Improved Lower Bounds for Coded Caching

Hooshang Ghasemi and Aditya Ramamoorthy, *Member, IEEE*

***Abstract*— Caching is often used in content delivery networks as a mechanism for reducing network traffic. Recently, the technique of coded caching was introduced whereby coding in the caches and coded transmission signals from the central server were considered. Prior results in this area demonstrate that carefully designing the placement of content in the caches and designing appropriate coded delivery signals from the server allow for a system where the delivery rates can be significantly smaller than conventional schemes. However, matching upper and lower bounds on the transmission rate have not yet been obtained. In this paper, we derive tighter lower bounds on the coded caching rate than were known previously. We demonstrate that this problem can equivalently be posed as a combinatorial problem of optimally labeling the leaves of a directed tree. Our proposed labeling algorithm allows for significantly improved lower bounds on the coded caching rate. Furthermore, we study certain structural properties of our algorithm that allow us to analytically quantify improvements on the rate lower bound for general values of the problem parameters. This allows us to obtain a multiplicative gap of at most four between the achievable rate and our lower bound.**

***Index Terms*— Coded caching, directed tree, optimal labeling, lower bounds, multiplicative gap.**

I. Introduction

CONTENT distribution over the Internet is an important problem and is the core business of several enterprises such as YouTube, Netflix, Hulu etc. The operation of such large scale systems presents several challenges, including (but not limited to) storage of the data, ensuring reliable availability and efficient content delivery. One commonly used technique to facilitate delivery is content caching [1]. The main idea in "conventional content caching" is to store relatively popular content in local memory either on the desired device or in a device at the edge of the network such as an intermediate router. This local memory is referred to as the cache. Upon request, this cached content is used to serve the clients, thus reducing the number of bits transmitted from the server and thereby reducing overall network congestion. Note that even web browsers, routinely cache the content of popular websites on a local machine to speed up the loading of webpages.

Manuscript received February 12, 2016; revised January 5, 2017; accepted April 7, 2017. Date of publication May 17, 2017; date of current version June 14, 2017. This work was supported by NSF under Grant CCF-1320416 and Grant CCF-1149860. This paper was presented in part at the 2015 IEEE International Symposium on Information Theory and the 2016 IEEE International Symposium on Information Theory.

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: ghasemi@iastate.edu; adityar@iastate.edu).

Communicated by K. Narayanan, Associate Editor for Coding Techniques.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2017.2705166

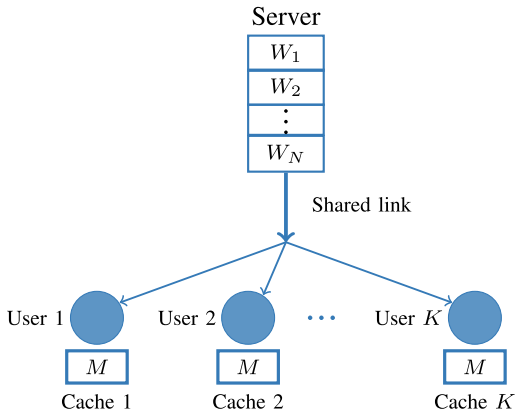

Fig. 1. Block diagram of coded caching system.

Historically, content caching algorithms have attempted to optimize the placement of content in the caches so that the average number of bits that are transmitted from the central server to the end users is minimized [2]–[5]. This often requires some knowledge on the popularity of file requests [6]–[8] made by the users. Moreover, the typical approach is to cache a certain fraction of the file and to obtain the remaining parts from the server when the need arises. Coding in the content of the cache and/or coding in the transmission from the server are typically not considered.

The work of [9] introduced the problem of coded caching, where there is a server with N files and K users each with a cache of size M. The users are connected to the server by a shared link (see Fig. 1). In each time slot each user requests one of the N files. There are two distinct phases in coded caching.

- *Placement phase.* In this phase, the content of caches is populated. This phase should not depend on the actual user requests (which are assumed to be arbitrary). Typically, the placement phase can be executed in the *off-peak* hours where the amount of network traffic is low.
- *Delivery phase.* In this phase, each of the K users request one of the N files. The server transmits a signal of rate R over the shared link that simultaneously serves to satisfy the demands of each of the users.

The work of [9] demonstrates that a carefully designed placement scheme and a corresponding delivery scheme achieves a rate that is significantly lower than conventional caching. While coded caching promises very significant gains in transmission rates, at this point we do not have matching upper and lower bounds on the (R, M) pairs for a given N and K.

In this work our main contribution is in developing improved lower bounds on the required rate for the coded

caching problem. We demonstrate that the computation of this lower bound can be posed as a combinatorial labeling problem on a directed tree. In particular, our method generates lower bounds on $\alpha R + \beta M$, where α and β are positive integers. We demonstrate that a careful analysis of the underlying combinatorial structure of the problem allows us to obtain significantly better lower bounds than those obtained in prior work [9]–[11]. In addition, our machinery allows us to show that the achievable rate of [9] is within a multiplicative factor of four of our proposed lower bound.

This paper is organized as follows. Section II discusses the background, related work and summarizes the main contributions of our work. Section III presents our proposed lower bound technique. The multiplicative gap between the achievable rate and our lower bound is outlined in Section IV. Our proposed strategy also applies to certain variants of the coded caching problem that have been discussed in the literature; this is explained in Section V. There have been some other approaches presented in the literature [9]–[11] for improving the lower bound on the coded caching rate. We present comparisons between our approach and the other approaches in Section VI. We conclude the paper with a discussion of opportunities for future work in Section VII.

II. Background, Related Work and Summary of Contributions

In a coded caching system there is a server that contains N files, denoted $W_i, i = 1, \ldots, N$, each of size F bits. There are K users that are connected to the central server by means of a shared link. Each user has a local cache memory of size MF bits; we denote the cache content by the symbol Z_i (which is a function of $W_1, \ldots, W_N$). In each time slot, the i-th user demands the file W_{d_i} where $d_i \in \{1, \ldots, N\}$. The coded caching problem has two distinct phases. In the *placement phase*, the content of caches is populated; this phase should not depend on the actual user requests (which are assumed to be arbitrary). In the *delivery phase*, the server transmits a potentially coded signal that serves to satisfy the demands of each of the users. A pair (M, R) is said to be achievable if for every possible request pattern (there are N^K of them), every user can recover its desired file with high probability for large enough F. We let $R^\star(M)$ denote the infimum of all such achievable rates for a given M.

The coded caching problem can be formally described as follows. Let $[m] = \{1, \ldots, m\}$, where m is a positive integer. Let $\{W_n\}_{n=1}^N$ denote N independent random variables (representing the files) each uniformly distributed over $[2^F]$. The i-th user requests the file W_{d_i}, where $d_i \in [N]$. A (M, R) system consists of the following.

- K caching functions, $Z_i \triangleq \phi_i(W_1, \ldots, W_N)$ where $\phi_i : [2^F] \to [2^{\lfloor FM \rfloor}]$.
- A total of N^K encoding functions $\varphi_{d_1,\ldots,d_K}(W_1, \ldots, W_N)$, so that the delivery phase signal $X_{d_1,\ldots,d_K} \triangleq \varphi_{d_1,\ldots,d_K}(W_1, \ldots, W_N)$. Here, $\varphi_{d_1,\ldots,d_K} : [2^F]^N \to [2^{\lfloor FR \rfloor}]$.
- For each delivery phase signal and each user, we define appropriate decoding functions. There are a total of KN^K of them. For the k-th user, we define $\mu_{d_1,\ldots,d_K;k}(X_{d_1,\ldots,d_K}, Z_k)$, where $k = 1, \ldots, K$ so that decoded file $\hat{W}_{d_1,\ldots,d_K;k} \triangleq \mu_{d_1,\ldots,d_K;k}(X_{d_1,\ldots,d_K}, Z_k)$. Here $\mu_{d_1,\ldots,d_K;k} : [2^{\lfloor RF \rfloor}] \times [2^{\lfloor FM \rfloor}] \to [2^F]$.

The probability of error is defined as

$$\max_{(d_1,\ldots,d_K)\in[N]^K} \max_{k\in[K]} P(\hat{W}_{d_1,\ldots,d_K;k} \neq W_{d_k}).$$

Definition 1: The pair (M, R) is said to be achievable if for $\epsilon > 0$, there exists a file size F large enough so that there exists a (M, R) caching scheme with probability of error at most ϵ. We define

$$R^\star(M) = \inf\{R : (M, R) \text{ is achievable}\}.$$

In this setting, it is not too hard to see that the best that a conventional caching system can do is to simply store an M/N fraction of each file in each of the caches. In order to satisfy the demands of the user, the server has to transmit the remaining $(1 - M/N)$ fraction of each of the requested files. Thus, the transmission rate (normalized by F) is given by

$$R_U(M) = \min(N, K)\left(1 - \frac{M}{N}\right). \tag{1}$$

Note that $\min(N, K)$ is the transmission rate in the absence of any caching. In [9], the factor $(1 - M/N)$ is referred to as the *local caching gain* as it is gain that is obtained purely from the cache, without any optimization of the transmission from the server. In the setting where we perform nontrivial coding in the cache and delivery phase encoding functions, [9] demonstrates that a carefully designed placement scheme and a corresponding delivery scheme achieves a rate

$$R_C(M) = K\left(1 - \frac{M}{N}\right) \cdot \min\left\{\frac{1}{1 + KM/N}, \frac{N}{K}\right\}, \tag{2}$$

where $M \in \{0, N/K, 2N/K, \ldots, N\}$. Other values of M are obtained by time-sharing between the solutions for integer multiples of N/K.

The factor $\frac{1}{1+KM/N}$ which definitely dominates when $N \geq K$ is referred to as the *global caching gain*. It is to be noted that the global caching gain depends on the overall cache size across all the users (owing to the term KM/N in the denominator) whereas the local caching gain only depends on the per-user cache size (owing to the term $1 - M/N$). Furthermore, they compare their achievable rate (*cf.* eq. (2)) to a cutset bound that can be expressed as follows.

$$R^\star(M) \geq \max_{s\in\{1,\ldots,\min(N,K)\}} \left(s - \frac{s}{\lfloor N/s \rfloor}M\right). \tag{3}$$

The work of [9] also shows that the rate $R_C(M)$ is within a factor of 12 of the cutset bound for all values of N, K and M.

A. Related Work

Coded caching is related to but different from the index coding problem [12]. In the index coding problem, there are N' sources such that i-th source has message W_i, $i = 1, \ldots, N'$. There are K terminals, each of which has some subset of $\{W_1, \ldots, W_{N'}\}$ available. In addition,

each terminal requests a certain subset of the messages $\{W_1, \ldots, W_{N'}\}$. The aim in the index coding problem is to minimize the number of bits that are transmitted on the shared link so that the demands of each user are satisfied. It is well recognized that the index coding problem for arbitrary side information is a computationally hard problem where nonlinear codes may be necessary [12], [13]. In particular, the optimal *linear* index code corresponds to minimizing the rank of an appropriately defined matrix over a finite field. This so called minrank problem [12] is also known to be computationally hard. It can be observed that for a fixed but *uncoded* cache content and a fixed set of demands of the various users, the problem of determining the optimal delivery phase signal in the coded caching problem is equivalent to an index coding problem. Note however, that in the coded caching problem, we allow the cache content to be coded.

Since the original work of [9], there have been several aspects of coded caching that have been investigated. Reference [14] considers the scenario of decentralized caching where the placement phase is driven by the users who randomly populate their caches with subsets of the files stored at the server. Approaches for updating the cache content are considered in [15] and the case of files with different popularity scores are considered in [16]–[18]. Security issues in this domain are considered in [19]. The work of [20] considers the more general case of hierarchical coded caching, where certain intermediate nodes in the network are equipped with potentially larger caches and investigates methods for minimizing the overall traffic in such networks (see also [21]). Coded caching where each user requests multiple files was investigated in [22]. The case of device-to-device (D2D) wireless networks where there is no central server was examined in [23] and [24]. Systems with files of differing sizes were examined in [25]. The work of [26]–[28], considers the problem of leveraging the rate gains of coded caching with reduced subpacketization levels. Synchronization issues and the problem setting where end users have deadlines was investigated in [29] and [30].

In addition to these contributions, there have been other lines of work that deal with content caching. In a parallel line of work [23], [31], [32] consider the problem of femtocaching in a wireless setting where in addition to a central server (or base station), there are helpers (with caches) interspersed in a cell that help the end users satisfy their demands. The goal is again to consider caching strategies that minimize the overall rate, but the solution approaches do not consider the worst case rate over all possible demand patterns; instead the popularity scores of the different files are explicitly taken into account. Moreover, while coding is considered, it is conceptually different in the sense that the coding is only restricted to parts of the same file and coding across different files is not considered. More recently, techniques inspired by coded caching have been employed for speeding up distributed computing [33], [34].

There has also been parallel work on establishing lower bounds for the coded caching problem. In [10], Han's inequality [35] was leveraged to obtain an improved lower bound. A multiplicative gap of eight between their lower bound and the achievable rate in eq. (2) was established. The work of [11] also presents a lower bound technique. As discussed in Section VI, their technique can be considered as a special case of our work. The specific case of $N = K = 3$ was considered in [36] via a computational approach. We present a detailed comparison of our technique with these other approaches in Section VI.

B. Summary of Our Contributions

In this work our main contribution is in developing improved lower bounds on the required rate for the coded caching problem. We show that the cutset based bound in eq. (3) is significantly loose and propose a larger class of lower bounds that are significantly tighter. Our specific contributions include the following.

- We demonstrate that the computation of our lower bound can be posed as a combinatorial labeling problem on a directed tree. Our method generates lower bounds on $\alpha R^\star + \beta M$, where α, β are positive integers. While the cutset bound only optimizes over at most $\min(N, K)$ choices, our technique allows us to consider many more (α, β) pairs.[1]
- We perform a careful analysis of the underlying combinatorial structure of the problem that allows us to obtain significantly better lower bounds than those obtained in prior work. For a given pair (α, β) and number of users K, it is intuitively clear that the lower bound on $\alpha R^\star + \beta M$ will be large if the number of files N is large. We define the notion of a saturated instance, which are directed trees and corresponding labelings that give the largest possible lower bound (using our technique) using as few files as possible. An analysis of saturated instances allows us to always improve on the cutset bound and in most ranges of M, our bound is strictly better.
- Our machinery allows us to show that the achievable rate of [9] is within a multiplicative factor of four of our proposed lower bound for all values of N and K. This is possible by analyzing some combinatorial properties of saturated instances.
- Our proposed technique also applies to other variants of coded caching problem. We discuss the application of our work to the case of D2D wireless networks and coded caching with multiple requests as well.

As an example, Fig. 2 illustrates the tightness of the proposed lower bound for a coded caching system with a server that contains $N = 9$ files and $K = 3$ users. Specifically, our proposed bound demonstrates the optimality of the achievable scheme for values of M that are integer multiples of N/K in this specific case.

III. Lower Bound on $R^\star(M)$

In this section we present our proposed lower bound on $R^\star(M)$. We begin with an example that demonstrates the core idea of our approach.

Example 1: Consider a coded caching system with $N = K = 3$. Then, the following sequence of information

[1]The cutset bound can be considered as a special case of our bound.

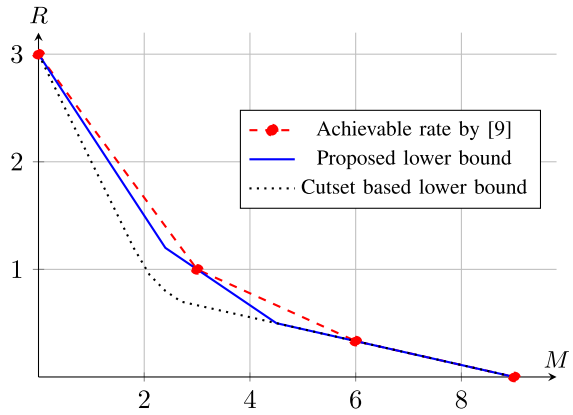

Fig. 2. An example of a coded caching system with $N = 9$ files, $K = 3$ users. Note that the proposed lower bound is better than the cutset bound and matches the achievable rate points at multiples of N/K.

theoretic inequalities hold.

$$
\begin{aligned}
&2R^\star F + 2MF \\
&\quad \geq H(Z_1, X_{123}) + H(Z_2, X_{312}) \\
&\quad \overset{(a)}{=} I(W_1; Z_1, X_{123}) + H(Z_1, X_{123}|W_1) \\
&\qquad + I(W_1; Z_2, X_{312}) + H(Z_2, X_{312}|W_1) \\
&\quad = H(W_1) - H(W_1|Z_1, X_{123}) + H(Z_1, X_{123}|W_1) \\
&\qquad + H(W_1) - H(W_1|Z_2, X_{312}) + H(Z_2, X_{312}|W_1) \\
&\quad \overset{(b)}{\geq} F(1-\epsilon) + F(1-\epsilon) + H(Z_1, Z_2, X_{123}, X_{312}|W_1) \\
&\quad = 2F(1-\epsilon) + I(W_2, W_3; Z_1, Z_2, X_{123}, X_{312}|W_1) \\
&\qquad + H(Z_1, Z_2, X_{123}, X_{312}|W_1, W_2, W_3) \\
&\quad \overset{(c)}{\geq} 2F(1-\epsilon) + 2F(1-\epsilon) = 4F(1-\epsilon),
\end{aligned}
$$

where equality (a) holds by the definition of mutual information. Inequality (b) holds by Fano's inequality since the file W_1 can be recovered with ϵ-error from the pairs (Z_1, X_{123}) and (Z_2, X_{312}) and by the fact that conditioning reduces entropy. Similarly, inequality (c) holds by Fano's inequality since the files W_2 and W_3 can be recovered with ϵ-error from $(Z_1, Z_2, X_{123}, X_{312})$. This holds for arbitrary $\epsilon > 0$ and F large enough. Dividing throughout by F we have the required result.

Thus, the key idea of the above bound is to choose the delivery phase signals in such a manner so that the various terms that are combined allow the "reuse" of the same file multiple times. For instance, in step (a) of the above bound, we use the definition of mutual information to rewrite the terms $H(Z_1, X_{123})$ and $H(Z_2, X_{312})$. Note that both pairs (Z_1, X_{123}) and (Z_2, X_{312}) allow the recovery of the *same* file W_1, resulting in a contribution of $2F$ to the lower bound. On the other hand, the files W_2 and W_3 are recovered only once. The overall result is a lower bound of $4F$.

Thus, our lower bound works with judiciously chosen labels for the delivery phase signals and combines them with the cache signals in an appropriate way such that a given file is recovered a large number of times. It turns out that doing this systematically and tractably requires the development of several new ideas. For instance, the aforementioned chain

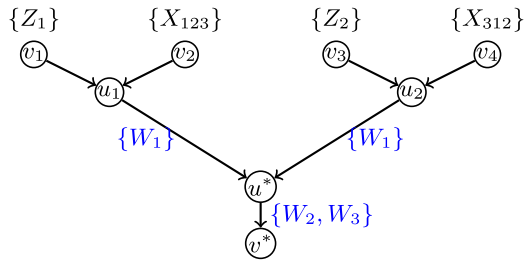

Fig. 3. Problem instance for Example 1. For clarity of presentation, only the $W_{new}(u)$ label has been shown on the edges.

of inequalities can be equivalently represented in terms of a directed tree with appropriate labels on its leaves and edges as shown in Fig. 3. In particular, the leaves of the tree are labeled with cache signals Z_1 and Z_2 and delivery phase signals X_{123} and X_{312}. Each internal node of the tree corresponds to the operation of combining the signals and its outgoing edge is labeled by the newly recovered file(s), e.g., at node u_1, the file W_1 is recovered. Likewise at node u^*, the files W_2 and W_3 are recovered. The lower bound can be obtained by summing the cardinalities of the edge labels. We note here that [9, Appendix] considers an application of a similar bound in the specific case of $K = N = 2$.

The next example shows another crucial point that is key to our approach. Namely, one can get the same lower bound by using different number of files. It turns out that using less files to obtain a specific lower bound can in turn be leveraged to improve the overall lower bound on the rate.

Example 2: Consider a coded caching system with $N = 4$, $K = 3$. Suppose that we are interested in deriving a lower bound of type $4R^\star + 4M \geq L$. Using the cutset bound in (3) for $s = 2$ we get $2R^\star + 2M \geq 4$, which in turn yields $4R^\star + 4M \geq 8$. The corresponding information theoretical inequalities to derive such a lower bound can be equivalently presented by the directed tree and labeled leaves and edges in Fig. 4 (b) (this is formalized in the Appendix). Note that there are no files labeled on the last edge (u^*, v^*).

On the other hand, consider the directed tree and the corresponding labels in Fig. 4 (a). The crucial difference is that the edge (u^*, v^*) recovers the file W_4 in Fig. 4 (a). Summing the cardinalities of the labels allows us to obtain the inequality $4R^\star + 4M \geq 9$ which is strictly better than the cutset bound. Intuitively, this can be explained as follows. It is not too hard to see that each subtree of the original directed tree can in turn yield an inequality by itself. For instance, consider the left subtree rooted at u^*, i.e., the subtree with v_1, v_2, v_5 and v_6 as leaves and (u_5, u^*) as its last edge. This subtree allows us to lower bound $2R^* + 2M$. Summing the cardinalities of the edges of this subtree yields the value 4; crucially, this subtree only uses three files W_1, W_2 and W_3. A similar statement holds for the right subtree rooted at u^*. This allows the remaining file W_4 to be recovered on the edge (u^*, v^*).

On the other hand an examination of Fig. 4 (b) shows that its subtrees also yield the value 4, but use four files $W_1, \ldots, W_4$. Thus, we conclude that the subtrees of Fig. 4 (a) are more efficient in using files. This allows one more file to

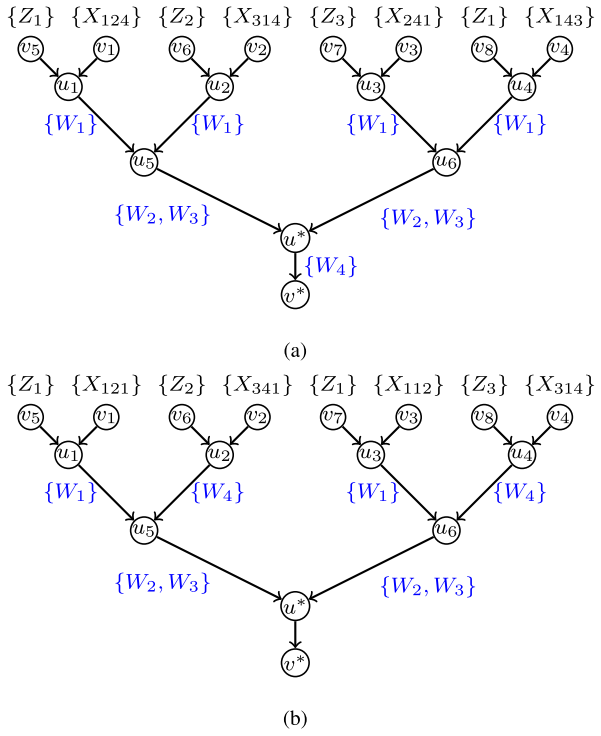

Fig. 4. Problem instances discussed in Example 2 where $N = 4$ and $K = 3$. The instance (a) has reused more files than the corresponding cutset bound derived from instance (b).

be recovered on the last edge (u^*, v^*) and translates into an overall better lower bound.

The key idea of our improved lower bounding technique is thus, to consider directed trees with appropriate labels that are efficient in using the number of files. We will formalize these notions in the subsequent discussion. As we have seen, there are new concepts that are needed in working with the directed trees with labeled leaves and edges. In what follows, we formally define these concepts.

Definition 2 (Directed in-Tree): A directed graph $\mathcal{T} = (V, A)$, is called a directed in-tree if there is one designated node called the root such that from any other vertex $v \in V$ there is exactly one directed path from v to the root.

The nodes in a directed in-tree that do not have any incoming edges are referred to as the leaves. The remaining nodes, excluding the leaves and the root are called internal nodes. Each node in a directed in-tree has at most one outgoing edge. We have the following definitions for a node $v \in V$.

$$\begin{aligned} out(v) &= \{u \in V : (v,u) \in A\}, \text{ (outgoing neighbor) and,} \\ in(v) &= \{u \in V : (u,v) \in A\} \text{ (incoming neighbor set).} \\ in\text{-}edge(v) &= \{e \in A : e = (u,v)\} \text{ (incoming edge set).} \end{aligned}$$

In this work, we exclusively work with trees which are such that the in-degree of the root equals 1. There is a natural topological order in $\mathcal{T}$ whereby for nodes $u \in \mathcal{T}$ and $v \in \mathcal{T}$, we say that $u \succ v$ if there exists a sequence of edges that can be traversed to reach v from u. This sequence of edges is denoted $path(u, v)$.

Algorithm 1 Lower Bound Algorithm

Input: $\mathcal{T} = (V, A)$ with leaves $v_1, \ldots, v_\ell$ and $\{label(v_i)\}_{i=1}^{\ell}$, such that $\mathbb{W}(v_i) = \emptyset, i = 1, \ldots, \ell$.

Initialization:

for $i \leftarrow 1, \ldots \ell$ **do**
  $W_{new}(v_i) = \Delta(v_i, v_i)$.
  $x_{(v_i, out(v_i))} = W_{new}(v_i)$.
  $y_{(v_i, out(v_i))} = |W_{new}(v_i)|$.
end for
while there exists an unlabeled edge **do**
  Pick an unlabeled node $u \in V$ such that all edges in $in - edge(u)$ are labeled.
  $\mathbb{W}(u) = \cup_{v \in in(u)} \mathbb{W}(v) \cup W_{new}(v)$.
  $\mathbb{Z}(u) = \cup_{v \in in(u)} \mathbb{Z}(v)$.
  $\mathbb{D}(u) = \cup_{v \in in(u)} \mathbb{D}(v)$.
  $W_{new}(u) = \Delta(u, u) \setminus \mathbb{W}(u)$.
  $x_{(u, out(u))} = W_{new}(u)$.
  $y_{(u, out(u))} = |W_{new}(u)|$.
end while

Output: $L = \sum_{e \in A} y_e$.

Definition 3: Meeting point of nodes in a directed tree. Consider nodes v_1 and v_2 in a directed in-tree $\mathcal{T} = (V, A)$. We say that v_1 and v_2 meet at node u if there exist $path(v_1, u)$ and $path(v_2, u)$ in $\mathcal{T}$ such that $path(v_1, u) \cap path(v_2, u) = \emptyset$. As there exists a path from any node in $\mathcal{T}$ to the root node, it follows that the existence of node u is guaranteed.

Let $D = \cup_{d_1 \in [N], \ldots, d_K \in [N]} \{X_{d_1, \ldots, d_K}\}$.

Definition 4 (Labeling of Directed in-Tree): Each node $v \in \mathcal{T}$ is assigned a label, denoted $label(v)$, which is a subset of $\{W_1, \ldots, W_N\} \cup \{Z_1, \ldots, Z_K\} \cup D$. Moreover, we also specify $\mathbb{W}(v) \subseteq \{W_1, \ldots, W_N\}$, $\mathbb{Z}(v) \subseteq \{Z_1, \ldots, Z_K\}$ and $\mathbb{D}(v) \subseteq D$ so that $label(v) = \mathbb{W}(v) \cup \mathbb{Z}(v) \cup \mathbb{D}(v)$.

In our formulation, the leaf nodes are denoted v_i, $i = 1, \ldots, \ell$ are such that $\mathbb{W}(v_i) = \emptyset$.

Definition 5 (Recoverability): We say that a singleton source subset $\{W_i\}$ is recoverable from the pair $(Z_j, X_{d_1, \ldots, d_K})$ if $d_j = i$. Similarly, for a given set of caches $Z' \subseteq \{Z_1, \ldots, Z_K\}$ and delivery phase signals $D' \subseteq D$, we define $Rec(Z', D') \subseteq \{W_1, \ldots, W_N\}$ to be the subset of the sources that can be recovered from pairs of the form (Z_i, X_J) where $Z_i \in Z'$ and J is a multiset of cardinality K with entries from $[N]$ such that $X_J \in D'$.

We let the entropy of a set of random variables equal the joint entropy of all the random variables in the set. We also let $[x]^+ = \max(x, 0)$.

Given a directed tree $\mathcal{T}$ with appropriate labels on its leaves we present an algorithm (see Algorithm 1) that generates an inequality of the form $\alpha R^\star + \beta M \geq L(\alpha, \beta)$. For nodes u, $v \in \mathcal{T}$, we define the following.

$$\begin{aligned} \Delta(u, v) &= Rec(\mathbb{Z}(u), \mathbb{D}(v)), \text{ and} \\ W_{new}(u) &= \Delta(u, u) \setminus \mathbb{W}(u). \end{aligned} \tag{4}$$

Algorithm 1 operates as follows. It takes as input a directed in-tree $\mathcal{T}$ where each leaf $v_i, i = 1, \ldots, \ell$ has labels $\mathbb{Z}(v_i)$ and $\mathbb{D}(v_i)$ ($\mathbb{W}(v_i)$ is set to $\emptyset$). The algorithm determines the

files that are recovered at each v_i and labels the corresponding outgoing edge with $W_{new}(v_i)$ and $|W_{new}(v_i)|$. Following this, the algorithm propagates the labels further down the tree in the following manner. For a given node u whose incoming edges are labeled, we set $\mathbb{Z}(u) = \cup_{v \in in(u)} \mathbb{Z}(v)$ and $\mathbb{D}(u) = \cup_{v \in in(u)} \mathbb{D}(v)$, i.e., each of these labels is set to the union of the corresponding labels of the nodes that belong to the incoming node set of u. Next, it sets $\mathbb{W}(u) = \cup_{v \in in(u)} \mathbb{W}(v) \cup W_{new}(v)$, i.e., in addition to the $\mathbb{W}$-labels of the incoming node set, $\mathbb{W}(u)$ also contains the new files that are recovered on the incident edges. Note that at each internal node certain cache signals and delivery phase signals *meet*, e.g., Z_1 and X_{123} meet at node u_1 in Fig. 3. The outgoing edge of an internal node is labeled by the *new* files that are recovered at the node, e.g., at u_1 the signals Z_1 and X_{123} recover the file W_1. We call a file new if it has not been recovered upstream of a given node. In a similar manner at u^* one can recover all the files $W_1, \ldots, W_3$; however only the set $\{W_2, W_3\}$ is labeled on edge (u^*, v^*) as W_1 was recovered upstream. This process is continued recursively, i.e., we label the outgoing edges with the new files that are recovered at node u, propagate the labels and continue thereafter. The algorithm continues until it labels the last outgoing edge.

It can be seen that the operation of Algorithm 1 is in one to one correspondence with the new files recovered in the sequence of inequalities in the lower bound. For example, the outgoing labels of u_1 and u_2 in Fig. 3 correspond to step (a) in the inequalities in Example 1. We formalize this statement in the Appendix (Lemma 5) where we show that a valid lower bound is always obtained when applying Algorithm 1. The complexity of this algorithm and the other algorithms used in this paper are discussed in Appendix E.

Definition 6 (Problem Instance): Consider a given tree $\mathcal{T}$ with leaves $v_i, i = 1, \ldots, \ell$ that are labeled as discussed above. Let $\alpha = \sum_{i=1}^{\ell} |\mathbb{D}(v_i)|$ and $\beta = \sum_{i=1}^{\ell} |\mathbb{Z}(v_i)|$. Suppose that the lower bound computed by Algorithm 1 equals L. We define the associated problem instance as $P(\mathcal{T}, \alpha, \beta, L, N, K)$. We also define $\hat{\alpha} = |\cup_{i=1}^{\ell} \mathbb{D}(v_i)|$ and $\hat{\beta} = |\cup_{i=1}^{\ell} \mathbb{Z}(v_i)|$. A problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ is said to be optimal if all instances of the form $P'(\mathcal{T}', \alpha, \beta, L', N, K)$ are such that $L' \leq L$.

It is worth emphasizing that $\hat{\alpha} \leq \alpha$ and $\hat{\beta} \leq \beta$ as some cache and delivery phase signals may be repeated.

In the subsequent discussion, we focus on understanding the characteristics of optimal problem instances. Towards this end, we shall often start with a problem instance P and modify it in appropriate ways to arrive at another instance P'. For ease of presentation, when needed we shall refer to quantities in instance $P(P')$ by using the corresponding superscripts. For example, for a node u in P (P'), we will denote the set of new files by $W^{P}_{new}(u)$ ($W^{P'}_{new}(u)$).

It is not too hard to see that it suffices to consider directed trees whose internal nodes have an in-degree at least two. In particular, if u has in-degree equal to 1, it is evident that $W_{new}(u) = \emptyset$ and thus, $|W_{new}(u)| = 0$. In addition, we claim that w.l.o.g. it suffices to consider trees where internal nodes have in-degree at most two. Therefore, we will assume that all internal nodes have degree equal to two. More specifically,

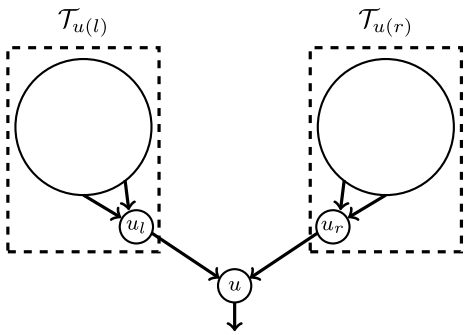

Fig. 5. For a given node $u \in \mathcal{T}$, its in-neighbors are denoted u_l and u_r. The corresponding subtrees are denoted $\mathcal{T}_{u(l)}$ and $\mathcal{T}_{u(r)}$ and are shown enclosed in the dotted boxes.

we can show the following property of problem instances (the proof appears in the Appendix).

Claim 1: Consider a problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ such that there exists a node $u \in \mathcal{T}$ with $|in(u)| \geq 3$. Then, there exists another instance $P'(\mathcal{T}', \alpha, \beta, L', N, K)$ where $L' \geq L$ and $|in(u)| \leq 2$ for all nodes $u \in \mathcal{T}'$.

Henceforth, we assume that all internal nodes in the problem instances under consideration have in-degree equal to two. Claim 1 can also be used to conclude that each leaf v in an instance P is such that either $|\mathbb{Z}(v)| = 1$ or $|\mathbb{D}(v)| = 1$ but not both. Indeed, if there exists a leaf v that violates this condition, we can use the modification in the proof of Claim 1 to replace v by a directed in-tree so that the condition is satisfied. If $|\mathbb{Z}(v)| = 1$, we call v a cache node; if $|\mathbb{D}(v)| = 1$ we call it a delivery phase node. In the subsequent discussion we will assume that the delivery phase nodes are labeled in an arbitrary order $v_1, \ldots, v_\alpha$ and the cache nodes from $v_{\alpha+1}, \ldots, v_{\alpha+\beta}$, where we note that $\alpha + \beta = \ell$. Moreover, we let $\mathcal{D} = \{v_1, \ldots, v_\alpha\}$ and $\mathcal{C} = \{v_{\alpha+1}, \ldots, v_{\alpha+\beta}\}$.

In the tree $\mathcal{T}$ corresponding to problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$, consider an internal node u and the edge $e = (u, v)$. In the subsequent discussion, we shall use $\mathcal{T}_u$ to refer to the subtree that has its last edge as $(u, out(u))$, i.e., the subtree that is rooted at $out(u)$. The incoming edges into u, denoted (u_l, u) and (u_r, u) are the last edges of the disjoint left and right subtrees denoted $\mathcal{T}_{u(l)}$ and $\mathcal{T}_{u(r)}$ respectively (see Fig. 5). Each of these subtrees defines a problem instance $P_l = P(\mathcal{T}_{u(l)}, \alpha_l, \beta_l, L_l, N, K)$ and $P_r = P(\mathcal{T}_{u(r)}, \alpha_r, \beta_r, L_r, N, K)$. We denote the set of delivery phase nodes and cache nodes in $\mathcal{T}_{u(r)}$ by

$$\mathcal{D}_{u(r)} = \{v \in \mathcal{D} : v \in \mathcal{T}_{u(r)}\} \text{ and}$$
$$\mathcal{C}_{u(r)} = \{v \in \mathcal{C} : v \in \mathcal{T}_{u(r)}\},$$

with similar definitions for $\mathcal{D}_{u(l)}$ and $\mathcal{C}_{u(l)}$. We also let

$$\mathcal{D}_u = \mathcal{D}_{u(l)} \cup \mathcal{D}_{u(r)}, \text{ and}$$
$$\mathcal{C}_u = \mathcal{C}_{u(l)} \cup \mathcal{C}_{u(r)}.$$

Let $\Gamma_l = \cup_{v \in \mathcal{T}_{u(l)}} W_{new}(v)$ and $\Gamma_r = \cup_{v \in \mathcal{T}_{u(r)}} W_{new}(v)$, i.e., Γ_l and Γ_r are the subsets of $\{W_1, \ldots, W_N\}$ that are used up in the problem instances P_l and P_r respectively. It can be observed that $\Gamma_l = \Delta(u_l, u_l)$ and $\Gamma_r = \Delta(u_r, u_r)$.

We shall often need to reason about the files recovered at the node u from the different subtrees. For instance, the set of cache nodes in $\mathcal{T}_{u(r)}$ and the delivery phase signals in $\mathcal{T}_{u(l)}$ meet and recover a subset of the files at u. This set of files corresponds to those recovered from $\mathbb{Z}(u_r) \setminus \mathbb{Z}(u_l)$ and $\mathbb{D}(u_l)$, and can be informally thought of as the *files recovered when going from right to left*. Accordingly, we have the following definitions.

$$\Delta_{rl}(u) = Rec(\mathbb{Z}(u_r) \setminus \mathbb{Z}(u_l), \mathbb{D}(u_l)), \text{ and}$$
$$\Delta_{lr}(u) = Rec(\mathbb{Z}(u_l) \setminus \mathbb{Z}(u_r), \mathbb{D}(u_r)).$$

Note that by definition, we have

$$\begin{aligned}
&\Delta(u,u)\\
&= Rec(\mathbb{Z}(u), \mathbb{D}(u))\\
&= Rec(\mathbb{Z}(u_l) \cup \mathbb{Z}(u_r), \mathbb{D}(u_l) \cup \mathbb{D}(u_r))\\
&= Rec(\mathbb{Z}(u_l), \mathbb{D}(u_l)) \cup Rec(\mathbb{Z}(u_r), \mathbb{D}(u_r))\\
&\cup\ Rec(\mathbb{Z}(u_l), \mathbb{D}(u_r)) \cup Rec(\mathbb{Z}(u_r), \mathbb{D}(u_l))\\
&\overset{(a)}{=} Rec(\mathbb{Z}(u_l), \mathbb{D}(u_l)) \cup Rec(\mathbb{Z}(u_r), \mathbb{D}(u_r))\\
&\cup\ Rec(\mathbb{Z}(u_l) \setminus \mathbb{Z}(u_r), \mathbb{D}(u_r)) \cup Rec(\mathbb{Z}(u_r) \setminus \mathbb{Z}(u_l), \mathbb{D}(u_l))\\
&= \underbrace{\Delta(\mathbb{Z}(u_l), \mathbb{D}(u_l))}_{\text{from } \mathcal{T}_{u(l)}} \cup \underbrace{\Delta(\mathbb{Z}(u_r), \mathbb{D}(u_r))}_{\text{from } \mathcal{T}_{u(r)}} \cup \Delta_{lr}(u) \cup \Delta_{rl}(u),
\end{aligned}$$

and $\mathbb{W}(u)$

$$= \Delta(\mathbb{Z}(u_l), \mathbb{D}(u_l)) \cup \Delta(\mathbb{Z}(u_r), \mathbb{D}(u_r)),$$

where (a) follows since the $Rec(\mathbb{Z}(u_l), \mathbb{D}(u_r))$ potentially contains some files that have already been recovered in $Rec(\mathbb{Z}(u_r), \mathbb{D}(u_r))$. The other equality holds because of similar reasoning. Therefore, it follows that

$$\begin{aligned}
W_{new}(u) &= \Delta(u,u) \setminus \mathbb{W}(u)\\
&= \Delta_{rl}(u) \cup \Delta_{lr}(u) \setminus \mathbb{W}(u). \qquad (5)
\end{aligned}$$

Note that based on Algorithm 1, we can conclude that

$$\begin{aligned}
\mathbb{W}(u) &= \cup_{v \in \{u_r, u_l\}} \mathbb{W}(v) \cup W_{new}(v)\\
&= \cup_{v \succ u} W_{new}(v) \text{ (by arguing inductively).} \qquad (6)
\end{aligned}$$

For the subsequent discussion, it will be useful to express the value of the lower bound L for an instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ in a functional form. In particular, we define the function $\psi : \mathcal{D} \times \mathcal{C} \to \{0,1\}$ that allows us to express L in another way. For nodes $v_i \in \mathcal{D}, v' \in \mathcal{C}$ we can define their meeting point $u \in \mathcal{T}$. The function $\psi(v_i, v')$ is determined by means of Algorithm 2, where the sequence in which we pick the nodes $v_1, \ldots, v_\alpha$ is fixed. Each element of $W_{new}(u)$ can be recovered from multiple pairs of nodes that meet there. The array $\Omega(u, \delta_u)$ keeps track of the first time the file δ_u is encountered. The function $\psi(v_i, v')$ takes the value 1 if the file W^* recovered from the pair $(\mathbb{Z}(v'), \mathbb{D}(v_i))$ at u belongs to $W_{new}(u)$ and has not been encountered before and 0 otherwise. A formal description is given in Algorithm 2.

Claim 2: For an instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ the following equality holds

$$L = \sum_{i=1}^{\alpha} \sum_{v' \in \mathcal{C}} \psi(v_i, v'). \qquad (7)$$

Algorithm 2 Computing ψ

Input: $P(\mathcal{T}, \alpha, \beta, L, N, K)$, Array $\Omega(u, \delta_u)$, where $u \in \mathcal{T}$, $\delta_u \subseteq W_{new}(u)$, $|\delta_u| = 1$.

Initialization
  for all $u \in \mathcal{T}$, $\delta_u \subseteq W_{new}(u)$ where $|\delta_u| = 1$ **do**
    $\Omega(u, \delta_u) \leftarrow 0$,
  end for
end Initialization
for $i \leftarrow 1$ to α **do**
  for all $v' \in \mathcal{C}$ **do**
    Let u be the meeting point of v_i and v'.
    $\delta_u = \Delta(v', v_i)$.
    if $\delta_u \in W_{new}(u)$ and $\Omega(u, \delta_u) == 0$ **then**
      $\psi(v_i, v') \leftarrow 1$, and $\Omega(u, \delta_u) \leftarrow 1$.
    else
      $\psi(v_i, v') \leftarrow 0$.
    end if
  end for
end for

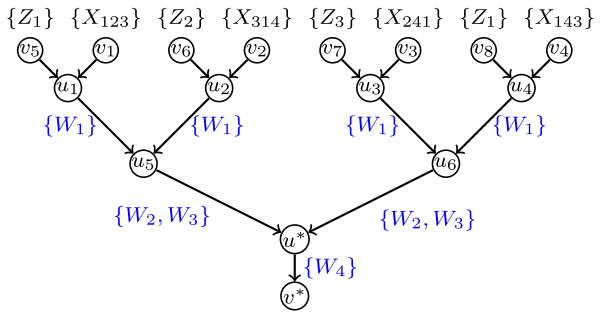

Fig. 6. Problem instance corresponding to Example 3. There are three users and the server contains four files.

Proof: We first note that at the end of Algorithm 2, we have $\Omega(u, \delta_u) = 1$ for all $u \in \mathcal{T}$ and all $\delta_u \subseteq W_{new}(u)$ such that $|\delta_u| = 1$. To see this suppose that there is a $u_1 \in \mathcal{T}$ and a singleton subset δ_{u_1} of $W_{new}(u_1)$ such that $\Omega(u_1, \delta_{u_1}) = 0$. Now δ_{u_1} is recovered from some delivery phase node and cache node, otherwise it would not be a subset of $W_{new}(u_1)$. As our algorithm considers all pairs of delivery phase nodes and cache nodes, at the end of the algorithm it has to be the case that $\Omega(u_1, \delta_{u_1}) = 1$.

Next, we note that for each pair (u_1, δ_{u_1}) where $u_1 \in \mathcal{T}$ and δ_{u_1} is singleton subset of $W_{new}(u_1)$, we can identify a unique pair of nodes (v_i, v') where $v_i \in \mathcal{D}$ and $v' \in \mathcal{C}$ such that $\psi(v_i, v')$ and $\Omega(u_1, \delta_{u_1})$ are set to 1 at the same step of the algorithm. The remaining pairs (v_i, v') that cannot be put in one to one correspondence with a pair (u_1, δ_{u_1}) are such that $\psi(v_i, v')$ are set to 0. Moreover as $\sum_{u \in \mathcal{T}} \sum_{\delta_u \subseteq W_{new}(u), |\delta_u| = 1} \Omega(u, \delta_u) = \sum_{u \in \mathcal{T}} |W_{new}(u)| = L$, it follows that $L = \sum_{i=1}^{\alpha} \sum_{v' \in \mathcal{C}} \psi(v_i, v')$. ∎

We now illustrate the definitions introduced above by means of the following example.

Example 3: The problem instance in Fig. 6 has seven internal nodes, $\{u_1, \ldots, u_6, u^*\}$. In the initialization step, Algorithm 2 sets $\Omega(u_i, \{W_1\}) = 0$ for $1 \le i \le 4$,

TABLE I
THE STEPS IN ALGORITHM 2 AFTER INITIALIZATION WHEN APPLIED TO EXAMPLE 3. THE STEPS FLOW FROM THE LEFTMOST TO THE RIGHTMOST COLUMN, AND IN EACH COLUMN FROM THE TOP TO THE BOTTOM ROW

setting	v_1	v_2	v_3	v_4
v_5	$\delta_{u_1} = W_1$ $\psi(v_1, v_5) = 1$ $\Omega(u_1, W_1) = 1$	$\delta_{u_5} = W_3$ $\psi(v_2, v_5) = 1$ $\Omega(u_5, W_3) = 1$	$\delta_{u^*} = W_2$ $\psi(v_3, v_5) = 0$	$\delta_{u^*} = W_1$ $\psi(v_4, v_5) = 0$
v_6	$\delta_{u_5} = W_2$ $\psi(v_1, v_6) = 1$ $\Omega(u_5, W_2) = 1$	$\delta_{u_2} = W_1$ $\psi(v_2, v_6) = 1$ $\Omega(u_2, W_1) = 1$	$\delta_{u^*} = W_4$ $\psi(v_3, v_6) = 0$ $\Omega(u^*, W_4) = 1$	$\delta_{u^*} = W_4$ $\psi(v_4, v_6) = 0$ $\Omega(u^*, W_4) = 1$
v_7	$\delta_{u^*} = W_3$ $\psi(v_1, v_7) = 0$	$\delta_{u^*} = W_4$ $\psi(v_2, v_7) = 1$ $\Omega(u^*, W_4) = 1$	$\delta_{u_3} = W_1$ $\psi(v_3, v_7) = 1$ $\Omega(u_3, W_1) = 1$	$\delta_{u_6} = W_2$ $\psi(v_4, v_7) = 1$ $\Omega(u_6, W_2) = 1$
v_8	$\delta_{u^*} = W_1$ $\psi(v_1, v_8) = 0$	$\delta_{u^*} = W_3$ $\psi(v_2, v_8) = 0$	$\delta_{u_6} = W_2$ $\psi(v_3, v_8) = 1$ $\Omega(u_6, W_2) = 1$	$\delta_{u_4} = W_3$ $\psi(v_4, v_8) = 1$ $\Omega(u_4, W_3) = 1$

$\Omega(u_i, \{W_2\}) = \Omega(u_i, \{W_3\}) = 0$ for $i = 5, 6$ and $\Omega(u^*, \{W_4\}) = 0$. In the next step, for node v_1 it sets $\psi(v_1, v_5) = 1$, $\Omega(u_1, \{W_1\}) = 1$ (for $v_5 \in \mathcal{C}$) and $\psi(v_1, v_6) = 1$, $\Omega(u_5, \{W_2\}) = 1$ (for $v_6 \in \mathcal{C}$). For $v_7 \in \mathcal{C}$ we have $\delta_{u^*} = \Delta(v_7, v_1) = \{W_3\}$ and since $W_3 \notin W_{new}(u^*) = \{W_4\}$ therefore $\psi(v_1, v_7) = 0$. By the same argument we have $\psi(v_1, v_8) = 0$. Thus, the contribution of v_1 to the lower bound, namely $\sum_{v' \in \mathcal{C}} \psi(v_1, v') = 2$. The complete description of the steps after the initialization, is shown in Table I. The table should be read in column order from left to right. Within a column, the order of the operations is from top to bottom. Note that there are two cases, $v_3 \in \mathcal{D}, v_6 \in \mathcal{C}$ and $v_4 \in \mathcal{D}, v_6 \in \mathcal{C}$ where $\psi(\cdot, \cdot)$ value is set to 0 (since the corresponding $\Omega(\cdot, \cdot)$ values are already 1). In both cases $\delta_{u^*} = \{W_4\}$ and since W_4 is recovered already, $\Omega(u^*, \{W_4\})$ has already been set to 1 when considering $v_2 \in \mathcal{D}, v_7 \in \mathcal{C}$. Therefore $\psi(v_4, v_6) = \psi(v_3, v_6) = 0$. Another point to be noted is that delivery phase node v_2 contributes three files towards L while the other delivery nodes contribute only two files each.

Corollary 1: For an instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$, we have $L \leq \alpha \min(\beta, K)$. Moreover, if $N \geq \alpha \min(\beta, K)$ there exists an instance such that $L = \alpha \min(\beta, K)$.

Proof: For a node v_i, where $1 \leq i \leq \alpha$, we have

$$\begin{aligned} \sum_{v' \in \mathcal{C}} \psi(v_i, v') &\leq |\cup_{v' \in \mathcal{C}} \mathbb{Z}(v')| \\ &= \hat{\beta}, \\ &\leq \min(\beta, K). \end{aligned} \tag{8}$$

Let u denote the meeting point of v' and v_i. The first inequality above holds since $\psi(v_i, v') = 1$ implies that $\delta_u = \Delta(v', v_i) \subseteq W_{new}(u)$ and

$$\begin{aligned} \sum_{v' \in \mathcal{C}} \psi(v_i, v') &\leq |\cup_{v' \in \mathcal{C}} Rec(\mathbb{D}(v_i), \mathbb{Z}(v'))| \\ &= |Rec(\mathbb{D}(v_i), \cup_{v' \in \mathcal{C}} \mathbb{Z}(v'))| \leq |\cup_{v' \in \mathcal{C}} \mathbb{Z}(v')|. \end{aligned}$$

From eq. (8) we can conclude that $L = \sum_{i=1}^{\alpha} \sum_{v' \in \mathcal{C}} \psi(v_i, v') \leq \alpha \min(\beta, K)$. If $N \geq \alpha \min(\beta, K)$, it is easy to construct an instance with $L = \alpha \min(\beta, K)$. We simply pick any directed tree on $\alpha + \beta$ leaves. Let the cache node indices be Z_1 repeated $\beta - \min(\beta, K) + 1$ times and $Z_2, Z_3, \ldots, Z_{\min(\beta,K)-1}, Z_{\min(\beta,K)}$. Suppose that node $v \in \mathcal{D}, v' \in \mathcal{C}'$ meet at node u. We label the delivery phase leaves such that $|\cup_{(v,v') \in \mathcal{D} \times \mathcal{C}'} \Delta(v', v)| = \alpha \min(\beta, K)$. This can be done since N is large enough so that we can choose the labels such that $Rec(\mathbb{Z}(v'_1), \mathbb{D}(v_1)) \cap Rec(\mathbb{Z}(v'_2), \mathbb{D}(v_2)) = \emptyset$ for $v'_1, v'_2 \in \mathcal{C}'$ and $v_1, v_2 \in \mathcal{D}$. For instance, initialize $\mathbb{D}(v) = X_{1,1,\ldots,1}$ for all $v \in \mathcal{D}$ and then set $\mathbb{D}(v_i) = X_{d_1,\ldots,d_K}$, $d_j = (i-1)\min(\beta, K) + j$ for $j = 1, \ldots, \min(\beta, K)$, and $i = 1, \ldots, \alpha$. ■

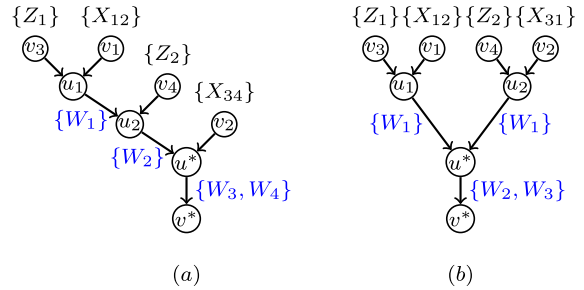

Fig. 7. (a) Problem instance $P'(\mathcal{T}', \alpha, \beta, L, N', K)$, (b) problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ where $\alpha = 2$, $\beta = 2$ and $K = 2$. Both instances reach $L = \alpha \min(\beta, K) = 4$ with different number of files $N = 3$ and $N' = 4$.

We illustrate the construction outlined above by means of the following example.

Example 4: Let $\alpha = \beta = 2$, $K = 2$, and $N = 4$. We arbitrarily pick a directed tree with v_1, v_2 as delivery nodes and v_3, v_4 as cache nodes. We label $\mathbb{Z}(v_3) = Z_1$ and $\mathbb{Z}(v_3) = Z_2$, and delivery nodes as $\mathbb{D}(v_1) = X_{1,2}$ and $\mathbb{D}(v_2) = X_{3,4}$. Such a problem instance is illustrated in Fig. 7 (a). It is evident that applying Algorithm 1 on this instance yields a lower bound of 4. However, as we will see later, this instance is not efficient in reusing files.

At this point we have established that for a given problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$, we can always generate an inequality of the form $\alpha R^\star + \beta M \geq L$. It is natural to therefore consider *optimal* problem instances that maximize the lower bound for a given value of α, β, N and K.

Definition 7: For given α, β, N and K, we say that a problem instance $P(\mathcal{T}^*, \alpha, \beta, L^*, N, K)$ is optimal if all problem instances $P'(\mathcal{T}, \alpha, \beta, L, N, K))$ are such that $L^* \geq L$.

Recall that $\hat{\beta} = |\cup_{i=1}^{\ell} \mathbb{Z}(v_i)|$. For a problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$, it may be possible that $\hat{\beta} < \min(\beta, K)$. However, given such an instance, we can convert it into another instance where $\hat{\beta} = \min(\beta, K)$ without reducing the value of L. In fact, the following stronger statement holds (see Appendix B for a proof).

Claim 3: For a problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ suppose that there exists an internal node u^* with associated problem instance $P^* = P(\mathcal{T}_{u^*}, \alpha^*, \beta^*, L^*, N^*, K)$ such that the following condition holds.

$$\hat{\beta}^* < \min(\beta^*, K).$$

Then, there exists another problem instance $P'(\mathcal{T}', \alpha, \beta, L', N, K)$ where $L' \geq L$ such that the above condition does not hold.

The next claim formalizes the intuitive fact that permuting the cache nodes and the delivery phase signals by the same permutation does not change the $\mathbb{W}$ labels and the lower bound of the instance.

Claim 4: Let $P(\mathcal{T}, \alpha, \beta, L, N, K)$ to be a problem instance and let $\pi : [K] \longrightarrow [K]$ to be a permutation with inverse σ. Assume that the problem instance $P'(\mathcal{T}', \alpha, \beta, L', N, K)$ is obtained from P by the following changes for all $v \in \mathcal{D}$ and $v' \in \mathcal{C}$.

- Let $\mathbb{Z}^P(v') = Z_i$, then set $\mathbb{Z}^{P'}(v') = Z_{\pi(i)}$.
- Let $\mathbb{D}^P(v) = X_{d_1,\ldots,d_K}$, then set $\mathbb{D}^{P'}(v) = X_{d_{\sigma(1)},\ldots,d_{\sigma(K)}}$.

Then $W^{P'}_{new}(u) = W^P_{new}(u)$, $\mathbb{W}^{P'}(u) = \mathbb{W}^P(u)$ for $u \in \mathcal{T}$, and $L' = L$.

Proof: We note that

$$\begin{aligned} Rec(Z_i, X_{d_1,\ldots,d_K}) &= W_{d_i} \\ &= W_{d_{\sigma(\pi(i))}} = Rec(Z_{\pi(i)}, X_{d_{\sigma(1)},\ldots,d_{\sigma(K)}}) \end{aligned}$$

for $i = 1, \ldots, K$. Therefore, for any $v \in \mathcal{D}$ and $v' \in \mathcal{C}$, we have $\Delta^{P'}(v', v) = \Delta^P(v', v)$ and more generally $\Delta^{P'}(u, u) = \Delta^P(u, u)$. Furthermore, $\mathbb{W}^P(u) = \Delta^P(u_l, u_l) \cup \Delta^P(u_r, u_r)$ and we have $\mathbb{W}^P(u) = \mathbb{W}^{P'}(u)$ for any $u \in \mathcal{T}$. Using eq. (4), we have $W^{P'}_{new}(u) = W^P_{new}(u)$ for all $u \in \mathcal{T}'$. It follows that $L' = L$. ∎

Henceforth, we will assume w.l.o.g. that $\hat{\beta} = \min(\beta, K)$ and that Claim 3 holds. Our next lemma shows a structural property of problem instances. Namely for an instance where $L < \alpha \min(\beta, K)$, increasing the number of files allows us to increase the value of L. This lemma is a key ingredient in our proof of the main theorem (the proof appears in the Appendix).

Lemma 1: Let $P = P(\mathcal{T}, \alpha, \beta, L, N, K)$ be an instance where $L < \alpha \min(\beta, K)$. Then, we can construct a new instance $P' = P(\mathcal{T}', \alpha, \beta, L', N + 1, K)$, where $L' = L + 1$.

Informally, another property of optimal problem instances is that the same file is recovered as many times as possible at the same level of the tree. For instance, in Fig. 3, W_1 is recovered in both $\mathcal{T}_{u^*(l)}$ and $\mathcal{T}_{u^*(r)}$. In fact, intuitively it is clear that the same set of files can be reused in any subtrees of an internal node. Our next claim formalizes this intuition. Recall that for a node u, $\Gamma_l = \cup_{v \in \mathcal{T}_{u(l)}} W_{new}(v)$ and $\Gamma_r = \cup_{v \in \mathcal{T}_{u(r)}} W_{new}(v)$.

Claim 5: Consider an instance $P = P(\mathcal{T}, \alpha, \beta, L, N, K)$. For all nodes $u \in \mathcal{T}$, suppose w.l.o.g. that $|\Gamma_l| \geq |\Gamma_r|$. Suppose that there exist a node $u \in \mathcal{T}$ such that such that $\Gamma_r \nsubseteq \Gamma_l$. Then there exists another instance $P'(\mathcal{T}', \alpha, \beta, L', N', K)$ such that $N' \leq N$, $L' \geq L$, and $\Gamma_r \subseteq \Gamma_l$ for all $u \in \mathcal{T}'$.

Next, we upper bound the maximum value of $|W_{new}(u)|$ for a node $u \in \mathcal{T}$.

Claim 6: In instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$, consider an internal node u. Let $\rho(u) = \hat{\alpha}_l[\min(\beta_r, K - \beta_l)]^+ + \hat{\alpha}_r[\min(\beta_l, K - \beta_r)]^+$. We have

$$|W_{new}(u)| \leq \min\left(\rho(u), [N - |\Gamma_l \cup \Gamma_r|]^+\right).$$

Proof: From eq. (5) it follows that

$$|W_{new}(u)| \leq |\Delta_{rl}(u) \setminus \mathbb{W}(u)| + |\Delta_{lr}(u) \setminus \mathbb{W}(u)|.$$

Next, we observe that

$$\begin{aligned} |\Delta_{rl}(u) \setminus \mathbb{W}(u)| &= |Rec(\mathbb{Z}(u_r) \setminus \mathbb{Z}(u_l), \mathbb{D}(u_l)) \setminus \mathbb{W}(u)| \\ &\leq |\mathbb{D}(u_l)| \times |\mathbb{Z}(u_r) \setminus \mathbb{Z}(u_l)| \\ &\overset{(a)}{\leq} \hat{\alpha}_l \times \min(\hat{\beta}_r, K - \hat{\beta}_l), \\ &\overset{(b)}{=} \hat{\alpha}_l \times [\min(\beta_r, K - \beta_l)]^+, \end{aligned}$$

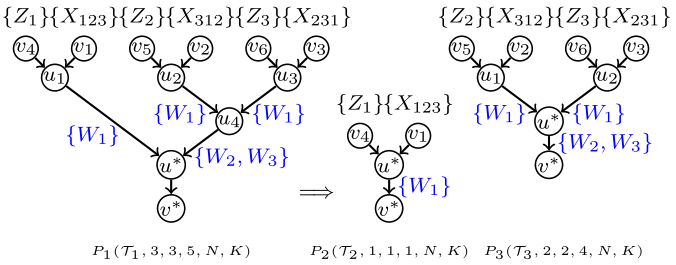

Fig. 8. Problem instances with $N = K = 3$. Instance P_1 is non-atomic as the corresponding lower bound can be obtained by summing the lower bounds from P_2 and P_3.

where inequality (a) holds, since $|\mathbb{D}(u_l)| = \hat{\alpha}_l$ and $|\mathbb{Z}(u_r) \setminus \mathbb{Z}(u_l)| \leq \min(\hat{\beta}_r, K - \hat{\beta}_l)$. Inequality (b) holds under the conditions $\hat{\beta}_l = \min(\beta_l, K)$ and $\hat{\beta}_r = \min(\beta_r, K)$ (see Claim 9 in Appendix). We can bound $|\Delta_{lr}(u) \setminus \mathbb{W}(u)|$ in a similar manner.

To conclude the proof we note that instances P_l and P_r recover a total of $|\Gamma_l \cup \Gamma_r|$ sources. As the total number of sources is N, $|W_{new}(u)| \leq [N - |\Gamma_l \cup \Gamma_r|]^+$. ∎

Definition 8 (Saturation Number): Consider an instance $P^*(\mathcal{T}^*, \alpha, \beta, L^*, N^*, K)$, where $L^* = \alpha \min(\beta, K)$, such that for all problem instances of the form $P(\mathcal{T}, \alpha, \beta, L^*, N, K)$, we have $N^* \leq N$. We call N^* the saturation number of instances with parameters (α, β, K) and denote it by $N_{sat}(\alpha, \beta, K)$.

In essence, for given α, β and K, saturated instances are most efficient in using the number of available files. It is easy to see that $N_{sat}(\alpha, \beta, K) \leq \alpha \min(\beta, K)$ since one can construct an instance with lower bound $\alpha \min(\beta, K)$ when $\alpha \min(\beta, K) \leq N$ (see Corollary 1).

Example 5: Consider the two problem instances P and P' with $\alpha = 2, \beta = 2$ and $K = 2$ that are shown in Fig. 7. The lower bound for both instances is $L = \alpha \min(\beta, K) = 4$. However, instance P uses one less file than P'. This reduction is accomplished by reusing file W_1 at both $\mathcal{T}_{u^*(l)}$ and $\mathcal{T}_{u^*(r)}$. The instance P' can be treated as a trivial instance constructed by the procedure suggested in the proof of Corollary 1 as it uses $N' = \alpha \min(\beta, K) = 4$ files. It can be verified by Algorithm 4 in Section III-B that P is one of the problem instances associated with $N_{sat}(2, 2, 2)$; therefore, $N_{sat}(2, 2, 2) = 3$.

Definition 9 (Atomic Problem Instance): For a given optimal problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ it is possible that there exist other optimal problem instances $P_i(\alpha_i, \beta_i, L_i, N, K), i = 1, \ldots, m$ with $m \geq 2$ such that $\sum_{i=1}^m \alpha_i = \alpha$, $\sum_{i=1}^m \beta_i = \beta$ and $\sum_{i=1}^m L_i = L$, i.e., the value of L follows from appropriately combining smaller problems. In this case we call the instance P non-atomic. Conversely, if such smaller problem instances do not exist, we call P an atomic problem instance.

Example 6: Consider the problem instance P_1 shown in Fig. 8 with $N = K = 3$. The lower bound associated with this instance, $3R^\star + 3M \geq 5$, can be obtained by combining the lower bounds acquired by P_2 and P_3. Specifically, instance P_2 yields $R^\star + M \geq 1$ and instance P_3 yields $2R^\star + 2M \geq 4$. Note that in P_1 the last edge (u^*, v^*) is such that $W_{new}(u^*) = \emptyset$.

Thus, the tree can be split into two separate instances at u^*. Thus it is non-atomic.

It is evident that instances where no new file is recovered in the last edge are non-atomic. However, we emphasize that there are other instances that are non-atomic as well. For example, consider instance P_1', obtained from P_1 where we change the label $\mathbb{D}(v_3)$ to X_{221}. In P_1', the labels of edges (u_4, u^*) and (u^*, v^*) will change to $\{W_2\}$ and $\{W_3\}$ respectively; none of the other labels will change. Even though $W_{new}(u^*)$ is nonempty in P_1', but we still call it non-atomic since the associated lower bound does not change.

The following theorem and its corollary are the main results of our paper and can be used to identify optimal problem instances.

Theorem 1: Suppose that there exists an optimal and atomic problem instance $P_o(\mathcal{T} = (V, A), \alpha, \beta, L_o, N, K)$. Then, there exists an optimal and atomic problem instance $P^*(\mathcal{T}^* = (V^*, A^*), \alpha, \beta, L^*, N, K)$ where $L^* = L_o$ with the following properties. Let us denote the last edge in P^* with (u^*, v^*). Let $P_l^* = P(\mathcal{T}^*_{u^*(l)}, \alpha_l, \beta_l, L_l^*, |\Gamma_l|, K)$ and $P_r^* = P(\mathcal{T}^*_{u^*(r)}, \alpha_r, \beta_r, L_r^*, |\Gamma_r|, K)$. Then, we have

$$\begin{aligned} L_l^* &= \alpha_l \min(\beta_l, K), \\ L_r^* &= \alpha_r \min(\beta_r, K), \text{ and} \\ L^* &= \min\left(\alpha \min(\beta, K), L_l^* + L_r^* + [N - N_0]^+\right), \end{aligned} \quad (9)$$

where $N_0 = \max(N_{sat}(\alpha_l, \beta_l, K), N_{sat}(\alpha_r, \beta_r, K))$.[2] Furthermore, $\min(\beta_l, \beta_r) < K$.

Proof: Note that we assume that the problem instance P_o is atomic. This implies that $W^{P_o}_{new}(u^*) \neq \emptyset$ and, consequently, $N > |\Gamma_l|, |\Gamma_r|$. Using Claim 3 we can assert that $\hat{\beta}_l = \min(\beta_l, K)$ and $\hat{\beta}_r = \min(\beta_r, K)$.

We denote by (u^*, v^*), the last edge in P_o. We let $P_l = P(\mathcal{T}_{u^*(l)}, \alpha_l, \beta_l, L_l, |\Gamma_l|, K)$ and $P_r = P(\mathcal{T}_{u^*(r)}, \alpha_r, \beta_r, L_r, |\Gamma_r|, K)$. It is easy to see that $L_o = L_l + L_r + |W^{P_o}_{new}(u^*)|$. Suppose that $L_l < \alpha_l \min(\beta_l, K)$. We apply the result of Lemma 1, by noting that $|\Gamma_l| < N$, and conclude that there exists another instance $P_l^{**} = P(\mathcal{T}^{**}_{u^*(l)}, \alpha_l, \beta_l, L_l^* + 1, |\Gamma_l| + 1, K)$ that can replace P_l, where the new file is denoted W^*. We also note that in P_o, $W^* \in W^{P_o}_{new}(u^*)$. Let us denote the new instance P_o'. We emphasize that the nature of the modification in Lemma 1 is such that $\Delta^{P_o'}(u^*, u^*) = \Delta^{P_o}(u^*, u^*)$. Moreover, we note that $\mathbb{W}^{P_o'}(u^*) = \mathbb{W}^{P_o}(u^*) \cup \{W^*\}$. Thus,

$$\begin{aligned} W^{P_o'}_{new}(u^*) &= \Delta^{P_o'}(u^*, u^*) \setminus \mathbb{W}^{P_o'}(u^*) \\ &= \Delta^{P_o'}(u^*, u^*) \setminus \mathbb{W}^{P_o}(u^*) \cup \{W^*\} \\ &= W^{P_o}_{new}(u^*) \setminus \{W^*\}. \end{aligned}$$

The problem instance P_o' is also optimal since L_l is increased by one and $|W^{P_o}_{new}(u^*)|$ is decreased by one, leaving L_o unchanged. Therefore, moving files from $W^{P_o}_{new}(u^*)$ to either P_l or P_r preserves optimality. In addition, from $L_o' = L_o$ and that P_o is atomic, P_o' is atomic. Based on this argument, we can immediately conclude that we cannot have $L_l < \alpha_l \min(\beta_l, K)$ and $L_r < \alpha_r \min(\beta_r, K)$ as the file W^* can be used to simultaneously modify the instance P_r. Upon this modification, we can conclude that L_o can be increased by one, which contradicts the optimality of the instance P_o. Thus we assume that $L_r = \alpha_r \min(\beta_r, K)$. We can repeatedly apply the operation of moving files from $W^{P_o}_{new}(u^*)$ to P_l until we have $L_l^* = \alpha_l \min(\beta_l, K)$. It has to be the case that $|W^{P_o}_{new}(u^*)| > \alpha_l \min(\beta_l, K) - |\Gamma_l|$ so that we can repeatedly apply the operation of moving the files, for if this were not true, the instance P_o would not be atomic.

We will denote the instance that we arrive at after completing these modification by P^* which is optimal and atomic. We can also observe at this point that if we have $\beta_l \geq K$ and $\beta_r \geq K$ so that $\hat{\beta}_l = \hat{\beta}_r = K$, then $W^{P^*}_{new}(u^*) = \emptyset$ (by Claim 6) which implies that the original instance P_o is not atomic. Thus, either β_l or β_r or both have to be strictly smaller than K. In the discussion below we assume w.l.o.g. that $\beta_r < K$. It is easy to see that

$$L^* = L_l^* + L_r^* + |W^{P^*}_{new}(u^*)|.$$

We define $\tilde{\rho}(u^*) = \alpha_l \times [\min(\beta_r, K - \beta_l)]^+ + \alpha_r \times [\min(\beta_l, K - \beta_r)]^+$ where $\tilde{\rho}(u^*) \geq \rho(u^*)$ due to the fact that $\alpha_l \geq \hat{\alpha}_l$ and $\alpha_r \geq \hat{\alpha}_r$. Using this and Claim 6, we have that

$$|W^{P^*}_{new}(u^*)| \leq \min\left(\tilde{\rho}(u^*), \left[N - \max(|\Gamma_l^*|, |\Gamma_r^*|)\right]^+\right).$$

For an optimal instance, we claim that the above inequality is met with equality. If $L^* = \alpha \min(\beta, K)$ there is nothing to prove. In this case, $|W^{P^*}_{new}(u^*)| = \alpha \min(\beta, K) - L_l^* - L_r^* = \tilde{\rho}(u^*)$ (see Claim 10 in Appendix) and the above inequality is met with equality.

Otherwise, we have $L^* < \alpha \min(\beta, K)$ which implies $\tilde{\rho}(u^*) > |W^{P^*}_{new}(u^*)|$ and $\tilde{\rho}(u^*) > N - \max(|\Gamma_l^*|, |\Gamma_r^*|)$. From the Claim 5, we can assume that either $\Gamma_l^* \subseteq \Gamma_r^*$ or $\Gamma_r^* \subseteq \Gamma_l^*$. In P^*, $N_{used} = \max\left(|\Gamma_l^*|, |\Gamma_r^*|\right) + |W^{P^*}_{new}(u^*)|$ files are used so far. Now, if $N > N_{used}$, we can use Lemma 1 to conclude that there exists a problem instance $P''(\mathcal{T}'', \alpha, \beta, L'', N'', K)$ where $N'' = N_{used} + 1 \leq N$ and $L'' = L^* + 1$. This is a contradiction since we assumed that P^* is optimal. Therefore, $N \leq N_{used}$. In addition, since the number of available files is N thus $N \geq N_{used}$. As a result, $N = N_{used} = \max\left(|\Gamma_l^*|, |\Gamma_r^*|\right) + |W^{P^*}_{new}(u^*)|$ and the inequality is met with equality. In both cases, we conclude that

$$|W^{P^*}_{new}(u^*)| = \min\left(\tilde{\rho}(u^*), \left[N - \max(|\Gamma_l^*|, |\Gamma_r^*|)\right]^+\right).$$

It follows that

$$L^* = \min\left(\alpha \min(\beta, K), L_l^* + L_r^* + \left[N - \max(|\Gamma_l^*|, |\Gamma_r^*|)\right]^+\right).$$

If $L^* = \alpha \min(\beta, K)$ the saturated instance associated with $N_{sat}(\alpha, \beta, K)$ is an optimal instance. Otherwise, $L^* < \alpha \min(\beta, K)$, and we have

$$\begin{aligned} |W^{P^*}_{new}(u^*)| &= \left[N - \max(|\Gamma_l^*|, |\Gamma_r^*|)\right]^+ \\ &\leq [N - \max(N_{sat}(\alpha_l, \beta_l, K), N_{sat}(\alpha_r, \beta_r, K))]^+. \end{aligned} \quad (10)$$

We claim that for P^* to be optimal, P_l^* and P_r^* have to be such that $\max(|\Gamma_l^*|, |\Gamma_r^*|) = \max(N_{sat}(\alpha_l, \beta_l, K), N_{sat}(\alpha_r, \beta_r, K))$. To see this we proceed as follows. Note that by the definition of saturation number, there exist problem instances

[2] As the instance is atomic, we have $N > N_0$.

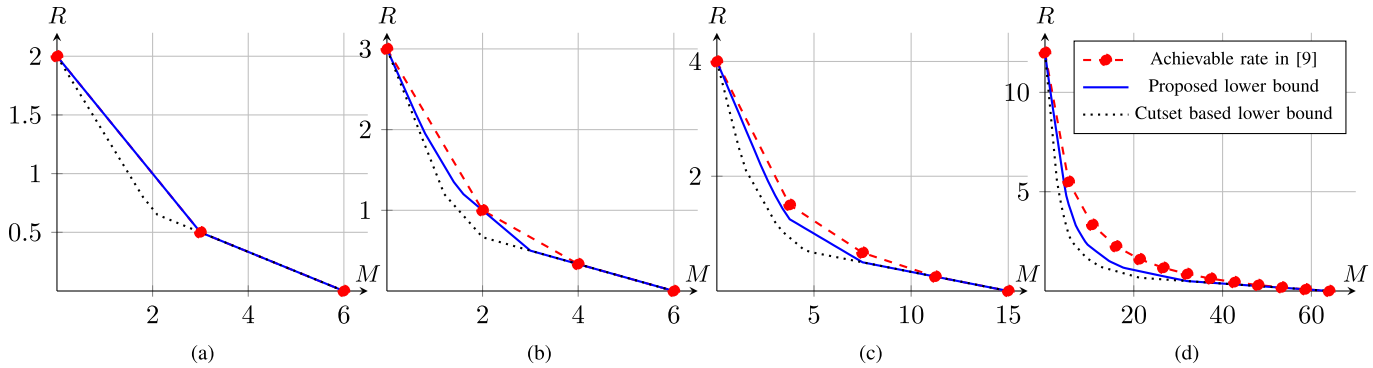

Fig. 9. Comparison of the proposed lower bound and the cutset bound. (a) Case I: $N = 6$, $K = 2$. (b) Case II: $N = 6$, $K = 3$. (c) Case III: $N = 15$, $K = 4$. (d) Case IV: $N = 64$, $K = 12$.

$P_l'(\mathcal{T}_l', \alpha_l, \beta_l, L_l', N_l', K)$ and $P_r'(\mathcal{T}_r', \alpha_r, \beta_r, L_r', N_r', K)$ such that $L_l' = L_l^*$, $L_r' = L_r^*$, $N_l' = N_{sat}(\alpha_l, \beta_l, K)$ and $N_r' = N_{sat}(\alpha_r, \beta_r, K)$. W.l.o.g. let assume $N_l' \geq N_r'$. By the Claims 3 and 5 problem instances P_l' and P_r' can be modified in such a way that $\hat{\beta}_l' = \min(\beta_l, K)$, $\hat{\beta}_r' = \min(\beta_r, K)$ and $\Gamma_l' \subseteq \Gamma_r'$. Also, by Claim 4 we can set $\cup_{v \in \mathcal{C}_l'} \mathbb{Z}(v) = \{Z_1, \ldots, Z_{\hat{\beta}_l'}\}$ and $\cup_{v \in \mathcal{C}_r'} \mathbb{Z}(v) = \{Z_{K-\hat{\beta}_r'+1}, \ldots, Z_K\}$. This ensures that $\hat{\beta}_l = \min(\beta_l, K)$, $\hat{\beta}_r = \min(\beta_r, K)$, and $\hat{\beta} = \min(\beta, K)$ hold in the defined problem instance. Now, consider the problem instance $P' = P(\mathcal{T}', \alpha, \beta, L', N, K)$ with last edge (u', v') where P_l' and P_r' are instances corresponding to u_l' and u_r' respectively. The instance P' uses $N_l' + |W_{new}^{P'}(u')|$ files. If $N - N_l' - |W_{new}^{P'}(u')| \geq 1$, then we are able to apply Lemma 1 $N - N_l' - |W_{new}^{P'}(u')|$ times and come up with a modified version of P' so that either $L' = \alpha \min(\beta, K)$ or $N - N_l' - |W_{new}^{P'}(u')| = 0$. The first case cannot happen since by assumption P^* is optimal and $L' \leq L^* < \alpha \min(\beta, K)$. Therefore, $|W_{new}^{P'}(u')| = N - N_l'$ and $L' = L_l^* + L_r^* + N - N_l'$. Finally, as $L' \leq L^*$ and $L^* \leq L_l^* + L_r^* + N - N_l'$, we conclude that $L' = L^*$. ∎

Corollary 2: Suppose that there exists an optimal and atomic problem instance $P_o(\mathcal{T} = (V, A), \alpha, \beta, L_o, N, K)$. Consider problem instances $P_l'(\alpha_l', \beta_l', L_l', N, K)$ and $P_r'(\alpha_r', \beta_r', L_r', N, K)$ such that $\alpha_l' + \alpha_r' = \alpha$ and $\beta_l' + \beta_r' = \beta$ such that $N \geq N_0' = \max(N_{sat}(\alpha_l', \beta_l', K), N_{sat}(\alpha_r', \beta_r', K))$. Then we have

$$L_o \geq \min\left(\alpha \min(\beta, K), L_l' + L_r' + N - N_0'\right).$$

Proof: The result follows by applying the arguments in the proof of Theorem 1, to the problem instance where P_l^* and P_r^* are replaced by P_l' and P_r' respectively. ∎

Lemma 2: Consider the class of coded caching systems where $K = 3$ and $N = 3n$ for $n = 1, 2, 3, \ldots$. For this class, the achievable scheme in [9] for $M \in \{0, n, 2n, 3n\}$ is optimal.

Proof: From the achievable scheme in [9] we have $R^\star(0) \leq 3$, $R^\star(n) \leq 1$, $R^\star(2n) \leq 1/3$, and $R^\star(3n) \leq 0$. It is easy to see that $N_{sat}(\alpha, 1, 3) = \alpha$ for any integer α. Then, the following inequalities hold,

$$\begin{aligned} 3nR^\star + M &\geq 3n, \\ nR^\star + 3M &\geq 3n, \text{ and} \\ 2nR^\star + 2M &\geq 4n. \end{aligned}$$

These inequalities are the result of Corollary 2 for $(\alpha, \beta) = (\alpha_l', \beta_l') = (3n, 1)$, $(\alpha, \beta) = (\alpha_l', \beta_l') = (n, 3)$ and $(\alpha, \beta) = (2n, 2)$ with $(\alpha_l', \beta_l') = (n, 1)$ respectively. The first two inequalities above can also be obtained by using the cutset bound while the third one cannot. Now, the second inequality for $M = 0$ implies that the achievable rate $R^\star(0) \leq 3$ is optimal. Similarly, the third inequality for $M = n$ implies that achievable rate $R^\star(n) \leq 1$ is optimal. Finally, the first inequality can be used to show that achievable rates $R^\star(2n) \leq 1/3$ and $R^\star(3n) \leq 0$ are optimal. ∎

The following example demonstrates the effectiveness of Corollary 2.

Example 7: Consider a system with $N = 64$, $K = 12$ and cache size $M = 16/3$. The cut-set bound for such a system provides a lower bound $R^\star(M) \geq 77/27 = 2.852$. Now, using the approach of Theorem 1 for $\alpha = 12$, $\beta = 8$, $(\alpha_l, \beta_l) = (\alpha_r, \beta_r) = (6, 4)$ yields $12R^\star + 8M \geq \min(12 \times 8, 24 + 24 + 64 - N_{sat}(6, 4, 12))$. It can be shown that $N_{sat}(6, 4, 12) = 17$ (see Algorithm 4 in Section III-B). Therefore, $R^\star(M) \geq 157/36 = 4.361$. This is significantly closer to the achievable rate of 5.5 (from [9]).

Theorem 1 can be leveraged effectively if it can also yield the optimal values of α_l, β_l and α_r, β_r. However, currently we do not have an algorithm for picking them in an optimal manner. Thus, we have to use Corollary 2 with either the exact value of $N_{sat}(\alpha, \beta, K)$ or an upper bound on it. Algorithm 4 in Section III-B is an algorithm to calculate the value of $N_{sat}(\alpha, \beta, K)$. Setting $\alpha_l = \lceil \alpha/2 \rceil$, $\beta_l = \lfloor \beta/2 \rfloor$ in Theorem 1 and using the corresponding values of the saturation numbers, we can obtain the results plotted in Fig. 9.

A. An Analytic Bound on the Saturation Number

Recall that the saturation number for a given α, β and K is the minimum value of N such that there exists a problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ with $L = \alpha \min(\beta, K)$. In particular, this implies that if we are able to construct a problem instance with N' files with a lower bound equal to $\alpha \min(\beta, K)$, then, $N_{sat}(\alpha, \beta, K) \leq N'$. In Algorithm 3, we create one such problem instance.

The basic idea of Algorithm 3 is as follows. The first part focuses on the construction of the tree, without labeling the leaves. For a given α and β, we first initialize a tree that just consists of a single edge (u^*, v^*). Following this, we

Algorithm 3 Instance Construction for Upper Bounding $N_{sat}(\alpha, \beta, K)$

Input: α, β and K.

Initialization
  Let (u^*, v^*) be last edge and set $U_{new} = \{u^*\}$.
  Set $\mathbb{Z}(u^*) = \{Z_1, Z_2, \ldots, Z_{\min(\beta,K)}\}$ and $b(u^*) = \beta$, $a(u^*) = \alpha$.
  $\mathcal{C} = \emptyset$ and $\mathcal{D} = \emptyset$.
end Initialization
procedure TREE CONSTRUCTION & CACHE NODES LABELING
  while U_{new} is nonempty **do**
    Pick $u \in U_{new}$, create nodes u_l and u_r, edges (u_l, u) and (u_r, u), add them to $\mathcal{T}_0$.
    Set $a(u_l) = \lceil a(u)/2 \rceil$, $b(u_l) = \lfloor b(u)/2 \rfloor$ and $a(u_r) = a(u) - a(u_l)$, $b(u_r) = b(u) - b(u_l)$.
    Set $\mathbb{Z}(u_l)$ and $\mathbb{Z}(u_r)$ be subsets of $\mathbb{Z}(u)$ of sizes $\min(b(u_l), K)$ and $\min(b(u_r), K)$ respectively with minimum intersection.
    Remove u from U_{new}.
    if $a(u_l) + b(u_l) \geq 2$ **then**
      Add u_l to U_{new}.
    else
      If $b(u_l) == 1$ add u_l to $\mathcal{D}$ otherwise to $\mathcal{C}$.
    end if
    if $a(u_r) + b(u_r) \geq 2$ **then**
      Add u_r to U_{new}.
    else
      If $b(u_r) == 1$ add u_r to $\mathcal{D}$ otherwise to $\mathcal{C}$.
    end if
  end while
end procedure
procedure DELIVERY NODES LABELING
  Let $\mathcal{D} = \{v_1, \ldots, v_\alpha\}$.
  for $r = 1, \ldots, \min(\beta, K)$ **do**
    Pick a node $v \in \mathcal{C}$ with $\mathbb{Z}(v) = \{Z_r\}$ and denote it by $v_{r+\alpha}$.
  end for
  Let $\mathcal{C} \setminus \{v_{\alpha+1}, \ldots, v_{\alpha+\min(\beta,K)}\} = \{v_{\alpha+\min(\beta,K)+1}, \ldots, v_\beta\}$.
  for $t = 1, \ldots, \alpha$ **do**
    for $r = 1, \ldots, \min(\beta, K)$ **do**
      $d_r = (t-1)\min(\beta, K) + r$.
    end for
    for $r = \min(\beta, K) + 1, \ldots, K$ **do**
      $d_r = 1$.
    end for
    Set $\mathbb{D}(v_t) = X_{d_1,\ldots,d_K}$
  end for
end procedure
procedure MODIFY DELIVERY PHASE SIGNALS
  Denote current instance by $P_0(\mathcal{T}_0, \alpha, \beta, L_0, N_0, K)$.
  Modify $P_0(\mathcal{T}_0, \alpha, \beta, L_0, N_0, K)$ by Claim 5 to obtain $P(\mathcal{T}, \alpha, \beta, L, \hat{N}_{sat}, K)$.
end procedure

Output: $\hat{N}_{sat}(\alpha, \beta, K) = |\Gamma(v^*)|$, $P(\mathcal{T}, \alpha, \beta, L, \hat{N}_{sat}, K)$.

partition α into two parts $\alpha_l = \lceil \alpha/2 \rceil$ and $\alpha_r = \alpha - \alpha_l$. On the other hand, β is split into $\beta_l = \lfloor \beta/2 \rfloor$ and $\beta_r = \beta - \beta_l$. The algorithm, then recursively constructs the left and right subtrees of u^*. It is important to note that the split in the (α, β) pair is done in such a manner that each subtree gets the floor and the ceiling of the one of the quantities. Moreover, the labeling of the cache node leaves is such that for a given node u, $|\mathbb{Z}(u_l) \cap \mathbb{Z}(u_r)|$ is as small as possible. The underlying reason for such a labeling is to ensure that the condition of Claim 3 doesn't hold for any $u \in \mathcal{T}$.

Following the construction of the tree, the second phase of the algorithm labels each of the delivery phase nodes, so that the computed lower bound is $L = \alpha\beta$. In this step we use $N = \alpha\beta$ files (see the procedure discussed in the proof of Corollary 1). In the third and final phase of the algorithm we modify the instance so that for any node $u \in \mathcal{T}$, we have that either $\Gamma_l \subseteq \Gamma_r$ or $\Gamma_r \subseteq \Gamma_l$; we use Claim 5 to achieve this. In the beginning all recovered files in the constructed instance are distinct so that $\Gamma(u_l) \cap \Gamma(u_r) = \emptyset$ for all nodes u. W.l.o.g. assume that $|\Gamma(u_r)| \leq |\Gamma(u_l)|$. An application of Claim 5 will thus cause a significant reduction in the number of files that are used. The following lemma quantifies this reduction.

Lemma 3: For given α, β and K if $\beta \leq K$ then,

$$N_{sat}(\alpha, \beta, K) \leq \left\lfloor \frac{2\alpha\beta + \alpha + \beta}{3} \right\rfloor.$$

Proof: We use Algorithm 3 to generate problem instance $P(\mathcal{T}, \alpha, \beta, L, \hat{N}_{sat}, K)$ so that $L = \alpha\beta$. By the definition of the saturation number we have $N_{sat}(\alpha, \beta, K) \leq \hat{N}_{sat}$ hence we just need to show that $\hat{N}_{sat} \leq \frac{2\alpha\beta+\alpha+\beta}{3}$.

First, we need to show that $L = \alpha\beta$. By line 32 of the algorithm the file $W_{(t-1)\beta+r}$ is recoverable in instance P_0 by the pair $(\mathbb{D}(v_t), \mathbb{Z}(v_{\alpha+r}))$ or equivalently $\Delta(v_t, v_{\alpha+r}) = W_{(t-1)\beta+r}$ for $1 \leq t \leq \alpha$ and $1 \leq r \leq \beta$. On the other hand, $\mathbb{W}(v^*) = \cup_{t=1}^{\alpha} \cup_{r=1}^{\beta} \Delta(v_t, v_{\alpha+r})$ therefore $\mathbb{W}(v^*) = \{W_1, \ldots, W_{\alpha\beta}\}$. Recall that $\mathbb{W}(v^*) = \cup_{u \in \mathcal{T}_0} W_{new}(u)$ and $L_0 = \sum_{u \in \mathcal{T}_0} |W_{new}(u)|$ so we have $L_0 \geq |\mathbb{W}(v^*)| = \alpha\beta$. But $L_0 \leq \alpha\beta$, by Corollary 2, therefore $L_0 = \alpha\beta$. In phase III of the Algorithm (Modify Delivery Phase Signals) using Claim 5, we have $L \geq L_0$ and since $L \leq \alpha\beta$ and $L_0 = \alpha\beta$ thus $L = \alpha\beta$.

W.l.o.g we set left incoming node such that $\Gamma(u_r) \subseteq \Gamma(u_l)$. Starting from the root node v^*, we let the set $\{u_0, u_1, \ldots, u_t\}$ and $\{w_0, \ldots, w_{t-1}\}$ to be the left and right incoming nodes respectively so that u_i is topologically higher than u_j for $i < j$, $u_t = u^*$ and u_0 to be a leaf. This is depicted in Fig. 10. Recall that $\Gamma(u) = W_{new}(u) \cup \Gamma(u_l) \cup \Gamma(u_r)$ and $W_{new}(u) \cap (\Gamma(u_l) \cup \Gamma(u_r)) = \emptyset$ for any $u \in \mathcal{T}$. Therefore, recursively we have,

$$\begin{aligned} \hat{N}_{sat} &= |\Gamma(v^*)| = |\Gamma(u_t)|, \\ &= |W_{new}(u_t)| + |\Gamma(u_{t-1})|, \\ &= \sum_{i=1}^{t} |W_{new}(u_i)|, \end{aligned} \tag{11}$$

where we used $W_{new}(u_0) = \emptyset$ since u_0 is a leaf.

In Algorithm 3, $a(u)$ and $b(u)$ denote the number of delivery phase nodes and the number cache nodes, respec-

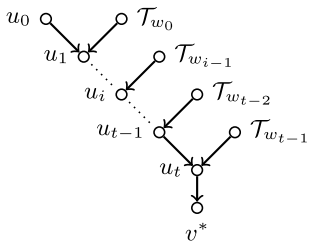

Fig. 10. Saturation path.

tively in the subtree rooted at u. Note that by definition, we have

$$L = |W_{new}(u_t)| + \sum_{u\in\mathcal{T}_{u_{t-1}}} |W_{new}(u)| + \sum_{u\in\mathcal{T}_{w_{t-1}}} |W_{new}(u)|.$$

We conclude that $\sum_{u\in\mathcal{T}_{u_{t-1}}} |W_{new}(u)| \leq a(u_{t-1})b(u_{t-1})$ and $\sum_{u\in\mathcal{T}_{w_{t-1}}} |W_{new}(u)| \leq a(w_{t-1})b(w_{t-1})$ by using Corollary 2. Similarly, using Claim 6, we have that $|W_{new}(u_t)| \leq a(u_{t-1})b(w_{t-1}) + a(w_{t-1})b(u_{t-1})$. In fact, all these inequalities are met with equality. This can be seen as follows. An application of Claim 5 does not change the lower bound, which implies that $L = \alpha\beta = a(u_t)b(u_t)$. But, $a(u_t) = a(u_{t-1}) + a(w_{t-1})$ and $b(u_t) = b(u_{t-1}) + b(w_{t-1})$ so that

$$\begin{aligned} L = {} & a(u_{t-1})b(w_{t-1}) + a(w_{t-1})b(u_{t-1}) \\ & + a(u_{t-1})b(u_{t-1}) + a(w_{t-1})b(w_{t-1}). \end{aligned}$$

An inductive argument can be made to show a similar result for u_i, $i = 1, \ldots, t-1$.

Using these results and the equality in (11) yields,

$$\begin{aligned} \alpha\beta &= L, \\ &= \sum_{u\in\mathcal{T}} |W_{new}(u)|, \\ &= \sum_{i=0}^{t} |W_{new}(u_i)| + \sum_{i=0}^{t-1} \sum_{u\in\mathcal{T}_{w_i}} |W_{new}(u)|, \\ &= \hat{N}_{sat} + \sum_{i=0}^{t-1} (a(w_i)b(w_i)), \\ \Rightarrow \hat{N}_{sat} &= \alpha\beta - \sum_{i=0}^{t-1} a(w_i)b(w_i). \end{aligned} \tag{12}$$

Considering our setting for $a(u)$ and $b(u)$ in the line 9 of Algorithm 3 we have

$$a(u_{i+1}) = a(u_i) + a(w_i), \quad b(u_{i+1}) = b(u_i) + b(w_i), \tag{13}$$

for $0 \leq i \leq t-1$ and either $(a(u_i), b(u_i)) = (\lceil a(u_{i+1})/2 \rceil, \lfloor b(u_{i+1})/2 \rfloor)$ or $(a(u_i), b(u_i)) = (\lfloor a(u_{i+1})/2 \rfloor, \lceil b(u_{i+1})/2 \rceil)$. In any case using eq. (13) we have

$$\begin{aligned} a(u_i) &\leq \lceil a(u_{i+1})/2 \rceil, \\ &\leq \frac{a(u_{i+1}) + 1}{2}, \\ &= \frac{a(u_i) + a(w_i) + 1}{2}, \\ \Rightarrow a(u_i) &\leq a(w_i) + 1. \end{aligned}$$

By a similar argument we have $b(u_i) \leq b(w_i) + 1$. Using eq. (13) recursively, it is easy to see that $\alpha = a(u_0) + \sum_{i=0}^{t-1} a(w_i)$ and $\beta = b(u_0) + \sum_{i=0}^{t-1} b(w_i)$. Therefore, using eq. (12) and (11),

$$\begin{aligned} \hat{N}_{sat} &= \alpha\beta - \sum_{i=0}^{t-1} a(w_i)b(w_i), \\ &= \sum_{i=0}^{t-1} (a(u_i)b(w_i) + a(w_i)b(u_i)), \\ &\leq \sum_{i=0}^{t-1} ([a(w_i) + 1]b(w_i) + a(w_i)[b(w_i) + 1]), \\ &\leq \sum_{i=0}^{t-1} (2a(w_i)b(w_i) + a(w_i) + b(w_i)), \\ &\leq \alpha + \beta + 2\sum_{i=0}^{t-1} a(w_i)b(w_i), \\ \Rightarrow \sum_{i=0}^{t-1} a(w_i)b(w_i) &\geq \frac{\alpha\beta - \alpha - \beta}{3}. \end{aligned}$$

Finally, using the above inequality and eq. (12), we have

$$\begin{aligned} N_{sat}(\alpha, \beta, K) &\leq \hat{N}_{sat}, \\ &= \alpha\beta - \sum_{i=0}^{t-1} \alpha(w_i)\beta(w_i), \\ &\leq \alpha\beta - \frac{\alpha\beta - \alpha - \beta}{3} = \frac{2\alpha\beta + \alpha + \beta}{3}. \end{aligned}$$

Furthermore as $N_{sat}(\alpha, \beta, K)$ is an integer we conclude that

$$N_{sat}(\alpha, \beta, K) \leq \left\lfloor \frac{2\alpha\beta + \alpha + \beta}{3} \right\rfloor.$$

■

The aforementioned upper bound on the saturation number is tight. To see this, let consider $\beta = 1$. It is easy to see that $N_{sat}(\alpha, 1, K) = \alpha$ and using Lemma 3 we have $N_{sat} \leq \lfloor \alpha + 1/3 \rfloor = \alpha$.

B. Best Lower Bound for a Fixed M

Theorem 1 and Corollary 2 characterize optimal and near-optimal problem instances for fixed α and β. In general, the best lower bound on the rate is obtained when we optimize over a range of choices for α and β. In our approach we restrict β to be less than $2K$, i.e., $\beta < 2K$. Our next result shows that an atomic problem instance with $\beta < 2K$ has $\alpha < 2N$. As a result, when $\beta < 2K$ the range of α, β pairs that we need to consider is limited.

Lemma 4: Any problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ with $\beta < 2K$ and $\alpha \geq 2N$ is non-atomic.

Proof: We let (u^*, v^*) to be the last edge in $\mathcal{T}$. If $\alpha \geq 2N$ then either $\alpha_l \geq N$ or $\alpha_r \geq N$ or both. W.l.o.g. we assume that $\alpha_l \geq N$. We note that $\beta_l < 2K$ as $\beta < 2K$. Claim 7 below shows that $N_{sat}(\alpha_l, \beta_l, K) \geq N$ for $\beta_l < 2K$. Therefore, $N \leq N_0 = \max\{N_{sat}(\alpha_l, \beta_l, K), N_{sat}(\alpha_r, \beta_r, K)\}$ and

from (10) we have $|W_{new}(u^*)| = 0$. This implies that the problem is non-atomic. ∎

Claim 7: $N_{sat}(\alpha, \beta, K) \geq \alpha$ for any $\beta < 2K$.

Proof: We use an inductive argument. Clearly, $N_{sat}(1, \beta', K) \geq 1$ for nonzero β' since at least one file must be used. Furthermore, by inspection we have $N_{sat}(\alpha', 1, K) = \alpha'$. Therefore, the base cases are established. Now, we assume that $N_{sat}(\alpha', \beta', K) \geq \alpha'$ for all $\alpha' \leq \alpha$ and $\beta' \leq \beta < 2K$.

We will first show that $N_{sat}(\alpha, \beta + 1, K) \geq \alpha$. Let $P(\mathcal{T}, \alpha, \beta + 1, L, N_s, K)$ be the problem instance associated with $N_{sat}(\alpha, \beta + 1, K)$ so that $N_s = N_{sat}(\alpha, \beta + 1, K)$ and $L = \alpha \min(\beta + 1, K)$. We also let (u^*, v^*) to be the last edge in $\mathcal{T}$ and P_l and P_r to be the problem instances corresponding to $\mathcal{T}_{u^*(l)}$ and $\mathcal{T}_{u^*(r)}$ respectively. By Claim 6, $|W_{new}(u^*)| \leq \rho(u^*) = \alpha_l[\min(\beta_r, K - \beta_l)]^+ + \alpha_r[\min(\beta_l, K - \beta_r)]^+$. We claim that $|W_{new}(u^*)| = \rho(u^*)$, $L_l = \alpha_l \min(\beta_l, K)$, and $L_r = \alpha_r \min(\beta_r, K)$ for problem instance P. This follows from the fact that $L = |W_{new}(u^*)| + L_l + L_r = \alpha \min(\beta + 1, K)$ and the limits on $|W_{new}(u^*)|$, L_l, and L_r discussed in Claim 6 and Corollary 1. The problem instances P_l and P_r are both saturated instances and each uses the minimum number of files. If this is not the case, replacing them in P with problem instances associated with $N_{sat}(\alpha_l, \beta_l, K)$ and $N_{sat}(\alpha_r, \beta_r, K)$ will result in a problem instance $P''(\mathcal{T}'', \alpha, \beta, L, N_s'', K)$ with $N_s'' < N_s$. But this contradicts our assumption that P is a problem instance associated with $N_{sat}(\alpha, \beta + 1, K)$. Thus we have $|\Gamma(v_l^*, v_l^*)| = N_{sat}(\alpha_l, \beta_l, K)$ and $|\Gamma(v_r^*, v_r^*)| = N_{sat}(\alpha_r, \beta_r, K)$. Then

$$\begin{aligned}
&N_{sat}(\alpha, \beta + 1, K)\\
&= |\Gamma(v^*, v^*)|,\\
&= |W_{new}(u^*) \cup \Gamma(u_l^*, u_l^*) \cup \Gamma(u_r^*, u_r^*)|,\\
&= |W_{new}(u^*)| + |\Gamma(u_l^*, u_l^*) \cup \Gamma(u_r^*, u_r^*)|,\\
&\geq |W_{new}(u^*)| + \max\left(N_{sat}(\alpha_l, \beta_l, K),\ N_{sat}(\alpha_r, \beta_r, K)\right),\\
&= \rho(u^*) + \max\left(N_{sat}(\alpha_l, \beta_l, K),\ N_{sat}(\alpha_r, \beta_r, K)\right). \qquad (14)
\end{aligned}$$

We note that we are guaranteed that either $[\min(\beta_r, K - \beta_l)]^+ \geq 1$ or $[\min(\beta_l, K - \beta_r)]^+ \geq 1$ or both must hold as $\beta + 1 < 2K$. Thus, we can assert that $\rho(u^*) \geq \min(\alpha_l, \alpha_r)$. Now, if we have $\beta_l > 0$ and $\beta_r > 0$, then using the induction hypothesis, we have $\max\left(N_{sat}(\alpha_l, \beta_l, K),\ N_{sat}(\alpha_r, \beta_r, K)\right) \geq \max(\alpha_l, \alpha_r)$ so that $N_{sat}(\alpha, \beta + 1, K) \geq \alpha$. On the other hand if w.l.o.g. $\beta_r = 0$, we have from eq. (14) that

$$\begin{aligned}
&N_{sat}(\alpha, \beta + 1, K)\\
&\quad \geq (\alpha - \alpha_l) \min(\beta + 1, K) + N_{sat}(\alpha_l, \beta + 1, K),\\
&\quad \geq \alpha - \alpha_l + N_{sat}(\alpha_l, \beta + 1, K).
\end{aligned}$$

One can argue recursively by considering the left and right branches of the instance associated with $N_{sat}(\alpha_l, \beta + 1, K)$ and arrive at the required result.

Next, we show that $N_{sat}(\alpha + 1, \beta, K) \geq \alpha$. In this case, as before let (u^*, v^*) be the last node of the instance and let P_l and P_r to be the problem instances associated with $\mathcal{T}_{u^*(l)}$ and $\mathcal{T}_{u^*(r)}$ respectively. Now, if $\alpha_l > 0$ and $\alpha_r > 0$, then the induction hypothesis can be applied to conclude that $\max\left(N_{sat}(\alpha_l, \beta_l, K),\ N_{sat}(\alpha_r, \beta_r, K)\right) \geq \max(\alpha_l, \alpha_r)$ so that the result holds. On the other hand, if w.l.o.g. $\alpha_r = 0$, then we have from eq. (14) that

$$N_{sat}(\alpha + 1, \beta, K) \geq N_{sat}(\alpha + 1, \beta_l, K),$$

where $\beta_l < \beta$. One can recursively argue by examining the left and right branches of the instance associated with $N_{sat}(\alpha_l + 1, \beta_l, K)$ and arrive at the required result, by using the fact that $N_{sat}(\alpha, 1, K) \geq \alpha$ for any α.

The results for $N_{sat}(\alpha, \beta + 1, K)$ and $N_{sat}(\alpha + 1, \beta, K)$ can be used to show the corresponding result for $N_{sat}(\alpha + 1, \beta + 1, K)$ in a similar manner. ∎

Thus far we have shown that the range of α is limited to $\alpha < 2N$ when β is limited to $\beta < 2K$. In fact, $\beta \geq 2K$ is a valid choice, though in our experiments it does not appear to yield any better lower bounds on the rate than the ones we have right now. If these choices of β are useful, they are likely to yield better lower bounds only in the regime when M is very small. The reason for this behavior is that for a fixed α the saturation number $N_{sat}(\alpha, \beta, K)$ takes maximum value at $\beta = K$ and starts decreasing once $\beta > K$.

Although Algorithm 3 is used to get an analytical upper bound on the saturation number, the exact saturation number $N_{sat}(\alpha, \beta, K)$ is recursively computable. It is not hard to see that the inequality in (14) is met with equality for a problem instance $P(\mathcal{T}, \alpha, \beta, L, N_s, K)$ associated with the saturation number $N_{sat}(\alpha, \beta, K)$. This is a consequence of the fact that either $\Gamma_l \subseteq \Gamma_r$ or $\Gamma_r \subseteq \Gamma_l$. For a fixed α, β, and K, there are limited possibilities for $0 \leq \alpha_l \leq \alpha$ and $0 \leq \beta_l \leq \beta$. Corresponding to each possible (α_l, β_l) we can construct a saturated problem instance. This also includes the problem instance associated with the saturation number $N_{sat}(\alpha, \beta, K)$. Therefore, the following recurrence holds

$$\begin{aligned}
&N_{sat}(\alpha, \beta, K)\\
&\quad = \min_{(\alpha_l, \beta_l) \in \mathcal{I}(\alpha, \beta)} \Big\{\rho(\alpha_l, \beta_l, \alpha, \beta)\\
&\qquad + \max\left(N_{sat}(\alpha_l, \beta_l, K),\ N_{sat}(\alpha - \alpha_l, \beta - \beta_l, K)\right)\Big\},
\end{aligned}$$

where $\rho(\alpha_l, \beta_l, \alpha, \beta) = \alpha_l[\min(\beta - \beta_l, K - \beta_l)]^+ + (\alpha - \alpha_l)[\min(\beta_l, K + \beta_l - \beta)]^+$ and $\mathcal{I}(\alpha, \beta) = \{(a, b) :\ 0 \leq a \leq \alpha,\ 0 \leq b \leq \beta\} \setminus \{(0, 0),\ (\alpha, \beta)\}$. We note that $(\alpha_l, \beta_l) \in \{(0, 0),\ (\alpha, \beta)\}$ are trivial and we ignore those cases. Using this recurrence, Algorithm 4 computes the saturation number in time which is polynomial in the (α, β) pair (see the analysis in Appendix E).

Thus, the overall process of computing the lower bound on the rate for a fixed value of M proceeds as follows. We consider $1 \leq \alpha \leq 2N$ and $1 \leq \beta \leq 2K - 1$. For each (α, β) in this range, we consider all possible (α_l, β_l) and (α_r, β_r) pairs and compute the lower bound on $\alpha R^\star + \beta M$. This procedure requires us to precompute $N_{sat}(a, b, K)$ for $1 \leq a \leq 2N$ and $1 \leq b \leq 2K$. The precomputation step has time-complexity $O(N^2K^2)$ (see Appendix E). After this step, we start computing the lower bounds over all possible (α, β) pairs. For each value of (α, β), and for a specific (α_l, β_l) and (α_r, β_r) such that $\alpha_l + \alpha_r = \alpha$ and $\beta_l + \beta_r = \beta$, the complexity of computing the lower bound is $O(1)$ since we can use the

Algorithm 4 Computing Saturation Number $N_{sat}(\alpha, \beta, K)$

Input: α, β and K.

Initialization:

1: For all $a \in \{0, \ldots, \alpha\}$ and $b \in \{0, \ldots, \beta\}$ set

$$N_{sat}(a, 0, K) = 0, \quad N_{sat}(0, b, K) = 0,$$
$$N_{sat}(a, 1, K) = a, \quad N_{sat}(1, b, K) = \min(b, K).$$

Main loop:

2: **for** $a = 2; a \leq \alpha; a++$ **do**

3: **for** $b = 2; b \leq \beta; b++$ **do**

4:

$$N_{sat}(a, b, K) = \min_{(\tilde{a}, \tilde{b}) \in \mathcal{I}(a,b)} \Big\{ \rho(\tilde{a}, \tilde{b}, a, b) + \max\Big(N_{sat}(\tilde{a}, \tilde{b}, K), N_{sat}(a - \tilde{a}, b - \tilde{b}, K) \Big) \Big\}$$

5: **end for**

6: **end for**

Output: $N_{sat}(\alpha, \beta, K)$

characterization of Theorem 1 and the saturation numbers are precomputed. Thus, for a value of (α, β), the complexity of computing the bound is $O(\alpha\beta) \leq O(NK)$. As, we consider a total of NK values of (α, β) in total, the time-complexity of our procedure is $O(N^2K^2)$.

IV. Multiplicative Gap Between Upper and Lower Bounds

We now show that for any set of problem parameters, our proposed lower bound and the achievable rate of [9] in eq. (2) are within a factor of four, i.e., we show the following result.

Theorem 2: Consider a coded caching system with N files and K users each with a normalized cache size M. Then,

$$\gamma(M) = \frac{R_c(M)}{R^\star(M)} \leq 4,$$

for $0 \leq M \leq N$.

The key idea in proving this result is to exploit the analytical upper bound on the saturation number $N_{sat}(\alpha, \beta, K)$ proposed in Section III-A. For a given N and K, we consider three distinct regions of M. For each range, an appropriate (α, β) pair allows us to obtain a lower bound on the rate that is within a factor of four of the achievable rate.

Proof: We use Corollary 2 with the 2α and 2β, so that P_l' and P_r' have parameters α and β. This gives us the following lower bound.

$$2\alpha R^\star(M) + 2\beta M \geq \min\left(2\alpha \min(2\beta, K), 2\alpha\beta + [N - N_0]^+\right),$$

Moreover, we restrict $2\beta \leq K$ so that,

$$2\alpha R^\star(M) + 2\beta M \geq \min\left(4\alpha\beta, 2\alpha\beta + [N - N_0]^+\right)$$
$$\implies R^\star(M) \geq \min\left(2\beta, \beta + \frac{[N - N_0]^+}{2\alpha}\right) - \frac{\beta}{\alpha} M. \quad (15)$$

Our first observation is that for $\min(N, K) \leq 4$, the bound is easily seen to be true. Towards this end, by setting $\alpha = N, \beta = 1$ in (15), we obtain

$$R^\star(M) \geq 1 - \frac{M}{N}.$$

where we used $N_{sat}(N, 1, K) = N$. Furthermore, from eq. (2),

$$R_c(M) \leq \min(N, K)\,(1 - M/N),$$

This means that $\gamma(M) = \min(N, K) \leq 4$ for $\min(N, K) \leq 4$.

Thus, in the subsequent discussion, we only consider $\min(N, K) \geq 5$. As in [9], we divide the M-axis to three separated regions. For given M, we explore the space of (α, β) pairs to obtain an appropriate lower bound that allows us to show the multiplicative gap of four.

A. Region I: $0 \leq M \leq \max(1, N/K)$

First, we consider the range $0 \leq M \leq 1$. In eq. (15) we set $\alpha = 1, \beta = \lfloor \min(N, K)/2 \rfloor$. By such a setting we have $2\beta \leq \min(N, K) \leq K$ and $N \geq N_{sat}(1, \beta, K) = \beta$. Therefore for $M \leq 1$,

$$\begin{aligned} R^\star(M) &\geq \min\left(2\beta, \frac{N + \beta}{2}\right) - \beta M \\ &\overset{(a)}{\geq} \min\left(\beta, \frac{N - \beta}{2}\right) \\ &\overset{(b)}{\geq} \min\left(\frac{\min(N, K) - 1}{2}, \frac{N - \min(N, K)/2}{2}\right) \\ &\overset{(c)}{\geq} \min\left(\frac{\min(N, K) - 1}{2}, \frac{\min(N, K)}{4}\right) \\ &\overset{(d)}{\geq} \frac{\min(N, K)}{4} \\ &\geq \frac{\min(N, K)(1 - M/N)}{4} \\ &\geq R_c(M)/4. \end{aligned}$$

Here, (a) holds since $M \leq 1$, (b) holds since $(\min(N, K) - 1)/2 \leq \beta \leq \min(N, K)/2$, (c) holds since $N \geq \min(N, K)$, and (d) holds since $\min(N, K) \geq 2$.

Next, consider the range $M \in [1, N/K]$. Note that we only need to consider the scenario where $N \geq K$. The achievable rate $R_c(M)$ in this interval is upper bounded by the convex combination of the rates $R_c(0)$ and $R_c(N/K)$ so that

$$R_c(M) \leq \lambda R_c(N/K) + (1 - \lambda) R_c(0) = K(1 - \lambda/2) - \lambda/2,$$

where $\lambda = KM/N$. Now, we set $\alpha = \lceil N/K \rceil, \beta = \lfloor K/2 \rfloor$ so that $\alpha\beta \leq (N/K + 1)K/2 = N/2 + K/2 \leq N$. As, $N_{sat}(\alpha, \beta, K) \leq \alpha\beta$, this means that $N \geq N_{sat}(\alpha, \beta, K)$. In addition, note that $2\beta \leq K$. Therefore, we can use eq. (15)

to obtain

$$\begin{aligned}
R^\star(M) &\geq \min\left\{2\beta,\ \beta+\frac{N-N_{sat}(\alpha,\beta,K)}{2\alpha}\right\}-\frac{\beta}{\alpha}M,\\
&\overset{(a)}{\geq} \min\left\{2\beta\left(1-\frac{M}{2\alpha}\right),\ \frac{2\beta}{3}+\frac{N-2\beta M}{2\alpha}-\frac{\beta}{6\alpha}-\frac{1}{6}\right\},\\
&\overset{(b)}{\geq} \min\left\{(K-1)\left(1-\frac{KM}{2N}\right),\ \frac{\beta}{2}+\frac{N-2\beta M}{4N/K}-\frac{1}{6}\right\},\\
&\geq \min\left\{\frac{K}{2}\left(1-\frac{\lambda}{2}\right),\ \frac{\beta}{2}(1-\lambda)+\frac{K}{4}-\frac{1}{6}\right\},\\
&\overset{(c)}{\geq} \min\left\{\frac{R_c(M)}{2},\ \frac{K}{2}\left(1-\frac{\lambda}{2}\right)-\frac{(1-\lambda)}{4}-\frac{1}{6}\right\},\\
&\overset{(d)}{\geq} \min\left\{\frac{R_c(M)}{2},\ \frac{R_c(M)}{4}+\frac{(K-3)}{4}\left(1-\frac{\lambda}{2}\right)+\frac{1}{3}\right\},\\
&\overset{(e)}{\geq} R_c(M)/4, \qquad (16)
\end{aligned}$$

where in (a) we used Lemma 3 to bound $N_{sat}(\alpha,\beta,K)$, in (b) we used $N-2\beta M\geq 0$, $1\leq\alpha\leq N/K+1\leq 2N/K$, $(K-1)/2\leq\beta$, and in (c) we used $\beta\geq(K-1)/2$, $\lambda=KM/N$ and the expression for the upper bound on $R_c(M)$ above. Next, (d) holds because of the achievable rate bound and (e) holds since $\min(N,K)\geq 5$. Therefore, $\gamma(M)\leq 4$ for $M\in[1,N/K]$ and $N\geq K$. Thus, we conclude that we have $\gamma(M)\leq 4$ for $M\in[0,\max(1,N/K)]$.

B. Region II: $\max(1,N/K)<M\leq N/2$

For M such that $\max(N/K,1)<M\leq N/2$ we define $t_0=\lfloor KM/N\rfloor$ so that $t_0N/K<M\leq(t_0+1)N/K$. Since $M\geq N/K$ thus $t_0\geq 1$. Using eq. (2), it turns out that,

$$\begin{aligned}
R_c(M) &\leq R_c(t_0N/K),\\
&= \frac{K}{t_0+1}-\frac{t_0}{t_0+1},\\
&\overset{(a)}{\leq} \frac{K}{KM/N}-\frac{1}{2},\\
&= \frac{N}{M}-\frac{1}{2},
\end{aligned}$$

where (a) holds since $t_0+1\geq KM/N$ and $t_0\geq 1$.

Now, consider setting $\alpha=\lfloor 2M\rfloor$ and $\beta=\lfloor N/2M\rfloor$. With this setting we have $\alpha\geq 2$ (since $M\geq 1$), $\beta\geq 1$ (since $M\leq N/2$), and $\beta\leq N/2M<K/2$ (since $M>N/K$). Furthermore, since $\alpha\beta\leq 2M\times N/2M=N$ and $N_{sat}(\alpha,\beta,K)\leq\alpha\beta$ therefore $N\geq N_{sat}(\alpha,\beta,K)$. This together with $2\beta\leq K$ implies that such a setting allows the usage of (15). Therefore, using Lemma 3 to bound $N_{sat}(\alpha,\beta,K)$, we have

$$R^\star(M)\geq\min\left\{2\beta,\frac{2\beta}{3}+\frac{N}{2\alpha}-\frac{\beta}{6\alpha}-\frac{1}{6}\right\}-\frac{\beta}{\alpha}M.$$

We claim that $2\beta\geq 2\beta/3+N/2\alpha-\beta/6\alpha-1/6$ or equivalently $8\alpha\beta+\alpha+\beta\geq 3N$. This can be seen as follows. When, $N/4<M\leq N/2$ we have $\alpha>N/2,\beta=1$, so that this holds. On the other hand when $\max(1,N/K)<M\leq N/4$, we have $\alpha\geq 2M-1$, $\beta\geq N/2M-1$, so that $8\alpha\beta+\alpha+\beta\geq 8N-7(N/2M+2M)+6$. It can been seen that $N/2M+2M\leq N/2+2$ for $1\leq M\leq N/4$ therefore $8\alpha\beta+\alpha+\beta\geq 9N/2-8\geq 3N$ for $N\geq 6$. For $N=5$, the claim trivially holds since $\alpha\geq 2,\beta\geq 1$ so that $8\alpha\beta+\alpha+\beta\geq 19\geq 3\times N=15$.

Thus, we have

$$\begin{aligned}
R^\star(M) &\geq \frac{2\beta}{3}+\frac{N-2\beta M}{2\alpha}-\frac{\beta}{6\alpha}-\frac{1}{6},\\
&\overset{(a)}{\geq} \frac{7\beta}{12}+\frac{N-2\beta M}{4M}-\frac{1}{6},\\
&= \frac{N}{4M}+\frac{\beta}{12}-\frac{1}{6},\\
&\overset{(b)}{\geq} \frac{N}{4M}-\frac{1}{12},\\
&\geq \frac{N}{4M}-\frac{1}{8}\\
&\geq \frac{R_c(M)}{4},
\end{aligned}$$

where in (a) we used $N-2\beta M\geq 0$, $\alpha\geq 2$ and $\alpha\leq 2M$ and in (b) we used $\beta\geq 1$. Eventually, $\gamma(M)\leq 4$ for $\max(N/K,1)\leq M\leq N/2$.

C. Region III: $N/2<M\leq N$

Let $t_0=\lfloor K/2\rfloor$ so that $M\geq t_0N/K$ for $M\in(N/2,N]$. For any $M\in(N/2,N]$ the convex combination of rate $R_c(t_0N/K)$ and $R_c(N)$ gives us $R_c(M)\leq\lambda R_c(t_0N/K)+(1-\lambda)R_c(N)=\lambda R_c(t_0N/K)$ where $M=\lambda t_0N/K+(1-\lambda)N$ or equivalently $\lambda=(1-M/N)/(1-t_0/K)$. According to this and eq. (2) we observe that,

$$\begin{aligned}
R_c(M) &\leq \lambda R_c(t_0N/K),\\
&= \frac{(1-M/N)}{(1-t_0/K)}\frac{(K-t_0)}{(t_0+1)},\\
&= \frac{K(1-M/N)}{(1+t_0)},\\
&\overset{(a)}{\leq} \frac{K(1-M/N)}{K/2},\\
&= 2(1-M/N),
\end{aligned}$$

where (a) holds since $1+t_0=1+\lfloor K/2\rfloor\geq K/2$.

Now if we set $\alpha=N$ and $\beta=1$ in (15) we obtain

$$\begin{aligned}
R^\star(M) &\geq 1-M/N\\
&\geq \frac{R_c(M)}{2}.
\end{aligned}$$

This implies that $\gamma(M)\leq 2\leq 4$ for $M\in[N/2,N]$ and concludes the proof. ∎

V. Lower Bounds on the Other Variants of the Coded Caching Problem

In addition to the original coded caching problem there are many variants of the problem including coded caching with multiple requests [22], decentralized coded caching [14] and caching in device to device wireless networks [23]. Our proposed strategy applies with minor changes for these problems.

A. Caching in Device to Device Wireless Networks

Wireless device to device (D2D) networks where communication is limited to be single-hop are studied in [23]. There are K users who are the nodes of the network. Each user has a cache of size M and N files are stored across the different user caches. Thus, in this setting we necessarily have $KM \geq N$. As in the coded caching problem there are placement and delivery phases. In the placement phase the caches are populated from a server; this phase does not depend on the user demands. The server then leaves the network. We let Z_i represent the cache content of the i-th user. In the delivery phase each user requests a file and the remaining users are informed about this request. Based on the requests, each user broadcasts a signal so that all demands can be satisfied. We denote by $X^{(i)}_{d_1,\ldots,d_K}$ the signal that is broadcasted in the delivery phase by the i-th user when the j-th user requests file $d_j \in [N]$ for $1 \leq j \leq K$. The delivery signal sent by each user is a function of its cache content so that $H(X^{(i)}_{d_1,\ldots,d_K}|Z_i) = 0$. We also denote by $X_{d_1,\ldots,d_K}$ the set of signals sent by all the users, i.e., $X_{d_1,\ldots,d_K} = \{X^{(1)}_{d_1,\ldots,d_K}, \ldots, X^{(K)}_{d_1,\ldots,d_K}\}$. The rate of the signal that the i-th user sends in the delivery phase is denoted by $R_{i,d_1,\ldots,d_K}(M)$. We are interested in lower bounding the worst case rate that denoted by $R^\star(M) = K \max_{i,d_1,\ldots,d_k} R_{i,d_1,\ldots,d_K}(M)$.

The cut-set technique and Han's inequality have been studied in [23] and [24] respectively to establish lower bounds on $R^\star(M)$. The multiplicative gap established in [23] depends on M and is not constant, whereas [24] shows a gap of at most 8.

The D2D setting is almost exactly the same as the coded caching setting studied in our work. Our technique for obtaining lower bounds is applicable here with essentially no change and we can use Theorem 1 and its corollary. Furthermore, since $H(X^{(i)}_{d_1,\ldots,d_K}|Z_i) = 0$ we can get lower bounds that are somewhat tighter. By treating $X_{d_1,\ldots,d_K}$ as the delivery signal of the original coded caching problem, we can apply our lower bound to show that the multiplicative gap between the achievable rate in [23] and our proposed lower bounds is at most 4. The proof is quite similar to that of Theorem 2 and is omitted.

B. Coded Caching With Multiple Requests

Coded caching with multiple requests is variation of the original problem in which each user requests l files from the server in the delivery phase. A straightforward achievable scheme in this setting is to apply the scheme of [9] l times. This problem is investigated in [22] where a new achievable scheme is proposed based on multiple groupcast index coding. Furthermore, [22] introduces a cut-set type lower bound and shows that their scheme is within a multiplicative factor of 18 of the lower bound. In contrast, using our approach we can demonstrate a multiplicative gap of 4 for this problem as well.

In this setting the only difference with respect to the original problem is that from a cache signal Z_i and delivery signal $X_{d_1,\ldots,d_K}$ one can recover up to l distinct files. Thus, d_i is a vector of size l containing information about the l files requested by i-th user. Therefore, all statements we presented for the original problem are applicable here, bearing in mind that $Rec(Z_i, X_{d_1,\ldots,d_K})$ can be as large as l. For instance, an extension of eq. (8) gives us $L \leq l\alpha \min(\beta, K)$. Similarly, the saturation number $N_{sat}(\alpha, \beta, K, l)$ is defined as the minimum N' among all problem instances $P(\mathcal{T}, \alpha, \beta, L, N', K, l)$ with $L = l\alpha \min(K, \beta)$. It is easy to verify that $N_{sat}(\alpha, \beta, K, l) \leq l\alpha \min(\beta, K)$ in a similar way. The following claim can be shown (we omit the proof as it is very similar to the previous discussion).

Claim 8: Consider a coded caching system with a server containing N files and K users. Each user has a cache of size M and demands l files in the delivery phase. The following lower bound holds for $N \geq N_0$ where $N_0 = N_{sat}(\alpha, \beta, K, l)$,

$$\alpha R^\star(M) + \beta M \geq \min\left\{2l\alpha \min(\beta, K),\ l\alpha \min(\beta, K) + (N - N_0)/2)\right\}.$$

Similarly, an extension of the Lemma 3 holds so that $N_{sat}(\alpha, \beta, K, l) \leq l(2\alpha\beta + \alpha + \beta)/3$ for $\beta \leq K$. Exploiting this upper bound and Claim 8, we are able to show that the multiplicative gap of the straightforward achievable scheme and our lower bound is at most 4. Let $R^l_c(M) = lR_c(M)$ where $R_c(M)$ is defined in eq. (2).

Theorem 3: Consider a coded caching system with a server containing N files and K users. Each user requests l files, and has a cache of size $0 \leq M \leq N$. Then

$$\frac{R^l_c(M)}{R^\star(M)} \leq 4.$$

Proof: We divide the M axis into three regions, $0 \leq M \leq \max(l, N/K)$, $\max(l, N/K) \leq M \leq N/2$, and $N/2 \leq M \leq N$. In each region we show $R^l_c(M)/R^\star(M) \leq 4$ for any N and K. In the following proof, $M = l$ plays the same role as $M = 1$ in proof of Theorem 2. Before embarking on the proof, we note that we only need to analyze the gap for $\min(N, lK) \geq 5$. Note that the lower bounds of the original problem are also valid here. Indeed, if each user requests the same file l times (instead of requesting l distinct files) the problem will be equivalent to the original one. Now, in (15) if we set $\alpha = N$ and $\beta = 1$ then we get $NR^\star + M \geq N$, or equivalently $R^\star(M) \geq (1 - M/N)$, which is applicable to the multiple request problem. Since $R^l_c(M) \leq \min(N, lK)(1 - M/N)$, therefore $R^l_c(M)/R^\star(M) \leq 4$ for $\min(N, lK) \leq 4$.

1) Region I: $0 \leq M \leq \max(l, N/K)$: For $0 \leq M \leq \max(l, N/K)$, we first show that the result holds for $M \leq l$. Since we separately analyze the gap for $M \geq N/2$ we assume $l \leq N/2$ so that $M \leq \max(l, N/K) \leq N/2$. We use result of the Claim 8 with setting $\alpha = 1$ and $\beta = \lfloor \min(N/2l, K/2) \rfloor$ where $\beta \geq 1$ from $l \leq N/2$. Following the exact same steps as in Section IV-A for $M \leq 1$, it turns out that $R^\star(M) \geq \min(N, lK)/4 \geq R^l_c(M)/4$ for $M \leq l$.

Now, we assume that $l \leq M \leq \max(l, N/K)$ which is nonempty if $N/K \geq l$. Therefore, we only need to analyze the gap for $N \geq lK$ and $l \leq M \leq N/K$. In this range of M the convex combination of $M = 0$ and $M = N/K$ is achievable so that $R^l_c(M) \leq \lambda R^l_c(N/K) + (1 - \lambda)R^l_c(0)$. From $R^l_c(0) = lK$ and $R^l_c(N/K) = l(K - 1)/2$ we have

$R_c^l(M) \leq lK(1-\lambda/2) - l\lambda/2$ where $\lambda = KM/N$. By setting $\alpha = \lceil N/lK \rceil$ and $\beta = \lfloor K/2 \rfloor$, we have $\alpha\beta \leq \alpha K/2 \leq N/2l + K/2 \leq N/l$ (from $lK \leq N$) and that $N_{sat}(\alpha, \beta, K, l) \leq l\alpha\beta \leq N$. This ensures that the setting is valid for using Claim 8. According to Claim 8 for such a setting we have,

$$
\begin{aligned}
&R^*(M) \\
&\geq \min\left(2l\beta, l\beta + \frac{N - N_{sat}(\alpha,\beta,K,l)}{2\alpha}\right) - \frac{\beta M}{\alpha}, \\
&\overset{(a)}{\geq} \min\left(\frac{lK}{2}\left(1-\frac{\lambda}{2}\right), \frac{lK(1-\lambda/2)}{2} - \frac{l(1-\lambda)}{4} - \frac{l}{6}\right), \\
&\overset{(b)}{\geq} \min\left(\frac{R_c(M)}{2}, \frac{lK(1-\lambda/2)}{4} + \frac{l(1-\lambda/2)}{2} - \frac{l(5-3\lambda)}{12}\right), \\
&= \min\left(\frac{R_c(M)}{2}, \frac{lK(1-\lambda/2)}{4} + \frac{l}{12}\right), \\
&\geq \min\left(\frac{R_c(M)}{2}, \frac{R_c(M)}{4}\right) \geq \frac{R_c(M)}{4},
\end{aligned}
$$

where inequality (a) can be obtained by making the same argument as we made in the first five lines of eq. (16) and (b) from $K \geq 2$.

2) Region II: $\max(l, N/K) \leq M \leq N/2$*:* In the first step, we try to get an upper bound on the achievable rate. Letting $t_0 = \lfloor KM/N \rfloor$ and following the argument we made in Section IV-B gives us $R_c^l(M) \leq lR_c(M) \leq l\,(N/M - 1/2)$ for M in this range. Next, by setting $\alpha = \lfloor 2M/l \rfloor$ and $\beta = \lfloor N/2M \rfloor$ we have $N_{sat}(\alpha, \beta, K, l) \leq l\alpha\beta \leq N$ and $\beta \leq N/2M \leq K/2$ (since $M \geq N/K$) which imply that the constraints of the Claim 8 are satisfied. Therefore,

$$
\begin{aligned}
R^\star &\geq \min\left(2l\beta,\ l\beta + \frac{N - N_{sat}(\alpha,\beta,K,l)}{2\alpha}\right) - \frac{\beta M}{\alpha}, \\
&\overset{(a)}{\geq} \min\left(2l\beta\left(1-\frac{M}{2l\alpha}\right),\ \frac{7l\beta}{12} + \frac{N-2\beta M}{2\alpha} - \frac{l}{6}\right), \\
&\overset{(b)}{\geq} \min\left(2l\beta\left(1-\frac{M}{2M}\right),\ \frac{7l\beta}{12} + \frac{N-2\beta M}{4M/l} - \frac{l}{6}\right), \\
&\overset{(c)}{\geq} \min\left(\frac{Nl}{4M},\ \frac{Nl}{4M} - \frac{l}{12}\right), \\
&\geq R_c^l(M)/4,
\end{aligned}
$$

where in (a) we used upper bound on $N_{sat}(\alpha, \beta, K, l)$ and that $\beta/\alpha \leq \beta/2$ (from $\alpha \geq 2$), in (b) we used $N - 2\beta M \geq 0$, $\alpha \leq 2M/l$, and $\alpha \geq 2M/l - 1 \geq M/l$ (from $M \leq l$). In (c) we used $\beta \geq K/4$ (for $K \geq 2$) and $\beta \geq 1$ (from $M \leq N/2$).

3) Region III: $N/2 \leq M \leq N$*:* Using the same argument we made in Section IV-C the achievable rate is bounded by $R_c^l(M) \leq lR_c(M) \leq 2l\,(1 - M/N)$. According to Claim 8 by setting $\alpha = \lfloor N/l \rfloor$ and $\beta = 1$ one may not recover all N files since $\alpha l \leq N$, but if we increase α to $\lceil N/l \rceil$ then all files will be recovered. Therefore $\alpha R^\star(M) + M \geq N$ or equivalently $R^\star(M) \geq (N - M)/\alpha$. From $N - M \geq 0$ and that $\alpha \leq N/l + 1 \leq 2N/l$ (since $l \leq N$) it turns out that $R^\star(M) \geq l(1 - M/N)/2 \geq 4R_c^l(M)$ for $N/2 \leq M \leq N$. This concludes the proof. ∎

C. Decentralized Coded Caching

In the original coded caching problem the placement phase is managed by a central server. However, in many scenarios such coordinated placement phase may be impractical. Instead, a decentralized placement phase was investigated in [14] where the users cache random subsets of the bits of each file while respecting the cache size constraint. Even in this setting a multiplicative gap of 12 to the cut-set lower bound was obtained. Note that the lower bounds established for the centralized coded caching problem are also applicable to the decentralized case. By similar techniques to those used in proof of Theorem 2 we can establish a multiplicative gap of 4. The proof is omitted as it is quite similar.

VI. Comparison With Existing Results

Lower bounds on the coding caching rate have been proposed in independent work as well. In this section we compare our lower bounds with other approaches.

A. Comparison With Cutset Bound

Our first observation is that the cutset bound in [9] is a special case of the bound in eq. (9). In particular, suppose that $\alpha = \lfloor N/s \rfloor$, $\beta = s$ for $s = 1, \ldots, \min(N, K)$. In this case, we have $\alpha\beta \leq N$. Thus, it is easy to construct a problem instance where $L = \alpha\beta$ (see Corollary 1). This also follows from observing that $N_{sat}(\alpha, \beta, K) \leq \alpha\beta$.

Our bound allows us to explore a larger range of (α, β) pairs that in turn lead to better lower bounds on $R^\star$. Suppose that for a coded caching system with N files and K users, we first apply the cutset bound with certain α_1 and β_1 such that $\alpha_1\beta_1 < N$. This would result in the inequality

$$\alpha_1 R^\star + \beta_1 M \geq \alpha_1\beta_1.$$

However, our approach can do strictly better. To see this note that $\alpha_1\beta_1 < N$ implies that $N_{sat}(\alpha_1, \beta_1, K) < N$. Now, using Corollary 2 we can instead attempt to lower bound $2\alpha_1 R^\star + 2\beta_1 M$ and obtain the following inequality.

$$
\begin{aligned}
&2\alpha_1 R^\star + 2\beta_1 M \\
&\quad\geq \min\left(4\alpha_1\beta_1,\ 2\alpha_1\beta_1 + N - N_{sat}(\alpha_1,\beta_1,K)\right) \\
&\implies \alpha_1 R^\star + \beta_1 M \\
&\quad\geq \min\left(2\alpha_1\beta_1,\ \alpha_1\beta_1 + (N - N_{sat}(\alpha_1,\beta_1,K))/2\right),
\end{aligned}
$$

which is strictly better than the cutset bound since $N - N_{sat}(\alpha_1, \beta_1, K) > 0$.

Example 8: Consider a system containing a server with four files and three users, $N = 4$ and $K = 3$. The cutset bounds corresponding to the given system are

$$
\begin{aligned}
4R^\star + M &\geq 4, \\
2R^\star + 2M &\geq 4, \text{ and} \\
R^\star + 3M &\geq 3.
\end{aligned}
$$

A simple calculation shows that if $M = 1$, the above inequalities, yield the lower bound $R^\star \geq 1$.

Now, consider the second bound, $2R^\star + 2M \geq 4$ and instead attempt to obtain a lower bound on $4R^\star + 4M$. In this case it

can be verified that $N_{sat}(2,2,3) = 3 < N$. Using Corollary 2, this results in the lower bound $L^* \geq \min(4 \times 3, 2 \times 4 + 4 - N_{sat}(2,2,3)) = 9$. Thus we can conclude $R^\star + M \geq 2.25$ which is better than the cutset bound $R^\star + M \geq 2$. Moreover, this inequality also yields a better lower bound $R^\star \geq 1.25$.

B. Comparison With Lower Bound of [10]

Sengupta *et al.* [10] use Han's inequality [35, Th. 17.6.1] to establish the following lower bounds on the coded caching problem.

$$\alpha R^\star(M) + \beta M \geq N - \frac{\mu}{\mu+\beta}[N-\alpha\beta]^+ - [N-\alpha K]^+, \tag{17}$$

where $\mu = \min(\lceil \frac{N-\alpha\beta}{\alpha} \rceil, K-\beta)$, $\beta \in \{1,\ldots,K\}$ and $\alpha \in \{1,\ldots,\lceil \frac{N}{\beta} \rceil\}$. This bound also provides more flexibility in the choice of α as compared to the cutset bound.

An analytical comparison between our bound and the bound in inequality (17) is hard, especially since a priori in all these bounds, for a given M, it is unclear which particular (α,β) pair gives the best lower bound. Thus, in the discussion below we attempt to analytically compare the bounds for given (α,β). We also present a numerical comparison in Section VI-E. The following conclusions can be drawn.

(a) Our bound is superior, when $1/\alpha + 1/\beta \leq 0.4$, i.e., when the values of α and β are large enough. Note that the best lower bounds on $R^\star(M)$ for systems with N and K reasonably large are obtained for higher values of α and β. Thus, for most parameter ranges our bounds are better.
(b) The bound in [10] is better when $\alpha = 1$ and $N \leq K$. This in turn means that their corresponding lower bound for small values of M is better than ours.
(c) We can demonstrate that our proposed lower bound is within a factor of four of the achievable rate, whereas [10] only demonstrates a multiplicative gap of eight.

In the remainder of this discussion we assume that $\alpha \geq 2$ and show these claims. Let L^* denote the value of our lower bound and let L_H denote the lower bound of [10].

Case 1 ($\alpha\beta > N$): Note that $\alpha \leq \lceil N/\beta \rceil$ in inequality (17). Furthermore, $\alpha \geq 2$ implies that $N \geq \beta$. Thus, we can conclude that $\alpha\beta \leq \lceil N/\beta \rceil \beta \leq 2N$. Now, we use Corollary 2 to compare the bounds. Specifically, set $\alpha_l = \lceil \alpha/2 \rceil$, $\beta_l = \lfloor \beta/2 \rfloor$, $\alpha_r = \lfloor \alpha/2 \rfloor$ and $\beta_r = \lceil \beta/2 \rceil$. This implies that

$$\max(\alpha_l\beta_l, \alpha_r\beta_r) \leq \frac{\alpha\beta}{2} \leq N.$$

Thus, we obtain $L^* = \min(\alpha\beta,\ \alpha_l\beta_l + \alpha_r\beta_r + N - N_0)$. Note that

$$\begin{aligned} N_0 &= \max\left(N_{sat}(\alpha_l,\beta_l,K),\ N_{sat}(\alpha_r,\beta_r,K)\right), \\ &\leq \max(\alpha_l\beta_l, \alpha_r\beta_r) \leq N, \end{aligned}$$

using the arguments made above. Thus,

$$\begin{aligned} L^* &= \min\{\alpha\beta,\ \alpha_l\beta_l + \alpha_r\beta_r + N - N_0\} \\ &\geq \min\{\alpha\beta,\ \alpha_l\beta_l + \alpha_r\beta_r + N - \max(\alpha_l\beta_l,\ \alpha_r\beta_r)\} \\ &= \min\{\alpha\beta, \min(\alpha_l\beta_l,\ \alpha_r\beta_r) + N\} \\ &> N. \end{aligned}$$

On the other hand note that L_H is at most N. Thus, our bound is strictly better.

Case 2(a) ($\alpha\beta \leq \alpha K \leq N$): As $N \geq \alpha\beta \geq N_{sat}(\alpha,\beta,K)$ we use (15) to obtain

$$L^* = \min\left(\alpha \min(K, 2\beta),\ \alpha\beta + (N-N_0)/2\right).$$

The corresponding bound L_H is obtained by setting $\mu = K - \beta$.

$$\begin{aligned} L_H &= \alpha K - (1-\beta/K)(N-\alpha\beta) \\ &= \alpha\beta(1+1/x-x) - (1-x)N, \ (\text{where } 0 \leq x = \beta/K \leq 1) \\ &\leq \alpha\beta(2-x), \ (\text{since, } N \geq \alpha K = \alpha\beta/x). \end{aligned}$$

Thus, we conclude that $L_H \leq \min(\alpha K, \alpha\beta(2-x)) \leq \alpha \min(K, 2\beta)$. As a result, we only need to examine whether $\alpha\beta + (N-N_0)/2 \geq L_H$. Now, using the fact that $N_0 \leq (2\alpha\beta + \alpha + \beta)/3$, we have that $L^* \geq L_H$ when

$$\begin{aligned} &2\alpha\beta/3 + N/2 - (\alpha+\beta)/6 \geq \alpha\beta(1+1/x-x) - (1-x)N \\ &\implies (3/2-x)N - (1/x + 1/3 - x)\alpha\beta - (\alpha+\beta)/6 \geq 0. \end{aligned} \tag{18}$$

As $N \geq \alpha K = \alpha\beta/x$, inequality (18) certainly holds if

$$(1/2x + x - 4/3)\alpha\beta - (\alpha+\beta)/6 \geq 0.$$

It can be verified that $1/2x + x - 4/3 \geq \sqrt{2} - 4/3 \geq 1/15$ for $0 \leq x \leq 1$, so that the above inequality will definitely hold if $0.4 \geq 1/\alpha + 1/\beta$ which is the case for $\alpha, \beta \geq 5$.

Case 2(b) ($\alpha\beta \leq N < \alpha K$): In this case $\mu = \lceil N/\alpha - \beta \rceil$, so that

$$\begin{aligned} L_H &\leq N - (1-\alpha\beta/N)(N-\alpha\beta) \\ &= \alpha\beta(2-x') \ (\text{where } 0 \leq x' = \alpha\beta/N \leq 1) \end{aligned}$$

As in the previous case, we conclude that $L^* \geq L_H$ if

$$2\alpha\beta/3 + N/2 - (\alpha+\beta)/6 \geq \alpha\beta(2-x').$$

Upon analysis similar to the previous case, we can conclude that our bound is better when $0.4 \geq 1/\alpha + 1/\beta$.

C. Comparison With Lower Bound of [11]

The work of [11] is closest in spirit to our proposed lower bound. In particular, we show that their lower bound corresponds to specific problem instance as defined in our work. We note however that the work of [11] does not analyze the multiplicative gaps between the achievable rates and lower bounds. The lower bounds in [11] can be rewritten as

$$\begin{aligned} 2mR^\star + 2tmM &\geq L_0, \quad \text{for } t \leq N,\ K \geq 2 \\ 2tmR^\star + 2mM &\geq L_0, \quad \text{for } t \leq N,\ K \geq 2t, \end{aligned} \tag{19}$$

where $L_0 = \min\{4tm^2,\ 2tm^2 + N - \tilde{N}_0\}$, $\tilde{N}_0 = t(m^2-m+1)$, $m = n - \gamma$ and $n = \lceil (t + \sqrt{t^2 + 12t(N-t)})/6t \rceil$. Also, $\gamma = \max(0,\ \lceil n - K/2t \rceil)$ and $\gamma = \max(0,\ \lceil n - K/2 \rceil)$ in the first and second lower bounds respectively. We present these bounds using our notation so that (α,β) is equal to $(2m, 2tm)$ and $(2tm, 2m)$ in the first and second lower bounds in (19) respectively. Note however, that in the above bound the only free parameter is t, i.e., m itself is dependent on t. It is easy

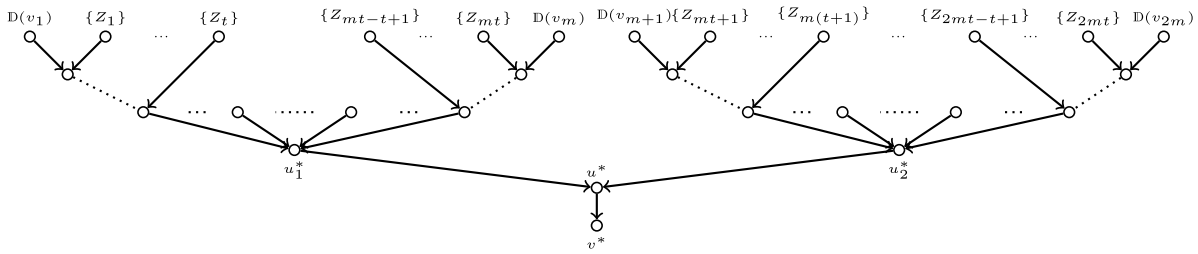

Fig. 11. Problem instance associated with the lower bounds in [11].

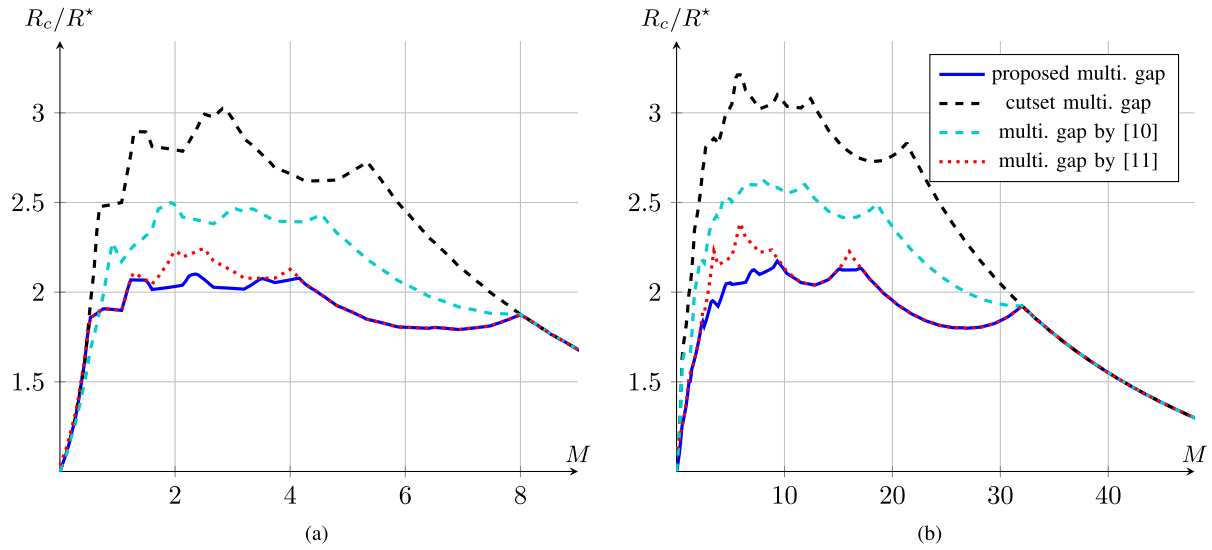

Fig. 12. The plot demonstrates the multiplicative gap between the achievable rate, $R_c(M)$, in [9] and lower bounds $R^\star(M)$ using different lower bounding techniques. For case II our lower bound results in the least multiplicative gap. In case I, where $N \leq K$, the multiplicative gap obtained by our proposed lower bound is lower than the others for $M \geq 1$. In the range $0 \leq M \leq 1$, [10] provides a slightly better result. (a) Case I: $N = 16$, $K = 30$. (b) Case II: $N = 64$, $K = 50$.

to see that $\beta \leq K$ therefore, unlike our method, this method cannot be used to obtain lower bounds when $\beta > K$.

The lower bound L_0 in eq. (19) above is reminiscent of our lower bound if the term $\tilde{N}_0$ is interpreted as a bound on the saturation number. In fact, for the specific setting of $(\alpha, \beta) = (m, mt)$, we can create a problem instance as described below, that is a saturated instance with exactly $t(m^2 - m + 1)$ files, so that we can infer that $N_{sat}(m, tm, K) \leq t(m^2 - m + 1)$. It turns out that this upper bound on the saturation number may be slightly stronger than the one we derived in Lemma 3 for general α and β when t and m are small. The associated problem instance of the first lower bound in (19) is depicted in Fig. 11. The corresponding instance for the second lower bound in (19) can be derived in a similar manner. In this figure, delivery phase signals $\mathbb{D}(v_1), \ldots, \mathbb{D}(v_{2m})$ are same as the delivery phase signals defined in [11]. For this tree, it can be verified that the instance can be saturated with $t(m^2-m+1)$ files, so that $N_{sat}(m, tm, K) \leq t(m^2 - m + 1)$.

However, an application of Algorithm 3 will result in even better upper bound on the saturation number as shown in the example below. In particular, Algorithm 3 will generate a different tree when trying to upper bound the saturation number.

Example 9: We consider a system with $N = 64$ files and $K = 8$ users and set $t = 2$ in eq. (19) so that $m = 4$ and $\tilde{N}_0 = 26$. Algorithm 4 for such a setting returns $N_{sat}(4, 8, 8) = 22$ which is smaller than $\tilde{N}_0$. On the other hand, it can be noted that in Fig. 11, node u_1^* is such that it has $m = 4$ incoming edges which makes the corresponding lower bound looser (*cf.* Claim 1).

D. Comparison With Results in [36]

Reference [36] presents lower bounds for the specific case of $N = K = 3$. The inequalities are generated via a computational technique that works with the entropic region of the associated random variables. Some of the bounds presented in [36] can be obtained via our approach as well. However, the specific inequalities $3R^\star + 6M \geq 8$, $18R^\star + 12M \geq 29$ and $6R^\star + 3M \geq 8$ cannot be obtained using our approach and strictly improves our region. Note however, that it is not clear whether these inequalities can be obtained in a computationally tractable manner for the case of large N and K.

E. Numerical Comparison of the Various Bounds

We conclude this section, by providing numerical results for two cases: (i) $N = 16, K = 30$ and (ii) $N = 64$, $K = 50$. In Fig. 12 the ratio $R_c(M)/R^\star(M)$ is plotted by lower bounding $R^\star(M)$ by different methods. In case I (see Fig. 12) we have $N = 16$ and $K = 30$. Our bound has the minimum

multiplicative gap except in the small range $0 \leq M \leq 1$. Specifically, as discussed previously, the bound in [10] is better than ours when $K \geq N$ and $\alpha = 1$ and $0 \leq M \leq 1$. In case II, where $N > K$ our bound has minimum multiplicative gap for all range of M.

VII. Conclusions and Future Work

In this work we considered a coded caching system with N files, K users each with a normalized cache of size M. We demonstrated an improved lower bound on the coded caching rate $R^\star(M)$. Our approach proceeds by establishing an equivalence between a sequence of information inequalities and a combinatorial labeling problem on a directed tree. Specifically, for given positive integers α and β, we generate an inequality of the form $\alpha R^\star + \beta M \geq L$. We showed that the best L that can be obtained using our approach is closely tied to how efficiently a given number of files can be used by our proposed algorithm. Formalizing this notion, we studied certain structural properties of our algorithm that allow us to quantify the improvements that our approach affords. In particular, we show a multiplicative gap of four between our lower bound and the achievable rate. An interesting feature of our algorithm is that it is applicable for general value of N, K and M and is strictly better than all prior approaches for most parameter ranges.

There are still gaps between the currently known lower bounds and the achievable rate and an immediate open question is whether this gap can be reduced or closed. It would also be of interest to better understand coded caching rates for more general network topologies.

Appendix

Lemma 5: Algorithm 1 always provides a valid lower bound on $\alpha R^\star + \beta M$ where $\alpha = \sum_{i=1}^{\ell} |\mathbb{D}(v_i)|$ and $\beta = \sum_{i=1}^{\ell} |\mathbb{Z}(v_i)|$.

Proof: Consider any internal node $v \in \mathcal{T}$. We have

$$\begin{aligned}
&\sum_{u \in in(v)} H(\mathbb{Z}(u) \cup \mathbb{D}(u) | \mathbb{W}(u) \cup W_{new}(u)), \\
&\overset{(a)}{\geq} \sum_{u \in in(v)} H(\mathbb{Z}(u) \cup \mathbb{D}(u) | \mathbb{W}(v)), \\
&\overset{(b)}{\geq} H(\mathbb{Z}(v) \cup \mathbb{D}(v) | \mathbb{W}(v)), \\
&\overset{(c)}{=} I(W_{new}(v); \mathbb{Z}(v) \cup \mathbb{D}(v) | \mathbb{W}(v)) \\
&\quad + H(\mathbb{Z}(v) \cup \mathbb{D}(v) | \mathbb{W}(v) \cup W_{new}(v)),
\end{aligned}$$

where inequality in (a) holds since $\mathbb{W}(u) \cup W_{new}(u) \subseteq \mathbb{W}(v)$ and conditioning reduces entropy, (b) holds since $\cup_{u \in in(v)} \mathbb{Z}(u) = \mathbb{Z}(v)$ and $\cup_{u \in in(v)} \mathbb{D}(u) = \mathbb{D}(v)$ and (c) holds by the definition of mutual information. Let V_{int} denote the set of internal nodes in $\mathcal{T}$. Let v^* denote the root and (u^*, v^*) denote its incoming edge. Then,

$$\begin{aligned}
&\sum_{v \in V_{int}} \sum_{u \in in(v)} H(\mathbb{Z}(u) \cup \mathbb{D}(u) | \mathbb{W}(u) \cup W_{new}(u)) \\
&\geq \sum_{v \in V_{int}} y_{(v, out(v))} + \sum_{v \in V_{int}} H(\mathbb{Z}(v) \cup \mathbb{D}(v) | \mathbb{W}(v) \cup W_{new}(v)),
\end{aligned}$$

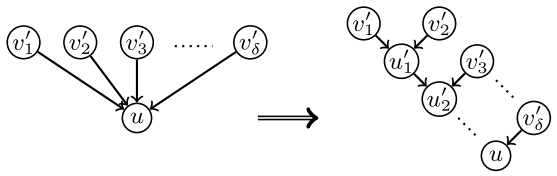

Fig. 13. Tree modification example.

where we have ignored the infinitesimal terms introduced due to Fano's inequality (for convenience of presentation). Note that the RHS of the inequality above contains terms of the form $H(\mathbb{Z}(v) \cup \mathbb{D}(v) | \mathbb{W}(v) \cup W_{new}(v))$ for all nodes $v \in V_{int}$ (including u^*). On the other hand the LHS contains terms of a similar form for all nodes including the leaf nodes but excluding the node u^*. Canceling the common terms, we obtain,

$$\begin{aligned}
&\sum_{i=1}^{\ell} H(\mathbb{Z}(v_i) \cup \mathbb{D}(v_i) | W_{new}(v_i)) \\
&\geq \left(\sum_{v \in V_i} y_{(v, out(v))} \right) + H(Z \cup \mathbb{D}(u^*) | \mathbb{W}(u^*), W_{new}(u^*)),
\end{aligned}$$

since $\mathbb{W}(v_i) = \phi$ for $i = 1, \ldots, \ell$. We can therefore conclude that

$$\sum_{i=1}^{\ell} H(\mathbb{Z}(v_i), \mathbb{D}(v_i)) \geq \sum_{v \in V} y_{(v, out(v))} \quad (20)$$

$$\implies \sum_{i=1}^{\ell} H(\mathbb{Z}(v_i)) + \sum_{i=1}^{\ell} H(\mathbb{D}(v_i)) \geq \sum_{v \in V} y_{(v, out(v))} \quad (21)$$

Noting that $M \geq H(\mathbb{Z}(v_i))$ and $R^\star \geq H(\mathbb{D}(v_i))$ we have the required result. ∎

A. Proof of Claim 1

Proof: We iteratively modify the problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ to arrive at an instance where every node has in-degree at most two. Towards this end, we first identify a node u with in-degree $\delta \geq 3$ such that no other node is topologically higher than it (such a node may not be unique).

We modify the instance P by replacing u with a directed in-tree where each node has in-degree exactly two. Specifically, arbitrarily number the nodes in $in(u)$ from $v'_1, \ldots, v'_\delta$. We replace the node u with a directed in-tree $\mathcal{T}_u$ with leaves $v'_1, \ldots, v'_\delta$ and root u. $\mathcal{T}_u$ has $\delta - 2$ internal nodes numbered $u'_1, \ldots, u'_{\delta-2}$ such that $in(u'_i) = \{u'_{i-1}, v'_{i+1}\}$ where $u'_0 = v'_1$ (see Fig. 13). Let us denote the new instance by $P_o = P_o(\mathcal{T}_o, \alpha, \beta, L_o, N, K)$. We claim that $L_o \geq L$. To see this, suppose that $W^* \in W^P_{new}(u)$. We show that $W^* \in \cup_{u' \in \mathcal{T}_u} W^{P_o}_{new}(u')$. This ensures that $L_o \geq L$. To see this we note that

$$\begin{aligned}
\mathbb{Z}^P(u) &= \mathbb{Z}^{P_o}(u) \\
\mathbb{D}^P(u) &= \mathbb{D}^{P_o}(u), \text{ and thus,} \\
\Delta^P(u, u) &= \Delta^{P_o}(u, u).
\end{aligned}$$

Thus, if $W^* \in W^P_{new}(u)$, there exists an internal node $u'_i \in \mathcal{T}_u$ with the smallest index $i \in \{1, \ldots, \delta - 2\}$ such that $W^* \in \Delta^{P_o}(u'_i, u'_i)$. Note that if $i > 1$, we have $W^* \in W^{P_o}_{new}(u'_i)$ since $W^* \notin \Delta^{P_o}(u'_{i-1}, u'_{i-1})$ which in turn implies that $W^* \notin \mathbb{W}^{P_o}(u'_i)$. On the other hand if $i = 1$, then a similar argument holds since it is easy to see that $W^* \notin \mathbb{W}^{P_o}(u'_1)$.

Note that the modification in the instance P can only affect nodes that are downstream of u. Now consider u' such that $u \in in(u')$. It is evident that $\mathbb{Z}^{P_o}(u') = \mathbb{Z}^P(u')$ and $\mathbb{D}^{P_o}(u') = \mathbb{D}^P(u')$. Moreover $\mathbb{W}^{P_o}(u') = \cup_{v \in in(u')} \mathbb{W}^{P_o}(v) \cup W^{P_o}_{new}(v)$. Now for $v \neq u$, $\mathbb{W}^{P_o}(v) = \mathbb{W}^P(v)$ and $W^{P_o}_{new}(v) = W^P_{new}(v)$ as there are no changes in the corresponding subtrees. Moreover, as $\Delta^P(u, u) = \Delta^{P_o}(u, u)$, we have that $\mathbb{W}^{P_o}(u) \cup W^{P_o}_{new}(u) = \mathbb{W}^P(u) \cup W^P_{new}(u)$. This implies that $\mathbb{W}^{P_o}(u') = \mathbb{W}^P(u')$. Thus, we can conclude that $W^{P_o}_{new}(u') = W^P_{new}(u')$. Applying an inductive argument we can conclude that the $W^{P_o}_{new}(u') = W^P_{new}(u')$ for all u' such that $u \succ u'$.

The above process can iteratively be applied to every node in the instance that is of degree at least three. Thus, we have the required result. ∎

B. Proof of Claim 3

Proof: We identify the set $\mathcal{U}$ as the set of all nodes in $\mathcal{T}$ such that the specified condition in the claim holds. Let $\mathcal{U}^* \subset \mathcal{U}$ denote the set of nodes that are highest in the topological ordering. We modify the instance in a way such that a node $u^* \in \mathcal{U}^*$ can be removed from $\mathcal{U}$, i.e., the specified condition no longer holds for it. Moreover, our modification procedure is such that a node $u \succ u^*$ cannot enter $\mathcal{U}$ at the end of the procedure.

We now discuss the modification procedure. In the discussion below, for a given node u, we can consider the instance obtained with tree $\mathcal{T}_u$. We let β_u denote the number of cache nodes in this instance. Note that for u^*, the condition $\hat{\beta}^* < \min(\beta^*, K)$ holds. This implies that there is a set of cache leaves in $\mathcal{T}_{u^*}$ denoted $\{v_{i_1}, \ldots, v_{i_m}\}$ such that $\mathbb{Z}(v_{i_1}) = \cdots = \mathbb{Z}(v_{i_m}) = \{Z_j\}$. Let $\Lambda = \{u \in \mathcal{T}_{u^*} : (v_{i_a}, v_{i_b}) \text{ meet at } u, \text{ for all distinct } v_{i_a}, v_{i_b} \in \{v_{i_1}, \ldots, v_{i_m}\}\}$. We identify $u_0 \in \Lambda$ such that no element of Λ is topologically higher than u_0 (note that u_0 may not be unique) and let $v^*_{i_a}$ and $v^*_{i_b}$ be one pair of the corresponding nodes in $\{v_{i_1}, \ldots, v_{i_m}\}$ that meet at u_0. W.l.o.g we assume that $v^*_{i_b} \in \mathcal{T}_{u_0(r)}$ and $v^*_{i_a} \in \mathcal{T}_{u_0(l)}$.

We claim that $u_0 = u^*$. Assume that this is not the case. Since $u_0 \in \mathcal{T}_{u^*}$ we have $u_0 \succeq u^*$. Using this and the fact that $u_0 \notin \mathcal{U}$ we have $|\cup_{v \in \mathcal{C}_{u_0}} \mathbb{Z}(v)| = \min(|\mathcal{C}_{u_0}|, K)$. Now, from $v^*_{i_a}, v^*_{i_b} \in \mathcal{C}_{u_0}$ and that $\mathbb{Z}(v^*_{i_a}) = \mathbb{Z}(v^*_{i_b})$ we conclude that $\min(|\mathcal{C}_{u_0}|, K) = K$. Moreover, as $\cup_{u \in \mathcal{T}_{u_0}} \mathbb{Z}(u) \subseteq \cup_{u \in \mathcal{T}_{u^*}} \mathbb{Z}(u)$ we have $\hat{\beta} = K$ which contradicts $\hat{\beta} < \min(\beta, K)$. Therefore $u_0 = u^*$.

We construct instance P' (with lower bound L') as follows. Choose a member of $\{Z_1, \ldots, Z_K\} \setminus \{\mathbb{Z}(v') : v' \in \mathcal{C}_{u^*}\}$ and denote it by Z_k. We set $\mathbb{Z}^{P'}(v^*_{i_b}) = \{Z_k\}$. Also, for any $u \in \mathcal{D}_{u_0(r)}$ and $\mathbb{D}^P(u) = X_{d_1,\ldots,d_K}$ we set $\mathbb{D}^{P'}(u) = X_{d'_1,\ldots,d'_K}$ such that $d'_j = d_k$ and $d'_k = d_j$ and $d'_i = d_i$ for $i \notin \{j, k\}$, i.e., we interchange the j-th and k-th labels and keep the other labels the same. With this modification, it can be seen that $\hat{\beta}^* = \min(\beta^*, K)$.

For nodes $u \succ u^*$, the change we applied to cache nodes in $\mathcal{C}_{u^*}$ to get P' is such that $\hat{\beta}_u$ continues to equal $\min(\beta_u, K)$ since Z_k is chosen from $\{Z_1, \ldots, Z_K\} \setminus \{\mathbb{Z}(v') : v' \in \mathcal{C}_{u^*}\}$

We now show that $L' \geq L$. In particular, for $u \in \mathcal{T}_{u_0(l)}$, we have $W^{P'}_{new}(u) = W^P_{new}(u)$, as there are no changes in the corresponding labels. Also we claim that $W^{P'}_{new}(u) = W^P_{new}(u)$ for $u \in \mathcal{T}_{u_0(r)}$. To see this, note that for $v \in \mathcal{D}_{u_0(r)}$ and $v' \in \mathcal{C}_{u_0(r)}$ we have $\Delta^{P'}(v', v) = \Delta^P(v', v)$ if $\mathbb{Z}(v') \notin \{Z_j, Z_k\}$. If $\mathbb{Z}^{P'}(v') = \{Z_k\}$ and $\mathbb{D}^{P'}(v) = X_{d'_1,\ldots,d'_K}$ then,

$$\begin{aligned}\Delta^{P'}(v', v) &= Rec(\{Z_k\}, \{X_{d'_1,\ldots,d'_K}\}) \\ &= \{W_{d'_k}\} = \{W_{d_j}\} \\ &= Rec(\{Z_j\}, \{X_{d_1,\ldots,d_K}\}) \\ &= \Delta^P(v', v).\end{aligned}$$

Furthermore, note that there does not exist any $v' \in \mathcal{C}_{u_0(r)}$ such that $\mathbb{Z}(v') = \{Z_j\}$ since we picked u_0 such that no element of Λ is topologically higher than u_0. From eq. (5) and (6), it is not hard to see that this in turn implies that $W^{P'}_{new}(u) = W^P_{new}(u)$ for $u \in \mathcal{T}_{u_0(r)}$.

It follows therefore that $\mathbb{W}^{P'}(u_0) = \mathbb{W}^P(u_0)$ (from eq. (6)). Let us now consider the other nodes. As the changes are applied only to $\mathcal{T}_{u_0(r)}$ so $label(u)$ changes only for nodes u such that $u_0 \succ u$. Consider the subset of internal nodes $U = \{u_0, u_1, \ldots, u_t\}$ such that (u_i, u_{i+1}) is an edge, i.e., the set of internal nodes including u_0 and all nodes downstream of u_0 such that u_t is the last internal node. W.l.o.g we assume that $u_{i-1} \in \mathcal{T}_{u_i(l)}$ for $i \geq 1$. We now show that $\cup_{u \in U} W^P_{new}(u) \subseteq \cup_{u \in U} W^{P'}_{new}(u)$. Towards this end we have the following observations for $u \in U$.

$$\begin{aligned}&\mathbb{Z}^{P'}(u) = \mathbb{Z}^P(u) \cup \{Z_k\} \text{ (from the construction of } P') \\ &\Delta^{P'}(u, u) = \cup_{v \in \mathcal{D}_u} \Delta^{P'}(u, v).\end{aligned}$$

Now, for $v \notin \mathcal{D}_{u_0(r)}$ we have $\mathbb{D}^{P'}(v) = \mathbb{D}^P(v)$ so that

$$\begin{aligned}\Delta^{P'}(u, v) &= Rec(\mathbb{Z}^{P'}(u), \mathbb{D}^{P'}(v)) \\ &= Rec(\mathbb{Z}^{P'}(u), \mathbb{D}^P(v)) \\ &\supseteq \Delta^P(u, v) (\text{ since } \mathbb{Z}^{P'}(u) \supseteq \mathbb{Z}^P(u)).\end{aligned}$$

Conversely, for $v \in \mathcal{D}_{u_0(r)}$ we have

$$Rec\left(\{Z_j, Z_k\}, \mathbb{D}^{P'}(v)\right) = Rec\left(\{Z_j, Z_k\}, \mathbb{D}^P(v)\right),$$

and

$$Rec\left(\{Z_i\}, \mathbb{D}^{P'}(v)\right) = Rec\left(\{Z_i\}, \mathbb{D}^P(v)\right) (\text{for } Z_i \notin \{Z_j, Z_k\}).$$

Now, note that $\{Z_k, Z_j\} \subseteq \mathbb{Z}^{P'}(u)$ so that

$$\begin{aligned}\Delta^{P'}(u, v) &= Rec\left(\mathbb{Z}^{P'}(u), \mathbb{D}^{P'}(v)\right) \\ &= Rec\left(\mathbb{Z}^{P'}(u), \mathbb{D}^P(v)\right), \\ &\supseteq Rec\left(\mathbb{Z}^P(u), \mathbb{D}^P(v)\right) = \Delta^P(u, v),\end{aligned}$$

since $\mathbb{Z}^{P'}(u) \supseteq \mathbb{Z}^P(u)$. We can therefore conclude that

$$\Delta^P(u, u) = \cup_{v \in \mathcal{D}_u} \Delta^P(u, v) \subseteq \cup_{v \in \mathcal{D}_u} \Delta^{P'}(u, v) = \Delta^{P'}(u, u).$$

Now we consider a $W^* \in W^P_{new}(u_i)$ so that $W^* \in \Delta^P(u_i, u_i)$ which by above condition means that $W^* \in \Delta^{P'}(u_i, u_i)$. Thus either $W^* \in W^{P'}_{new}(u_i)$ or $W^* \in \mathbb{W}^{P'}(u_i)$. In the latter case there exists a node $u_{i'}$ where $0 \leq i' < i$ such that $W^* \in W^{P'}_{new}(u_{i'})$ since $W^* \notin \mathbb{W}(u_0)$ and we have shown that $\mathbb{W}^{P'}(u_0) = \mathbb{W}^P(u_0)$. Thus, we observe that

$$\begin{aligned} L' &= |\cup_{u\in U} W^{P'}_{new}(u)| + \sum_{u\in\mathcal{T}', u\notin U} |W^{P'}_{new}(u)|, \\ &\geq |\cup_{u\in U} W^{P}_{new}(u)| + \sum_{u\in\mathcal{T}, u\notin U} |W^{P}_{new}(u)|, \\ &= L, \end{aligned}$$

where the second inequality holds since $\sum_{u\in\mathcal{T}', u\notin U} |W^{P'}_{new}(u)| = \sum_{u\in\mathcal{T}, u\notin U} |W^{P}_{new}(u)|$ and $|\cup_{u\in U} W^{P'}_{new}(u)| \geq |\cup_{u\in U} W^{P}_{new}(u)|$.

As discussed before, the modification procedure is such that at the end of the operation $u^* \notin \mathcal{U}$. Moreover nodes $u \succ u^*$ are not in $\mathcal{U}$ either. For each node $u \in \mathcal{U}$ let $d(u)$ denote the number of edges in the path connecting u to the root node. Our modification procedure is such that $d^* = \max_{u\in\mathcal{U}} d(u)$ is guaranteed to decrease over the course of the iterations. Indeed, if $|\mathcal{U}^*| = 1$, then at the end of the iteration d^* will definitely decrease. If $|\mathcal{U}^*| > 1$, then d^* will definitely decrease after the modification procedure is applied to all the nodes in $\mathcal{U}^*$. Thus, the sequence of iterations is guaranteed to terminate. This observation concludes the proof. ∎

C. Proof of Lemma 1

Proof: Given the conditions of the theorem, from Corollary 1 we can conclude that there exists an index $i^* \in \{1, \ldots, \alpha\}$ such that $\sum_{v'\in\mathcal{C}} \psi(v_{i^*}, v') < \min(\beta, K)$. We set i^* to be the smallest such index. Let $\Pi^1(v_{i^*}) = \{v' \in \mathcal{C} : \psi(v_{i^*}, v') = 1\}$ and $\Pi^0(v_{i^*}) = \{v' \in \mathcal{C} : \psi(v_{i^*}, v') = 0, \mathbb{Z}(v') \nsubseteq \cup_{v\in\Pi^1(v_{i^*})}\mathbb{Z}(v)\}$. Note that $\Pi^0(v_{i^*})$ is non-empty since $|\cup_{v'\in\mathcal{C}} \mathbb{Z}(v')| = \min(\beta, K)$ and $\sum_{v'\in\mathcal{C}} \psi(v_{i^*}, v') < \min(\beta, K)$.

Next, we determine the set of nodes where v_{i^*} and the nodes in $\Pi^0(v_{i^*})$ meet, i.e., we define $\Lambda^0(v_{i^*}) = \{u \in \mathcal{T} : \exists v' \in \Pi^0(v_{i^*})$ such that v_{i^*} and v' meet at $u.\}$. Note that there is a topological ordering on the nodes in $\Lambda^0(v_{i^*})$. Pick the node $u^* \in \Lambda^0(v_{i^*})$ such that no element of $\Lambda^0(v_{i^*})$ is topologically higher than u^* (u^* is in the path from v_{i^*} to the root node). Let the corresponding node in $\Pi^0(v_{i^*})$ be denoted by v_{j^*} where $j^* \in \{\alpha+1, \ldots, \alpha+\beta\}$. Note that v_{j^*} might not be unique.

Suppose that $\mathbb{Z}(v_{j^*}) = \{Z_k\}$ and that $\mathbb{D}(v_{i^*}) = X_{d_1,\ldots,d_K}$. We modify the instance P as follows. Set $d_k = N+1$ (i.e., the index of the $N+1$ file). Thus, the only change is in $\mathbb{D}(v_{i^*})$. Let us denote the new instance by $P' = P(\mathcal{T}', \alpha, \beta, L', N+1, K)$.

We now analyze the value of L'. W.l.o.g. we assume that $v_{i^*} \in \mathcal{T}'_{u^*(l)}$ and $v_{j^*} \in \mathcal{T}'_{u^*(r)}$. Note that $W^{P'}_{new}(u) = W^P_{new}(u)$ for $u \in \mathcal{T}'_{u^*(r)}$ as the subtree $\mathcal{T}'_{u^*(r)}$ is identical to $\mathcal{T}_{u^*(r)}$. We also have

$$W^{P'}_{new}(u) = W^P_{new}(u) \text{ for } u \in \mathcal{T}'_{u^*(l)}.$$

To see this suppose that this is not true. This implies that the file W_{N+1} is recovered at some node in $\mathcal{T}'_{u^*(l)}$, i.e., there exists $v' \in \mathcal{C}$ such that $v' \in \mathcal{T}'_{u^*(l)}$, $\mathbb{Z}(v') = \{Z_k\}$, and that v' and v_{i^*} meet at some $u \succ u^*$. From $v_{j^*} \in \Pi^0(v_{i^*})$ we can conclude that $\{Z_k\} \nsubseteq \cup_{v\in\Pi^1(v_{i^*})}$ and $v' \in \Pi^0(v_{i^*})$ (as $\mathbb{Z}(v') = \{Z_k\}$). However this is a contradiction, since this implies the existence of node u that is topologically higher than u^* in the set $\Lambda^0(v_{i^*})$. It follows from eq. (6) that $\mathbb{W}^{P'}(u^*) = \mathbb{W}^P(u^*)$.

Next, we claim that $W^{P'}_{new}(u^*) = W^P_{new}(u^*) \cup \{W_{N+1}\}$. To see this consider the following series of arguments. Let the singleton subset $\Delta^P(v_{i^*}, v_{j^*}) = \{W^*\}$. Note that $\psi^P(v_{i^*}, v_{j^*}) = 0$. This implies that there exist $v \in \mathcal{D}_{u^*}$ and $v' \in \mathcal{C}_{u^*}$ such that v and v' meet above u^* and recover the file W^* where $(v, v') \neq (v_{i^*}, v_{j^*})$. Thus, as $\mathbb{Z}^{P'}(u^*) = \mathbb{Z}^P(u^*)$, we can conclude that

$$\begin{aligned} \Delta^{P'}(u^*, u^*) &= Rec(\mathbb{Z}^{P'}(u^*), \mathbb{D}^{P'}(u^*)) \\ &= Rec(\mathbb{Z}^{P}(u^*), \mathbb{D}^{P'}(u^*)) \\ &= \Delta^P(u^*, u^*) \cup \{W_{N+1}\}. \end{aligned}$$

Furthermore, we have

$$\begin{aligned} W^{P'}_{new}(u^*) &= \Delta^{P'}(u^*, u^*) \setminus \mathbb{W}^{P'}(u^*) \\ &= \Delta^P(u^*, u^*) \cup \{W_{N+1}\} \setminus \mathbb{W}^P(u^*) \\ &= W^P_{new}(u^*) \cup \{W_{N+1}\}, \text{ (since } W_{N+1} \notin \mathbb{W}^P(u^*)). \end{aligned}$$

For u such that $u^* \succ u$ we inductively argue that $W^{P'}_{new}(u) = W^P_{new}(u)$. To see this suppose that $u^* = u_r$. It is evident that $\Delta^{P'}_{rl}(u) = \Delta^P_{rl}(u)$. Next, $\Delta^{P'}_{lr}(u) = \Delta^P_{lr}(u)$ since $Z_k \notin \mathbb{Z}(u_l) \setminus \mathbb{Z}(u_r)$. Thus,

$$\begin{aligned} &W^{P'}_{new}(u) \\ &= \Delta^{P'}_{rl}(u) \cup \Delta^{P'}_{lr}(u) \setminus \mathbb{W}^{P'}(u) \\ &= \Delta^{P}_{rl}(u) \cup \Delta^{P}_{lr}(u) \setminus \mathbb{W}^{P'}(u) \\ &= \Delta^{P}_{rl}(u) \cup \Delta^{P}_{lr}(u) \setminus \mathbb{W}^{P}(u) \cup \{W_{N+1}\} \\ &= \Delta^{P}_{rl}(u) \cup \Delta^{P}_{lr}(u) \setminus \mathbb{W}^{P}(u) \text{ (since } W_{N+1} \notin \Delta^P_{rl}(u) \cup \Delta^P_{lr}(u)) \\ &= W^P_{new}(u). \end{aligned}$$

Next, we note that $\mathbb{W}(u) = \mathbb{W}(u_r) \cup W_{new}(u_r) \cup \mathbb{W}(u_l) \cup W_{new}(u_l)$. It is evident that $\mathbb{W}^{P'}(u_l) = \mathbb{W}^P(u_l)$ and $W^{P'}_{new}(u_l) = W^P_{new}(u_l)$. Next, $\mathbb{W}^{P'}(u_r) = \mathbb{W}^{P'}(u^*) = \mathbb{W}^P(u^*)$ (from above) and $W^{P'}_{new}(u^*) = W^P_{new}(u^*) \cup \{W_{N+1}\}$, so that $\mathbb{W}^{P'}(u) = \mathbb{W}^P(u) \cup \{W_{N+1}\}$.

As the induction hypothesis we assume that for any node u downstream of u^*, we have $W^{P'}_{new}(u) = W^P_{new}(u)$ and $\mathbb{W}^{P'}(u) = \mathbb{W}^P(u) \cup \{W_{N+1}\}$. Consider a node u' such that $u'_r = u$. As before we have $\mathbb{W}^{P'}(u'_l) = \mathbb{W}^P(u'_l)$, $W^{P'}_{new}(u'_l) = W^P_{new}(u'_l)$. Moreover, we have $\mathbb{W}^{P'}(u'_r) = \mathbb{W}^P(u'_r) \cup \{W_{N+1}\}$ and $W^{P'}_{new}(u'_r) = W^P_{new}(u'_r)$, by the induction hypothesis, so that $\mathbb{W}^{P'}(u') = \mathbb{W}^P(u') \cup \{W_{N+1}\}$.

Next, we argue similarly as above that $\Delta^{P'}_{rl}(u') = \Delta^P_{rl}(u')$ and $\Delta^{P'}_{lr}(u') = \Delta^P_{lr}(u')$ and the sequence of equations above can be used to conclude to that $W^{P'}_{new}(u') = W^P_{new}(u')$.

We conclude that $L' = L + 1$. ∎

D. Proof of Claim 5

Proof: W.l.o.g we assume that $|\Gamma_l| \geq |\Gamma_r|$ for all $u \in \mathcal{T}$. We identify the set $\mathcal{U}$ as the set of nodes in $\mathcal{T}$ such that

$\Gamma_r \nsubseteq \Gamma_l$. Let $\mathcal{U}^* \subset \mathcal{U}$ denote the set of nodes in $\mathcal{U}$ that are highest in the topological ordering.

Consider a node $u^* \in \mathcal{U}^*$. Note that since $|\Gamma_l| \geq |\Gamma_r|$, there exists an injective mapping $\phi : \Gamma_r \setminus \Gamma_l \to \Gamma_l \setminus \Gamma_r$. Let $\mathbb{Z}(u_r^*) = \{Z_{i_1}, \ldots, Z_{i_m}\}$. We construct the instance P' as follows. For each $v \in \mathcal{D}_{u_r^*}$ suppose $\mathbb{D}(v) = \{X_{d_1,\ldots,d_K}\}$. For $j = 1, \ldots, m$, if $d_{i_j} \in \Gamma_r \setminus \Gamma_l$, we replace it by $\phi(d_{i_j})$; otherwise, we leave it unchanged. In other words, we modify the delivery phase signals so that the files that are recovered in $\mathcal{T}_{u^*(r)}$ are a subset of those recovered in $\mathcal{T}_{u^*(l)}$.

As our change amounts to a simple relabeling of the sources, for $u \in \mathcal{T}_{u^*(r)}$ we have $|W_{new}^{P'}(u)| = |W_{new}^{P}(u)|$. For any $u \succ u^*$ we have $\Gamma_r^P(u) \subseteq \Gamma_l^P(u)$. Similarly, we can show that $\Gamma_r^{P'}(u) \subseteq \Gamma_l^{P'}(u)$. We note that $\Gamma^{P'}$ and Γ^P only differ in files such as W_d where d is in the domain of $\phi(\cdot)$, i.e., if $W_d \in \Gamma^P$ then $W_{\phi(d)} \in \Gamma^{P'}$. If there exist a file $W_d \in \Gamma_r^P(u)$ with d in domain of $\phi(\cdot)$ then $W_{\phi(d)} \in \Gamma_r^{P'}(u)$ and from $\Gamma_r^P(u) \subseteq \Gamma_l^P(u)$ we have $W_{\phi(d)} \in \Gamma_l^{P'}(u)$. Thus, we have $\Gamma_r^{P'}(u) \subseteq \Gamma_l^{P'}(u)$. This indicates that after applying this change, the property $\Gamma_r \subseteq \Gamma_l$ still holds in P' for all nodes u that are upstream of u^*. Furthermore, the relabeling of the sources only affects $u \in \mathcal{T}'$ such that $u^* \succ u$. Note that $\mathbb{W}^{P'}(u^*) \subset \mathbb{W}^P(u^*)$ (the inclusion is strict since at least one source in $\Gamma_r \setminus \Gamma_l$ is mapped to $\Gamma_l \setminus \Gamma_r$) since we have $\Gamma_r^{P'} \subseteq \Gamma_l^{P'}$ and $\Gamma_l^{P'} = \Gamma_l^P$.

Now, we note that

$$\Delta_{rl}^{P'}(u^*) = \Delta_{rl}^{P}(u^*), \text{ and}$$
$$\Delta_{lr}^{P'}(u^*) = \Delta_{lr}^{P}(u^*),$$

where the first equality holds since $\mathbb{Z}^P(u_r^*) = \mathbb{Z}^{P'}(u_r^*)$, $\mathbb{Z}^P(u_l^*) = \mathbb{Z}^{P'}(u_l^*)$ and $\mathbb{D}^P(u_l^*) = \mathbb{D}^{P'}(u_l^*)$. The second equality holds since our modification to the delivery phase signals in $\mathcal{T}_{u^*(r)}$ does not affect files that are recovered from $\mathbb{Z}^P(u_l^*) \setminus \mathbb{Z}^P(u_r^*)$. It follows therefore that $|W_{new}^{P'}(u^*)| \geq |W_{new}^{P}(u^*)|$.

We make an inductive argument for nodes u that are downstream of u^*; w.l.o.g. we assume that $u^* \in \mathcal{T}_{u(r)}$. Specifically, our induction hypothesis is that for a node u that is downstream of u^*, we have $\mathbb{W}^{P'}(u) \subseteq \mathbb{W}^P(u)$, $\Delta_{rl}^{P'}(u) = \Delta_{rl}^{P}(u)$ and $\Delta_{lr}^{P'}(u) = \Delta_{lr}^{P}(u)$.

Now consider a node u' downstream of u such that $u'_r = u$. We have, $\mathbb{W}(u') = \mathbb{W}(u'_l) \cup W_{new}(u'_l) \cup \mathbb{W}(u) \cup W_{new}(u)$. Note that we can express $\mathbb{W}(u) \cup W_{new}(u) = \mathbb{W}(u) \cup \Delta_{rl}(u) \cup \Delta_{lr}(u)$. It is evident that $\mathbb{W}^{P'}(u'_l) = \mathbb{W}^P(u'_l)$ and $W_{new}^{P'}(u'_l) = W_{new}^{P}(u'_l)$. Moreover, by the induction hypothesis, $\mathbb{W}^{P'}(u) \subseteq \mathbb{W}^P(u)$ and $\Delta_{rl}^{P'}(u) \cup \Delta_{lr}^{P'}(u) = \Delta_{rl}^{P}(u) \cup \Delta_{lr}^{P}(u)$. Thus, the induction step is proved.

We have shown that after applying the changes for u^*, the condition $\Gamma_r \nsubseteq \Gamma_l$ will not hold for $u \succeq u^*$. For each node $u \in \mathcal{U}$ let $d(u)$ denote the number of edges in path connecting u to the root node. Our modification procedure is such that $d^* = \max_{u \in \mathcal{U}} d(u)$ is guaranteed to decrease over the course of the iterations. Indeed, if $|\mathcal{U}^*| = 1$, then at the end of the iteration d^* will definitely decrease. If $|\mathcal{U}^*| > 1$, then d^* will definitely decrease after the modification procedure is applied to all the nodes in $\mathcal{U}^*$. Thus, the sequence of iterations is guaranteed to terminate. This observation concludes the proof.

As we have shown, the modification procedure is such that at the end of the operation u^* is removed from $\mathcal{U}$. Therefore, each node in $\mathcal{T}$ will be involved in the modification procedure at most once. In Appendix E, we show that there are $2(\alpha+\beta)$ nodes in $\mathcal{T}$. Thus, the modification procedure requires at most $2(\alpha+\beta)$ iterations to terminate. At each iteration we only need to apply the mapping $\phi(\cdot)$ to the indices of the delivery nodes connected to u^*. The complexity of this step is at most $\alpha\beta$. Therefore, the complexity of the modification is at most $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$. ∎

Claim 9: When $\hat{\beta}_l = \min(\beta_l, K)$ and $\hat{\beta}_r = \min(\beta_r, K)$ we have $\min(\hat{\beta}_l, K - \hat{\beta}_r) = [\min(\beta_l, K - \beta_r)]^+$ and $\min(\hat{\beta}_r, K - \hat{\beta}_l) = [\min(\beta_r, K - \beta_l)]^+$.

Proof: First, we consider the case where $\beta_l + \beta_r \leq K$ so $\beta_l \leq K - \beta_r$ and $[\min(\beta_l, K - \beta_r)]^+ = \beta_l$. By assumption, $\beta_l + \beta_r \leq K$ implies $\hat{\beta}_l + \hat{\beta}_r \leq K$ thus $\min(\hat{\beta}_l, K - \hat{\beta}_r) = \hat{\beta}_l = \beta_l$. We now consider the $\beta_l + \beta_r \geq K$ case which in turns leads to $\hat{\beta}_l + \hat{\beta}_r \geq K$. Therefore,

$$\min(\hat{\beta}_l, K - \hat{\beta}_r) = K - \hat{\beta}_r = K - \min(K, \beta_r)$$
$$= \max(0, K - \beta_r) = [K - \beta_r]^+ = [\min(\beta_l, K - \beta_r)]^+.$$

The same argument will show that $\min(\hat{\beta}_r, K - \hat{\beta}_l) = [\min(\beta_r, K - \beta_l)]^+$. ∎

Claim 10: Consider the integers $\alpha, \alpha_l, \alpha_r, \beta, \beta_l, \beta_r, K$ so that $\alpha = \alpha_l + \alpha_r$ and $\beta = \beta_l + \beta_r$. Then

$$\alpha \min(\beta, K)$$
$$= \alpha_l \min(\beta_l, K) + \alpha_r \min(\beta_r, K)$$
$$+ \alpha_l[\min(\beta_r, K - \beta_l)]^+ + \alpha_r[\min(\beta_l, K - \beta_r)]^+.$$

Proof: First, we consider the case where $\beta \leq K$ thus $\beta_l \leq K - \beta_r$ and $\beta_r \leq K - \beta_l$. Then, the above relation reduces to $\alpha\beta = \alpha_l\beta_l + \alpha_r\beta_r + \alpha_l\beta_r + \alpha_r\beta_l$ which is true. For the case $\beta \geq K$, the relation reduces to $\alpha K = \alpha_l \left(\min(\beta_l, K) + [K - \beta_l]^+\right) + \alpha_r \left(\min(\beta_r, K) + [K - \beta_r]^+\right)$. However $\min(\beta_l, K) = K - [K - \beta_l]^+$ and $\min(\beta_r, K) = K - [K - \beta_r]^+$ and the result follows. ∎

E. Complexity of the Algorithms 1, 2, 3, and 4

In this part we discuss the time-complexity of the algorithms used in this paper. Before proceeding, we note that the directed in-tree corresponding to the problem instance $P(\mathcal{T}, \alpha, \beta, L, N, K)$ contains $\alpha + \beta$ leaves and a single root. The degree (total number of incoming and outgoing edges) of the leaves and the root is 1. Based on Claim 1 the intermediate nodes have a total degree of 3. Thus,

$$2|A| = \alpha + \beta + 1 + 3(|V| - \alpha - \beta - 1)$$
$$= 3|V| - 2\alpha - 2\beta - 2.$$

On the other hand, since the undirected version of $\mathcal{T}$ is a tree we have $|A| = |V| - 1$. Solving these two equations yields

$$|V| = 2(\alpha + \beta),$$
$$|A| = 2(\alpha + \beta) - 1.$$

1) Complexity of the Algorithm 1: The complexity of computing $\Delta(v_i, v_i)$ in second line of the algorithm is less than $\alpha\beta$. As there are $\alpha + \beta$ leaves thus the complexity of lines 1-5 of the algorithm is $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$. The while loop in the algorithm goes over all nodes except the leaves exactly once. Thus, the while loop is executed $\alpha + \beta$ times. At each phase of the while loop, computing $\Delta(u, u)$ has the largest running time among the other operation and its complexity is less than $\alpha\beta$. Therefore, the complexity of the while loop is also $\mathcal{O}(\alpha^2\beta+\alpha\beta^2)$. Thus, this algorithm has a time-complexity of $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$.

2) Complexity of the Algorithm 2: As there are $2(\alpha + \beta)$ nodes in $\mathcal{T}$ and $|W_{new}(u)| \leq \alpha\beta$, the complexity of the initialization part is $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$. In the remaining steps of the algorithm, the main complexity of the inner for loop is in finding the meeting point of v_i and v'. It is not hard to see that the complexity of finding this meeting point is at most $(\alpha+\beta)$, i.e., number of edges in $\mathcal{T}$. Therefore, the complexity of this part is $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$. Putting these together, complexity of the algorithm is $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$.

3) Complexity of the Algorithm 3: The initialization part of the Algorithm 3 takes $\mathcal{O}(1)$ running time. The while loop at "Tree Construction and Cache Nodes Labeling" goes over all nodes in $\mathcal{T}$ exactly once. As the operation inside the loop takes $\mathcal{O}(1)$ time, the complexity of this part of the algorithm is $\mathcal{O}(\alpha + \beta)$. The third part of the algorithm is "Delivery Nodes Labeling". It is not difficult to see that the first for loop in this part requires at most β running times. Also, the second for loop takes $\mathcal{O}(\alpha\beta)$ running time. Thus, complexity of this part is at most $\mathcal{O}(\alpha\beta)$. Finally, as we have shown in proof of the Claim 5, the complexity of "Modifying Delivery Phase Signals" is $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$. Putting all these together, complexity of Algorithm 3 is $\mathcal{O}(\alpha^2\beta + \alpha\beta^2)$.

4) Complexity of the Algorithm 4: The algorithm needs $\mathcal{O}(\alpha\beta)$ memory units to save $N_{sat}(a, b, K)$ for $0 \leq a \leq \alpha$ and $0 \leq b \leq \beta$. Once $N_{sat}(\tilde{a}, \tilde{b}, K)$ is known for $(\tilde{a}, \tilde{b}) \in \mathcal{I}(a, b)$ then we are able to compute $N_{sat}(a + 1, b + 1, K)$ by using the recursive relationship.

The time complexity of populating the N_{sat} values can be determined as follows. At the initialization step we fill the first two rows and columns of the matrix N_{sat} corresponding to $a = 0, 1$ and $b = 0, 1$ respectively. Following this initialization step, the remaining rows and columns are populated. It is clear the that the initialization takes $\mathcal{O}(\alpha+\beta)$ time. In the main loop we compute each entry of matrix N_{sat} once. This computation takes at most $\mathcal{O}(\alpha\beta)$ operation as we look for minimum over set $\mathcal{I}(a, b)$ whose size is at most $\mathcal{O}(\alpha\beta)$. As we compute all entries of the matrix N_{sat} and each entry takes $\mathcal{O}(\alpha\beta)$ running times thus time complexity of the algorithm is $\mathcal{O}(\alpha^2\beta^2)$. The required memory is $\mathcal{O}(\alpha\beta)$ as determined above.

REFERENCES

[1] D. Wessels, *Web Caching*. O'Reilly Media, 2001.

[2] A. Meyerson, K. Munagala, and S. Plotkin, "Web caching using access statistics," in *Proc. 12th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2001, pp. 354–363.

[3] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Placement algorithms for hierarchical cooperative caching," *J. Algorithms*, vol. 38, no. 1, pp. 260–302, 2001.

[4] S. C. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1478–1486.

[5] B. Tan and L. Massoulié, "Optimal content placement for peer-to-peer video-on-demand systems," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 566–579, Apr. 2013.

[6] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, "On the scale and performance of cooperative Web proxy caching," in *Proc. 17th ACM Symp. Oper. Syst. Principles*, 1999, pp. 16–31.

[7] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 126–134.

[8] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale VoD system," in *Proc. ACM 6th Int. Conf. Emerg. Netw. Experim. Technol. (Co-NEXT)*, 2010.

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

[10] A. Sengupta, R. Tandon, and T. C. Clancy, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2015, pp. 1691–1695.

[11] N. Ajaykrishnan, N. S. Prem, V. M. Prabhakaran, and R. Vaze, "Critical database size for effective caching," in *Proc. IEEE 21st Nat. Conf. Commun.*, Feb. 2015, pp. 1–6.

[12] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, Mar. 2011.

[13] E. Lubetzky and U. Stav, "Nonlinear index coding outperforming the linear optimum," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3544–3551, Aug. 2009.

[14] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2014.

[15] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2014, pp. 1878–1883.

[16] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb. 2017.

[17] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order optimal coded caching-aided multicast under zipf demand distributions," in *Proc. 11th Int. Symp. Wireless Commun. Syst.*, 2014, pp. 1–5.

[18] J. Hachem, N. Karamchandani, and S. Diggavi, "Multi-level coded caching," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 56–60.

[19] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.

[20] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 2142–2146.

[21] J. Hachem, N. Karamchandani, and S. N. Diggavi. (2014). "Coded caching for heterogeneous wireless networks with multi-level access." [Online]. Available: http://arxiv.org/abs/1404.6560

[22] M. Ji, A. M. Tulino, J. Llorca, and G. Caire. (2014). "Order optimal coded delivery and caching: Multiple groupcast index coding." [Online]. Available: http://arxiv.org/abs/1402.4572

[23] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in D2D wireless networks," in *Proc. IEEE Inf. Theory Workshop*, Sep. 2013, pp. 1–5.

[24] A. Sengupta and R. Tandon, "Beyond cut-set bounds-the approximate capacity of D2D networks," in *Proc. IEEE Inf. Theory Workshop*, Jun. 2015, pp. 78–83.

[25] J. Zhang, X. Lin, C.-C. Wang, and X. Wang, "Coded caching for files with distinct file sizes," in *Proc. IEEE Intl. Symp. Inf. Theory*, Sep. 2015, pp. 1686–1690.

[26] L. Tang and A. Ramamoorthy, "Low subpacketization schemes for coded caching," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2017.

[27] L. Tang and A. Ramamoorthy, "Coded caching with low subpacketization levels," in *Proc. IEEE Workshop Netw. Coding (NetCod)*, Dec. 2016, pp. 1–6.

[28] Q. Yan, M. Cheng, X. Tang, and Q. Chen. (2015). "On the placement delivery array design in centralized coded caching scheme." [Online]. Available: http://arxiv.org/abs/1510.05064.

[29] U. Niesen and M. A. Maddah-Ali, "Coded caching for delay-sensitive content," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 5559–5564.

[30] H. Ghasemi and A. Ramamoorthy, "Asynchronous Coded Caching," in *Proc. IEEE Inl. Symp. Inf. Theory*, Apr. 2017.
[31] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
[32] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.
[33] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr. (20156). "A fundamental tradeoff between computation and communication in distributed computing." [Online]. Available: http://arxiv.org/abs/1604.07086
[34] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. (2015). "Speeding up distributed machine learning using codes." [Online]. Available: http://arxiv.org/abs/1512.02673
[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
[36] C. Tian. (2015). "A note on the fundamental limits of coded caching." [Online]. Available: http://arxiv.org/abs/1503.00010

Hooshang Ghasemi is a Ph. D. student in the Department of Electrical and Computer Engineering at Iowa State University. He obtained his M. Sc. degree from Sharif University of Technology, Tehran, Iran in 2012 and his B. Sc. degree from Amirkabir University of Technology, Tehran, Iran in 2009. His research interests are in the area of information theory and signal processing.

Aditya Ramamoorthy (M'05) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, in 1999, and the M.S. and Ph.D. degrees from the University of California, Los Angeles (UCLA), in 2002 and 2005, respectively. He was a systems engineer with Biomorphic VLSI Inc. until 2001. From 2005 to 2006, he was with the Data Storage Signal Processing Group of Marvell Semiconductor Inc. Since fall 2006, he has been with the Electrical and Computer Engineering Department at Iowa State University, Ames, IA 50011, USA. His research interests are in the areas of network information theory, channel coding and signal processing for bioinformatics and nanotechnology. Dr. Ramamoorthy served as an editor for the IEEE TRANSACTIONS ON COMMUNICATIONS from 2011 2015. He is currently serving as an associate editor for the IEEE TRANSACTIONS ON INFORMATION THEORY. He is the recipient of the 2012 Iowa State Universitys Early Career Engineering Faculty Research Award, the 2012 NSF CAREER award, and the Harpole-Pentair professorship in 2009 and 2010.