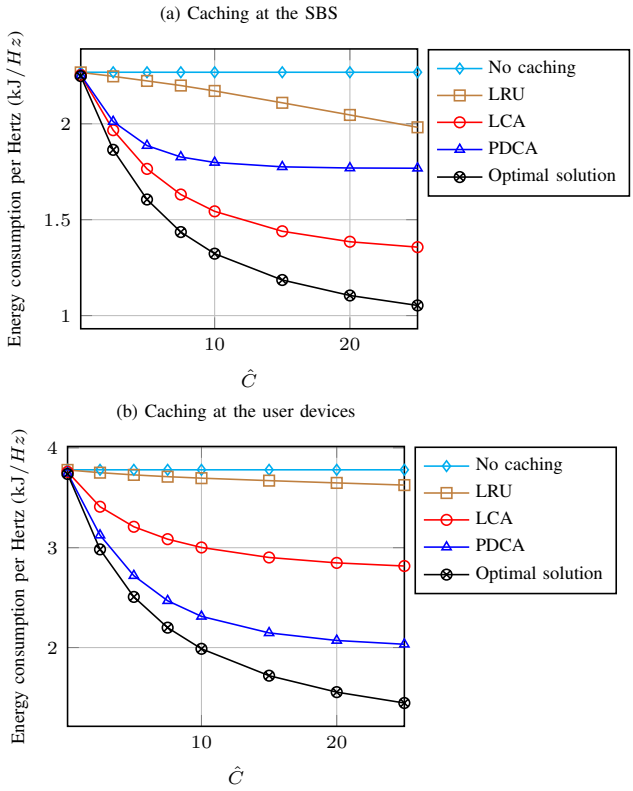


Fig. 7. Energy consumption of the MBS with respect to cache capacity ($\gamma = 1$, $U = 3$). In (a), the files are cached at the SBS as explained in Section II. In (b), the files are cached at the users as explained in Section III.

MBS when users cooperate in exchange of an economical incentive. Figs. 7(a) and 7(b) evaluate the energy consumption of the MBS for different sizes of the cache capacity in the SBS and D2D scenarios, respectively. It is observed that in both scenarios the MBS energy consumption decreases with the cache capacity for all the caching strategies. When the jointly optimal transmission and caching strategy is compared with traditional non-caching solutions, it is observed that the optimal policy reduces the MBS energy consumption by 53.59% in the SBS scenario, and by 61.78% in the D2D scenario. This reduction is obtained when the cache capacity is 25% of the average user traffic, i.e., $\hat{C} = 25$; however, further energy savings can be achieved by increasing the total cache capacity. The performance of the online LRU caching policy is far from the optimal; however, this was expected as it does not exploit any information about the future requests. Next, we assess the performance of policies that exploit only one of the caching gains, namely, pre-downloading (PDCA) or local caching gains (LCA). Interestingly, it is observed in the SBS scenario that LCA achieves more energy savings than PDCA (see Fig. 7(a)); while in the D2D scenario, PDCA requires a lower energy consumption than LCA (see Fig. 7(b)). This different behavior is later argued in the following paragraph as it depends on the value of γ. Finally, if we globally compare the two scenarios, we observe that the deployment of SBSs leads to more reduction in the MBS energy consumption compared to the D2D scenario. The rationale behind this is two-fold. First, the expressions for energy consumption in

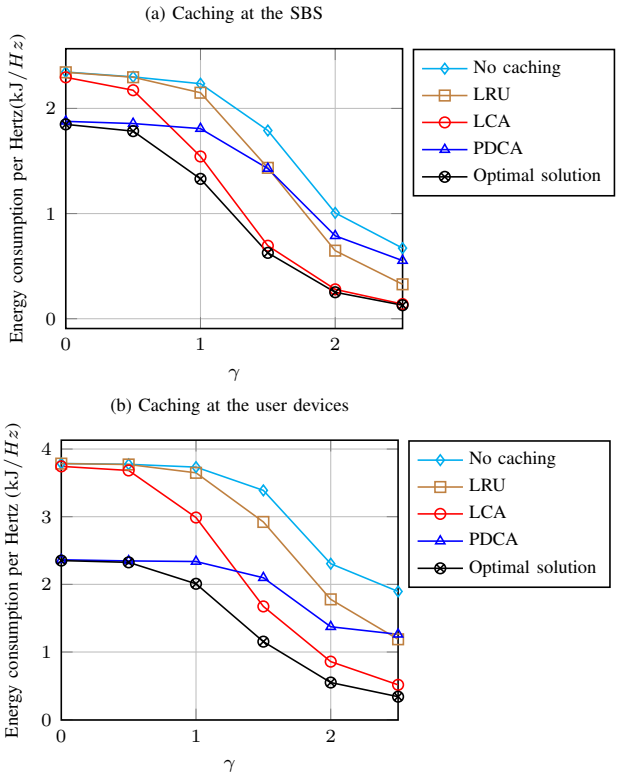


Fig. 8. Energy consumption of the MBS for different values of the Zipf distribution parameter γ ($\hat{C} = 10$, $U = 3$). In (a) the files are cached at the SBS as explained in Section II. In (b) the files are cached at the users as explained in Section III.

the objective functions of problems (2) and (8) are different; specifically, the energy consumption in the SBS scenario is always smaller (or equal) than the consumption in the D2D scenario, which can be proved by using Jensen's inequality. Second, in the D2D scenario the total storage capacity is distributed across users instead of being centralized and, as a result, the energy consumption increases.

Figs. 8(a) and 8(b) evaluate the impact of the file popularity distribution, which is controlled by γ, on the energy consumption of the MBS for the scenarios and algorithms mentioned above. We observe that the energy consumption is dramatically reduced when γ increases (and thus, the popularity distribution becomes skewed) as more file repetitions are encountered. By focusing on the policies *No caching* and *PDCA*, we observe that both follow a similar trend when γ increases; the energy consumption of these policies decreases with γ because it is more likely that two users request the same file within the same time slot. Indeed, for the case of a single user connected to the SBS, $U = 1$, it can be observed that the energy consumption of these policies is not affected by γ. Under a uniform popularity distribution of the files ($\gamma = 0$), PDCA outperforms LRU and LCA as file repetitions are unlikely; however, there is a crossing point between the performances of these policies. Interestingly, this crossing point occurs later in the D2D scenario, and it moves to larger values of γ as the number of users increases. We believe that this is because the more distributed the cache is, the more the local caching gain is penalized. In contrast, the pre-downloading gain is not

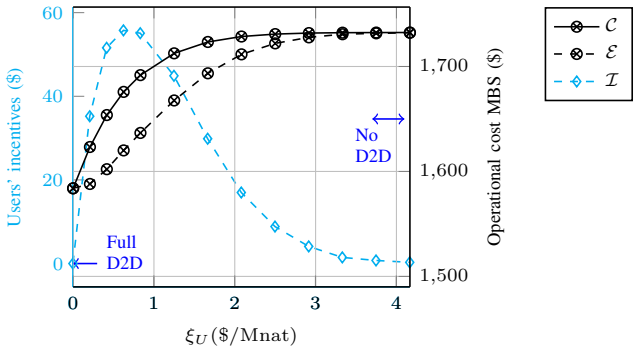


Fig. 9. Economical cost at the MBS for different economical incentives per transmitted data in the D2D links ($U = 3$, $\gamma = 1$, $\hat{C} = 10$, and $\xi_{MBS} = 0.3\$/KWh$).

affected much by distributing the cache across users. Note that the optimal policy outperforms all the other policies by adapting to the best available gain for any value of γ.

In the previous simulations we have considered that users altruistically cooperate with the MBS. Next, we consider that users transmit over the D2D links in exchange of an economical incentive of ξ_U dollars per transmitted data. In this context, we minimize the economical cost at the MBS defined as the sum of the electricity bill, $\mathcal{E}$, and the incentives paid to users $\mathcal{I}$, i.e., $\mathcal{C} = \mathcal{E} + \mathcal{I}$, where $\mathcal{E} = \int_0^T \xi_{MBS} \left(\sum_{u=1}^{U} W/U(\exp(r_u(\tau)U/W) - 1) \right) \mathrm{d}\tau$ and $\mathcal{I} = \int_0^T \xi_U \sum_{u=1}^{U} \sum_{u' \neq u} \hat{r}_u(\tau, u') \mathrm{d}\tau$. We have set $\xi_{MBS} = 0.3$ \$/KWh, which is a typical electricity price. Fig. 9 depicts the costs obtained under the optimal transmission and caching policy as ξ_U varies in the x-axis. The plot $\mathcal{I}$ refers to the left y-axis, whereas the plots $\mathcal{C}$ and $\mathcal{E}$ refer to the right y-axis. The D2D links are used at its greatest possible extent when $\xi_U = 0$ as D2D transmissions do not incur any cost to the MBS. Then, the users incentives first increases with ξ_U until it starts decreasing as the usage of the D2D links is reduced. Finally, for very large values of ξ_U, the MBS serves all the traffic and no data is transmitted over the D2D links.

V. Conclusions

This paper has investigated the opportunities that caching offers to reduce a generic cost function of the transmission rates, e.g., the required energy, bandwidth, or traffic to serve the users. Two different scenarios have been considered where the information is either cached at an SBS, or directly at the user terminals, which then use D2D communications to share the cached contents. It has been shown that, when the transmission and caching policies are jointly designed, the cache offers two possible gains, namely, the *pre-downloading* and *local* caching gains. In both scenarios, the jointly optimal transmission and caching policy has been obtained by demonstrating that constant rate caching within a time slot is optimal, which allows to reformulate the infinite-dimensional optimization problem as a solvable convex program. The numerical results have focused on minimizing the energy consumption at the MBS. It has been shown that the proposed solutions achieve substantial energy savings. Specifically, when the cache capacity is only 25% of the average traffic of a single

user, energy savings of more than 53% have been obtained. It has been observed that the pre-downloading gain is greater than the local caching gain when the file popularity distribution is uniform, and vice versa when the file popularity is skewed. The proposed optimal offline transmission and caching policies can be used as a lower bound to evaluate the cost of any online policy. In particular, it has been observed that, in the considered wireless setting, the popular LRU online algorithm performs far from the optimal strategy as it does not exploit the pre-downloading caching gain. To conclude, our results motivate the design of novel online algorithms that can better approach the performance of the optimal offline solution, which is left for future work. These algorithms can be designed by exploiting the partial knowledge of some of the subsequent file requests (e.g., when some users are watching long video content that span several time slots), or by learning users' daily behaviors.

APPENDIX

In the statement of Lemma 2, we assume that we know the optimal number of data units to be cached for each request, $q^\star_{nu}$. Thus, we also know the optimal value of the maximum and minimum data departure curves at slot transitions, i.e., $b_n \triangleq B(nT_s, \mathbf{c}^\star) = C + \sum_{\ell=1}^{n}\sum_{u=1}^{U} T_s s_{nu} - q^\star_{nu}$ and $a_n \triangleq A(nT_s, \mathbf{c}^\star) = \sum_{\ell=1}^{n}\sum_{u=1}^{U} T_s s_{nu} - q^\star_{\bar{\boldsymbol{\rho}}(n,u)}$ (c.f. Definitions 5 and 6). $\bar{\boldsymbol{\rho}}(n,u)$ is defined after the problem in (2). However, we do not know the actual values of $B(t,\mathbf{c}^\star)$ and $A(t,\mathbf{c}^\star)$ for $t \neq nT_s$ since it depends on the shape of the optimal caching policy.

We first relax the problem in (1) by considering the constraints in (1b) and (1c) only at slot transitions ($t = nT_s$, $\forall n$). Thus, we consider the following relaxed problem:

$$\min_{\{r(t),\mathbf{c}(t)\}_{t\in[0,T]}} \int_0^T g(r(\tau))\mathrm{d}\tau \tag{9a}$$

$$\text{s. t.} \quad a_n \leq D(nTs, r) \leq b_n, \qquad \forall n, \tag{9b}$$

$$r(t) \geq 0, \qquad \forall t \in [0,T], \tag{9c}$$

$$\mathbf{0} \preceq \mathbf{c}(t) \preceq \mathbf{s}(t), \qquad \forall t \in [0,T], \tag{9d}$$

$$B(nT_s, \mathbf{c}) = b_n, A(nT_s, \mathbf{c}) = a_n, \quad \forall n, \tag{9e}$$

where the values of a_n and b_n are known $\forall n$ as argued above.

Note that any caching policy, $\mathbf{c}$, satisfying (9d)-(9e) is optimal to the relaxed problem.

This problem is represented in Fig. 10. Define $\bar{r}^\star(t)$ as the optimal transmission rate to the relaxed problem in (9). The optimal data departure curve, $D(t,\bar{r}^\star)$, is a piece-wise linear function that can be obtained as the tightest string whose ends are tied to $(0,0)$ and $(0,a_N)$. This statement is proved in [22] by using the integral version of the Jensen's inequality and the convexity of the cost function $g(\cdot)$. Accordingly, for any feasible caching policy $\mathbf{c}$, we know that the transmission rate of this relaxed problem might only change at slot transitions. Due to this, the optimal data departure curve to the problem in (9), $D(t,\bar{r}^\star)$, satisfies $\bar{A}(t) \leq D(t,\bar{r}^\star) \leq \bar{B}(t)$, $\forall t \in [0,T]$, where $\bar{A}(t)$ is the piece-wise linear curve obtained by joining the points (nT_s, a_n) for $n = 0,\ldots,N$; and $\bar{B}(t)$ is the piece-wise linear curve obtained by joining the points (nT_s, b_n) for $n = 0,\ldots,N$ (see Fig. 10).

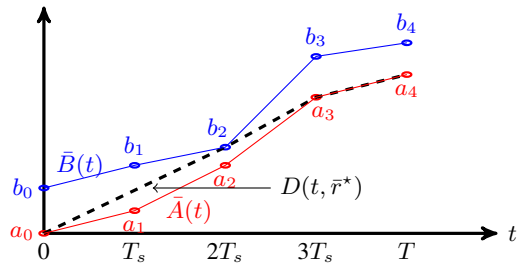


Fig. 10. Representation of the relaxed problem in (9).

Next consider the caching policy that caches each file at a constant rate, $\mathbf{c}^\star$, as defined in Lemma 2. First note that $\mathbf{c}^\star$ satisfies (9d)-(9e); and thus, it is an optimal caching policy of the relaxed problem in (9). Additionally, the maximum and minimum data departure curves associated to the caching policy $\mathbf{c}^\star$ are piece-wise linear and satisfy

$$A(t,\mathbf{c}^\star) = \bar{A}(t) \leq D(t,\bar{r}^\star) \leq \bar{B}(t) = B(t,\mathbf{c}^\star), \forall t \in [0,T]. \tag{10}$$

Thus, $\{\bar{r}^\star, \mathbf{c}^\star\}$ is the optimal solution to the relaxed problem in (9).

Note that, from (10), the pair $\{\bar{r}^\star, \mathbf{c}^\star\}$ satisfies the constraints that had been relaxed in the original problem in (1). Accordingly, since $\{\bar{r}^\star, \mathbf{c}^\star\}$ is a feasible solution to the original problem, and the objective function is the same in both problems, it is also an optimal solution. ■

REFERENCES

[1] "Mobile traffic forecasts 2010- 2020," Tech. Rep. UMTS Forum, MSU-CSE-00-2, Jan. 2011.

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

[3] S. Goebbels and R. Jennen, "Enhancements in wireless broadband networks using smart caching an analytical evaluation," in *Proceedings of the IEEE Int'l Symp on Personal, Indoor and Mobile Radio Commun.*, Sep. 2008, pp. 1–5.

[4] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proceedings of the IEEE Conf. on Computer Commun.*, Apr. 2015, pp. 936–944.

[5] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[6] 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.

[7] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Multicast-aware caching for small cell networks," in *Proceedings of the IEEE Wireless Commun. and Netw. Conf.*, Apr. 2014, pp. 2300–2305.

[8] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[9] P. Blasco and D. Günd"uz, "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.

[10] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Cache-aware user association in backhaul-constrained small cell networks," in *Proceedings of the IEEE Modeling and Optimization in Mobile, Ad Hoc, and Wireless Net.*, 2014, pp. 37–42.

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

[12] N. Golrezaei, P. Mansourifard, A. Molisch, and A. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[13] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *arXiv preprint arXiv:1502.03124*, 2015.

[14] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[15] P. Li and S. Guo, "Incentive mechanisms for device-to-device communications," *IEEE Netw.*, vol. 29, no. 4, pp. 75–79, Jul. 2015.

[16] P. Ostovari, A. Khreishah, and J. Wu, "Cache content placement using triangular network coding," in *Proceedings of the IEEE Wireless Commun. and Netw. Conf.*, Apr. 2013, pp. 1375–1380.

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

[18] 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.

[19] M. Dräxler, J. Blobel, P. Dreimann, S. Valentin, and H. Karl, "Anticipatory buffer control and quality selection for wireless video streaming," in *Int. Conf. on Networked Systems*, Mar. 2015.

[20] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Trans. Netw.*, vol. PP, no. 99, pp. 1–10, 2015.

[21] M. Gregori, J. Gómez-Vilardebó, J. Matamoros, and D. Gündüz, "Joint transmission and caching policy design for energy minimization in the wireless backhaul link," in *Proceedings of the IEEE Int'l Symp on Inform Theory*, 2015, pp. 1004–1008.

[22] 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.

[23] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "In-network caching and content placement in cooperative small cell networks," in *Proceedings of the IEEE Int'l Conf. on 5G for Ubiquitous Connectivity*, 2014, pp. 128–133.

[24] H. O. Fattorini, *Infinite Dimensional Optimization and Control Theory*. Cambridge University Press, 1999, vol. 54.

[25] 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.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ Press, 2004.

[27] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*, ser. Athena Scientific Optimization and Computation Series. Athena Scientific, 2003. [Online]. Available: http://books.google.es/books?id=DaOFQgAACAAJ

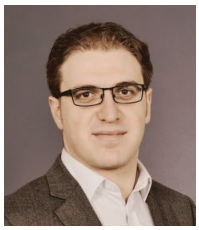
[28] A. Nedic and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1757–1780, Feb. 2009.

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

Maria Gregori received the B.Sc., M.Sc., and Ph.D. degrees in telecommunications from the Universitat Politècnica de Catalunya (UPC), Barcelona, in 2009, 2011, and 2014, respectively. From September 2008 until June 2009, she hold a visiting researcher position at the Georgia Institute of Technology with a fellowship from the Vodafone Foundation. From November 2009 until September 2010, she worked as research assistant at Intel-UPC. From September 2010 until September 2014, she pursued her Ph.D. studies at Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). During her PhD, she hold a visiting researcher position at the Hong Kong University of Science and Technology from September 2013 until February 2014. She received predoctoral grants from the Generalitat de Catalunya and from the CTTC fellowship program. Since October 2014, she holds a researcher position at CTTC. Dr. Gregori received the NEWCOM# 2013 Best young researcher's paper award and was designated exemplary reviewer of the IEEE Wireless Communications Letters in 2013. She participated in the National and European projects INTENSYV, E-CROPS, GRE3N, P2P-SmarTest, and NEWCOM#. She served as chair of the IEEE WCNC 2016 Workshop on Wireless Powered Communication Networks: From Theory to Industry Challenges. Her primary research interests are in the fields of energy harvesting for wireless communications and energy-aware communication system design.

Jesús Gómez-Vilardebó received his M.Sc. and Ph.D. degrees in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC) in October 2003 and July 2009, respectively. In September 2005, he was granted by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) to obtain the Ph. D. on Signal Theory and Communications at the UPC. He is now with the CTTC holding a Research Associate position. His current research interests include information theory, stochastic signal processing, and their applications in wireless multi-user communications and information privacy.

Javier Matamoros holds a researcher position at CTTC. He received the M.Sc. degree in Telecommunications and the Ph.D. degree in signal theory and communications from the Polytechnic University of Catalonia in 2005 and 2010, respectively. He has participated in several National and EC-funded projects (JUNTOS, NEWCOM#, E2SG, EXALTED, ADVANTAGE). He is now the principal investigator at CTTC of the H2020 project P2P-SmarTest and the national project INTENSYV. He is active in several research topics of current interest: smart grid, distributed signal processing, compressive sensing, wireless sensor networks and energy harvesting for wireless communications.



Deniz Gündüz [S'03-M'08-SM'13] received the B.S. degree in electrical and electronics engineering from METU, Turkey in 2002, and the M.S. and Ph.D. degrees in electrical engineering from NYU Polytechnic School of Engineering in 2004 and 2007, respectively. After his PhD he served as a postdoctoral research associate at the Department of Electrical Engineering, Princeton University, and as a consulting assistant professor at the Department of Electrical Engineering, Stanford University. He was a research associate at CTTC in Spain until September 2012, when he joined the Electrical and Electronic Engineering Department of Imperial College London, UK, as a Lecturer. He also held a visiting researcher position at Princeton University from November 2009 until November 2011.

Dr. Gunduz is an Associate Editor of the IEEE Transactions on Communications, and an Editor of the IEEE Journal on Selected Areas in Communications (JSAC) Series on Green Communications and Networking. He is the recipient of a Starting Grant of the European Research Council (ERC), the 2014 IEEE Communications Society Best Young Researcher Award for the Europe, Middle East, and Africa Region, and the Best Student Paper Award at the 2007 IEEE International Symposium on Information Theory (ISIT).

He is the General Co-chair of the 2016 IEEE Information Theory Workshop. He served as a Technical Program Chair of the Network Theory Symposium at the 2013 and 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), and General Co-chair of the 2012 IEEE European School of Information Theory (ESIT). His research interests lie in the areas of communication theory and information theory with special emphasis on joint source-channel coding, multi-user networks, energy efficient communications and privacy in cyber-physical systems.