Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Caching in Heterogeneous Satellite Networks with Fountain Codes

Estefanía Recayte | Francisco Lázaro | Gianluigi Liva

Institute of Communication and Navigation, DLR - German Aerospace Center, Wessling, 82234 Germany [†]

Correspondence
*Estefanía Recayte Email: estefania.recayte@dlr.de

Summary

In this paper we investigate the performance of caching schemes based on fountain codes in a heterogeneous satellite network. We consider multiple cache-aided hubs which are connected to a geostationary satellite through backhaul links. With the aim of reducing the average number of transmissions over the satellite backhaul link, we propose the use of a caching scheme based on fountain codes. We derive a simple analytical expression of the average backhaul transmission rate and provide a tight upper bound on it. Furthermore, we show how the performance of the fountain code based caching scheme is similar to that of a caching scheme based on maximum distance separable codes.

KEYWORDS:
caching, fountain codes, satellite heterogeneous networks, backhaul transmissions

1 | INTRODUCTION

Cache-aided delivery protocols represent a promising solution to counteract the dramatic increase in demand for multimedia content in wireless networks. Caching techniques have been widely studied in literature with the aim of reducing the congestion in the backhaul link, the energy consumption and the latency. In cache-enabled networks content is pre-fetched close to the user during network off-peak periods in order to directly serve the users when the network is congested. In their seminal work, Maddah-Ali *et al.*[1] aim at reducing the transmission rate in a network where each user has an individual cache memory. In their work, the idea of *coded caching* is introduced, i.e. the cache memory does not only provide direct local access to the content but also generates the so called coded multicasting opportunities among users requesting different files, making it possible to serve several users with a single transmission.

Maximum distance separable (MDS) codes have been proposed to minimize the use of the backhaul link during the delivery phase in networks with caches at the transmitter side only[2,3,4]. A delayed offloading scheme based on MDS codes was proposed to spare backhaul link resources in a network with mobile users[5]. Caching schemes leveraging on MDS codes have also been proposed for device to device communication in order to reduce the latency[6].

Codes are classified as *fixed rate* codes when their codeword blocklength is a priory fixed. MDS codes are optimal fixed rate codes in sense that they achieve the Singleton bound. The drawback of MDS codes is that the order of the field used for their construction increases with the blocklength. Furthermore, when using MDS codes in a caching scheme[3] the blocklength (and, possibly, the field order) must be chosen depending on the number of transmitters and the topology of the network. Unlike MDS codes, fountain codes[7] are *rateless*, i.e., their rate can be adapted on-the-fly. This has the advantage of adding flexibility to the network, allowing a dynamic resource management.

[†] "Caching in Heterogeneous Satellite Networks with Fountain Codes", E. Recayte, F. Lázaro, G. Liva, International Journal of Satellite Communications and Networking, Special Issue, John Wiley Sons, Ltd Copyright ©2019

Extensive studies regarding caching for terrestrial applications can be found in the literature. However, the number of works considering caching in heterogeneous satellite networks is limited. Caching has been studied by de Cola et al.[8] in satellite-assisted emergency communications to reduce end-to-end delay. A two-layer caching model for content delivery services in satellite-terrestrial networks has also been proposed[9]. Content placement in LEO satellite constellation networks has been also proposed in order to minimize user terminals content access delay[10]. An off-line caching approach over a hybrid satellite-terrestrial network has also been proposed for reducing the traffic of terrestrial network[11]. However, sparing backhaul resources is of particular importance not only for terrestrial networks but also in satellite systems.

In this paper we propose the application of fountain codes for caching content in satellite heterogeneous networks. We consider a heterogeneous network, in which different transmitter types coexist. In particular, we consider a satellite which acts as central entity and has direct connectivity to the cache-enabled transmitters (hubs). Despite the fact that a satellite architecture is assumed, our work can be adapted to terrestrial heterogeneous networks.

We show how the performance of a fountain code based caching scheme approaches that of a scheme that uses MDS codes in terms of backhaul transmission rate. To this end, among the class of fountain codes, we analyze the performance of linear random fountain codes (LRFCs) which represents a benchmark for extending in future the analysis to other types of fountain codes (i.e. LT codes, raptor codes). In particular, we study and optimize the performance of fountain codes for caching-enabled networks with satellite backhauling. We extend our previous work[12] by introducing a novel and simpler derivation of the average backhaul transmission rate[†]. Furthermore, we derive a new upper bound to the average backhaul transmission rate which is tighter than the one presented in our previous work[12]. Furthermore, we present additional simulation results, which show that the performance of the caching system using binary LRFCs is close to that of a system based on MDS codes when we consider a sufficient number of input symbols.

The rest of the paper is organized as follows. Section 2 introduces the system model, while in Section 3 some preliminaries on LRFCs are presented. The expression of the achievable backhaul rate is presented in Section 4. The optimization problem related to the number of coded symbol to be memorized at each cache is presented in Section 5. In Section 6 the numerical results are presented. Finally, Section 7 contains the conclusions.

2 | SYSTEM MODEL

We consider a two-tier heterogeneous network composed of a geostationary Earth orbit (GEO) satellite, a number of hubs (e.g. terrestrial repeaters or High Altitude Platforms (HAPs)) with cache capabilities and fixed users, as shown in Fig. 1. Each hub is connected to the GEO satellite through a backhaul link. Users are assumed to have a limited antenna gain so that a direct connection to the GEO satellite is not possible. Depending on their location, users may be connected to one or multiple hubs. We denote by γ_h the probability that a user is connected to h hubs, and we assume that γ_h can be easily derived from the geometry of the network.

The GEO has access to a library of n files (e.g. video clips) $\mathcal{F} = \{\mathsf{f}_1 \dots, \mathsf{f}_n\}$, all having identical size. We assume that users request files from the library independently at random. Furthermore, we assume that the probability of file f_j being requested, θ_j, follows a Zipf distribution[13] with parameter α leading to

$$\theta_j = \frac{1/j^{\alpha}}{\sum_{i=1}^{n} 1/i^{\alpha}}. \tag{1}$$

We assume that the caching process in the satellite heterogeneous network consists of two phases: a placement phase and a delivery phase. In the placement phase, each file is fragmented into k *input symbols* (packets) and the GEO satellite encodes each file independently, using a linear random fountain code. During this phase, the GEO satellite fills up each cache by transmitting w_j output symbols from file f_j. Each hub has storage capability for M files, i.e., for Mk packets, such that the following holds for every cache

$$\sum_{j=1}^{n} w_j = Mk.$$

We want to highlight that in our scheme each hub caches for each file the same number of *output symbols* (encoded packets). However, different sets of output symbols are cached at different hubs. We also remark that the placement phase is carried out

[†] We define the average backhaul transmission rate as the average number of coded packets (output symbols) that the GEO needs to send through the backhaul link during the delivery phase to serve the request of a user.

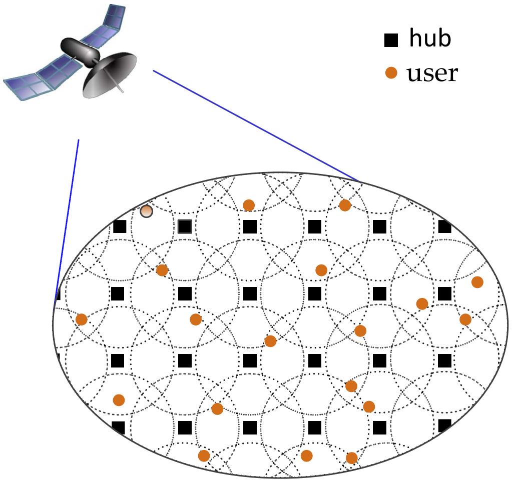

FIGURE 1 System model for a satellite heterogeneous network composed of a satellite, a number of hub transmitters with caching capability and multiple users.

offline. In the delivery phase, users request files at random. In a first stage, the user downloads $m = w_j \cdot h$ different output symbols of f_j cached in the h hubs he is connected to. Whenever the number of symbols received is not enough for decoding f_j successfully, additional output symbols must be sent through the backhaul link to one of the neighbouring hubs, which forwards them to the users. For simplicity we assume that all transmissions are error-free.

3 | LINEAR RANDOM FOUNTAIN CODES

In this work we consider the use of LRFCs for the delivery of the different files in the library. Each file is fragmented into k input symbols, $(u_1, u_2, \dots, u_k)$. For simplicity, we assume that the input symbols u_i belong to a finite field of order q i.e., $u_i \in \mathbb{F}_q$. The case of $u_i \in \mathbb{F}_q^m$, i.e. the case in which packets are m symbols long, can be addressed as a straightforward extension. The LRFC encoder generates a sequence of output symbols $\mathbf{c} = (c_1, c_2, \dots, c_\ell)$, where the number of outputs symbols ℓ can grow indefinitely. In particular, the i-th output symbols is generated as

$$c_i = \sum_{a=1}^{k} g_{a,i}\, u_a,$$

where the coefficients $g_{a,i}$ are picked independently at random with uniform probability in $\mathbb{F}_q$. For fixed ℓ, LRFC encoding can be expressed as a vector matrix multiplication

$$\mathbf{c} = \mathbf{u}\mathbf{G}$$

where $\mathbf{u}$ is the vector of input symbols and $\mathbf{G}$ is a $k \times \ell$ matrix with entry $g_{a,i}$. In order to download a file, a user must collect a set of $m \geq k$ output symbols $\mathbf{y} = (y_1, y_2, \dots, y_m)$. If we denote by $\mathcal{I} = (i_1, i_2, \dots, i_m)$ the set of indices corresponding to the m output symbols collected by the receiver we have

$$y_r = c_{i_r}.$$

The user attempts decoding by solving the system of equations

$$\mathbf{y} = \mathbf{u}\tilde{\mathbf{G}}$$

where $\tilde{\mathbf{G}}$ is a matrix corresponding to the m columns of $\mathbf{G}$ associated to the collected output symbols i.e., the columns of $\mathbf{G}$ with indices in $\mathcal{I}$. If the system of equations admits a unique solution (i.e., if $\tilde{\mathbf{G}}$ is full rank), decoding is declared successful after recovering $\mathbf{u}$, for example by means of Gaussian elimination. If $\tilde{\mathbf{G}}$ is rank deficient, a decoding failure is declared. In the latter case the receiver reattempts decoding after collecting one or more additional output symbols.

Let us define δ as the receiver overhead $\delta = m - k$, that is, the number of output symbols in excess to k that the receiver has collected. Given k, δ and q, the probability of decoding failure of an LRFC is given by

$$P_f(k,\delta,q) = 1 - \prod_{i=1}^{k}\left(1 - \frac{q^{i-1}}{q^{k+\delta}}\right)$$

and can be tightly lower and upper bounded as[14]

$$l(\delta,q) \leq P_f(k,\delta,q) < u(\delta,q) \tag{2}$$

where

$$l(\delta,q) := q^{-\delta-1}$$

and

$$u(\delta,q) := \frac{1}{q-1}q^{-\delta}.$$

Note that the bounds are independent from the number of input symbols k and become tighter for increasing q.

For notational convenience, in the remaining of the paper we denote $P_f(k,\delta,q)$ by $P_F(\delta)$, that is, we do not indicate explicitly the dependency on k and q.

4 | AVERAGE BACKHAUL TRANSMISSION RATE

We define the average backhaul transmission rate $\mathbb{E}[T]$ as the average number output coded symbols that the GEO satellite has to transmit during the delivery phase in order to fulfill a user request.

4.1 | Overhead Decoding Probability

We define the overhead decoding probability P_δ as the probability that a user needs *exactly* $k+\delta$ coded symbols to successfully decode the requested file. Hence, P_δ corresponds to the probability that the matrix $\tilde{\mathbf{G}}$ is full rank when $m = k+\delta$ output symbols have been collected, conditioned to the fact that matrix $\tilde{\mathbf{G}}$ was not full rank when $m = k + \delta - 1$ symbols had been collected. For a fixed number of input symbols k and a fixed field order q, P_δ can be written as

$$P_\delta = P_F(\delta - 1) - P_F(\delta).$$

4.2 | Overhead Average

Let us denote as Δ the random variable associated to the average number of symbols in excess to k that a user needs in order to recover the requested content and let us also denote as δ its realization. We can calculate the average overhead as follows

$$\begin{aligned}\mathbb{E}[\Delta] &= \sum_{\delta=1}^{\infty}\delta \cdot \left[P_F(\delta-1) - P_F(\delta)\right]\\ &= \sum_{\delta=0}^{\infty}(\delta+1)\cdot P_F(\delta) - \sum_{\delta=0}^{\infty}\delta \cdot P_F(\delta)\\ &= \sum_{\delta=0}^{\infty}P_F(\delta).\end{aligned} \tag{3}$$

By using (2) in (3), $\mathbb{E}[\Delta]$ can be upper bounded as

$$\begin{aligned}\mathbb{E}[\Delta] &< \sum_{\delta=0}^{\infty}\frac{q^{-\delta}}{q-1}\\ &\overset{\text{(a)}}{=} \frac{q}{(q-1)^2} := \delta_u.\end{aligned} \tag{4}$$

4.3 | Backhaul Rate

Let us now consider a generic user requesting a file. Let Z be the random variable associated to the number of output symbols from the requested file which are available at the hubs that the user is connected to. Also let z be the realization of Z. Let H be the random variable associated to the number of hubs a user is connected to, being h its realization. Finally, let J be the random variable associated to the index of the file requested by the user, being j its realization. We have

$$P_{Z|J,H}(z|j,h) = \begin{cases} 1 & \text{if } z = w_j\, h \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where we recall that w_j stands for the number of coded symbols from file j stored in every hub. The probability mass function of Z is

$$\begin{aligned} P_Z(z) &= \sum_j \sum_h P_{Z|J,H}(z|j,h) P_J(j) P_H(h) \\ &= \sum_j \sum_h \theta_j\, \gamma_h\, P_{Z|J,H}(z|j,h). \end{aligned} \tag{6}$$

We are interested in deriving the average backhaul transmission rate, i.e, the average number of output symbols which have to be sent over the backhaul link to serve the request of a user, which we denote by random variable T. The conditional probability of T given Z corresponds to the decoding success probability when exactly t output symbols have been received from the backhaul link. A user collects in total $m = z + t$ output symbols where z symbols are transmitted from the caches through local links. In order to derive $P_{T|Z}(t|z)$ we shall distinguish two cases.
If $z \le k$, then

$$P_{T|Z}(t|z) = \begin{cases} P_F(z-k+t-1) - P_F(z-k+t) & \text{if } t \ge 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

If $z > k$, we have

$$P_{T|Z}(t|z) = \begin{cases} 1 - P_F(z-k) & \text{if } t = 0, \\ P_F(z-k+t-1) - P_F(z-k+t) & \text{if t} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The expectation $\mathbb{E}[T]$ is given by

$$\mathbb{E}[T] = \sum_t t \left(\sum_{z=0}^{k} P_{T|Z}(t|z) P_Z(z) + \sum_{z=k+1}^{\infty} P_{T|Z}(t|z) P_Z(z) \right). \tag{9}$$

As shown in the Appendix, it is possible to upper bound (9) as

$$\mathbb{E}[T] \le \delta_u + \sum_{z=0}^{k} (k-z) P_Z(z). \tag{10}$$

5 | LRFC PLACEMENT OPTIMIZATION PROBLEM

The LRFC placement problem calls for minimizing the average backhaul transmission rate during the delivery phase. In particular, we would like to determine the number of coded symbols per file that each hub has to cache, so that the average backhaul transmission rate is minimized. We present in this section the placement optimization problem adapted to a LRFC cached scheme based on the optimization problem proposed for MDS codes in[3].

TABLE 1 Average overhead required for successful decoding for a LRFC

q	$\mathbb{E}[\Delta]$	δ_u
2	1.6047	2
4	0.4211	0.4444
8	0.1610	0.1633
16	0.0708	0.0711
32	0.0333	0.0333
64	0.0161	0.0161
128	0.0079	0.0079

The optimization problem can be written as

$$\begin{aligned} \min_{w_1,\ldots,w_n} \quad & \mathbb{E}[T] \\ \text{subject to} \quad & \sum_{j=1}^{n} w_j = Mk \\ & w_j \in \mathbb{N}_0. \end{aligned} \tag{11}$$

The first constraint specifies that the total number of stored coded symbols should be equal to the size cache. The second constraints accounts for the discrete nature of the optimization variable.

Solving exactly the optimization problem requires evaluating (9), which is complex. Hence, as an alternative to minimizing the average backhaul transmission rate, we propose minimizing its upper bound in (23), which leads to the following optimization problem

$$\begin{aligned} \min_{w_1,\ldots,w_n} \quad & \left[\delta_u + \sum_{z=0}^{k}(k-z)P_Z(z)\right] \\ \text{subject to} \quad & \sum_{j=1}^{n} w_j = Mk \\ & w_j \in \mathbb{N}_0. \end{aligned} \tag{12}$$

Since the upper bound on $\mathbb{E}[T]$ in (23) relies on the upper bound in (2), which is tight, we expect the result of the optimization problem in (12) to be close to the result of the optimization problem in (11).

6 | RESULTS

In this section, we numerically evaluate the normalized average backhaul transmission rate, which we define as

$$\widehat{T} = \frac{\mathbb{E}[T]}{k}.$$

In all the setups, we consider that users are uniformly distributed within the coverage area of the satellite and border effects are neglected. We consider that each hub covers a circular area of radius r centered around the hub. For simplicity, we assume that the hubs are arranged according to a uniform two dimensional grid, with spacing d. Unless otherwise specified, we assume $r = 60$ km and $d = 45$ km. Thus, the coverage areas of different hubs partially overlap, as it can be observed in Fig. 1. With geometrical calculations the following connectivity distribution can be obtained

$$\begin{aligned} \gamma_1 = 0.2907,\ \gamma_2 = 0.6591, \\ \gamma_3 = 0.0430,\ \gamma_4 = 0.0072. \end{aligned} \tag{13}$$

We first evaluate the tightness of the upper bound (4) on the average overhead. Table 1 shows $\mathbb{E}[\Delta]$ for different values of q. The values in the second column were numerically derived from equation (3) while values in the third column were derived from the bound in equation (4). We can see that the bound becomes tighter for increasing q.

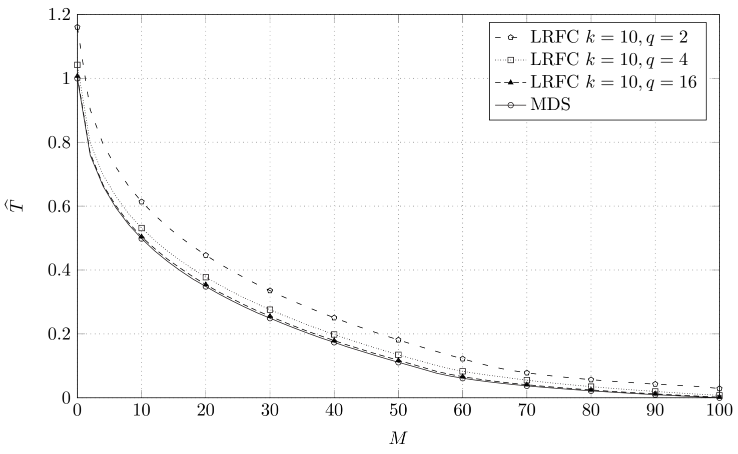

FIGURE 2 Normalized average backhaul transmission rate as a function of memory size M for LRFC codes over $\mathbb{F}_q$ for $q = 2, 4, 16$ with $k = 10$ and for an MDS code, given $n = 100$, $\alpha = 0.8$ and $\gamma_1 = 0.2907, \gamma_2 = 0.6591, \gamma_3 = 0.0430, \gamma_4 = 0.0072$. Solid curves represent LRFC schemes while the dashed curve represents the MDS scheme.

In the first scenario, we study the impact of the cache size M on the average backhaul transmission rate. In this setup, we consider the connectivity distribution given in (13) and file popularity distribution given by (1) with parameter $\alpha = 0.8$. The library size is set to $n = 100$. We optimized the number of LRFC coded symbols w_j cached at each hub by solving the problem (12) for $q = 2$, $q = 4$ and $q = 16$, and we calculated numerically the average backhaul transmission rate of our fountain coding caching scheme by applying (22). As a benchmark, we used the MDS caching scheme from Bioglio et al.[3]. We would also like to remark that the performance of a scheme without caching is characterized in our setting by $\widehat{T} = 1$, since all the content has to be transmitted through the backhaul link.

In Fig. 2 the normalized average backhaul transmission rate is shown as a function of the memory size M when each file is fragmented into $k = 10$ input symbols. We can observe how the penalty on the average rate for using LRFC with respect to a MDS code becomes smaller for increasing q and for $q = 16$ is almost negligible. We remark that for $M = 100$ the cache size coincides with the library size, hence, the backhaul rate for the MDS scheme becomes zero, whereas for the LRFC schemes the average backhaul transmission rate coincides with the average overhead. Note that since a MDS code achieves the best possible performance, this result shows implicitly that solving the optimization problem in (12) yields a solution that is close to that of solving the optimization problem in (11). We further observe that LRFC caching with storage capabilities equal to 10% of the library size can reduce the average backhaul rate for at least 40% with respect to a system with no caching ($M = 0$).

Next, we consider the same parameters as the previous scenario but we evaluate the average backhaul transmission rate when each file is fragmented into $k = 100$ input symbols. In Fig. 3 we show the impact of memory size M on the normalized average backhaul transmission rate for different code caching schemes when $k = 100$. We can observe that the optimal cache placing computed for an LRFC caching scheme built in a field of order $q = 2$ reaches the performance of the MDS scheme and slightly better performance is given when $q = 4$. Thus, the LRFC caching scheme can approach the best performance in the binary field just by increasing the number of input symbols.

For the same connectivity distribution, library size $n = 100$ and fixed memory at each cache of $M = 10$, we investigate how the file distribution impacts on the average backhaul transmission rate. In Fig. 4 the normalized average backhaul transmission rate is shown as a function of the file parameter distribution α. As expected, when α increases, caching schemes become more efficient since the majority of the requests is concentrated in a small number of files. Looking at the figure we can observe how for $\alpha = 0.2$, a LRFC in $\mathbb{F}_2$ with $k = 10$ requires roughly 12% more transmissions over backhaul link than a LRFC in $\mathbb{F}_{32}$ with $k = 10$. We can also observe that a binary LRFC with $k = 100$ performs as good as the optimal scheme. For $\alpha = 1.5$ the LRFC of order $q = 2$ with $k = 10$ requires only 4.7% more than in $q = 2$ with $k = 100$.

In our next setup we consider $\alpha = 0.8$, $M = 10$, $k = 10$ or $k = 100$ and the distribution given in (13). We evaluate the average backhaul transmission rate for different cardinalities of the library. In Fig. 5 the normalized average backhaul transmission rate

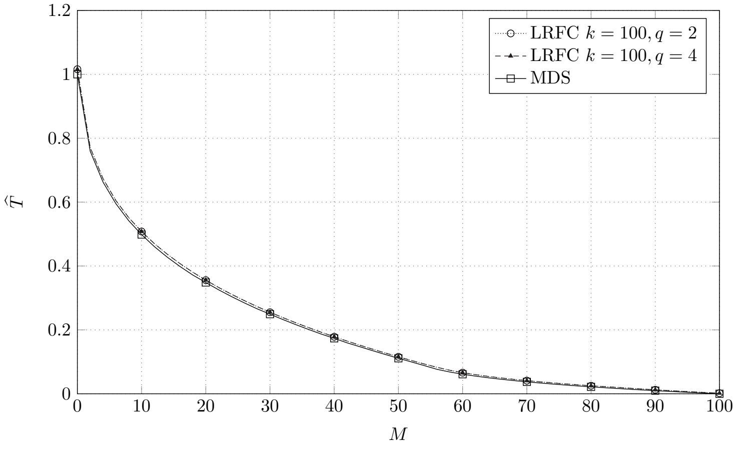

FIGURE 3 Normalized average backhaul transmission rate as a function of memory size M for LRFC codes over $\mathbb{F}_q$ for $q = 2, 4$ with $k = 100$ for and for an MDS code, given $n = 100$, $\alpha = 0.8$ and $\gamma_1 = 0.2907$, $\gamma_2 = 0.6591$, $\gamma_3 = 0.0430$, $\gamma_4 = 0.0072$.

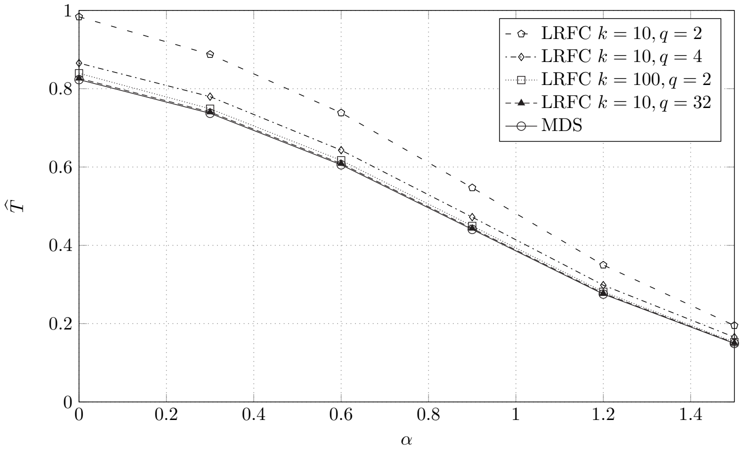

FIGURE 4 Normalized average backhaul transmission rate as a function of the file parameter distribution α for LRFC codes over $\mathbb{F}_q$ for $q = 2, 4, 32$ with $k = 10$ or $k = 100$ and for an MDS code, given $n = 100$, $M = 10$ $\gamma_1 = 0.2907$, $\gamma_2 = 0.6591$, $\gamma_3 = 0.0430$, $\gamma_4 = 0.0072$.

is shown as a function of the library size. For a fixed memory size the average backhaul transmission rate increases as the library size increases. As it can be observed, also in this case the proposed LRFC caching scheme performs similarly to a MDS scheme.

In our last setup we consider $M = 10$, $k = 100$ and $\alpha = 0.8$. We compute the average backhaul transmission rate in function of the radius of coverage r of the hubs. The plot in Figure 6 shows that gain of using a MDS caching scheme with respect of the binary LRFC caching scheme is at maximum of the 0.0126% when the hubs has radius of coverage $r = 32$ km and becomes smaller for larger values.

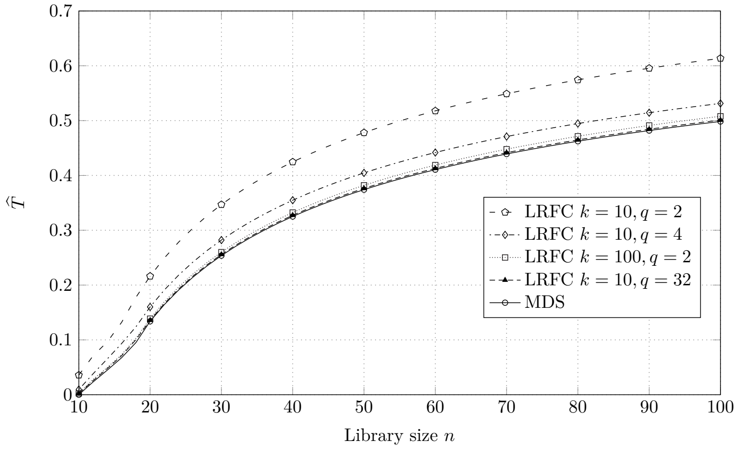

FIGURE 5 Normalized average backhaul transmission rate as a function of the library size n for LRFC codes over $\mathbb{F}_q$ for $q = 2, 4, 32$ with $k = 10$ and for an MDS code, given $\alpha = 0.8$, $M = 10$ and $\gamma_1 = 0.2907, \gamma_2 = 0.6591, \gamma_3 = 0.0430, \gamma_4 = 0.0072$.

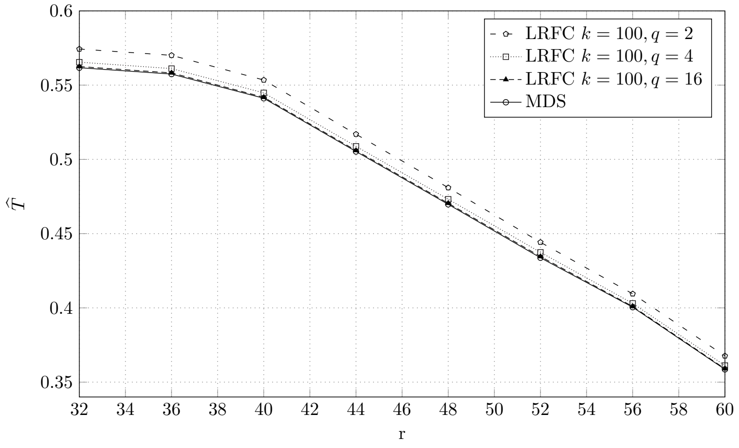

FIGURE 6 Normalized average backhaul transmission rate as a function of coverage radius r for LRFC codes over $\mathbb{F}_q$ for $q = 2, 4, 16$ with $k = 100$ and for an MDS code, given $\alpha = 0.8$ and $M = 10$.

7 | CONCLUSIONS

In this work, we analyze a caching scheme based on fountain codes, which are characterized by a high degree of flexibility. In particular, we derived the analytical expression of the average backhaul transmission rate and a simple yet tight upper bound to it. The optimization problem related to which content to place in the cache memories was formalized and solved. Numerical results were presented comparing the fountain code caching scheme with a caching scheme based on MDS codes available in literature. The results indicate that the fountain code caching scheme approaches the performance of the MDS caching scheme when the number of input symbols is high, or when the fountain code is built over finite field of moderate high order.

APPENDIX

Let us define $\bar{T}_1$ and $\bar{T}_2$ as

$$\bar{T}_1 := \sum_t t \sum_{z=0}^{k} P_{T|Z}(t|z) P_Z(z)$$
$$\bar{T}_2 := \sum_t t \sum_{z=k+1}^{\infty} P_{T|Z}(t|z) P_Z(z)$$

so that

$$\mathbb{E}[T] = \bar{T}_1 + \bar{T}_2. \tag{14}$$

If we introduce the variable change $\delta = z - k + t$ in the expression of $\bar{T}_1$, we obtain

$$\begin{aligned}
\bar{T}_1 &= \sum_{z=0}^{k} P_Z(z) \sum_{\delta=z-k}^{\infty} (\delta - z + k)\Big[P_F(\delta-1) - P_F(\delta)\Big] \\
&\overset{(a)}{=} \sum_{z=0}^{k} P_Z(z) \sum_{\delta=0}^{\infty} (\delta - z + k)\Big[P_F(\delta-1) - P_F(\delta)\Big] \\
&= \sum_{z=0}^{k} P_Z(z)\left(\sum_{\delta=0}^{\infty} \delta\Big[P_F(\delta-1) - P_F(\delta)\Big] + (k-z)\sum_{\delta=0}^{\infty}\Big[P_F(\delta-1) - P_F(\delta)\Big]\right) \\
&\overset{(b)}{=} \sum_{z=0}^{k} P_Z(z)\Big(\mathbb{E}[\Delta] + k - z\Big) \\
&= (\mathbb{E}[\Delta] + k)\Pr\{Z \le k\} - \sum_{z=0}^{k} z P_Z(z)
\end{aligned} \tag{15}$$

where equality (a) is due to $[P_F(\delta-1) - P_F(\delta)] = 0$ for $\delta < 0$, and equality (b) is due to

$$\sum_{\delta=0}^{\infty}[P_F(\delta-1) - P_F(\delta)] = 1.$$

If we introduce the same variable change in the expression of $\bar{T}_2$ we have

$$\begin{aligned}
\bar{T}_2 &= \sum_{z=k+1}^{\infty} P_Z(z) \sum_{\delta=z-k}^{\infty} (\delta - z + k)\Big[P_F(\delta-1) - P_F(\delta)\Big] \\
&= \sum_{z=k+1}^{\infty} P_Z(z)\left(\sum_{\delta=z-k}^{\infty} \delta\Big[P_F(\delta-1) - P_F(\delta)\Big] + \sum_{\delta=z-k}^{\infty} (k-z)\Big[P_F(\delta-1) - P_F(\delta)\Big]\right).
\end{aligned} \tag{16}$$

Let us rewrite (16) as follow

$$\bar{T}_2 = \sum_{z=k+1}^{\infty} P_Z(z)\Big(\bar{T}_{21}(z) + \bar{T}_{22}(z)\Big) \tag{17}$$

where

$$\begin{aligned}\bar{T}_{21}(z) &= \sum_{\delta=z-k}^{\infty} \delta\Big[P_F(\delta-1) - P_F(\delta)\Big] \\ &= \sum_{\delta=0}^{\infty} \delta\Big[P_F(\delta-1) - P_F(\delta)\Big] - \sum_{\delta=0}^{z-k-1} \delta\Big[P_F(\delta-1) - P_F(\delta)\Big] \\ &= \sum_{\delta=0}^{\infty} P_F(\delta) - \left[\sum_{\delta=0}^{z-k-1} P_F(\delta) - (z-k)P_F(z-k-1)\right] \\ &= \mathbb{E}[\Delta] - \sum_{\delta=0}^{z-k-1} P_F(\delta) + (z-k)P_F(z-k-1) \end{aligned} \tag{18}$$

and

$$\begin{aligned}\bar{T}_{22}(z) &= \sum_{\delta=z-k}^{\infty} (k-z)\Big[P_F(\delta-1) - P_F(\delta)\Big] \\ &= (k-z)\left\{\sum_{\delta=0}^{\infty} \Big[P_F(\delta-1) - P_F(\delta)\Big] - \sum_{\delta=0}^{z-k-1} \Big[P_F(\delta-1) - P_F(\delta)\Big]\right\} \\ &= (k-z)\Big\{1 - \Big[1 - P_F(z-k-1)\Big]\Big\} \\ &= (k-z)P_F(z-k-1). \end{aligned} \tag{19}$$

By inserting (18) and (19) in (17) and sum we obtain

$$\bar{T}_2 = \sum_{z=k+1}^{\infty} P_Z(z)\Big[\mathbb{E}[\Delta] - \sum_{\delta=0}^{z-k-1} P_F(\delta)\Big] \tag{20}$$

$$\leq \sum_{z=k+1}^{\infty} P_Z(z)\mathbb{E}[\Delta] \tag{21}$$

where the inequality is due to

$$\sum_{\delta=0}^{z-k-1} P_F(\delta) \geq 0.$$

If we replace (15) and (20) in (14), the expression of the average backhaul transmission rate becomes

$$\begin{aligned}\mathbb{E}[T] &= \Big(\mathbb{E}[\Delta] + k\Big)\Pr\{Z \leq k\} - \sum_{z=0}^{k} zP_Z(z) + \sum_{z=k+1}^{\infty} P_Z(z)\Big[\mathbb{E}[\Delta] - \sum_{\delta=0}^{z-k-1} P_F(\delta)\Big] \\ &= \mathbb{E}[\Delta] + \sum_{z=0}^{k}(k-z)P_Z(z) - \sum_{z=k+1}^{\infty} P_Z(z)\left[\sum_{\delta=0}^{z-k-1} P_F(\delta)\right]. \end{aligned} \tag{22}$$

Finally, we can upper bound the average transmission rate by use of (4) and (21) in (22) as

$$\mathbb{E}[T] \leq \delta_u + \sum_{z=0}^{k}(k-z)P_Z(z). \tag{23}$$

References

1. Maddah-Ali M, Niese U. "Cache-aided interference channels". In: Proc. of IEEE Int. Symp. on Inf. Theory (ISIT); 2015; Hong Kong, China: 809–813.

2. Shanmugam K, Golrezaei B, Dimakis A, Malisch A, Caire G. "Fempto Caching: wireless content delivery through distributed caching helpers". *IEEE Trans. Inf. Theory* 2013; 59(12): 8402-8413.

3. Bioglio V, Gabry F, Land I. "Optimizing MDS codes for caching at the edge". In: Proc. IEEE Global Comm. Conf. (GLOBECOM); 2015; San Diego, USA.

4. Liao J, Wong K, Khandaker M, Zheng Z. "Optimizing cache placement for heterogeneous small cell networks". *IEEE Commun. Lett.* 2017; 21(1): 120-123.

5. Ozafatura E, Gündüz E. "Mobility and popularity-aware coded small-cell caching". *IEEE Commun. Lett.* 2018; 22(2): 288-291.

6. Piemontese A, Graell i Amat A. "MDS-coded distributed storage for low delay wireless content delivery". In: 2016 9th Int. Symp. on Turbo Codes and Iterative Inf. Processing (ISTC); 2016; Brest, France.

7. MacKay D. "Fountain Codes". *IEEE Proc. Commun.* 2005; 152: 1062-1068.

8. De Cola T, Gonzalez G, Vicente E. Mujica V. "Applicability of ICN-based network architectures to satellite-assisted emergency communications". In: Proc. IEEE Global Comm. Conf. (GLOBECOM); 2016; Washington, DC, USA.

9. Wu H, Li J, Lu H, Hong P. "A two-layer caching model for content delivery services in satellite-terrestrial networks". In: Proc. IEEE Global Comm. Conf. (GLOBECOM); 2016; Washington, DC, USA.

10. Liu S, Hu X, Wang Y, Cui G, Wang W. "Distributed caching based on matching game in LEO satellite constellation networks". *IEEE Commun. Lett.* 2018; 22(2): 300-303.

11. Kalantari H, Fittipaldi M, Chatzinotas S, Vu T, Ottersten B. "Cache-assisted hybrid satellite-terrestrial backhauling for 5G cellular networks". In: Proc. IEEE Global Comm. Conf. (GLOBECOM); 2017; Singapore, Singapore.

12. Recayte E, Lázaro F, Liva G. "Caching at the edge with fountain codes". In: Proc. of Advanced Sat. Mobile Systems Conf. (ASMS); 2018; Berlin, Germany.

13. Breslau L, Pei Cao , Li Fan , Phillips G, Shenker S. "Web caching and Zipf-like distributions: evidence and implications". In: Proc. of IEEE Conf. on Computer Commun. (INFOCOM); 1999; New York, NY, USA: 126–134.

14. Liva G, Paolini E, Chiani M. "Performance versus overhead for fountain codes over $\mathbb{F}_q$". *IEEE Commun. Lett.* 2018; 14(2): 178-180.

□