entropy

MDPI

Article

# Content Delivery in Fog-Aided Small-Cell Systems with Offline and Online Caching: An Information—Theoretic Analysis

**Seyyed Mohammadreza Azimi [1,*], Osvaldo Simeone [2] and Ravi Tandon [3]**

[1] CWiP, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
[2] Centre for Telecommunications Research, Department of Informatics, King's College London, London WC2R 2LS, UK; osvaldo.simeone@kcl.ac.uk
[3] Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721, USA; tandonr@email.arizona.edu
* Correspondence: sa677@njit.edu

Received: 28 May 2017; Accepted: 14 July 2017; Published: 18 July 2017

**Abstract:** The storage of frequently requested multimedia content at small-cell base stations (BSs) can reduce the load of macro-BSs without relying on high-speed backhaul links. In this work, the optimal operation of a system consisting of a cache-aided small-cell BS and a macro-BS is investigated for both offline and online caching settings. In particular, a binary fading one-sided interference channel is considered in which the small-cell BS, whose transmission is interfered by the macro-BS, has a limited-capacity cache. The delivery time per bit (DTB) is adopted as a measure of the coding latency, that is, the duration of the transmission block, required for reliable delivery. For offline caching, assuming a static set of popular contents, the minimum achievable DTB is characterized through information-theoretic achievability and converse arguments as a function of the cache capacity and of the capacity of the backhaul link connecting cloud and small-cell BS. For online caching, under a time-varying set of popular contents, the long-term (average) DTB is evaluated for both proactive and reactive caching policies. Furthermore, a converse argument is developed to characterize the minimum achievable long-term DTB for online caching in terms of the minimum achievable DTB for offline caching. The performance of both online and offline caching is finally compared using numerical results.

**Keywords:** edge caching; interference channel; information theory; latency; cloud RAN

## 1. Introduction

*Edge* or *femto-caching* relies on the storage of popular multimedia content at small-cell base stations (BSs) of a cellular system. This approach has been widely studied in recent years as a means to deliver video files with reduced latency and limited overhead on backhaul connections to the "cloud" [1,2]. Caching at the edge can be seen as an instance of fog networking, whereby storage, computing and communication capabilities are moved closer to the end users [2]. Edge caching has been initially studied for wireless channel models in which small-cell BSs and macro-BSs cannot coordinate their transmissions and hence cannot cooperatively manage their mutual interference (see [1,2] and references therein). In contrast, recent work in [3,4] addresses the possibility of interference management among edge nodes, such as small-cell and macro-BSs, based on the respective cached contents.

The papers [3,4] proposed caching and transmission schemes that enables coordination and cooperation at the BSs based on the cached contents for a system with three BSs and three users.

The performance of these schemes was evaluated in terms of the information-theoretic high signal-to-noise ratio (SNR) metric of the degrees of freedom (DoF), or, more precisely, of its inverse, as a function of the cache capacity of the BSs. More recent research in [5] provided an operational meaning for the inverse of the degrees of freedom metric used in [3,4] in terms of delivery latency, and derived a lower bound on the resulting metric, known as Normalized Delivery Time (NDT), for a general system with any number of BSs and users. The delivery coding latency, henceforth delivery latency, measures the duration of the transmission block. A scenario in which both BSs and users have cache storage is considered in [6,7] under one-shot linear transmission and in [8] under several transmission schemes for both centralized and decentralized caching strategies. It is proved that both BSs and users' caches have the same quantitative contribution to the achievable sum-DoF. Naderializadeh et al. [9] prop osed a universal scheme for content placement and delivery which is independent of underlying communication networks and is order-optimal in the high-SNR regime. In [10], upper and lower bounds on the NDT of cache-aided MIMO interference channels are provided.

In [11,12] the analysis in [3–5] was generalized to study a system in which a cloud server is connected to the BSs via finite-capacity backhaul links and can compensate for partial caching of the library of files at the BSs. This system was referred to as *Fog-Radio Access Networks* (F-RAN). The minimum NDT latency metric was characterized within a multiplicative factor of 2 in [12] as a function of the cache and backhaul capacity by developing achievability and converse arguments. Other works on NDT characterization include [13–16]. In [13], a scenario with a multicast fronthaul is studied. In [14], decentralized content placement and file delivery are considered for a F-RAN system with caching at both BSs and users. Reference [15] studies the achievable NDT region to account for heterogenous requirements on the delivery of different files. In [16], the NDT performance of F-RAN systems is considered within a set-up characterized by a time-varying set of popular files. Reference [17] characterized the delivery time per bit of a cache-aided small-cell system by considering binary fading interference channels. Kakar et al. [18] considered the set-up in [17] under linear deterministic channel model to provide upper and lower bounds on the NDT. The optimization of linear processing and often signal processing aspects of F-RAN systems are considered in [19–23].

In this work, we consider the F-RAN model in Figure 1, which includes a small-cell BS and a macro-BS, represented by Encoder 1 and Encoder 2, respectively. The small-cell BS (Encoder 1) is equipped with a cache of finite capacity and can serve a small-cell mobile user, represented by Decoder 1. The macro-BS (Encoder 2) can serve a macro-cell user, namely Decoder 2, as well as, possibly, also Decoder 1. The transmission from the macro-BS (Encoder 2) to Decoder 2 interferes with Decoder 1. It is assumed that the small-cell BS transmits with sufficiently small power so as not to create interference at Decoder 2, which is modeled here as a partially connected wireless channel. We investigate both scenarios with offline and online caching.

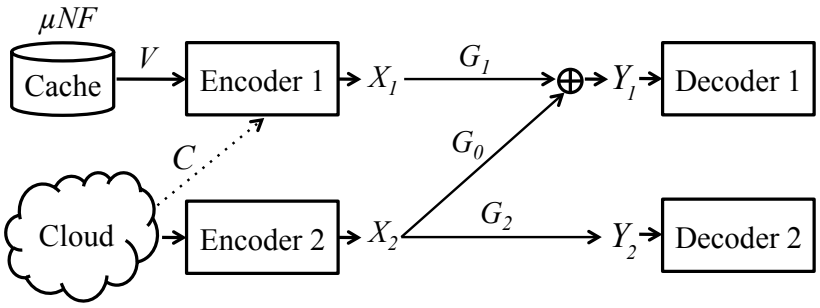


**Figure 1.** Cloud and edge-aided data delivery over binary fading interference channels.

The main contributions of this article are as follows:

- An information-theoretic formulation for the analyses of the system in Figure 1 is presented that centers on the characterization of the minimum delivery coding latency measured in terms of

the Delivery Time per Bit (DTB), for both offline and online caching. The system model is based on a one-sided interference channel.
- Assuming a fixed set of popular contents, the minimum DTB for the system in Figure 1 is obtained as a function of the cache capacity at Encoder 1 and the capacity of the backhaul link that connects the cloud to Encoder 1 in the offline setting.
- Online caching and delivery schemes based on both reactive and proactive caching principles (see, e.g., [2]) are proposed in the presence of a time-varying set of popular files, and bounds on the corresponding achievable long-term DTBs are derived.
- A lower bound on the achievable long-term DTB is obtained, which is a function of the time-variability of the set of popular files. The lower bound is then utilized to compare the achievable DTBs under offline and online caching.
- Numerical results are provided in which the DTB performance of reactive and proactive online caching schemes is compared with offline caching. In addition, different eviction mechanisms, such as random eviction, Least Recently Used (LRU) and First In First Out (FIFO) (see, e.g., [24]), are evaluated.

The rest of the paper is organized as follows. In Section 2 we present the system model for offline caching, including the definition of the key performance metric of DTB. The minimum DTB for offline caching is then derived and discussed in Section 3. The online caching scenario for the system in Figure 1 is studied in Section 4 in terms of the long-term DTB metric. The comparison between online and offline caching is explored in Section 5. Numerical results are provided in Section 6 and, finally, Section 7 concludes the paper. This work was presented in part in [17].

**Notation 1.** *Given $a > 0$, we define the set $[a] = \{1, 2, ..., \lceil a \rceil\}$. For any probability $p$, we define $\bar{p} = 1 - p$.*

## 2. System Model for Offline Caching

In this section, we study the fog-aided system depicted in Figure 1. We consider a static library of $N$ files denoted by $\mathcal{L} = \{W_1, ..., W_N\}$. Each file is independent and identically distributed according to uniform distribution, so that we have $W_i \sim \mathcal{U}([2^F])$, for $i \in [N]$, where $F$ is the file size in bits. Encoder 1, which models a small-cell BS, has a local cache and is able to store $\mu NF$ bits. The parameter $\mu$, with $0 \leq \mu \leq 1$, is hence the fractional cache size and represents the portion of library that can be stored at the cache. Encoder 2, which models a macro-BS, can access the entire library $\mathcal{L}$ thanks to its direct connection to the cloud. Encoder 1 is also connected to the cloud but only through a rate-limited link of capacity $C$ bits per channel use. We will first consider the scenario of edge-aided offline caching in which $C = 0$, i.e., Encoder 1 does not have access to the cloud, and then extend the analysis to cloud and edge-aided offline caching, i.e., when $C \geq 0$.

It is assumed that encoders and decoders are connected by a binary fading interference channel, previously studied in [17,25,26]. This model represents a special case of the deterministic linear model of [27] as generalized to account for random fading (see [28]). As illustrated in Figure 1, the signal received at Decoder 1 and Decoder 2 at time $t$ can be written as

$$\begin{aligned} Y_1(t) &= G_1(t)X_1(t) \oplus G_0(t)X_2(t) \\ Y_2(t) &= G_2(t)X_2(t), \end{aligned} \tag{1}$$

where $\mathbf{G}(t) = (G_0(t), G_1(t), G_2(t)) \in \{0,1\}^3$ is the vector of binary channel coefficients at time $t$, and $X_1(t)$ and $X_2(t)$ are the binary transmitted signals from Encoder 1 and Encoder 2, respectively. In (1), all operations are in the binary field. The channel gains are distributed as $G_1(t) \sim \text{Bernoulli}(\epsilon_1)$ and $G_0(t), G_2(t) \sim \text{Bernoulli}(\epsilon_2)$, are mutually independent and change independently over time. The parameters $\epsilon_1$ and $\epsilon_2$ describes the average quality of the communication links originating at Encoder 1 and Encoder 2, respectively, and are hence in practice related to the transmission powers of Encoder 1 and Encoder 2. We remark that a more general model with different erasure probabilities for

the links $G_0(t)$ and $G_2(t)$ could also be considered but at the expense of a more cumbersome notation and analysis, which is not further pursued here.

Each user, or decoder, $k$ requests a file $W_{d_k}$ from the library $\mathcal{L}$ at every transmission interval for $k = 1, 2$. The demand vector is defined as $\mathbf{d} = (d_1, d_2) \in [N]^2$. In the next two subsections, we described first the edge-aided scenario and then we generalize it to the cloud and edge-aided system.

### 2.1. Edge-Aided Offline Caching

The edge-aided small-cell system corresponds to the case with $C = 0$ in Figure 1. The system operates according to the following two phases.

(1) *Placement phase*: The placement phase is defined by functions $\phi_i(\cdot)$, at Encoder 1, which maps each file $W_i \in \mathcal{L}$ to its cached version $V_i$

$$V_i = \phi_i(W_i) \quad \forall i \in \{1, ..., N\}. \tag{2}$$

To satisfy cache storage constraint, it is required that

$$H(V_i) \leq \mu F. \tag{3}$$

The total cache content at Encoder 1 is given by

$$V = (V_1, ..., V_N). \tag{4}$$

Note that, as in [5,11], we concentrate on caching strategies that allow for arbitrary intra-file coding but not for inter-file coding as per (2). Furthermore, the caching policy is kept fixed over multiple transmission intervals and is thus independent of the receivers' requests and of the channel realizations in the transmission intervals.

(2) *Delivery phase*: The delivery phase is in charge of satisfying the given request vector $\mathbf{d}$ in each transmission interval given the current channel realization. We assume the availability of full Channel State Information (CSI) throughout the transmission block for simplicity of exposition, although this is not required by achievable schemes that will be proven to be optimal (see Remark 1). Note that in practice non-causal CSI for the coding block can be justified for multi-carrier transmission schemes, such as OFDM, in which index $t$ runs over the subcarriers. It is defined by the following two functions.

- *Encoding*: Encoder 1 uses the encoding function

$$\psi_1 : [2^{\mu NF}] \times [N]^2 \times \{0, 1\}^{3T} \to \{0, 1\}^T \tag{5}$$

which maps the cached content $V$, the demand vector $\mathbf{d}$ and the CSI sequence $\mathbf{G}^T = (\mathbf{G}(1), ..., \mathbf{G}(T))$ to the transmitted codeword $X_1^T = (X_1[1], ..., X_1[T]) = \psi_1(V, \mathbf{d}, \mathbf{G}^T)$. Note that $T$ represents the duration of transmission in channel uses. Encoder 2 uses the following encoding function

$$\psi_2 : [2^{NF}] \times [N]^2 \times \{0, 1\}^{3T} \to \{0, 1\}^T \tag{6}$$

which maps the library $\mathcal{L}$ of all files, the demand vector $\mathbf{d}$, and the CSI vector $\mathbf{G}^T$ to the transmitted codeword $X_2^T = (X_2[1], ..., X_2[T]) = \psi_2(\mathcal{L}, \mathbf{d}, \mathbf{G}^T)$.

- *Decoding*: Each decoder $j \in \{1, 2\}$ is defined by the following mapping

$$\eta_j : \{0, 1\}^T \times [N]^2 \times \{0, 1\}^{3T} \to [2^F] \tag{7}$$

which outputs the detected message $\hat{W}_{d_j} = \eta_j(Y_j^T, \mathbf{d}, \mathbf{G}^T)$ where $Y_j^T = (Y_j(1), ..., Y_j(T))$ is the received signal (1) at receiver $j$.

We refer to a selection of caching, encoding, and decoding functions in (5)–(7) as a policy. The probability of error is evaluated with respect to the worst-case demand vector and decoder as

$$P_e^F = \max_{\mathbf{d}\in[N]^2} \max_{j\in\{1,2\}} \Pr(\hat{W}_{d_j} \neq W_{d_j}). \tag{8}$$

The delivery time per bit (DTB) of a code is defined as $T/F$ and is measured in channel symbols per bit. A DTB $\delta$ is said to be *achievable* if there exists a sequence of policies indexed by the file size $F$ for which the limits

$$\lim_{F\to\infty} \frac{T}{F} = \delta(\mu) \tag{9}$$

and $P_e^F \to 0$ as $F \to \infty$ hold. The *minimum DTB* $\delta^*(\mu)$ is the infimum of all achievable DTB when the fractional cache capacity at Encoder 1 is equal to $\mu$.

### 2.2. Cloud and Edge-Aided Offline Caching

In this section, we generalize the model described above to the case in which there is a link with capacity $C \geq 0$ between Cloud and Encoder 1. The content placement phase is the same as Section 2.1. In the delivery phase, the Cloud implements an encoding function

$$\psi_C : [2^{NF}] \times [N]^2 \times \{0,1\}^{3T} \to [2^{T_C C}], \tag{10}$$

which maps the library $\mathcal{L}$ of all files, the demand vector $\mathbf{d}$ and the CSI vector $\mathbf{G}^T$ to the signal $U^{T_C} = (U_1, ..., U_{T_C}) = \psi_C(\mathcal{L}, \mathbf{d}, \mathbf{G}^T)$ to be delivered to Encoder 1. Here, parameter $T_C$ represents the duration of the transmission from Cloud to Encoder 1 in terms of number of channel uses of the fading channel from encoders to decoders. We have the inequality $H(U_i) \leq C$ for $i \in [T_C]$ by the capacity limitations on the Cloud-to-Encoder 1 link. Furthermore, Encoder 1 uses the encoding function

$$\psi_1 : [2^{\mu NF}] \times [2^{T_C C}] \times [N]^2 \times \{0,1\}^{3T} \to \{0,1\}^T, \tag{11}$$

which maps the cached content $V$, the received signal $U^{T_C}$, the demand vector $\mathbf{d}$ and the CSI sequence $\mathbf{G}^T = (\mathbf{G}(1), ..., \mathbf{G}(T))$ to the transmitted codeword $X_1^T = (X_1[1], ..., X_1[T]) = \psi_1(V, U^{T_C}, \mathbf{d}, \mathbf{G}^T)$. Note that, as for the edge-aided case, we assume non-causal CSI at both cloud and edge for simplicity of exposition. As discussed, this is a sensible assumption for multi-carrier modulation schemes. However, as indicated in Remark 2, it will be proven that the optimal strategy requires only causal CSI at the encoders and no CSI at the cloud. As above, $T$ represents the duration of transmission on the binary fading channel in channel uses.

Decoding and probability of error are defined as in Section 2.1. Instead, a DTB $\delta$ is said to be achievable if there exists a sequence of policies, defined by (2), (6), (7), (10) and (11) and indexed by $F$, such that the limits:

$$\lim_{F\to\infty} \frac{T + T_C}{F} = \delta(\mu, C) \tag{12}$$

and $P_e^F \to 0$ as $F \to \infty$ hold. The *minimum DTB* $\delta^*(\mu, C)$ is the infimum of all achievable DTBs when the fractional cache size at Encoder 1 is equal to $\mu$ and the Cloud-to-Encoder 1 capacity is equal to $C$.

## 3. Minimum DTB under Offline Caching

In this section, we first characterize minimum DTB for edge-aided under offline caching scenario. Then, we derive the minimum DTB for the cloud and edge-aided system.

### 3.1. Edge-Aided System ($C = 0$)

In this subsection, we derive the minimum DTB $\delta^*(\mu)$ for the system in Figure 1 by assuming $C = 0$.

**Proposition 1.** *The minimum DTB for the cache and cloud-aided system in Figure 1 with $C = 0$ is*

$$\delta^*(\mu) = \begin{cases} \frac{2-\mu}{1-\epsilon_2^2} & \text{if } \mu \leq \mu_0 \\ \delta_0 & \text{if } \mu \geq \mu_0, \end{cases} \tag{13}$$

*where $\mu_0$ and $\delta_0$ are given by*

$$\mu_0 = \begin{cases} 1-\epsilon_2 & \text{if } \bar{\epsilon}_1\epsilon_2 > \bar{\epsilon}_2^2\epsilon_1 \\ \frac{2(1-\epsilon_1)(\epsilon_2^2-\epsilon_2+1)}{2-\epsilon_1-\epsilon_2+\epsilon_1\epsilon_2-\epsilon_1\epsilon_2^2} & \text{if } \bar{\epsilon}_1\epsilon_2 \leq \bar{\epsilon}_2^2\epsilon_1 \end{cases} \tag{14}$$

*and*

$$\delta_0 = \max\Big(\frac{1}{1-\epsilon_2}, \frac{2}{2-\epsilon_1-\epsilon_2+\epsilon_1\epsilon_2-\epsilon_1\epsilon_2^2}\Big). \tag{15}$$

**Proof.** The converse is presented in Appendix A, and the achievable scheme is presented next. □

To provide some insights obtained from the result in Proposition 1, consider first the set-up in which Encoder 1 has no caching capabilities, i.e., $\mu = 0$. In this case, Encoder 2 needs to deliver the requested files to both decoders on a binary erasure broadcast channel. Considering the worst-case in which two different files are requested by two decoders, the minimum average time to serve both users is $T = 2F/(1-\epsilon_2^2)$, since with probability $(1-\epsilon_2^2)$ a bit can be delivered to either Decoder 1 or Decoder 2 by Encoder 2, yielding a minimum DTB of $\delta^*(0) = 2/(1-\epsilon_2^2)$. In contrast, when the entire library is available at Encoder 1, i.e., $\mu = 1$, depending on the relative values of $\epsilon_1$ and $\epsilon_2$, two different cases should be distinguished. Roughly speaking, if the channel between Encoder 2 and the Decoders is weaker on average than the channel between Encoder 1 and Decoder 1, or more precisely if $\bar{\epsilon}_1 \geq \bar{\epsilon}_2$, then the minimum DTB is limited by transmission delay to Decoder 2 and the minimum DTB is $\delta^*(1) = 1/(1-\epsilon_2)$. Instead, when the channel between Encoder 1 and Decoder 1 is weaker on average than the channel between Encoder 2 and both decoders, or $\bar{\epsilon}_1 \leq \bar{\epsilon}_2$, the resulting minimum DTB depends on both $\epsilon_1$ and $\epsilon_2$. In both cases, Encoder 2 serves a fraction $(1-\mu_0)$ of the requested file to Decoder 1, so that Encoder 1 only needs to deliver a fraction $\mu_0$ of the requested file by Decoder 1.

As will be detailed below, a key element of the transmission policies is that, in the channel state in which all three links are active, the presence of the cache at Encoder 1 allows the latter to coordinate its transmission with Encoder 2 and cancel the interference caused by Encoder 2 to Decoder 1. Furthermore, from the discussion above, a fractional cache size $\mu \geq \mu_0$ is sufficient to achieve the same DTB $\delta_0$ as with full caching. Figure 2 shows the value $\mu_0$ as a function of $\epsilon_1$ for different values of $\epsilon_2$. It is observed that, for fixed $\epsilon_2$, the fraction $\mu_0$ decreases with $\epsilon_1$, showing that an Encoder 1 with a low channel quality cannot benefit from a large cache size. Furthermore, as the channel from Encoder 2 becomes more reliable, i.e., for small $\epsilon_2$, a larger cache at Encoder 1 enables the latter to coordinate more effectively with Encoder 2, hence improving the DTB.

**Remark 1.** *The achievable schemes proposed above only require the encoders to know the current state of the CSI, i.e., at each time $t$, only the CSI $\mathbf{G}(t)$ is needed. As a result, even if the encoders know only the current CSI, as well as the CSI statistics, the optimal performance is the same as for the case in which the entire sequence $\mathbf{G}^T$ is known as per definition (5) and (6).*

Proof of Achievability

Here, we provide details on the policies that achieve the minimum DTB identified in Proposition 1. We start by proving that the minimum DTB $\delta^*(\mu)$ is a convex function of $\mu$. The proof leverages the splitting of files into subfiles delivered using different strategies via time sharing.

**Lemma 1.** *The minimum DTB $\delta^*(\mu)$ is a convex function of $\mu \in [0,1]$.*

**Proof.** Consider two policies that require fractional cache sizes $\mu_1$ and $\mu_2$ and achieve DTBs $\delta_1$ and $\delta_2$, respectively. Given a fractional cache size $\mu = \alpha\mu_1 + (1-\alpha)\mu_2$ for any $\alpha \in [0,1]$, the system can operate by splitting each file into two parts, one of size $\alpha F$ and the other of size $(1-\alpha)F$, while satisfying the cache constraints. The first fraction of the files is delivered following the first policy, while the second fraction is delivered using the second policy. Since the delivery time is additive over the two file fractions, the DTB $\delta = \alpha\delta_1 + (1-\alpha)\delta_2$ is achieved. □

By the convexity of $\delta^*(\mu)$ proved in Lemma 1, it suffices to prove that the corner points ($\mu = 0$, $\delta^*(0) = 2/(1-\epsilon_2^2)$) and ($\mu = \mu_0, \delta_0$) are achievable. In fact, the minimum DTB $\delta^*(\mu)$ can then be achieved, following the proof of Lemma 1, by file splitting and time sharing between the optimal policies for $\mu = 0$ and $\mu = \mu_0$ in the interval $0 \le \mu \le \mu_0$ and by using the optimal policy for $\mu = \mu_0$ in the interval $\mu_0 \le \mu \le 1$ (see Figure 3).

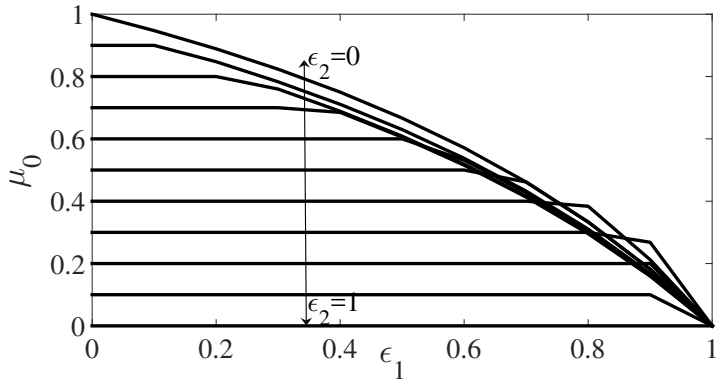


**Figure 2.** Optimum fractional cache size $\mu_0$ as a function of $\epsilon_1$ for different values of $\epsilon_2$, which ranges from 0 to 1 with step size 0.1.

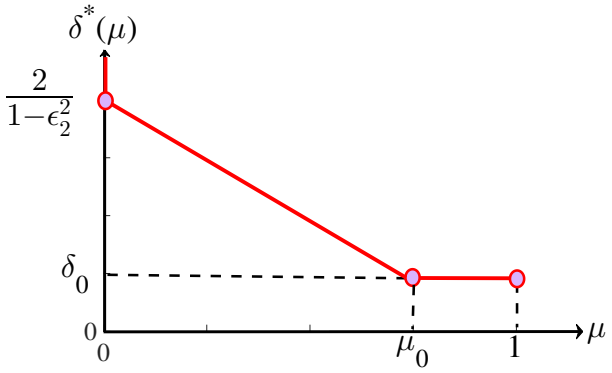


**Figure 3.** Minimum Delivery Time per Bit (DTB) $\delta^*(\mu)$ for the system in Figure 1 with $C = 0$.

In the following, we use the notation $(g_0, g_1, g_2) \in \{0,1\}^3$ to identify the channel realization $(G_0 = g_0, G_1 = g_1, G_2 = g_2)$. For instance, $(0,1,1)$ represents the channel realization in which $Y_1 = X_1$ and $Y_2 = X_2$, and $(1,0,1)$ that in which $Y_1 = X_2$ and $Y_2 = X_2$.

- **No Caching ($\mu = 0$)**: We first consider the corner point $(\mu = 0, \delta^*(0) = 2/(1-\epsilon_2^2))$. In this setting, in which Encoder 1 has no caching capabilities, the model reduces to a broadcast erasure channel from Encoder 2 to both decoders. The worst-case demand vector is any one in which the decoders request different files. In fact, if the same file is requested, it can always be treated as two distinct files achieving the same latency as for a scenario with distinct files. Focusing on this worst-case scenario, we adopt the following delivery policy.

Encoder 1 always transmits $X_1 = 0$. Encoder 2 transmits 1 bit of information to Decoder 1 in the states $(1,0,0)$ and $(1,1,0)$, in which the channel from Encoder 2 to Decoder 1 is on while the channel to Decoder 2 is off. It transmits 1 bit of information to Decoder 2 in the states $(0,0,1)$ and $(0,1,1)$, in which the channel to Decoder 2 is on while the channel to decoder 1 is off. Instead, in states $(1,0,1)$ and $(1,1,1)$, in which both channels to Decoder 1 and Decoder 2 are on, Encoder 2 transmits 1 bit of information to Decoder 1 or to Decoder 2 with equal probability.

Consider now the time $T_1$ required for Decoder 1 to decode successfully $F$ bits. We can write this random variable as

$$T_1 = \sum_{k=1}^{F} T_{1,k}, \tag{16}$$

where $T_{1,k}$ denotes the number of channel uses required to transmit the $k$th bit. Given the discussion above, the variables $T_{1,k}$ are independent for $k \in [F]$ and have a Geometric distribution with mean $(\Pr[\mathbf{G} = (1,0,0)] + \Pr[\mathbf{G} = (1,1,0)] + 1/2\Pr[\mathbf{G} = (1,0,1)] + 1/2\Pr[\mathbf{G} = (1,1,1)])^{-1} = 2/(1-\epsilon_2^2)$. By the strong law of large numbers we now have the limit

$$\lim_{F\to\infty} \frac{T_1}{F} = E[T_1] = \frac{2}{1-\epsilon_2^2} \tag{17}$$

with probability 1. In a similar manner, the resulting delivery time for Decoder 2 for any given bit has a Geometric distribution with mean $(\Pr[\mathbf{G} = (0,0,1)] + \Pr[\mathbf{G} = (0,1,1)] + \frac{1}{2}\Pr[\mathbf{G} = (1,0,1)] + \frac{1}{2}\Pr[\mathbf{G} = (1,1,1)])^{-1} = 2/(1-\epsilon_2^2)$; and, by the strong law of large numbers, we obtain that the time $T_2$ needed to transmit $F$ bits to Decoder 2 satisfies the limit $\lim_{F\to\infty} \frac{T_2}{F} = E[T_2] = \frac{2}{1-\epsilon_2^2}$ almost surely. Using this limit along with (17) allows to conclude that there exists a sequence of policies with $T/F \to 2/(1-\epsilon_2^2)$ for any arbitrarily small probability of error.

- **Partial Caching ($\mu = \mu_0$) with $\bar{\epsilon}_1\epsilon_2 \geq \epsilon_1\bar{\epsilon}_2^2$**: Next, we consider the corner point $(\mu_0, \delta_0)$ under the condition $\bar{\epsilon}_1\epsilon_2 \geq \epsilon_1\bar{\epsilon}_2^2$. In this case, in which Encoder 1 has a better channel than Decoder 2 in the average sense discussed above, our findings show that Encoder 2 should communicate to Decoder 1 only in the channel states in which the channel to Decoder 2 is off. Using these states, Encoder 2 sends $(1-\mu_0)F$ bits to Decoder 1. Encoder 1 cache a fraction $\mu_0$ of each file in the library and delivers $\mu_0 F$ bits of the requested file to Decoder 1. For this purpose, coordination between Encoder 1 and Encoder 2 is needed to manage interference in the state $(1,1,1)$ in which all links are on.

  A detailed description of the transmission strategy is provided below as a function of the channel state $\mathbf{G}$.

  (1) $\mathbf{G} = (0,0,1)$: Only the channel between Encoder 2 and Decoder 2 is active, and Encoder 2 transmits 1 bit of information to Decoder 2.
  (2) $\mathbf{G} = (0,1,0)$: The only active channel is between Encoder 1 and Decoder 1, and Encoder 1 transmits 1 information bit to Decoder 1.
  (3) $\mathbf{G} = (0,1,1)$: The cross channel is off, and each encoder transmits 1 bit of information to its decoder.
  (4) $\mathbf{G} = (1,0,0)$: Only the channel between Encoder 2 and Decoder 1 is active, and Encoder 2 transmits 1 bit of information to Decoder 1.
  (5) $\mathbf{G} = (1,0,1)$: The direct channel between Encoder 1 and Decoder 1 is off, while two other channels are on. Encoder 2 transmits 1 bit of information to Decoder 2.
  (6) $\mathbf{G} = (1,1,0)$: Both channels from Encoder 1 and Encoder 2 to Decoder 1 are on. Encoder 1 transmits $X_1 = 0$ and Encoder 2 transmits 1 bit of information to Decoder 1.
  (7) $\mathbf{G} = (1,1,1)$: Encoder 2 transmits 1 bit $X_2$ of information to Decoder 2. Encoder 1 transmits $X_1 = \tilde{X}_1 \oplus X_2$, where $\tilde{X}_1$ is an information bit for Decoder 1. This form of coordination is enabled by the fact that Encoder 1 knows the bit $X_2$, since it is part of the $\mu_0 F$ cached bits from the file requested by Decoder 2. In this way, interference from Encoder 2 is cancelled at Decoder 1.

From the discussion above, Encoder 2 transmits 1 bit of information to Decoder 2 in the states (1), (3), (5) and (7). For large $F$, the normalized transmission delay for transmitting the requested file to Decoder 2 is then equal to

$$\begin{aligned}\delta_{22} &= \Big(\Pr[\mathbf{G}=(0,0,1)+\Pr[\mathbf{G}=(0,1,1)] \\ &\quad +\Pr[\mathbf{G}=(1,0,1)]+\Pr[\mathbf{G}=(1,1,1)]\Big)^{-1} \\ &= \tfrac{1}{\bar{\epsilon}_2}.\end{aligned} \tag{18}$$

Furthermore, Encoder 2 transmits $(1-\mu_0)F$ bits to decoder 1 in the states at (4) and (6). The required normalized time for large $F$ is hence

$$\delta_{21} = \frac{1-\mu_0}{\epsilon_2\bar{\epsilon}_2} \tag{19}$$

Finally, Encoder 1 transmits $\mu_0 F$ bits to Decoder 1 in the states at (2), (3) and (7). The required time is thus

$$\delta_{11} = \frac{\mu_0}{\bar{\epsilon}_1\bar{\epsilon}_2+\bar{\epsilon}_1\epsilon_2^2} \tag{20}$$

It can be shown that $\delta_{11} \leq \delta_{21} = \delta_{22} = \delta_0$ under the given condition $\bar{\epsilon}_1\epsilon_2 \geq \epsilon_1\bar{\epsilon}_2^2$, and hence the DTB is given by $\max(\delta_{11},\delta_{21},\delta_{22}) = \delta_0$.

- **Partial Caching ($\boldsymbol{\mu=\mu_0}$) with $\boldsymbol{\bar{\epsilon}_1\epsilon_2 \leq \epsilon_1\bar{\epsilon}_2^2}$**: Finally, we consider the corner point $(\mu_0,\delta_0)$ under the complementary condition $\bar{\epsilon}_1\epsilon_2 \leq \epsilon_1\bar{\epsilon}_2^2$, in which Encoder 2 has better channels to the decoders. In this case, as above, Encoder 1 caches a fraction $\mu_0$ of all files. Transmission take place as described in the previous case except for state (5) which is modified as follows: (5) $\mathbf{G}=(1,0,1)$: Encoder 2 transmits 1 bit of information to either Decoder 1 or Decoder 2 with probabilities $\alpha = (1-\bar{\epsilon}_1\epsilon_2/\epsilon_1\bar{\epsilon}_2^2)/2$ and $1-\alpha$, respectively.

Encoder 2 hence transmits 1 bit of information to Decoder 2 in the states at cases (1), (3) and (7) and also with probability $1-\alpha$ in case (5). For large $F$, the normalized transmission delay for transmitting the requested file to Decoder 2 tends to

$$\begin{aligned}\delta_{22} &= \Big(\Pr[\mathbf{G}=(0,0,1)+\Pr[\mathbf{G}=(0,1,1)]+\Pr[\mathbf{G}=(1,1,1)]+(1-\alpha)\Pr[\mathbf{G}=(1,0,1)]\Big)^{-1} \\ &= \frac{2}{2-\epsilon_1-\epsilon_2+\epsilon_1\epsilon_2-\epsilon_1\epsilon_2^2}.\end{aligned} \tag{21}$$

In addition, Encoder 2 transmits 1 bit to Decoder 1 in cases (4) and (6) as well as in case (5) with probability $\alpha$. The required time to transmit $(1-\mu_0)F$ bits from Encoder 2 to Decoder 1 is hence

$$\delta_{21} = \frac{1-\mu_0}{\epsilon_2\bar{\epsilon}_2+\frac{1}{2}(\epsilon_1\bar{\epsilon}_2^2-\bar{\epsilon}_1\epsilon_2)}. \tag{22}$$

It can be shown that $\delta_{11} = \delta_{21} = \delta_{22} = \delta_0$, where $\delta_{11}$ is given in (20) under the given condition $\bar{\epsilon}_1\epsilon_2 \leq \epsilon_1\bar{\epsilon}_2^2$, yielding the DTB $\max(\delta_{11},\delta_{21},\delta_{22}) = \delta_0$. This concludes the proof of achievability.

### *3.2. Cloud and Edge-Aided System ($C \geq 0$)*

In the following proposition, we derive the minimum DTB $\delta^*(\mu,C)$ for the system in Figure 1 with $C \geq 0$.

**Proposition 2.** *The minimum DTB for the cache and cloud-aided system in Figure 1 is:*

$$\delta^*(\mu,C) = \delta^*(\mu), \tag{23}$$

*if* $C \leq 1-\epsilon_2^2$. *Otherwise, it is given by:*

$$\delta^*(\mu, C) = \begin{cases} \frac{2-\mu}{C} + \left(1 - \frac{1-\epsilon_2^2}{C}\right)\delta_0 & \text{if } \mu \leq \mu_0 \\ \delta_0 & \text{if } \mu \geq \mu_0, \end{cases} \tag{24}$$

*where* $\delta^*(\mu)$, $\mu_0$ *and* $\delta_0$ *are defined in* (13), (14) *and* (15), *respectively.*

**Proof.** See below and Appendix B. □

Figure 4 shows the minimum DTB as a function of $\mu$ and $C$. To elaborate on the results in Proposition 2, we focus first on the setting in which Encoder 1 has no caching capability, i.e., $\mu = 0$. In this case, unlike the scenario studied in the previous section, Encoder 1 can deliver part of the file requested by Decoder 1 through the connection to the Cloud. Nevertheless, if $C \leq 1-\epsilon_2^2$, that is, if the average delay for transmission of 1 bit from cloud to Encoder 1, namely $1/C$, is larger than the corresponding delay between Encoder 2 and both decoders, namely $1/(1-\epsilon_2^2)$, then it is optimal to neglect Encoder 1 and operate as discussed in Section 3.1. Instead, if $C \geq 1-\epsilon_2^2$, it is optimal for Encoder 1 to transmit parts of the requested files, or functions thereof, which are received from the cloud. In fact, as discussed below, it is necessary for the cloud to transmit a coded signal obtained from both the files requested by the users in order to obtain the DTB in Proposition 2. Moreover, if the fractional cache size satisfies the inequality $\mu \geq \mu_0$, then the cache size at Encoder 1 is sufficient to achieve the DTB $\delta_0$ corresponding to full caching and the Cloud-to-Encoder 1 link can be neglected with no loss of optimality.

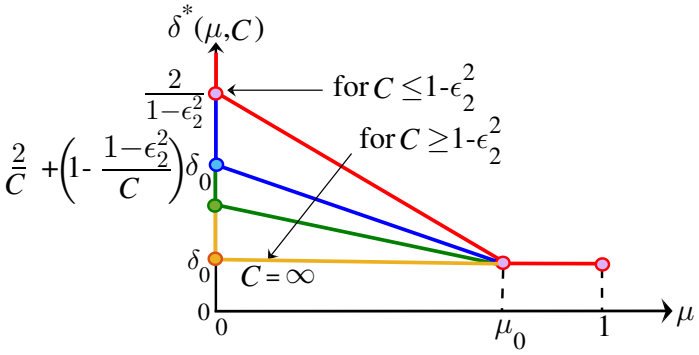


**Figure 4.** Minimum Delivery Time per Bit (DTB) $\delta^*(\mu, C)$ for the system in Figure 1.

Proof of Achievability

In this section, we detail the policies that achieve the minimum DTB described in Proposition 2. We start by noting that for $C \leq 1-\epsilon_2^2$, the achievability of the DTB follows from Proposition 1, and hence we can concentrate on the case $C \geq 1-\epsilon_2^2$. We first note that the minimum DTB $\delta^*(\mu, C)$ is a convex function of $\mu$ for any value of $C$. The proof follows as in Lemma 1 by file splitting and time sharing and is hence omitted.

**Lemma 2.** *The minimum DTB* $\delta^*(\mu, C)$ *is a convex function of* $\mu \in [0,1]$ *for any given value of* $C \geq 0$.

By the convexity of $\delta^*(\mu, C)$ in Lemma 2, and by the achievability of the DTB in Proposition 1 with $C = 0$, and hence also for $C \geq 0$, it suffices to prove that the corner point $\delta^*(0, C) = 2/C + (1 - (1-\epsilon_2^2)/C)\delta_0$ is achievable for $C \geq 1-\epsilon_2^2$. To this end, we consider the worst case in which each decoder requests a different file, and we adopt the following policy.

The Cloud-to-Encoder 1 link is used for a normalized time $\delta_C = T_C/F = (2 - \delta_0(1-\epsilon_2^2))/C$ to transmit $\rho F$ bits from the file requested by Encoder 1, with

$$\rho = 2 - \delta_0(1-\epsilon_2^2). \tag{25}$$

Of these bits, $\rho F \bar{\epsilon}_1 \epsilon_2/(\bar{\epsilon}_1\epsilon_2 + \bar{\epsilon}_1\bar{\epsilon}_2^2)$ bits are sent to Encoder 1 by the Cloud in an uncoded form. Instead, the remaining $\rho F \bar{\epsilon}_1 \bar{\epsilon}_2^2/(\bar{\epsilon}_1\epsilon_2 + \bar{\epsilon}_1\bar{\epsilon}_2^2)$ bits are transmitted by XORing each bit of the file with the corresponding bit of the file requested by Decoder 2. The mentioned $\rho F$ bits are sent to Decoder 1 by Encoder 1, while the remaining $(1-\rho)F$ bits are sent by Encoder 2 to Decoder 1, as discussed next.

The transmission strategy follows the approach described in Section 3.1. As for (20) the transmission of uncoded bits from Encoder 1 to Decoder 1 requires a normalized time on the channel

$$\delta_{11}^u = \frac{\rho}{\bar{\epsilon}_1\epsilon_2 + \bar{\epsilon}_1\bar{\epsilon}_2^2}. \tag{26}$$

while the transmission of coded bits requires time

$$\delta_{11}^c = \frac{\rho}{\bar{\epsilon}_1\epsilon_2 + \bar{\epsilon}_1\bar{\epsilon}_2^2}. \tag{27}$$

Similar to (19) and (22), the time required for Encoder 2 to transmit to Decoder 1 is

$$\delta_{21} = \begin{cases} \frac{1-\rho}{\epsilon_2\bar{\epsilon}_2} & \text{if } \bar{\epsilon}_1\epsilon_2 > \bar{\epsilon}_2^2\epsilon_1 \\ \frac{1-\rho}{\epsilon_2\bar{\epsilon}_2 + \frac{1}{2}(\epsilon_1\bar{\epsilon}_2^2 - \bar{\epsilon}_1\epsilon_2)} & \text{if } \bar{\epsilon}_1\epsilon_2 \leq \bar{\epsilon}_2^2\epsilon_1 \end{cases} \tag{28}$$

while $\delta_{22} = \delta_0$ is sufficient to communicate to Decoder 2. Under the channel conditions $\bar{\epsilon}_1\epsilon_2 > \bar{\epsilon}_2^2\epsilon_1$, from (25), (26) and (28), it can be shown that $\delta_{11}^u = \delta_{11}^c \leq \delta_{21} = \delta_{22} = \delta_0$. Therefore, the normalized time required on the edge channel is $\delta_E = \max(\delta_{11}^u, \delta_{11}^c, \delta_{21}, \delta_{22}) = \delta_0$. Instead, under the condition $\bar{\epsilon}_1\epsilon_2 \leq \bar{\epsilon}_2^2\epsilon_1$, using the same equations, it can be seen that $\delta_{11}^c = \delta_{11}^u = \delta_{21} = \delta_{22} = \delta_0$. It follows that $\delta_E = \max(\delta_{21}, \delta_{11}^c, \delta_{11}^u, \delta_{22}) = \delta_0$. We can conclude that DTB is $\delta_C + \delta_E = \delta_0 + (2 - \delta_0(1-\epsilon_2^2))/C$, which is equal to $\delta^*(0, C)$ in (24).

**Remark 2.** *In a manner similar to the edge-aided case, the optimal scheme described above requires only causal CSI at the encoders, and, furthermore, it requires no CSI at the Cloud (but only knowledge of the channel statistics.) This shows that the assumption of non-causal CSI is not needed to obtain optimal performance.*

## 4. Online Caching

Section 2 focused on an offline caching scenario in which there is a fixed set $\mathcal{L}$ of popular contents and the operation of the system is divided between a placement phase and a delivery phase. In this section, instead, we consider an online caching set-up in which the set of popular files varies from one time slot to the next. As a result, both content delivery and cache update should be generally performed in every time slot, where the latter is needed to ensure the timeliness of the cached content.

### *4.1. System Model*

Let $\mathcal{L}_t$ be the set of $N$ popular files at time slot $t$. As in [29], we assume that with probability $1-p$, the popular set is unchanged and we have $\mathcal{L}_t = \mathcal{L}_{t-1}$; while, with probability $p$, the set $\mathcal{L}_t$ is constructed by randomly and uniformly selecting one of the files in the set $\mathcal{L}_{t-1}$ and replacing it by a new popular file. At each time slot $t$, users request files $\mathbf{d}_t$, which are drawn uniformly at random from the set $\mathcal{L}_t$ without replacement. We consider two cases, namely: (i) *known popular set*: the Cloud is informed about the set $\mathcal{L}_t$ at time $t$, e.g., by leveraging data analytics tools; (ii) *unknown popular set*: the set $\mathcal{L}_t$ may only be inferred at the Cloud via the observation of the users' requests. We note that the latter assumption is typically made in the networking literature [24].

Define as $T_{C,t}$ the duration of the transmission from Cloud to Encoder 1 and as $T_t$ the duration of the transmission from both encoders to decoders at time slot $t$. As in the previous section, durations are measured in terms of number of channel uses of the binary fading channel. Since the set of popular files $\mathcal{L}_t$ is time-varying, both cache update and file delivery are generally performed at each time slot $t$. To this end, at time slot $t$, the Cloud encodes via the function:

$$\psi_C : [2^{NF}] \times [N]^2 \times \{0,1\}^{3T_t} \to [2^{T_{C,t}C}], \tag{29}$$

which maps the library $\mathcal{L}_t$ of all files, the demand vector $\mathbf{d}_t$ and the CSI vector $\mathbf{G}^{T_t}$ to the signal $U^{T_{C,t}} = (U_1, ..., U_{T_{C,t}}) = \psi_C(\mathcal{L}_t, \mathbf{d}_t, \mathbf{G}^{T_t})$ to be delivered to Encoder 1. We have the inequality $H(U^{T_{C,t}}) \leq T_{C,t}C$ according to the capacity constraints on the Cloud-to-Encoder 1 link. Moreover, Encoder 1 uses the encoding function

$$\psi_1 : [2^{\mu NF}] \times [2^{T_{C,t}C}] \times [N]^2 \times \{0,1\}^{3T_t} \to \{0,1\}^{T_t}, \tag{30}$$

which maps the cached content $V_t$, the received signal $U^{T_{C,t}}$, the demand vector $\mathbf{d}_t$ and the CSI sequence $\mathbf{G}^{T_t} = (\mathbf{G}(1), ..., \mathbf{G}(T_t))$ to the transmitted codeword $X_1^{T_t} = (X_1[1], ..., X_1[T_t]) = \psi_1(V_t, U^{T_{C,t}}, \mathbf{d}_t, \mathbf{G}^{T_t})$.

The probability of error is defined as

$$\mathrm{P}_{e,t}^{F} = \max_{j \in \{1,2\}} \Pr(\hat{W}_{d_{j,t}} \neq W_{d_{j,t}}), \tag{31}$$

where $d_{j,t}$ is the index of the requested file by $j$th user at time slot $t$ so that we have $\mathbf{d}_t = (d_{1,t}, d_{2,t})$. The probability of error in (31) is evaluated with respect to the distribution of the popular set $\mathcal{L}_t$ and of the request vector $\mathbf{d}_t$. A sequence of policies indexed by $t$ is said to be feasible if $\mathrm{P}_{e,t}^{F} \to 0$ as $F \to \infty$ for all $t$. In a manner similar to the offline case, we define DTB at time slot $t$ as

$$\delta_t(\mu, C) = \lim_{F \to \infty} \frac{\mathrm{E}[T_t + T_{C,t}]}{F}, \tag{32}$$

where the average is taken over the distribution of the popular set $\mathcal{L}_t$ and of the request vector $\mathbf{d}_t$. To measure the performance of online caching, we define the long-term DTB as

$$\bar{\delta}_{\mathrm{on}}(\mu, C) = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \delta_t(\mu, C). \tag{33}$$

We denote the minimum long-term DTB over all feasible policies under the known popular set assumption as $\bar{\delta}^*_{\mathrm{on,k}}(\mu, C)$, while $\bar{\delta}^*_{\mathrm{on,u}}(\mu, C)$ denotes the minimum long-term DTB under the unknown popular set assumption. By definition, we have the inequality $\bar{\delta}^*_{\mathrm{on,k}}(\mu, C) \leq \bar{\delta}^*_{\mathrm{on,u}}(\mu, C)$. Furthermore, both DTBs $\bar{\delta}^*_{\mathrm{on,k}}(\mu, C)$ and $\bar{\delta}^*_{\mathrm{on,u}}(\mu, C)$ are not smaller than the offline DTB $\delta^*(\mu, C)$, given that in the offline set-up caching takes place in a separate phase with no overhead on the Cloud-to-Encoder 1 link. In the rest of this section, we evaluate the performance of two proposed online caching schemes and we also provide a lower bound on the the minimum long-term DTB. The treatment is inspired by the prior work [16], which focuses on the F-RAN model studied in [11].

### *4.2. Proactive Online Caching*

If the popular set $\mathcal{L}_t$ is known, the cloud can proactively cache any new content at the small-cell BS by replacing the outdated file. Specifically, we propose to transfer a $\mu$-fraction of the new popular file from the Cloud to Encoder 1 in order to update the cache content at the small-cell BS. Since, after this update, the cache configuration with respect to the current set $\mathcal{L}_t$ of popular files is the same as in the offline case with respect to $\mathcal{L}$, delivery can then be performed by following the offline delivery policy detailed in Section 3.2. The following proposition presents the resulting achievable long-term DTB of proactive online caching.

**Proposition 3.** *The proposed proactive online caching for the cache and cloud-aided system in Figure 1 achieves the long-term DTB*

$$\bar{\delta}_{\text{on,pro}}(\mu, C) = \delta^*(\mu, C) + \frac{p\mu}{C}, \tag{34}$$

*with $\delta^*(\mu, C)$ is given by (23) and (24). We hence have the upper bound $\bar{\delta}^*_{\text{on,k}}(\mu, C) \leq \bar{\delta}_{\text{on,pro}}(\mu, C)$.*

**Proof.** With probability $p$, there is a new file in the popular set $\mathcal{L}_t$ and hence a $\mu$-fraction of the new content is sent on the cloud-to-Encoder 1 link resulting in a latency of $T_{C,t} = \mu F/C$. The achievable scheme in Section 3.2 is then used to deliver both requested files. As a result, the DTB at time slot $t$ is $\delta_t = p(\delta^*(\mu, C) + \mu/C) + (1-p)\delta^*(\mu, C)$. Using (33), the long-term DTB is given by (34). □

### *4.3. Reactive Online Caching*

When the popular set is highly time-varying, the proactive scheme sends a large number of new contents on the Cloud-to-Encoder 1 link to update the cache content at small-cell BS. However, only a subset of these files will generally be requested before becoming outdated. To potentially solve this problem, the Cloud can update the small-cell BS's cache by means of a reactive scheme. Accordingly, the Cloud updates the cache only if the files requested by Decoder 1 and/or Decoder 2 are not (partially) cached at the small-cell BS.

The reactive strategy, unlike the proactive one, can operate under the unknown popular set assumption. It is also possible to define a reactive strategy that leverages knowledge of the set of popular files to outperform proactive caching. This will be discussed in our future work.

To elaborate, in a manner similar to [29], in each time slot $t$, small-cell BS stores a $(\mu/\alpha)$-fraction of $N' = \alpha N$ files for some $\alpha > 1$. Note that the set of $N' > N$ cached files in the cached contents of small-cell BS generally contains files that are no longer in the set $\mathcal{L}_t$ of $N$ popular files. Caching $N' > N$ files is instrumental in keeping the intersection between the set of cached files and $\mathcal{L}_t$ from vanishing [29]. To update the cache content, a $(\mu/\alpha)$-fraction of the requested and uncached files is sent on the Cloud-to-Encoder 1 link and is cached at the small-cell BS by randomly and uniformly evicting the same number of cached files. The following proposition presents an achievable long-term DTB for the proposed reactive online caching policy.

**Proposition 4.** *The proposed reactive online caching for the cache and cloud-aided system in Figure 1 achieves a long-term DTB that is upper bounded as*

$$\bar{\delta}_{\text{on,react}}(\mu, C) \leq \delta^*\left(\frac{\mu}{\alpha}, C\right) + \frac{p\mu}{C(1-p/N)(\alpha-1)}, \tag{35}$$

*for any $\alpha > 1$. This yields the upper bound $\bar{\delta}^*_{\text{on,u}}(\mu, C) \leq \bar{\delta}_{\text{on,react}}(\mu, C)$.*

**Proof.** Denoting $Y_t \in \{0, 1, 2\}$ the number of requested and uncached files at time slot $t$, the cloud send a $(\mu/\alpha)$-fraction of the $Y_t$ requested and uncached files to the small-cell BS. Hence, the achievable DTB at each time slot $t$ is

$$\delta_t(\mu, C) = \delta^*\left(\frac{\mu}{\alpha}, C\right) + \frac{\mu \mathrm{E}(Y_t)}{\alpha C}. \tag{36}$$

By plugging (36) into the definition of long-term DTB (33), we have

$$\bar{\delta}_{\text{on,react}}(\mu, C) = \delta^*\left(\frac{\mu}{\alpha}, C\right) + \left(\frac{\mu}{\alpha C}\right) \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathrm{E}[Y_t]. \tag{37}$$

Noting the fact that content placement and random eviction are the same as [29], the result of ([29] Lemma 3) can be invoked to obtain the upper bound

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathrm{E}[Y_t]\leq\frac{p}{(1-p/N)(1-1/\alpha)}. \tag{38}$$

Plugging (38) into (37) completes the proof. □

### *4.4. Lower Bound on the Minimum Long-Term DTB*

We now provide a lower bound on the the minimum long-term DTB.

**Proposition 5.** *(Lower bound on the Long-Term DTB of Online Caching). For the cache and cloud-aided system in Figure 1 with $N\geq 2$, the long-term DTB is lower bounded as*

$$\bar{\delta}^{*}_{\mathrm{on,u}}(\mu,C)\geq\bar{\delta}^{*}_{\mathrm{on,k}}(\mu,C)\geq\Big(1-\frac{2p}{N}\Big)\delta^{*}(\mu,C)+\Big(\frac{2p}{N}\Big)\delta^{*}(0,C) \tag{39}$$

*with $\delta^{*}(\mu,C)$ given in (23) and (24).*

**Proof.** See Appendix C. □

The lower bound (39) will be leveraged in the next section to relate the performance of offline and online caching.

## 5. Comparison between Online and Offline Caching

In this section, we compare the performance of the offline caching system studied in Section 3 and of the online caching system introduced in Section 4. The following proposition presents that the minimum long-term DTB can be upper and lower bounded in terms of the minimum DTB of offline caching.

**Proposition 6.** *For the cache and cloud-aided system in Figure 1 with $N\geq 2$, the long-term DTB satisfies the inequalities*

$$\Big(1-\frac{2p}{N}\Big)\delta^{*}(\mu,C)+\frac{2p}{N}\frac{2}{1-\epsilon_2^2}\leq\bar{\delta}^{*}_{\mathrm{on,k}}(\mu,C)\leq\bar{\delta}^{*}_{\mathrm{on,u}}(\mu,C)\leq 2\delta^{*}(\mu,C) \tag{40}$$

*if $C\leq 1-\epsilon_2^2$, and*

$$\Big(1-\frac{2p}{N}\Big)\delta^{*}(\mu,C)+\frac{2p}{N}\Big(\frac{2-(1-\epsilon_2^2)\delta_0}{C}+\delta_0\Big)\leq\bar{\delta}^{*}_{\mathrm{on,k}}(\mu,C)\leq\bar{\delta}^{*}_{\mathrm{on,u}}(\mu,C)\leq\delta^{*}(\mu,C)+\frac{4}{C} \tag{41}$$

*if $C\geq 1-\epsilon_2^2$.*

**Proof.** The upper bound is obtained by comparing the performance (34) of the proposed reactive scheme with the minimum offline DTB in Proposition 2, while the lower bound is from Proposition 5. Details are provided in Appendix D. □

Proposition 6 shows that the long-term DTB with online caching is no larger than twice the minimum offline DTB in the regime of low capacity $C$. Instead for larger values of $C$, the minimum online DTB is proportional to minimum offline DTB with an additive gap that decreases as $1/C$. Informally, these results demonstrate that the additive loss of online caching decreases as $1/C$ for sufficiently large $C$, while, for lower values of $C$, the performance gap is bounded. This stands in contrast to [16], in which the performance gap between offline and online caching increases as the inverse of the capacity of the link between Cloud and BSs when the latter becomes smaller. The key

distinction here is that the macro-BS has direct access to the set of popular files and can directly serve the users, while in [16] the Cloud can only access the users through the finite-capacity links.

## 6. Numerical Results

In this section, we evaluate the performance of the proposed online caching schemes numerically. We specifically consider the long-term DTB achievable by the proposed proactive scheme (34) and the proposed reactive scheme (35). For the latter, we evaluate the expectation in (36) via Monte Carlo simulations by averaging over a large number of realizations, i.e., 10,000, of the random process $Y_t$. It is assumed that the small-cell cache is empty at the start of simulation, i.e., at time $t = 1$.

The impact of the cloud-to-Encoder 1 capacity $C$ is first considered in Figure 5. As a reference, we also plot the minimum DTB for offline caching in (23) and (24) and the performance with no caching, that is, $\delta^*(0, C)$ in (24). For reactive caching, we assume random eviction for reactive caching. Parameters are set as $\mu = 0.5$, $p = 0.5$, $\epsilon_1 = \epsilon_2 = 0.5$ and $N = 10$. It is seen that both proactive and reactive caching can significantly improve over the no caching scheme by updating the content stored at the small-cell BS. However, as the capacity of Cloud-to-Encoder 1 link decreases, it is deleterious in terms of delivery latency to use the link in order to update the cache content. As a result, if $C$ is small enough, the performance of reactive and proactive caching coincides with the no caching system. When $C$ is large enough, instead, the latency of cache update is negligible and both proactive and reactive schemes achieve the same DTB, which tends to the minimum offline DTB.

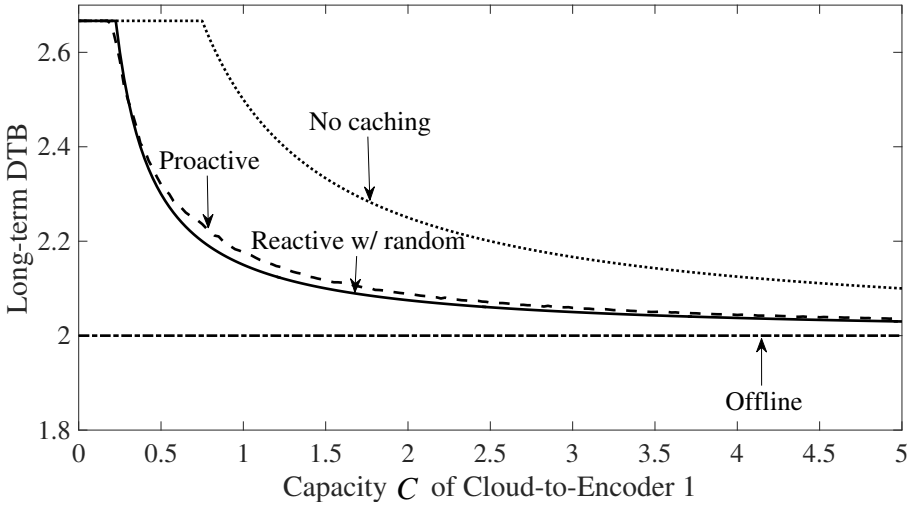


**Figure 5.** Achievable long-term DTB versus the capacity $C$ of the Cloud-to-Encoder 1 for proactive scheme (34) and reactive caching with random eviction (35). For reference, the DTB with no caching, namely $\delta^*(0, C)$, and the offline minimum DTB (23) and (24) are also shown ($p = 0.5$, $\mu = 0.5$, $\epsilon_1 = \epsilon_2 = 0.5$, $N = 10$).

Next, we compare the performance of reactive and proactive online caching schemes as a function of the probability $p$ of new content. As shown in Figure 6 for $\mu = 0.5$, $C = 0.5$, $\epsilon_1 = \epsilon_2 = 0.5$ and $N = 10$, when $p$ is small, proactive caching outperforms reactive caching, since it uses the Cloud-to-Encoder 1 connection only with rare event that there is a new popular file. On the other hand, when $p$ is large, as explained in the previous section, the reactive approach yields a smaller latency than the proactive scheme. It is also seen that the LRU eviction strategy, whereby the replaced file is the one that has been least recently requested by any user, and FIFO eviction strategy, whereby the file that has been in the caches for the longest time is replaced, are both able to improve over randomized eviction.

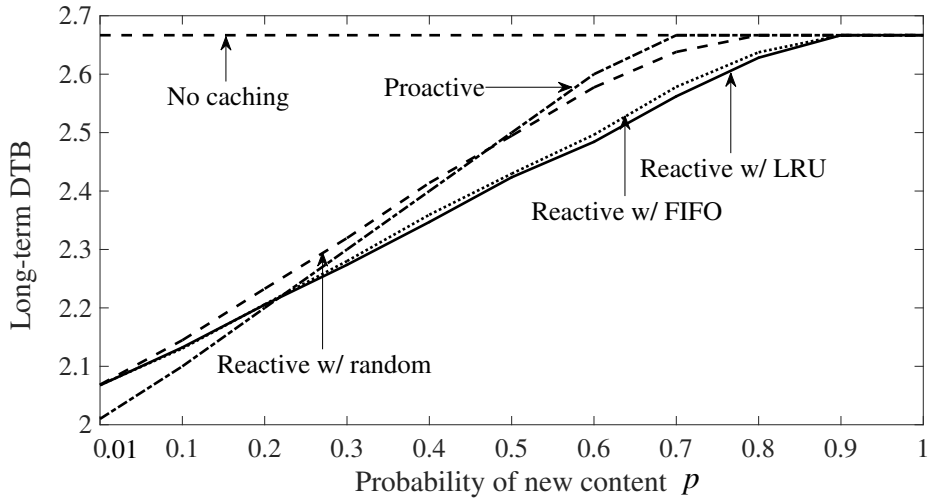


**Figure 6.** Achievable long-term DTB versus probability $p$ of new content for the proactive scheme (34) and reactive caching scheme with random, LRU or FIFO eviction (35). For reference, the DTB with no caching, namely $\delta^*(0, C)$, and the offline minimum DTB (23) and (24) are also shown ($C = 0.5$, $\mu = 0.5$, $\epsilon_1 = \epsilon_2 = 0.5$, $N = 10$).

## 7. Conclusions

Motivated by recent advances in cache and cloud-aided wireless network architectures, we have considered a fog-assisted system for content delivery. The system model includes a macro-BS that coexists with a cache and cloud-aided small-cell BS whose user can also be served by the macro-BS. Using the minimum delivery latency as performance measure, the trade-off between latency and system resources has been studied. A characterization of this optimal trade-off has been derived under a binary fading interference channel and in the presence of full CSI when the set of popular contents is fixed. For the alternative online scenario with time-varying set of popular files, the average DTB within a long time horizon is shown to be at most two times larger than for the offline scenario case when the capacity of the link used to update the cache content is small and to have otherwise a gap inversely proportional to this capacity.

**Author Contributions:** Osvaldo Simeone and Ravi Tandon conceived the system model and some of the key ideas behind achievability and converse; Seyyed Mohammadreza Azimi performed the analysis, carried out the numerical experiments, and wrote the paper with comments from Osvaldo Simeone and Ravi Tandon. All authors have read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Proof of Converse for Proposition 1

Consider any request vector $\mathbf{d}$ containing two arbitrary, different files $W_1$ and $W_2$, and any coding scheme satisfying $P_e^F \to 0$ as $F \to \infty$. The following set of inequalities is based on the fact that, under any such coding scheme, a hypothetical decoder provided with the CSI vector $\mathbf{G}^T$, with the cached contents $V_1$ and $V_2$ in (2) relative to files $W_1$ and $W_2$, and with the signal $\tilde{G}^T X_2^T$, to be described below, must be able to decode both messages $W_1$ and $W_2$. The signal $\tilde{G}^T X_2^T = (\tilde{G}(1)X_2(1), ..., \tilde{G}(T)X_2(T))$ is such that $\tilde{G}(t) = 0$ if $G_0(t) = G_2(t) = 0$ and $\tilde{G}(t) = 1$ otherwise. Note, therefore, that $\tilde{G}(t)X_2(t) = X_2(t)$ as long as either or both $G_0(t)$ and $G_2(t)$ are equal to one. The intuition here is that from $\tilde{G}^T X_2^T$ and $G^T$, the hypothetical decoder can recover $Y_2^T$ and hence $W_2$; while from $\tilde{G}^T X_2^T$, $G^T$ and $V_1$, the decoder can reconstruct $Y_1^T$ and hence decode $W_1$. Details are as follows

$$\begin{aligned}
2F &= H(W_1, W_2) \\
&= I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T) \\
&\quad + H(W_1, W_2 | \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T) \\
&= I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T) \\
&\quad + H(W_1 | \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T) \\
&\quad + H(W_2 | \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T, W_1) \\
&\overset{(a)}{=} I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T) \\
&\quad + H(W_1 | \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T, Y_1^T) \\
&\quad + H(W_2 | \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T, W_1, Y_2^T) \\
&\overset{(b)}{\leq} I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, \mathbf{G}^T) + F\gamma_F \\
&= I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2 | \mathbf{G}^T) + F\gamma_F \\
&\overset{(c)}{\leq} H(V_1) + H(\tilde{G}^T X_2^T | \mathbf{G}^T) + F\gamma_F \\
&\overset{(d)}{\leq} \mu F + T(1 - \epsilon_2^2) + F\gamma_F,
\end{aligned} \tag{A1}$$

where $\gamma_F$ indicates any function that satisfies $\gamma_F \to 0$ as $F \to \infty$. In above derivation, (a) follows from the facts that: (*i*) $Y_1^T$ is a function of $V_1$, $V_2$, $\mathbf{G}^T$, and $\tilde{G}^T X_2^T$, since $X_1^T$ can be assumed to depend on without loss of generality only on $V_1$ and $V_2$, and the vector $G_0^T X_2^T$ can be obtained from $\tilde{G}^T X_2^T$ and $\mathbf{G}^T$; (*ii*) $Y_2^T$ is a function of $(\mathbf{G}^T, \tilde{G}^T X_2^T)$; (b) follows from Fano's inequality; (c) follows from the fact that the messages are independent of channel realization and from Fano inequality $H(V_2 | \tilde{G}^T X_2^T, \mathbf{G}^T) \leq F\gamma_F$; (d) hinges on the cache constraint (3) and by the following bounds

$$\begin{aligned}
H(\tilde{G}^T X_2^T | \mathbf{G}^T) &\leq \sum_{t=1}^{T} H(\tilde{G}(t) X_2(t) | \mathbf{G}(t)) \\
&\leq T \sum_{\mathbf{g} \in \mathcal{G}} p(\mathbf{g}) \max_{p(X_2)} H(\tilde{G} X_2 | \mathbf{G} = \mathbf{g}) \\
&\leq T(1 - \epsilon_2^2),
\end{aligned} \tag{A2}$$

where $\mathcal{G}$ is the set of all channel states and the last inequality follows from the fact that the entropy in all states $\mathbf{G} = \mathbf{g}$ is maximized for $X_2 \sim \text{Bernoulli}(1/2)$. For $F \to \infty$, (A1) yields the bound on the minimum DTB

$$\delta^*(\mu) \geq \frac{2 - \mu}{1 - \epsilon_2^2}. \tag{A3}$$

Based on the fact that requested files should be retrieved from the received signals, another bound can be derived as follows:

$$\begin{aligned}
2F &= H(W_1, W_2) \\
&= I(W_1, W_2; Y_1^T, Y_2^T, \mathbf{G}^T) + H(W_1, W_2 | Y_1^T, Y_2^T, \mathbf{G}^T) \\
&\overset{(a)}{\leq} I(W_1, W_2; Y_1^T, Y_2^T, \mathbf{G}^T) + F\gamma_F \\
&\overset{(b)}{\leq} I(W_1, W_2; Y_1^T, Y_2^T | \mathbf{G}^T) + F\gamma_F \\
&= H(Y_1^T, Y_2^T | \mathbf{G}^T) + F\gamma_F \\
&\overset{(c)}{\leq} T \sum_{\mathbf{g} \in \mathcal{G}} p(\mathbf{g}) \max_{p(X_1, X_2)} H(Y_1, Y_2 | \mathbf{G} = \mathbf{g}) + F\gamma_F \\
&\overset{(d)}{=} T(2 - \epsilon_1 - \epsilon_2 + \epsilon_1\epsilon_2 - \epsilon_1\epsilon_2^2) + F\gamma_F,
\end{aligned} \tag{A4}$$

where (a) follows from Fano's inequality; (b) follows from the fact that channel gains are independent from files; (c) follows in a manner similar to (A2); and (d) is due to the fact that the entropy terms in

the previous step are maximized by choosing $X_1$ and $X_2$ to be independent and identically distributed as Bernoulli$(1/2)$. With $F \to \infty$, we obtain the bound

$$\delta^*(\mu) \geq \frac{2}{2-\epsilon_1-\epsilon_2+\epsilon_1\epsilon_2-\epsilon_1\epsilon_2^2}. \tag{A5}$$

Considering Decoder 2, the file $W_2$ should be decodable from $Y_2^T$, leading to the following bounds

$$\begin{aligned} F = H(W_2) &= I(W_2; Y_2^T, \mathbf{G}^T) + H(W_2|Y_2^T, \mathbf{G}^T) \\ &\overset{(a)}{\leq} I(W_2; Y_2^T|\mathbf{G}^T) + F\gamma_F \\ &\leq H(Y_2^T|\mathbf{G}^T) + F\gamma_F \\ &\overset{(b)}{\leq} T(1-\epsilon_2) + F\gamma_F, \end{aligned} \tag{A6}$$

where (a) follows from Fano's inequality and (b) follows in a manner similar to (A2) and the independence of channel gains from files. Therefore, based on (A6) as $F \to \infty$, we obtain the bound

$$\delta^*(\mu) \geq \frac{1}{1-\epsilon_2}. \tag{A7}$$

Combining (A3), (A5) and (A7) yields the desired lower bound.

## Appendix B. Proof of Converse for Proposition 2

Let us denote $\delta_C = T_C/F$ the normalized latency on the Cloud-to-Encoder 1 link and $\delta_E = T/F$ the normalized latency on the channel between encoders and decoders. We first observe that, following the same argument as in (A4)–(A7), we have the bound

$$\delta_E \geq \delta_0 \tag{A8}$$

for any sequence of feasible policies. We now obtain a lower bound on both normalized delays $\delta_E$ and $\delta_C$ by observing that a hypothetical decoder provided with the CSI vector $\mathbf{G}^T$, with the cached content $V_1$ and $V_2$ in (2), with the cloud-aided message $U^{T_C}$, and with the signal $\tilde{G}^T X_2^T$ described in Appendix A can decode both messages $W_1$ and $W_2$. Details are as follows

$$\begin{aligned} 2F &= H(W_1, W_2) \\ &= I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, U^{T_C}, \mathbf{G}^T) \\ &\quad + H(W_1, W_2|\tilde{G}^T X_2^T, V_1, V_2, U^{T_C}, \mathbf{G}^T) \\ &\overset{(a)}{\leq} I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, U^{T_C}, \mathbf{G}^T) + F\gamma_F \\ &= I(W_1, W_2; \tilde{G}^T X_2^T, V_1, V_2, U^{T_C}|\mathbf{G}^T) + F\gamma_F \\ &\overset{(b)}{\leq} H(V_1) + H(U^{T_C}) + H(\tilde{G}^T X_2^T|\mathbf{G}^T) + F\gamma_F \\ &\overset{(c)}{\leq} \mu F + T_C C + T(1-\epsilon_2^2) + F\gamma_F, \end{aligned} \tag{A9}$$

where, as in Appendix A, $\gamma_F$ indicates any function that satisfies $\gamma_F \to 0$ as $F \to \infty$. In above derivation, steps (a)–(b) follow as steps (a)–(b) in (A1), where we note that the inequality $H(V_2|\tilde{G}^T X_2^T, \mathbf{G}^T) \leq F\gamma_F$ by Fano inequality, while (c) hinges on the cache constraint (3) and the bound $H(U^{T_C}) \leq \sum_{i=1}^{T_C} H(U_i) \leq T_C C$ due to the capacity constraint on the Cloud-to-Encoder 1 link. As $F \to \infty$, the inequality (A9) yields the bound on the latency components $\delta_c$ and $\delta_E$

$$\frac{1-\epsilon_2^2}{C}\delta_E + \delta_C \geq \frac{2-\mu}{C}. \tag{A10}$$

To complete the proof, we combine bounds (A8) and (A10) as follows.

- For $C \leq 1-\epsilon_2^2$, the bound (A10), directly yields

$$\delta^*(\mu, C) = \delta_E + \delta_C \geq \delta_E + \frac{C}{1-\epsilon_2^2}\delta_C \geq \frac{2-\mu}{1-\epsilon_2^2}. \tag{A11}$$

- For $C \geq 1-\epsilon_2^2$, two scenarios are possible. If $\mu \leq \mu_0$, multiplying (A8) by the positive coefficient $1-(1-\epsilon_2^2)/C$ and summing the result with (A10), provides the corresponding result in (24). Instead, if $\mu \geq \mu_0$, from (A8), we directly obtain $\delta^*(\mu, C) \geq \delta_E \geq \delta_0$.

## Appendix C. Proof of Proposition 5

To obtain a lower bound on the long-term DTB, following [16], we consider an enhanced system in which, at each time slot $t$, the small-cell BS is informed of the optimal cache content of an offline scheme tailored to the current popular set $\mathcal{L}_t$. In this system, at each time slot $t$, with probability of $p$ there is a new file in the set of popular files, and hence the probability that an uncached file is requested by one of the users is $2p/N$. As a result, the DTB in time slot $t$ for the genie-aided system can be lower bounded as

$$\delta_t \geq \left(1-\frac{2p}{N}\right)\delta^*(\mu, C) + \left(\frac{2p}{N}\right)\delta_{\text{on,lb}}(C), \tag{A12}$$

where $\delta^*(\mu, C)$ is the minimum DTB for the offline caching set-up in Proposition 2, while $\delta_{\text{on,lb}}(C)$ is a lower bound on the minimum DTB for offline caching in which all files but one can be cached.

To obtain the lower bound $\delta_{\text{on,lb}}(C)$, we start by noting that the set-up is equivalent to that for the proof in Appendix B with the only difference is that one of the requested files by users cannot be cached at the small-cell BS. Since the probability of error (31) should be small for any request vector, in order to obtain a lower bound, we assume that the message $W_1$ requested by user 1 cannot be cached at the small-cell BS. Using the resulting condition $H(V_1) = 0$ in step (b) of (50) yields the inequality

$$2F \leq T_{C,t}C + T(1-\epsilon_2^2) + F\gamma_F, \tag{A13}$$

and hence, letting $\gamma_F \to 0$ as $F \to \infty$, we have the inequality

$$\frac{1-\epsilon_2^2}{C}\delta_E + \delta_C \geq \frac{2}{C}. \tag{A14}$$

To complete the proof, we combine bounds (A8) and (A14) as follows.

- For $C \leq 1-\epsilon_2^2$, the bound (A14), directly yields

$$\delta_{\text{on,lb}}(C) = \delta_E + \delta_C \geq \delta_E + \frac{C}{1-\epsilon_2^2}\delta_C \geq \frac{2}{1-\epsilon_2^2}. \tag{A15}$$

- For $C \geq 1-\epsilon_2^2$, multiplying (A8) by the positive coefficient $1-(1-\epsilon_2^2)/C$ and summing the result with (A14) yields the lower bound

$$\delta_{\text{on,lb}}(C) \geq \frac{2}{C} + \left(1-\frac{1-\epsilon_2^2}{C}\right)\delta_0. \tag{A16}$$

We note that comparing (A15) and (A16) with Propositions 1 and 2 reveals that when one of the requested files is not available at the small-cell BS, the system degrades to the case with zero caching at small-cell BS and hence we have

$$\delta_{\text{on,lb}}(C) \geq \delta^*(0, C). \tag{A17}$$

Plugging (A17) into (A12) and then using (33) completes the proof.

## Appendix D. Proof of Proposition 6

The lower bound follows directly from Proposition 5. To prove the upper bound, we leverage the following lemma.

**Lemma A1.** *For any $\alpha > 1$, we have the following inequality*

$$\delta^*\left(\frac{\mu}{\alpha}, C\right) \leq \delta^*(\mu, C) + \max\left(\frac{2}{C}, \frac{\mu\left(1-\frac{1}{\alpha}\right)}{C}\right). \tag{A18}$$

**Proof.** See Appendix E. □

Using Proposition 4 and Lemma A1, an upper bound on the long-term DTB of the proposed reactive caching scheme is obtained as

$$\bar{\delta}_{\text{on,react}}(\mu, C) \leq \delta^*(\mu, C) + f(\alpha), \tag{A19}$$

where

$$f(\alpha) = \frac{p\mu}{C(1-p/N)(\alpha-1)} + \max\left(\frac{2}{C}, \frac{\mu\left(1-\frac{1}{\alpha}\right)}{C}\right). \tag{A20}$$

Since the additive gap (A20) is a decreasing function of $N$ and an increasing function of $p$ and $\mu$, it can be further upper bounded by setting $N = 2$, $p = 1$ and $\mu = 1$. By plugging $\alpha = 2$, we have

$$\bar{\delta}^*_{\text{on,k}}(\mu, C) \leq \bar{\delta}^*_{\text{on,u}}(\mu, C) \leq \bar{\delta}_{\text{on,react}}(\mu, C) \leq \min\left(\delta^*(0,0), \delta^*(\mu, C) + \frac{4}{C}\right). \tag{A21}$$

The upper bound in (A21) is obtained using the fact that the maximum delivery latency namely, $\delta^*(0,0)$ is achieved when both requested files are delivered by transmission from macro-BS. To complete the proof, we consider the following regimes

- Low capacity regime ($C \leq 1 - \epsilon_2^2$): In this regime, using Propositions 2 and 5, the lower bound is

$$\left(1 - \frac{2p}{N}\right)\delta^*(\mu, C) + \frac{2p}{N}\frac{2}{1-\epsilon_2^2} \leq \bar{\delta}^*_{\text{on,k}}(\mu, C) \leq \bar{\delta}^*_{\text{on,u}}(\mu, C). \tag{A22}$$

  To prove the upper bound, we consider the following two sub-regimes

  - Low cache regime ($\mu \leq \mu_0$): In this case, using Proposition 2 and (A21), we have

$$\bar{\delta}^*_{\text{on,k}}(\mu, C) \leq \bar{\delta}^*_{\text{on,u}}(\mu, C) \leq \min\left(\delta^*(0,0), \delta^*(\mu, C) + \frac{4}{C}\right) = \frac{2}{1-\epsilon_2^2}. \tag{A23}$$

    Using Proposition 2, the minimum offline DTB is $\delta^*(\mu, C) = (2-\mu)/(1-\epsilon_2^2)$ and therefore we have

$$\frac{\bar{\delta}^*_{\text{on,u}}(\mu, C)}{\delta^*(\mu, C)} \leq \frac{2}{2-\mu} \overset{(a)}{\leq} \frac{2}{2-\mu_0} \overset{(b)}{\leq} 2, \tag{A24}$$

    where $(a)$ follows from $\mu \leq \mu_0$ and $(b)$ follows from $0 \leq \mu_0 \leq 1$.
  - High cache regime ($\mu \geq \mu_0$): In this regime, using Proposition 2, the minimum offline DTB is $\delta^*(\mu, C) = \delta_0$ with $\delta_0$ given by (15). Using (A21), we have

$$\frac{\bar{\delta}^*_{\text{on,u}}(\mu, C)}{\delta^*(\mu, C)} \leq \frac{2}{\delta_0(1-\epsilon_2^2)} \overset{(a)}{\leq} \frac{2}{1+\epsilon_2} \overset{(b)}{\leq} 2, \tag{A25}$$

    where $(a)$ follows from the definition of $\delta_0$ in (15) and $(b)$ follows from $0 \leq \epsilon_2 \leq 1$.

Combining (A22), (A24) and (A25) results in (40).

- High capacity regime ($C \geq 1-\epsilon_2^2$): In this regime, using Propositions 2 and 5, the lower bound is

$$\Big(1-\frac{2p}{N}\Big)\delta^*(\mu,C)+\frac{2p}{N}\Big(\frac{2-(1-\epsilon_2^2)\delta_0}{C}+\delta_0\Big) \leq \bar{\delta}^*_{\text{on,k}}(\mu,C) \leq \bar{\delta}^*_{\text{on,u}}(\mu,C). \tag{A26}$$

To prove the upper bound, using (A21) and Proposition 2, we have

$$\bar{\delta}^*_{\text{on,u}}(\mu,C) \leq \delta^*(\mu,C)+\frac{4}{C}. \tag{A27}$$

Combining (A26) and (A27) results in (41) and completes the proof.

## Appendix E. Proof for Lemma A1

To prove Lemma A1, for any given $\alpha > 1$, we first define

$$\mu_1 = \min(1, \alpha\mu_0), \tag{A28}$$

where $\mu_0$ given by (14). Then, we consider separately small-cache regime with $\mu \in [0,\mu_0]$, medium-cache regime $\mu \in [\mu_0,\mu_1]$ and the high-cache regime with $\mu \in [\mu_1,1]$.

- Small-cache Regime ($\mu \in [0,\mu_0]$): Using (24), we have the following upper bound

$$\begin{aligned}\delta^*\Big(\frac{\mu}{\alpha},r\Big) &= \frac{2-\frac{\mu}{\alpha}}{C}+\Big(1-\frac{1-\epsilon_2^2}{C}\Big)\delta_0\\ &= \frac{2-\mu}{C}+\Big(1-\frac{1-\epsilon_2^2}{C}\Big)\delta_0+\frac{\mu\left(1-\frac{1}{\alpha}\right)}{C}\\ &\overset{(a)}{=} \delta^*(\mu,C)+\frac{\mu\left(1-\frac{1}{\alpha}\right)}{C},\end{aligned} \tag{A29}$$

where $(a)$ follows from (24) in the regime of interest.

- Medium-cache Regime ($\mu \in [\mu_0,\mu_1]$): Using (24), we have the following upper bound

$$\begin{aligned}\delta^*\Big(\frac{\mu}{\alpha},r\Big)-\delta^*(\mu,C) &= \frac{2-\frac{\mu}{\alpha}}{C}+\Big(1-\frac{1-\epsilon_2^2}{C}\Big)\delta_0-\delta_0\\ &\overset{(a)}{\leq} \frac{2}{C},\end{aligned} \tag{A30}$$

where $(a)$ is obtained by omitting the negative terms.

- High-cache Regime ($\mu \in [\mu_1,1]$): Using (24), we have

$$\delta^*\Big(\frac{\mu}{\alpha},r\Big) = \delta^*(\mu,C) = \delta_0. \tag{A31}$$

Finally, using (A29), (A30) and (A31) concludes the proof.

## References

1. Shanmugam, K.; Golrezaei, N.; Dimakis, A.G.; Molisch, A.F.; Caire, G. Femtocaching: Wireless content delivery through distributed caching helpers. *IEEE Trans. Inf. Theory* **2013**, *59*, 8402–8413.
2. Bastug, E.; Bennis, M.; Debbah, M. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Commun. Mag.* **2014**, *52*, 82–89.
3. Maddah-Ali, M.A.; Niesen, U. Cache Aided Interference Channels. 2015. Available online: http://arxiv.org/abs/1510.06121 (accessed on 21 October 2015).

4. Maddah-Ali, M.A.; Niesen, U. Cache aided interference channels. In Proceedings of the 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, China, 7–12 July 2015; pp. 809–813.
5. Sengupta, A.; Tandon, R.; Simeone, O. Cache-aided wireless networks: Tradeoffs between storage and latency. In Proceedings of the Annual Conference on Information Science and Systems (CISS), Princeton, NJ, USA, 16–18 March 2016; pp. 320–325.
6. Naderializadeh, N.; Maddah-Ali, M.A.; Avestimehr, A.S. Fundamental limits of cache-aided interference management. *IEEE Trans. Inf. Theory* **2017**, *63*, 3092–3107.
7. Xu, F.; Liu, K.; Tao, M. Cooperative Tx/Rx caching in interference channels: A storage-latency tradeoff study. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 7–12 July 2016; pp. 2034–2038.
8. Roig, J.S.P.; Gunduz, D.; Tosato, F. Interference Networks with Caches at Both Ends. 2017. Available online: http://arxiv.org/abs/1703.04349 (accessed on 13 March 2017).
9. Naderializadeh, N.; Maddah-Ali, M.A.; Niesen, U. On the Optimality of Separation between Caching and Delivery in General Cache Networks. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 1–5.
10. Cao, Y.; Tao, M.; Xu, F.; Liu, K. Fundamental Storage-Latency Tradeoff in Cache-Aided MIMO Interference Networks. Available online: http://arxiv.org/abs/1609.01826 (accessed on 7 September 2016).
11. Tandon, R.; Simeone, O. Cloud aided wireless networks with edge caching: Fundamental latency trade offs in Fog radio access networks. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 7–12 July 2016; pp. 2029–2033.
12. Sengupta, A.; Tandon, R.; Simeone, O. Fog-Aided Wireless Networks for Content Delivery: Fundamental Latency Trade-Offs. 2016. Available online: http://arxiv.org/abs/1605.01690 (accessed on 5 May 2016).
13. Koh, J.; Simeone, O.; Tandon, R.; Kang, J. Cloud-aided edge caching with wireless multicast fronthauling in fog radio access networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
14. Girgis, M.; Ercetiny, O.; Nafie, M.; ElBatt, T. Decentralized coded caching in wireless networks: Trade-off between storage and latency. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 1–5.
15. Goseling, J.; Simeone, O.; Popovski, P. Delivery Latency Regions in Fog-RANs with Edge Caching and Cloud Processing. 2017. Available online: http://arxiv.org/abs/1701.06303 (accessed on 23 January 2017).
16. Azimi, S.M.; Simeone, O.; Tandon, R. Online Edge Caching in Fog-Aided Wireless Network. 2017. Available online: http://arxiv.org/abs/1701.06188 (accessed on 22 January 2017).
17. Azimi, S.M.; Simeone, O.; Tandon, R. Fundamental limits on latency in small-cell caching systems: An information-theoretic analysis. In Proceedings of the IEEE Global Communication Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6.
18. Kakar, J.; Gherekhloo, S.; Sezgin, A. Fundamental Limits on Delivery Time in Cloud- and Cache-Aided Heterogeneous Networks. 2017. Available online: http://arxiv.org/abs/1706.07627 (accessed on 23 June 2017).
19. Peng, X.; Shen, J.C.; Zhang, J.; Kang, J.; Letaief, K.B. Joint data assignment and beamforming for backhaul limited caching networks. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Washington, DC, USA, 2–5 September 2014; pp. 1370–1374.
20. Park, S.-H.; Simeone, O.; Shamai, S. Joint optimization of cloud and edge processing for Fog radio access networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 7621–7632.
21. Tao, M.; Chen, E.; Zhou, H.; Yu, W. Content-Centric Sparse Multicast Beamforming for Cache-Enabled Cloud RAN. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 6118–6131.
22. Azari, B.; Simeone, O.; Spagnolini, U.; Tulino, A.M. Hypergraph-based analysis of clustered cooperative beamforming with application to edge caching. *IEEE Wirel. Commun. Lett.* **2016**, *5*, 84–87.
23. Park, S.H.; Simeone, O.; Shamai, S. Joint cloud and edge processing for latency minimization in fog radio access networks. In Proceedings of the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Edinburgh, UK, 3–6 July 2016; pp. 1–5.
24. Martina, V.; Garetto, M.; Leonardi, E. A unified approach to the performance analysis of caching systems. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Toronto, CA, Canada, 27 May–2 April 2014; pp. 2040–2048.

25. Zhu, Y.; Guo, D. Ergodic fading Z-interference channels without state information at transmitters. *IEEE Trans. Inf. Theory* **2011**, *57*, 2627–2647.
26. Vahid, A.; Maddah-Ali, A.S.; Avestimehr, A.S. Capacity results for binary fading interference channels with delayed CSIT. *IEEE Trans. Inf. Theory* **2014**, *60*, 6093–6130.
27. Avestimehr, A.S.; Diggavi, S.N.; Tse, D.N.C. Wireless Network Information Flow: A Deterministic Approach. *IEEE Trans. Inf. Theory* **2011**, *57*, 1872–1905.
28. Tse, D.N.C.; Yates, R.D. Fading Broadcast Channels With State Information at the Receivers. *IEEE Trans. Inf. Theory* **2012**, *58*, 3453–3471.
29. Pedarsani, R.; Maddah-Ali, M.A.; Niesen, U. Online coded caching. *IEEE Trans. Inf. Theory* **2016**, *24*, 836–845.