Coded Caching with Full Heterogeneity: Exact Capacity of The Two-User/Two-File Case

Chih-Hua Chang, Borja Peleato, and Chih-Chun Wang

***Abstract*—The most commonly used setting in the coded caching literature consists of the following four elements: (i) homogeneous file sizes, (ii) homogeneous cache sizes, (iii) user-independent homogeneous file popularity (i.e., all users share the same file preference), and (iv) worst-case rate analysis. While recent results have relaxed some of these assumptions, deeper understanding of the full heterogeneity setting is still much needed since traditional caching schemes place little assumptions on file/cache sizes and almost always allow each user to have his/her own file preference through individualized file request prediction. Taking a microscopic approach, this paper characterizes the *exact capacity* of the smallest 2-user/2-file ($N = K = 2$) problem but under the most general setting that simultaneously allows for (i) heterogeneous files sizes, (ii) heterogeneous cache sizes, (iii) user-dependent file popularity, and (iv) average-rate analysis. Solving completely the case of $N = K = 2$ could shed further insights on the performance and complexity of optimal coded caching with full heterogeneity for arbitrary N and K.**

***Index Terms*—Coded caching, average rate, non-uniform demands, heterogeneous file size, heterogeneous cache size.**

I. Introduction

NOWADAYS high-definition video streaming motivates the demand for high-throughput internet traffic with small delay. One way to contain the peak load within the underlying communication channel capacity is to use caching to re-distribute some of the peak traffic to off-peak hours by prefetching (some of) the content in advance. The design tasks of a caching scheme consist of two parts: how to place the content during off-peak hours and how to satisfy the requests by delivering the additional packets during peak hours. Caching is especially attractive under the model of *broadcast channels* for which a single packet transmission could simultaneously benefit/reach multiple destinations.

Content caching has been studied in various settings [2], such as exploiting the opportunities of user population, file correlation, and time correlation. These traditional techniques usually divide each file into multiple (uncoded) pieces, prefetch some of them, and transmit the rest when needed. Recently, coded caching was proposed [3], which reduces delivery time by substituting the uncoded pieces with a coded version and taking advantage of multicasting capabilities. The results show that coded caching can shorten the *worst-case* delivery time by a factor of $(\frac{1}{1+KM/FN})$ when compared to the traditional uncoded caching schemes, where N is the number of files, K is the number of users, M is the individual cache size and F is the individual file size. While the capacity of the general coded caching problem remains an open problem, the optimal coded caching scheme (exact capacity) has been characterized for some special cases [3]–[7] and order-optimal capacity characterization has been found for several more general scenarios [3], [8]–[15].

Most existing results are based on the settings of (i) homogeneous file sizes, (ii) homogeneous cache sizes, (iii) user-independent homogeneous file popularity, and (iv) worst-case analysis. These settings are not 100% compatible with the traditional uncoded caching solutions. Specifically, the basic design principle of traditional uncoded schemes is to first predict the likelihood of the next file request for each individual user separately (i.e., user-dependent heterogeneous file popularity), and then let each user store the most likely file(s) until his/her cache is full (which is naturally applicable to heterogeneous file and cache sizes). The rationale behind this simple design is that such a probability-based greedy solution would reduce the *average rate* during delivery, even though there is no information-theoretic optimality guarantee.

Because of the aforementioned differences between their settings, a coded scheme designed for the homogeneous, worst-case setting could have significantly worse average-rate performance in practice when compared to a traditional scheme, especially for the scenarios in which the individualized file request prediction is very effective and the file and cache sizes are highly heterogeneous. In principle, since coded caching is a strict generalization of any uncoded solution, an optimal coded caching solution should outperform its non-coded counterpart under *any setting*. This potential loss of performance[1] is mainly due to the mismatch between practical scenarios and the homogeneous and worst case settings for which existing coded caching schemes are optimized.

Motivated by this observation, this work studies the exact capacity region and the corresponding optimal coded caching schemes under (i) heterogeneous file sizes, (ii) heterogeneous cache sizes, (iii) user-dependent heterogeneous file popularity, and (iv) average-rate analysis. Such results could allow the system designers to accurately assess the performance gain of coded caching (the ultimate capacity minus the achievable rate of traditional uncoded schemes) in a practical heterogeneous

This work was supported in parts by NSF under Grant CCF-1422997, Grant ECCS-1407604, Grant CCF-1618475, Grant CCF-1816013, Grant CCF-2008527, and Grant CNS-2107363; and by the Comunidad Autónoma de Madrid under Grant 2020-T1/TIC-20698.

Part of this work was presented in ISIT 2019 [1].

C.-H. Chang and C.-C. Wang are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA (e-mail: {chang377, chihw}@purdue.edu).

B. Peleato is with the Signal Theory and Communications Department, Universidad Carlos III de Madrid, Spain (e-mail: bpeleato@ing.uc3m.es).

[1] In practice, there are other issues that need to be addressed, e.g., synchronization [16]. Our statement disregards the implementation overhead and focuses exclusively on the theoretic performance under heterogeneous settings.

setting. While the problem remains open for general N and K values, we characterize the exact capacity for $N = K = 2$. The results can shed further insights for general N and K.

A. Comparison to Existing Results

Several existing works relaxed parts of the above conditions (i) to (iv). Table I provides a non-comprehensive list of several related results. For example, the authors in [3] assume homogeneous file and cache size and, under those conditions, characterize the exact worst-case capacity when $N = K = 2$ and propose a scheme that achieves order-optimal worst-case rate for arbitrary N and K. [12] provides a new information-theoretic lower bound and a corresponding order-optimal scheme of average rate with homogeneous file size, homogeneous cache size, and user-independent popularity.

As can be seen in Table I, finding the exact capacity of coded caching remains a difficult task. Most existing exact capacity results [3]–[5], [17] are based on small K (i.e., $K = 2$ or $K = 3$) and focus on the *worst-case* rate rather than a general probabilistic average-rate model. One of the most general heterogeneous setting results is [18], which uses linear programming results to search for better achievable rates without deriving any converse bounds, and is not focused on the general user-dependent file popularity setting. By focusing on the average-rate setting with heterogeneous file and cache sizes as well as user-dependent file popularity, our $N = K = 2$ results represent the first step toward fully characterizing the capacity of coded caching with full heterogeneity.

II. Problem Formulation

We consider the simplest non-trivial coded caching system with $N = 2$ files and $K = 2$ users. A central server has access to two files W_1 and W_2 of file sizes F_1 and F_2 bits, respectively. We sometimes write F_1 and F_2 as some non-integer values, e.g., $F_1 = 1.5$ and $F_2 = \frac{1}{3}$. One way to interpret this real-valued file-size expression is to assume F_1 and F_2 are sufficiently large so that we can express F_1 and F_2 by their normalized values instead. The cache content of user k is denoted by Z_k and is of size M_k bits for $k \in \{1, 2\}$. Without loss of generality, we assume real-valued $M_k \in [0, F_1 + F_2]$ for all k.

The operation of the system contains two phases, the *placement phase* and the *delivery phase*. In the *placement phase*, user k populates its cache by

$$Z_k = \phi_k(W_1, W_2),\ \forall k \in \{1, 2\}, \tag{1}$$

where ϕ_k is the caching function of user k. In the *delivery phase*, the two users send a demand request $\vec{d} \triangleq (d_1, d_2) \in \{1, 2\}^2$ to the server, i.e., user k demands file W_{d_k}. The probability mass function of the demand request $\vec{d}$ is denoted by $p_{\vec{d}}$, which satisfies $\sum_{\vec{d}\in\{1,2\}^2} p_{\vec{d}} = 1$. We assume $\{p_{\vec{d}} : \vec{d} \in \{1, 2\}^2\}$ is known to the server.

One popular choice of $p_{\vec{d}}$ is to assume that the demands of user-1 and user-2 are *statistically independent*, i.e., $p_{(d_1,d_2)} = p^{[1]}_{d_1} p^{[2]}_{d_2}$ where $p^{[k]}_{\vec{d}}$ is the marginal probability that user-k requests file W_d. In this work, we allow for arbitrary $p_{\vec{d}}$ that can be statistically independent or not.

After receiving $\vec{d}$, the server *broadcasts* an encoded signal

$$X_{\vec{d}} = \psi(\vec{d}, W_1, W_2) \tag{2}$$

of $R_{\vec{d}}$ bits with encoding function ψ through an *error-free* broadcast channel. Each user k then uses its cache content Z_k and the received signal $X_{\vec{d}}$ to decode his/her desired file

$$\hat{W}_{d_k} = \mu_k(\vec{d}, X_{\vec{d}}, Z_k),\ \forall k \in \{1, 2\}, \tag{3}$$

where μ_k is the decoding function of user k. Herein we assume that each user k knows the network-wide request pattern $\vec{d}$, which can be easily achieved by piggybacking the 2-bit vector $\vec{d}$ to the encoded symbol $X_{\vec{d}}$.

Definition 1. *A coded caching scheme for $N = K = 2$ is specified completely by its five functions $\{\phi_1, \phi_2, \psi, \mu_1, \mu_2\}$. The scheme is* zero-error feasible *if $\hat{W}_{d_k} = W_{d_k}$ for all $\vec{d} \in \{1, 2\}^2$, all $k \in \{1, 2\}$, and all $W_k \in \{0, 1\}^{F_k}$.*

Definition 2. *The worst-case rate of a zero-error coded caching scheme is*

$$R^* = \max_{\vec{d}\in\{1,2\}^2} R_{\vec{d}}. \tag{4}$$

The worst-case capacity is the infimum of the worst-case rates of all zero-error schemes.

Definition 3. *The average rate of a zero-error coded caching scheme is*

$$\bar{R} = \sum_{\vec{d}\in\{1,2\}^2} p_{\vec{d}} R_{\vec{d}}. \tag{5}$$

The average-rate capacity is the infimum of the average rates of all zero-error schemes.

For simplicity, we slightly abuse the above notation and directly use R^* and $\bar{R}$ to denote the worst-case and the average-rate capacities, respectively, even though their original definitions in (4) and (5) are for the achievable rates instead.

III. Main Results

To solve the worst-case and the average-rate capacities R^* and $\bar{R}$, we first define the following strictly more general concept.

Definition 4. *The* per-request capacity region *(PRCR) is the closure of the rate vectors $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ of all zero-error coded caching schemes.*

The PRCR is the most fundamental performance limit of coded caching since it captures the optimal trade-off needed to simultaneously satisfy different request patterns.

In Section III-A we describe 7 basic coded caching schemes for the 2-file/2-user setting ($N = K = 2$). Section III-B then provides the basic lower bounds of the 4-dimensional coded caching rate $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$. Finally, Section III-C shows that the 7 basic schemes can achieve the 4-dimensional rate lower bounds. The end result is thus a complete characterization of the PRCR for arbitrary (F_1, F_2, M_1, M_2) values. The exact characterization of PRCR will naturally lead to new closed form expressions for the capacities R^* and $\bar{R}$ under any arbitrary file popularity distribution $p_{\vec{d}}$. Further discussion on how to use the new PRCR characterization to derive the average-rate capacity $\bar{R}$ is provided at the end of Section III-C.

TABLE I
COMPARISONS OF EXISTING RESULTS

	Worst-case rate	Average rate
Homo. file sizes and homo. cache sizes	Arbitrary K and N, order-optimal rate [3], [8], [10] $K = 2$ and arbitrary N, exact capacity [3], [4] $K = 3$ and $N = 2$, exact capacity [4]	Arbitrary K and N, order-optimal rate [10]–[13], [19], [20] Arbitrary K and N, achievable rate only [21]–[23] Arbitrary K and $N = 2$, uncoded placement, exact capacity [24]
Homo. file sizes and heter. cache sizes	$K = 2$ and arbitrary N, exact capacity [5] Arbitrary K and N, achievable rate only [25]	Arbitrary K and N, achievable rate only [18] $K = 2$ and $N = 3$, exact capacity [6]
Heter. file sizes and homo. cache sizes	Arbitrary K and N, order-optimal rate [26]–[28]	Arbitrary K and N, achievable rate only [18]
Heter. file sizes and heter. cache sizes	$N = K = 2$, exact capacity [17]	Arbitrary K and N, achievable rate only [18] $N = K = 2$, exact capacity [This work]

A. Basic Zero-Error Coded Caching Schemes

We first describe 7 coded caching schemes for the 2-file/2-user setting ($N = K = 2$), which later serve as the basis for all our achievability proofs when characterizing the 4-dimensional PRCR. Consider user 1 and 2 of cache memory size m_1 and m_2 with two files w_1 and w_2 of sizes f_1 and f_2, respectively. The 7 basic schemes of parameters (f_1, f_2, m_1, m_2) are listed in Table II and can be described as follows.

1) Mix.Emp: Consider two files of equal size $f_1 = f_2 = f$, and two users of memory sizes $m_1 = f$ and $m_2 = 0$. In the placement phase, user 1 caches $w_1 \oplus w_2$ and user 2 caches none. In the delivery phase, the transmitted signals for the demands are $X_{(1,1)} = w_1$, $X_{(1,2)} = w_2$, $X_{(2,1)} = w_1$, and $X_{(2,2)} = w_2$. One can easily verify that for any $\vec{d}$, the transmitted symbol $X_{\vec{d}}$ satisfies the demands of both users. Since $X_{\vec{d}}$ is of size f for all $\vec{d}$, the corresponding achievable rate vector is $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f, f, f, f)$. The first row of Table II summarizes the achievable rate vector and the condition $f = f_1 = f_2 = m_1$, $m_2 = 0$ for this scheme to be zero-error feasible. Since user 1 stores an XORed packet and user 2 stores none, we call this scheme Mix.Emp.

2) Emp.Mix: The scheme is user-symmetric to Mix.Emp by swapping the roles of users 1 and 2. Since this time user 1 stores none and user 2 stores an XORed packet, we call this scheme Emp.Mix.

3) Ha.Fi: Consider two files of equal size $f_1 = f_2 = f$, and two users of equal memory size $m_1 = m_2 = f$. We divide the file $w_1 = (u_1, u_2)$ into two subfiles of size $(f/2, f/2)$ and divide the file $w_2 = (v_1, v_2)$ into two subfiles of size $(f/2, f/2)$. In the placement phase, user 1 caches (u_1, v_1) and user 2 caches (u_2, v_2). In the delivery phase, the transmitted signals for the demands are $X_{(1,1)} = u_1 \oplus u_2$, $X_{(1,2)} = u_2 \oplus v_1$, $X_{(2,1)} = u_1 \oplus v_2$, and $X_{(2,2)} = v_1 \oplus v_2$. Since $X_{\vec{d}}$ is of size $f/2$ for all $\vec{d}$, the achievable rates are $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f/2, f/2, f/2, f/2)$. Since each user stores half of file k for all k, we call this scheme Ha.Fi.

4) 1.1.Cov: In this scheme both users cache as much as possible from file 1. Consider two users of cache memory size $\max(m_1, m_2) \leq f_1$ and arbitrary f_2. If $m_1 \geq m_2$, we divide $w_1 = (u_1, u_2, u_3)$ into three subfiles of file size $(m_2, m_1 - m_2, f_1 - m_1)$. In the placement phase, user 1 caches (u_1, u_2) and user 2 caches u_1. In the delivery phase, the transmitted signals for the different demands $\vec{d}$ are $X_{(1,1)} = (u_2, u_3)$, $X_{(1,2)} = (u_3, w_2)$, $X_{(2,1)} = (u_2, u_3, w_2)$, and $X_{(2,2)} = w_2$. One can easily verify that both users can decode their desired files under any demand $\vec{d}$. By quantifying the size of $X_{\vec{d}}$ for all $\vec{d}$, the achievable rates are $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f_1 - m_2, f_1 + f_2 - m_1, f_1 + f_2 - m_2, f_2)$.

If $m_1 < m_2$, we can implement the same scheme by swapping the roles of users 1 and 2. By taking into account both scenarios ($m_1 \geq m_2$ and $m_1 < m_2$), we can write the achievable rate vector in the following more general form:

$$(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f_1 - \min(m_1, m_2),\\ f_1 + f_2 - m_1, f_1 + f_2 - m_2, f_2). \tag{6}$$

Since the strategy of both users is “to cover as much file 1 as possible”, we call this scheme 1.1.Cov.

5) 1.2.Cov: In this scheme user 1 caches file 1 and user 2 caches file 2. Consider two users of memory size $m_1 \leq f_1$ and $m_2 \leq f_2$. If $m_1 \geq m_2$, we divide $w_1 = (u_1, u_2, u_3)$ into three subfiles of size $(m_2, m_1 - m_2, f_1 - m_1)$ and divide $w_2 = (v_1, v_2)$ into two subfiles of file size $(m_2, f_2 - m_2)$. In the placement phase, user 1 caches (u_1, u_2) and user 2 caches v_1. In the delivery phase, the transmitted signals for the different demands are $X_{(1,1)} = w_1$, $X_{(1,2)} = (u_3, v_2)$, $X_{(2,1)} = (u_1 \oplus v_1, u_2, u_3, v_2)$, and $X_{(2,2)} = w_2$, which results in the achievable rate vector being $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f_1, f_1 + f_2 - m_1 - m_2, f_1 + f_2 - m_2, f_2)$.

If $m_1 < m_2$, a symmetric scheme can be implemented by dividing w_1 into two subfiles of size $(m_1, f_1 - m_1)$ and w_2 into three subfiles of size $(m_1, m_2 - m_1, f_2 - m_2)$. By taking into account both scenarios ($m_1 \geq m_2$ and $m_1 < m_2$), we can write the achievable rate vector in the following more general form:

$$(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)}) = (f_1,\\ f_1 + f_2 - m_1 - m_2, f_1 + f_2 - \min(m_1, m_2), f_2). \tag{7}$$

Since the strategy of user 1 is “to cover as much file 1 as possible” and the strategy of user 2 is “to cover as much file 2 as possible”, we call this scheme 1.2.Cov.

6) 2.1.Cov: The scheme is user-symmetric to 1.2.Cov by swapping the roles of users 1 and 2. Since the strategy of user 1 is “to cover as much file 2 as possible” and the strategy of user 2 is “to cover as much file 1 as possible”, we call this scheme 2.1.Cov.

7) 2.2.Cov: The scheme is file-symmetric to 1.1.Cov by swapping the roles of files 1 and 2. Since the strategy of user 1 is “to cover as much file 2 as possible” and so is user 2’s strategy, we call this scheme 2.2.Cov.

TABLE II
BASIC CODED CACHING SCHEMES FOR TWO FILES OF SIZE (f_1, f_2) AND TWO USERS OF MEMORY (m_1, m_2). IT IS POSSIBLE TO HAVE $f_1 \geq f_2$, OR $f_1 < f_2$ AND $m_1 \geq m_2$, OR $m_1 < m_2$.

Scheme	Feasibility Condition	Achievable Rate Vector $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$	Intuition
Mix.Emp	$f = f_1 = f_2 = m_1$; $m_2 = 0$	(f, f, f, f)	Premixing at d_1
Emp.Mix	$f = f_1 = f_2 = m_2$; $m_1 = 0$	(f, f, f, f)	Premixing at d_2
Ha.Fi	$f = f_1 = f_2 = m_1 = m_2$	$(f/2, f/2, f/2, f/2)$	Splitting files in halves
1.1.Cov	$\max(m_1, m_2) \leq f_1$	$(f_1 - \min(m_1, m_2), f_1 + f_2 - m_1, f_1 + f_2 - m_2, f_2)$	Covering $\vec{d} = (1, 1)$
1.2.Cov	$m_1 \leq f_1$, $m_2 \leq f_2$	$(f_1, f_1 + f_2 - m_1 - m_2, f_1 + f_2 - \min(m_1, m_2), f_2)$	Covering $\vec{d} = (1, 2)$
2.1.Cov	$m_1 \leq f_2$, $m_2 \leq f_1$	$(f_1, f_1 + f_2 - \min(m_1, m_2), f_1 + f_2 - m_1 - m_2, f_2)$	Covering $\vec{d} = (2, 1)$
2.2.Cov	$\max(m_1, m_2) \leq f_2$	$(f_1, f_1 + f_2 - m_2, f_1 + f_2 - m_1, f_2 - \min(m_1, m_2))$	Covering $\vec{d} = (2, 2)$

It is worth noting that none of the 7 basic schemes can be achieved by space-sharing the rest of 6 schemes and they thus will serve as the basis of our achievability proofs [29].

B. Lower Bounds of the PRCR

We derive the following lower bounds for arbitrary file and cache sizes (F_1, F_2, M_1, M_2).

Instance 0: Nonnegative rates:

$$R_{\vec{d}} \geq 0, \quad \forall \vec{d} \in \{1, 2\}^2.$$

By varying $\vec{d}$, there are a total of 4 inequalities in Instance 0.

$$R_{(1,1)} \geq 0 \quad \text{(O-1)} \qquad R_{(1,2)} \geq 0 \quad \text{(O-2)}$$
$$R_{(2,1)} \geq 0 \quad \text{(O-3)} \qquad R_{(2,2)} \geq 0 \quad \text{(O-4)}$$

Instance 1: For any $i, j \in \{1, 2\}$, there are two inequalities:

$$\begin{aligned} R_{(i,j)} + M_1 \geq & H(X_{(i,j)}, Z_1) = H(X_{(i,j)}, Z_1, W_i) \geq F_i, \\ R_{(i,j)} + M_2 \geq & H(X_{(i,j)}, Z_2) = H(X_{(i,j)}, Z_2, W_j) \geq F_j. \end{aligned}$$

By varying i, j, there are a total of 8 inequalities in Instance 1.

$$R_{(1,1)} + M_1 \geq F_1 \quad \text{(I-1)} \qquad R_{(1,1)} + M_2 \geq F_1 \quad \text{(I-2)}$$
$$R_{(1,2)} + M_1 \geq F_1 \quad \text{(I-3)} \qquad R_{(1,2)} + M_2 \geq F_2 \quad \text{(I-4)}$$
$$R_{(2,1)} + M_1 \geq F_2 \quad \text{(I-5)} \qquad R_{(2,1)} + M_2 \geq F_1 \quad \text{(I-6)}$$
$$R_{(2,2)} + M_1 \geq F_2 \quad \text{(I-7)} \qquad R_{(2,2)} + M_2 \geq F_2 \quad \text{(I-8)}$$

Instance 2: For any $(i, j) = (1, 2)$ or $(2, 1)$,

$$R_{(i,j)} + M_1 + M_2 \geq H(X_{(i,j)}, Z_1, Z_2) \geq F_1 + F_2.$$

By varying (i, j), there are a total of 2 inequalities in Instance 2.

$$R_{(1,2)} + M_1 + M_2 \geq F_1 + F_2 \quad \text{(II-1)}$$
$$R_{(2,1)} + M_1 + M_2 \geq F_1 + F_2. \quad \text{(II-2)}$$

Instance 3: For any $i, j \in \{1, 2\}$, there are two inequalities:

$$\begin{aligned} R_{(i,1)} + & R_{(j,2)} + M_2 \geq H(X_{(i,1)}, X_{(j,2)}, Z_2) \geq F_1 + F_2, \\ R_{(1,i)} + & R_{(2,j)} + M_1 \geq H(X_{(1,i)}, X_{(2,j)}, Z_1) \geq F_1 + F_2. \end{aligned}$$

By varying i, j, there are a total of 8 inequalities in Instance 3.

$$\begin{aligned} R_{(1,1)} + R_{(1,2)} + M_2 \geq & F_1 + F_2, && \text{(III-1)} \\ R_{(1,1)} + R_{(2,1)} + M_1 \geq & F_1 + F_2, && \text{(III-2)} \\ R_{(1,1)} + R_{(2,2)} + M_1 \geq & F_1 + F_2, && \text{(III-3)} \\ R_{(1,1)} + R_{(2,2)} + M_2 \geq & F_1 + F_2, && \text{(III-4)} \\ R_{(1,2)} + R_{(2,1)} + M_1 \geq & F_1 + F_2, && \text{(III-5)} \\ R_{(1,2)} + R_{(2,1)} + M_2 \geq & F_1 + F_2, && \text{(III-6)} \\ R_{(1,2)} + R_{(2,2)} + M_1 \geq & F_1 + F_2, && \text{(III-7)} \\ R_{(2,1)} + R_{(2,2)} + M_2 \geq & F_1 + F_2. && \text{(III-8)} \end{aligned}$$

Instance 4 uses a more refined technique[2] and thus we provide the detailed derivation.

Instance 4: For any $(i, j) = (1, 2), (2, 1)$, or $(2, 2)$,

$$\begin{aligned} & R_{(i,1)} + R_{(1,j)} + M_1 + M_2 && (8) \\ & \geq H(X_{(i,1)}) + H(Z_2) + H(X_{(1,j)}) + H(Z_1) && (9) \\ & \geq H(X_{(i,1)}, Z_2) + H(X_{(1,j)}, Z_1) && (10) \\ & \geq H(X_{(i,1)}, Z_2, W_1) + H(X_{(1,j)}, Z_1, W_1) && (11) \\ & \geq H(X_{(i,1)}, X_{(1,j)}, Z_1, Z_2, W_1) + H(W_1) && (12) \\ & \geq H(X_{(i,1)}, X_{(1,j)}, Z_1, Z_2, W_1, W_2) + H(W_1) && (13) \\ & = H(W_1, W_2) + H(W_1) = 2F_1 + F_2 && (14) \end{aligned}$$

where (10) follows from that the sum of marginal entropies is no less than the joint entropy; (11) follows from that user 2 can decode W_1 based on $X_{(i,1)}$ and Z_2, and user 1 can decode W_1 based on $X_{(1,j)}$ and Z_1; (12) follows from the Shannon-type inequality; (13) follows from that we can decode W_2 from $X_{(i,1)}$, $X_{(1,j)}$, Z_1, and Z_2 since we choose $(i, j) \in \{(1, 2), (2, 1), (2, 2)\}$ to begin with; and (14) follows from that X's and Z's are functions of (W_1, W_2).

Symmetrically for any $(i, j) = (1, 2), (2, 1)$, or $(1, 1)$

$$R_{(i,2)} + R_{(2,j)} + M_1 + M_2 \geq F_1 + 2F_2.$$

[2] A more general version of the techniques can be found in [3], [4], [30], [31].

Varying (i,j), there are a total of 6 inequalities in Instance 4.

$$R_{(1,1)} + R_{(1,2)} + M_1 + M_2 \geq 2F_1 + F_2, \quad \text{(IV-1)}$$
$$R_{(1,1)} + R_{(2,1)} + M_1 + M_2 \geq 2F_1 + F_2, \quad \text{(IV-2)}$$
$$R_{(1,2)} + R_{(2,1)} + M_1 + M_2 \geq 2F_1 + F_2, \quad \text{(IV-3)}$$
$$R_{(1,2)} + R_{(2,1)} + M_1 + M_2 \geq F_1 + 2F_2, \quad \text{(IV-4)}$$
$$R_{(1,2)} + R_{(2,2)} + M_1 + M_2 \geq F_1 + 2F_2, \quad \text{(IV-5)}$$
$$R_{(2,1)} + R_{(2,2)} + M_1 + M_2 \geq F_1 + 2F_2. \quad \text{(IV-6)}$$

Totally, there are 28 linear inequalities in Instances 0 to 4.

C. Coded Caching Capacity for $N = K = 2$

The derivation of the aforementioned lower bounds is relatively straightforward, see [3], [4], [17], [30] for similar derivations. A significantly more important contribution of this work is to show that these lower bounds indeed characterize the exact 4-dimensional PRCR.

Proposition 1. *Consider arbitrary (F_1, F_2, M_1, M_2). For any $\vec{R}$ that satisfies the 28 lower bounds in Section III-B simultaneously, we can find a zero-error scheme attaining such $\vec{R}$.*

Proposition 1 leads to the following self-explanatory corollary:

Corollary 1. *Given arbitrary (F_1, F_2, M_1, M_2) values, the average-rate capacity can be characterized by solving a linear programming (LP) problem using the 28 lower bounds in Section III-B.*

Further discussion of Corollary 1 will be provided in the remark after Proposition 3. Proposition 1 follows directly from the following propositions.

Proposition 2. *The 4-dimensional polytope formed by the 28 linear inequalities has either 2 or 4 or 6 distinct corner points. The actual number depends on the underlying (F_1, F_2, M_1, M_2) values. An exhaustive list of all the corner points under arbitrary (F_1, F_2, M_1, M_2) is provided jointly in Fig. 1 and Table III.*

Proposition 3. *All 28 corner points listed in Fig. 1 and Table III can be achieved by space-sharing the 7 basic schemes described in Section III-A.*

The proofs of Propositions 2 and 3 are relegated to Appendices A and B, respectively. In the proof of Proposition 3, we explicitly find 28 constructions that achieve the 28 corner points, respectively.

Remark: The statement in Proposition 1 has already cast the coded caching capacity problem as an LP problem involving 4 variables $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ and 28 inequalities, which can be solved numerically. Nonetheless, the constructive and explicit statements in Propositions 2 and 3 go one step further. By exhaustively characterizing all corner points of the lower bounds and then proving their achievability, one can use Proposition 3 to devise the coded caching scheme of any feasible $\vec{R}$ and the formulas in Proposition 2, i.e., the expressions listed in Table III, can be used to derive the closed-form expression of the capacity without any numerical solver. Compared to the implicit statement in Proposition 1, Propositions 2 and 3 uncover new, cleaner results that shed further insight to the problem at hand.

For example, since the average capacity $\bar{R}$ is achieved by the vector $\vec{R}$ in the PRCR that has the smallest linear objective value $\sum p_{\vec{d}} R_{\vec{d}}$ and since the minimum of a linear programming problem can only happen at the corner points, we can easily use the corner points in Fig. 1 and Table III to characterize the average-rate capacity with an arbitrary popularity vector $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)})$. Namely, given any (F_1, F_2, M_1, M_2), we first use Fig. 1 to find all the corner points in the PRCR (at most 6 of them). Then for each corner point, we plug in the closed-form expression in Table III to the objective function $\sum p_{\vec{d}} R_{\vec{d}}$. Repeat this process for each corner point. Finally the smallest objective value must be the average-rate capacity under the given (F_1, F_2, M_1, M_2) and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)})$. Two examples of this general procedure are provided as follows.

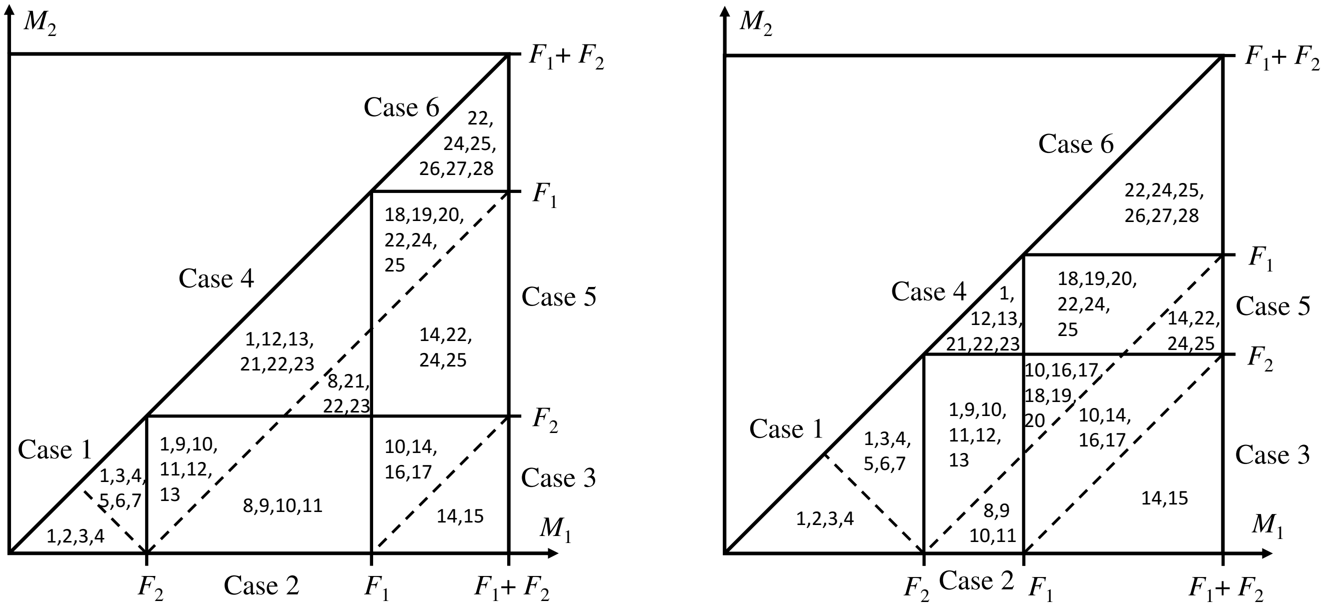
Example 1. *Suppose $(F_1, F_2) = (1.5, 1)$ and the demands of the users are statistically independent, with user 1 demanding files 1 and 2 with probability 2/3 and 1/3, respectively, and user 2 demanding files 1 and 2 with probability 2/5 and 3/5, respectively. The corresponding average-rate capacity for arbitrary (M_1, M_2) is described in Fig. 2.*

As discussed in the introduction, the main motivation of our study is to compare the optimal coded caching capacity with the performance of the naïve likelihood-based uncoded caching solution. For this particular example, we thus compare in Fig. 3 the optimal average-rate coded caching capacity with the performance of (i) the naïve likelihood-based uncoded caching, and (ii) the coded caching scheme in [17] that is optimized for the worst-case performance. As expected, the optimal coded caching capacity is always the smallest of the three. The largest rate reduction over the uncoded scheme is at v_{13} for which the optimal coded caching scheme uses only $\frac{1/2}{11/15} \simeq 68.2\%$ of the bandwidth of the uncoded solution.

It is also worth noting that at the corner point v_3, the worst-case-optimal coded caching scheme[3] actually performs worse than the uncoded scheme (5% worse) while the optimal coded scheme still exhibits 10% improvement over the uncoded solution.

Example 2. *Suppose $(F_1, F_2) = (1.5, 1)$ and the demands of the users are dependent with popularity $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (\frac{2}{15}, \frac{8}{15}, \frac{4}{15}, \frac{1}{15})$. Namely, user 1 requests files 1 and 2 with probabilities 2/3 and 1/3 and user 2 requests files 1 and 2 with probability 2/5 and 3/5 but their demands are no longer statistically independent. Instead, the demands are negatively correlated with correlation coefficient $-2\sqrt{5}/15$. The corresponding average-rate capacity for arbitrary (M_1, M_2) is described in Fig. 4.*

[3] In general, the optimal scheme for the worst-case capacity may not be unique. A more precise statement should thus be "one worst-case optimal coded scheme actually performs ...". It is worth mentioning that it is an open problem how a system designer should choose from the set of optimal worst-case coded scheme since currently there is little study about what is the set of optimal worst-case coded schemes.



(a) $F_1 \geq 2F_2$

(b) $2F_2 > F_1 \geq F_2$

Fig. 1. Description of the regions of (M_1, M_2) and the corresponding corner points. The x-axis (resp. y-axis) is for the M_1 (resp. M_2) value. In this figure we assume $F_1 \geq F_2$ and only describe the cases when $M_1 \geq M_2$, thus the lower-half of the line $M_1 = M_2$. The cases of $F_1 < F_2$ and $M_1 < M_2$ can be obtained by swapping the file and user indices, respectively. Two scenarios are considered: (a) $F_1 \geq 2F_2$; (b) $2F_2 > F_1 \geq F_2$. In both scenarios, there are 6 major regions described by solid lines, which are labeled as Cases 1 to 6. The 6 major regions are further partitioned into 11 sub-regions by three 45-degree dotted lines. The numbers within each sub-region are the indices of the corner points of the 4-dimensional PRCR polytope when (M_1, M_2) falls into the sub-region. For example, in both (a) and (b), the triangular subregion corresponding to "$0 \leq M_2 \leq M_1$ and $M_1 + M_2 \leq F_2$" are labeled by "1,2,3,4". This means that when "$0 \leq M_2 \leq M_1$ and $M_1 + M_2 \leq F_2$" holds, the 4 corner points are vertices 1, 2, 3, and 4 described in Table III.

TABLE III
THE EXPRESSIONS OF ALL 28 POSSIBLE CORNER POINTS.

Vertex	Corresponding rate vector $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$
1	$(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$
2	$(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$
3	$(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$
4	$(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$
5	$(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$
6	$(F_1, F_1, F_1 + F_2 - M_1 - M_2, F_2)$
7	$(F_1 + \frac{1}{2}(F_2 - M_1 - M_2), F_1 + \frac{1}{2}(F_2 - M_1 - M_2), F_1 + \frac{1}{2}(F_2 - M_1 - M_2), F_2 + \frac{1}{2}(F_2 - M_1 - M_2))$
8	$(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$
9	$(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$
10	$(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$
11	$(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$
12	$(F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_1 - M_2, F_2)$
13	$(F_1 + \frac{1}{2}(F_2 - M_1 - M_2), F_1 + \frac{1}{2}(F_2 - M_1 - M_2), F_1 + \frac{1}{2}(F_2 - M_1 - M_2), \frac{1}{2}(F_2 + M_1 - M_2))$
14	$(F_1 - M_2, F_2, F_1 - M_2, F_2)$
15	$(F_1, F_2 - M_2, F_1, F_2 - M_2)$
16	$(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$
17	$(\frac{1}{2}(F_1 + M_1 - M_2), F_2 + \frac{1}{2}(F_1 - M_1 - M_2), \frac{1}{2}(F_1 + M_1 - M_2), F_2 + \frac{1}{2}(F_1 - M_1 - M_2))$
18	$(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$
19	$(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$
20	$(F_1 + \frac{1}{2}(F_2 - M_1 - M_2), \frac{1}{2}(F_2 + M_1 - M_2), F_1 + \frac{1}{2}(F_2 - M_1 - M_2), \frac{1}{2}(F_2 + M_1 - M_2))$
21	$(F_1 + F_2 - M_2, F_1 - M_1, F_1 + F_2 - M_2, F_2)$
22	$(F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0)$
23	$(F_1 + \frac{F_2}{2} - M_2, F_1 + \frac{F_2}{2} - M_1, F_1 + \frac{F_2}{2} - M_2, \frac{F_2}{2})$
24	$(F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2, F_1 + F_2 - M_1)$
25	$(\frac{1}{2}(F_1 + F_2 + M_1) - M_2, \frac{1}{2}(F_1 + F_2 - M_1), \frac{1}{2}(F_1 + F_2 + M_1) - M_2, \frac{1}{2}(F_1 + F_2 - M_1))$
26	$(0, F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2)$
27	$(F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2)$
28	$(\frac{1}{2}(F_1 + F_2 - M_1), \frac{1}{2}(F_1 + F_2 + M_1) - M_2, \frac{1}{2}(F_1 + F_2 - M_1), \frac{1}{2}(F_1 + F_2 + M_1) - M_2)$

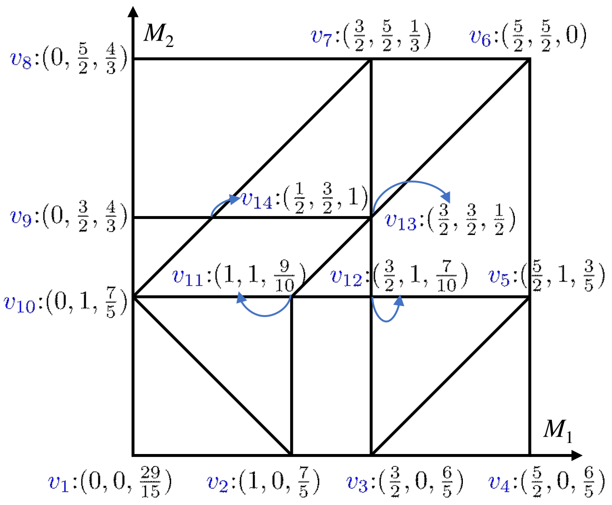

M_2
v_8:$(0,\frac{5}{2},\frac{4}{3})$
v_7:$(\frac{3}{2},\frac{5}{2},\frac{1}{3})$
v_6:$(\frac{5}{2},\frac{5}{2},0)$
v_9:$(0,\frac{3}{2},\frac{4}{3})$
v_{14}:$(\frac{1}{2},\frac{3}{2},1)$
v_{13}:$(\frac{3}{2},\frac{3}{2},\frac{1}{2})$
v_{10}:$(0,1,\frac{7}{5})$
v_{11}:$(1,1,\frac{9}{10})$
v_{12}:$(\frac{3}{2},1,\frac{7}{10})$
v_5:$(\frac{5}{2},1,\frac{3}{5})$
M_1
v_1:$(0,0,\frac{29}{15})$
v_2:$(1,0,\frac{7}{5})$
v_3:$(\frac{3}{2},0,\frac{6}{5})$
v_4:$(\frac{5}{2},0,\frac{6}{5})$

Fig. 2. The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (\frac{4}{15}, \frac{2}{5}, \frac{2}{15}, \frac{1}{5})$. There are 12 facets and 14 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) give the location and the third coordinate specifies the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity.

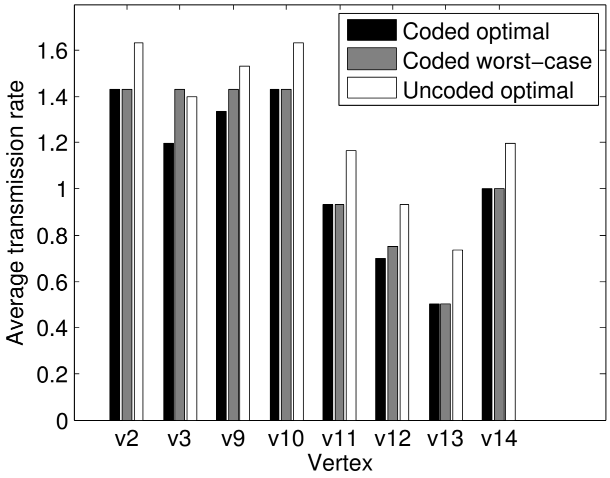


Fig. 3. Comparison of the average-rate capacity with the average rate of naïve likelihood-based uncoded caching, and the coded caching scheme in [17] that is optimized for the worst-case performance on some of the vertices in Fig. 2.

Comparing Example 1 and 2, one can see that even with the same marginal distribution, the optimal coded caching scheme can take into account the negative correlation, which results in a different capacity region.

We also compare the average rates of the optimal coded solution, the uncoded solution, and the worst-case-optimal coded solution with the setting of Example 2 in Fig. 5. The largest rate reduction over the uncoded scheme happens at v_{12} for which the optimal coded caching scheme uses only $\frac{7}{15}/\frac{11}{15} \simeq 63.6\%$ of the bandwidth of the uncoded solution.

The above examples consider user-dependent file popularity. If we relax that constraint and consider only uniform file popularity, we can derive a closed form capacity expression for any arbitrary (F_1, F_2, M_1, M_2).

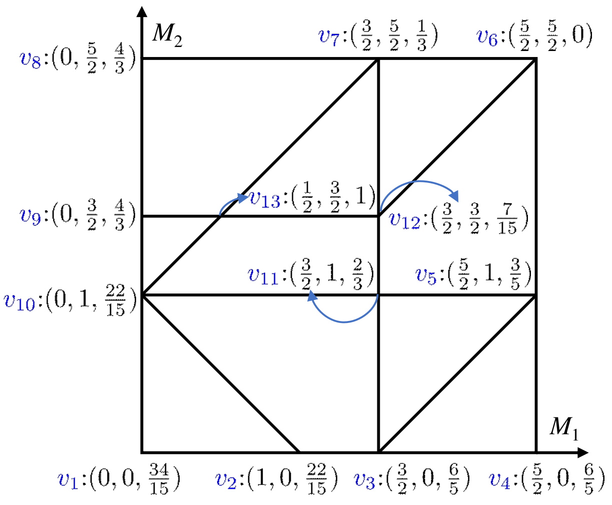

M_2
v_8:$(0,\frac{5}{2},\frac{4}{3})$
v_7:$(\frac{3}{2},\frac{5}{2},\frac{1}{3})$
v_6:$(\frac{5}{2},\frac{5}{2},0)$
v_9:$(0,\frac{3}{2},\frac{4}{3})$
v_{13}:$(\frac{1}{2},\frac{3}{2},1)$
v_{12}:$(\frac{3}{2},\frac{3}{2},\frac{7}{15})$
v_{10}:$(0,1,\frac{22}{15})$
v_{11}:$(\frac{3}{2},1,\frac{2}{3})$
v_5:$(\frac{5}{2},1,\frac{3}{5})$
M_1
v_1:$(0,0,\frac{34}{15})$
v_2:$(1,0,\frac{22}{15})$
v_3:$(\frac{3}{2},0,\frac{6}{5})$
v_4:$(\frac{5}{2},0,\frac{6}{5})$

Fig. 4. The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (\frac{2}{15}, \frac{8}{15}, \frac{4}{15}, \frac{1}{15})$. There are 10 facets and 13 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) give the location and the third coordinate specifies the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity.

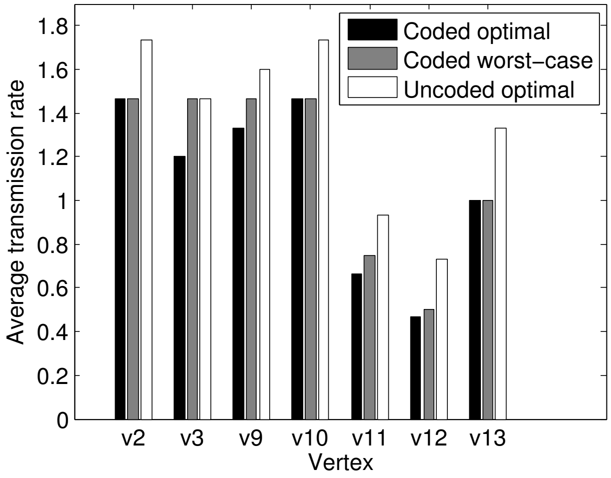


Fig. 5. Comparison of the average-rate capacity with the average rate of naïve likelihood-based uncoded caching, and the coded caching scheme in [17] that is optimized for the worst-case performance on some of the vertices in Fig. 4.

Corollary 2. *For arbitrary* (F_1, F_2) *satisfying* $F_1 \geq F_2$ *and uniform file popularity (i.e.,* $p_{\vec{d}} = 0.25$, $\forall \vec{d}$*), the average-rate capacity for arbitrary* (M_1, M_2) *is described in Fig. 6, which contains exactly 5 facets.*

The proof of Corollary 2 is relegated to Appendix C.

The exact PRCR characterization can also be used to easily rederive the worst-case capacity R^* with arbitrary (F_1, F_2, M_1, M_2), previously found by examining the outer bounds of entropic cones [17]. See Appendix D for details.

The closed form expressions of $\bar{R}$ and R^* as functions of (F_1, F_2, M_1, M_2) and $\{p_{\vec{d}}\}$, i.e., Corollary 2 and Corollary 3 in Appendix D, can be used to solve other design optimization problems. For example, we can solve the 2-user/2-file *memory allocation problem* [32] optimally by finding the (M_1^*, M_2^*)

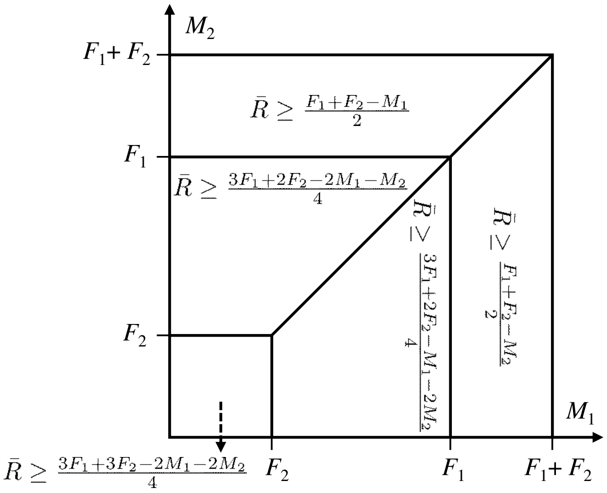

M_2
$F_1 + F_2$
$\bar{R} \geq \frac{F_1+F_2-M_1}{2}$
F_1
$\bar{R} \geq \frac{3F_1+2F_2-2M_1-M_2}{4}$
$\bar{R} \geq \frac{3F_1+2F_2-M_1-2M_2}{4}$
$\bar{R} \geq \frac{F_1+F_2-M_2}{2}$
F_2
M_1
$\bar{R} \geq \frac{3F_1+3F_2-2M_1-2M_2}{4}$
F_2
F_1
$F_1 + F_2$

Fig. 6. The average-rate capacity of uniform popularity, described for the case of $F_1 \geq F_2$.

that minimizes $\bar{R}$ (or R^*) subject to the total memory constraint $M_1 + M_2 \leq M_{\text{total}}$. That is, we evaluate the coded caching capacity over the line $(M_1, M_2) = (m, M_{\text{total}} - m)$ for all $m \in [0, M_{\text{total}}]$. Then, the optimal allocation is simply $(M_1, M_2) = (m^*, M_{\text{total}} - m^*)$, where m^* denotes the value that leads to the smallest capacity rate.

IV. Per-Request Capacity Region for $N > 2$ and $K > 2$

The same methodology that we have used to characterize the PRCR of coded caching systems with 2-user/2-file can be applied to more general settings, i.e., $N > 2$ and $K > 2$ as follows.

(i) Derive all the converse bounds of the N^K-dimensional PRCR.

(ii) Find all the vertices of the PRCR polytope formed by the converse bounds for all $\{F_1, \ldots, F_N\}$ and $\{M_1, \ldots, M_K\}$ values.

(iii) Find the achievable schemes for each of the vertices on the polytope. Such a process can be further simplified if there is any symmetry/homogeneity that can be exploited in the process. Also the construction of basic coded caching schemes can be used to characterize the achievability region.

The generalization is straightforward, but obtaining tight converse bounds for the PRCR may require complicated derivations. Furthermore, the number of inequalities, vertices, regions, and basic coded caching schemes in N^K dimensions increases exponentially with the number of files N and users K.

Despite these complications, we can still characterize some new PRCR for $N > 2$ and $K > 2$ by extending our 2-user/2-file results. For instance, the four families of inequalities (instances I through IV) described in Section III-B can be easily generalized without further conditions on file size, cache size, and file popularity: there would be N^K inequalities in Instance 0 (one for each possible request), $N \cdot N^K$ in instance 1, etc. Though unable to yield a complete characterization of PRCR for $N > 2$ and $K > 2$, the four families of inequalities are guaranteed to provide lower bounds for the PRCR with any $N \geq 2$ and $K \geq 2$ values.

We now show three examples of how our 2-user/2-file derivations can be extended to a scenario with three users and three files ($N = K = 3$). Examples 3 and 4 share the same file and cache sizes, but differ in the users' file popularities; Example 5 uses homogeneous file sizes and cache sizes. It is shown that our inequalities suffice to find the new 3-user/3-file PRCR results in all three cases.

Example 3. *Consider a 3-user/3-file coded caching system with three files with sizes* $(F_1, F_2, F_3) = (1, 2, 3)$ *and three cache memory sizes* $(M_1, M_2, M_3) = (1, 2, 3)$. *Assume that the file popularity is*

$$p_{(1,2,3)} = \tfrac{1}{2} \qquad p_{(2,1,3)} = \tfrac{1}{8} \qquad p_{(3,1,2)} = \tfrac{1}{32}$$
$$p_{(1,3,2)} = \tfrac{1}{4} \qquad p_{(2,3,1)} = \tfrac{1}{16} \qquad p_{(3,2,1)} = \tfrac{1}{32}$$

and $p_{\vec{d}} = 0$ *otherwise. The minimum achievable average rate (capacity) of this system is* $\bar{R} = \frac{25}{16}$.

Example 4. *Keeping the same cache and file sizes as in the previous example,* $(M_1, M_2, M_3) = (1, 2, 3)$, $(F_1, F_2, F_3) = (1, 2, 3)$, *we now invert the file popularity:*

$$p_{(1,2,3)} = \tfrac{1}{32} \qquad p_{(2,1,3)} = \tfrac{1}{16} \qquad p_{(3,1,2)} = \tfrac{1}{4}$$
$$p_{(1,3,2)} = \tfrac{1}{32} \qquad p_{(2,3,1)} = \tfrac{1}{8} \qquad p_{(3,2,1)} = \tfrac{1}{2}$$

and $p_{\vec{d}} = 0$ *otherwise. The minimum achievable average rate (capacity) of this system is* $\bar{R} = \frac{5}{2}$.

Example 5. *Now assume that all the files and caches have the same size, i.e.,* $(M_1, M_2, M_3) = (1, 1, 1)$, $(F_1, F_2, F_3) = (1, 1, 1)$, *and the file popularity is*

$$p_{(1,2,3)} = p_{(2,1,3)} = p_{(3,1,2)} = p_{(1,3,2)} = p_{(2,3,1)} = p_{(3,2,1)} = \tfrac{1}{6}$$

and $p_{\vec{d}} = 0$ *otherwise. The minimum achievable average rate (capacity) of this system is* $\bar{R} = 1$.

The proofs of Examples 3, 4, and 5 are relegated to Appendices E-A, E-B, and E-C, respectively. Unlike previous proofs for general M and F, which required a complete characterization of the vertices in the PRCR polytope and an achievability scheme for each of them, these examples seek to minimize the average rate for specific cache and file sizes, therefore each example has a unique optimal vertex. For the sake of simplicity[4], the proofs in Appendix E will only list the inequalities which are active at such vertex (labeled with the corresponding Instance from Section III-B) and a corresponding achievability scheme.

V. Conclusion

The per-request capacity region (PRCR) is the most fundamental performance metric in the information-theoretic studies of coded caching. Given the PRCR of a coded caching problem, we can find the optimal coded caching schemes for

[4] The PRCR polytope for the general $(N, K) = (3, 3)$ case has 27 dimensions and Instances 0-4 would provide over 933 inequalities.

any convex objective function. In this work, we have characterized the exact PRCR of the 2-user/2-file setting with full heterogeneity and used it to derive the average-rate capacity with heterogeneous demand popularity, file sizes, and cache sizes, and to re-derive the worst-case capacity previously found in [17]. By explicitly charactering the capacity and finding the capacity-achieving schemes, the results in this work allow the system designer to accurately evaluate the gain that optimal coded caching offers over naïve uncoded solutions under any general scenarios. The $N = K = 2$ results also represent the first step toward fully characterizing the average-rate/worst-case capacity of coded caching with full heterogeneity for general N and K values. We provide examples to show that the $N = K = 2$ results can be further generalized to derive PRCR results for some useful scenarios of $N = K = 3$.

Appendix A
Proof of Proposition 2

Section III-B shows that any achievable 4-dimensional rate vector $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ must satisfy the 28 inequalities for (O-1) to (IV-6), which can be succinctly summarized into the following two groups.

Group A: Bounds of a single variable, which combine Instances 0 to 2.

$$R_{(1,1)} \geq a_1 \triangleq \max(0, F_1 - M_1, F_1 - M_2) \tag{A1}$$

$$R_{(1,2)} \geq a_2 \triangleq \max(0, F_1 - M_1, F_2 - M_2, F_1 + F_2 - M_1 - M_2) \tag{A2}$$

$$R_{(2,1)} \geq a_3 \triangleq \max(0, F_2 - M_1, F_1 - M_2, F_1 + F_2 - M_1 - M_2) \tag{A3}$$

$$R_{(2,2)} \geq a_4 \triangleq \max(0, F_2 - M_1, F_2 - M_2) \tag{A4}$$

Group B: Bounds of two variables, which combine Instances 3 and 4.

$$R_{(1,1)} + R_{(1,2)} \geq b_1 \triangleq \max(F_1 + F_2 - M_2, 2F_1 + F_2 - M_1 - M_2) \tag{B1}$$

$$R_{(1,1)} + R_{(2,1)} \geq b_2 \triangleq \max(F_1 + F_2 - M_1, 2F_1 + F_2 - M_1 - M_2) \tag{B2}$$

$$R_{(1,1)} + R_{(2,2)} \geq b_3 \triangleq \max(F_1 + F_2 - M_1, F_1 + F_2 - M_2) \tag{B3}$$

$$R_{(1,2)} + R_{(2,1)} \geq b_4 \triangleq \max(F_1 + F_2 - M_1, F_1 + F_2 - M_2, 2F_1 + F_2 - M_1 - M_2, F_1 + 2F_2 - M_1 - M_2) \tag{B4}$$

$$R_{(1,2)} + R_{(2,2)} \geq b_5 \triangleq \max(F_1 + F_2 - M_1, F_1 + 2F_2 - M_1 - M_2) \tag{B5}$$

$$R_{(2,1)} + R_{(2,2)} \geq b_6 \triangleq \max(F_1 + F_2 - M_2, F_1 + 2F_2 - M_1 - M_2) \tag{B6}$$

Note that the values a_1 to a_4 and b_1 to b_6 are computed by evaluating the max operations in (A1) to (B6). For example, if $M_2 < M_1 < F_1$, then $a_1 = F_1 - M_2$ in (A1). However, if $F_1 < M_2 < M_1$, then $a_1 = 0$ in (A1). The key observation is that once we fix the (F_1, F_2, M_1, M_2) value, the 28 linear inequalities immediately collapse to 10 linear inequalities.

We now discuss some perquisite of the detailed proof.

Tight inequalities: There are 10 inequalities in (A1) to (B6). Each corner point in this 4-dimensional polytope must satisfy at least 4 of them with equalities and sometimes more. If an inequality is satisfied with equality, we say such an inequality is *tight*. Therefore, we need to have at least 4 tight inequalities. One main contribution of this proof is to analyze the relationship among these 10 inequalities for any arbitrary (F_1, F_2, M_1, F_2) so that we do not need to exhaustively examining all $\binom{10}{4}$ combinations for every (F_1, F_2, M_1, F_2). For simplicity we use the notation $\overline{(\cdot)}$ to represent an inequality being tight. For example, $\overline{\text{(A1)}}$ represents (A1) being tight. Another example is that the four equalities $\overline{\text{(A1)}}$, $\overline{\text{(A3)}}$, $\overline{\text{(B1)}}$, and $\overline{\text{(B6)}}$ jointly imply $R_{(1,1)} = a_1$, $R_{(1,2)} = b_1 - a_1$, $R_{(2,1)} = a_3$, and $R_{(2,2)} = b_6 - a_3$.

Global conditions: Without loss of generality, we assume implicitly the following conditions throughout Appendix A.

$$M_2 \geq 0 \quad \text{(G1)} \qquad F_2 \geq 0 \quad \text{(G2)}$$

$$M_1 \geq M_2 \quad \text{(G3)} \qquad F_1 \geq F_2 \quad \text{(G4)}$$

$$M_1 \leq F_1 + F_2 \quad \text{(G5)}$$

These technical assumptions are without loss of generality. Specifically, (G1) and (G2) ensure non-negativity; (G3) and (G4) always hold after swapping the user and file indices; and (G5) holds since there is no need to store more than the total file size $F_1 + F_2$. In the future, we refer these 5 inequalities as the global conditions $\mathcal{G}$:

$$\mathcal{G} \triangleq \{\text{(G1)}, \text{(G2)}, \text{(G3)}, \text{(G4)}, \text{(G5)}\}$$

Additional notation: For any set of (linear) inequalities $\mathcal{A}$, we use $\vec{\mathcal{R}}_{\mathcal{A}}$ to denote the set of $\vec{R}$ vectors that satisfy simultaneously *all* inequalities of $\mathcal{A}$. For any two sets of inequalities $\mathcal{A}$ and $\mathcal{B}$, we say $\mathcal{A}$ *implies* $\mathcal{B}$ if $\vec{\mathcal{R}}_{\mathcal{A}} \subseteq \vec{\mathcal{R}}_{\mathcal{B}}$. We use $\mathcal{A} \Rightarrow \mathcal{B}$ as shorthand.

We say the two sets of inequalities $\mathcal{A}$ and $\mathcal{B}$ are *equivalent*, denoted by $\mathcal{A} \Leftrightarrow \mathcal{B}$, if $\mathcal{A} \Rightarrow \mathcal{B}$ and $\mathcal{B} \Rightarrow \mathcal{A}$. Sometimes the equivalence and implication relationships hold only under some additional conditions $\mathcal{C}$. To that end, we use

$$\mathcal{A} \overset{\mathcal{C}}{\Rightarrow} \mathcal{B}$$

to represent $\mathcal{A}$ implies $\mathcal{B}$ under conditions[5] $\mathcal{C}$. Similarly, the notation $\mathcal{A} \overset{\mathcal{C}}{\Leftrightarrow} \mathcal{B}$ represents conditional equivalence under $\mathcal{C}$.

In the following, we only provide the proof for the vertices of Case 1 in Fig. 1, similar steps can be applied for proving the remaining Cases 2–5. See [33] for detailed steps.

Case 1: We assume

$$M_1 \leq F_2. \tag{c1}$$

Ineq. (c1) and $\mathcal{G}$ jointly describe the scenario when the (M_1, M_2) value falls into the lower-left triangle in Fig. 1 with

[5] A more rigorous notation of conditional implication should be $(\mathcal{A} \cup \mathcal{C}) \Rightarrow \mathcal{B}$. However, by writing $\mathcal{A} \overset{\mathcal{C}}{\Rightarrow} \mathcal{B}$ it is clearer what are the inequalities of interest (i.e., $\mathcal{A}$ and $\mathcal{B}$) and what are extra conditions being considered (i.e., $\mathcal{C}$).

solid edges and being marked as "Case 1". In this case, the a_1 to b_6 values of (A1) to (B6) become

$$\begin{aligned} &a_1 = F_1 - M_2, \quad a_2 = a_3 = F_1 + F_2 - M_1 - M_2, \\ &a_4 = F_2 - M_2, \quad b_1 = b_2 = b_4 = 2F_1 + F_2 - M_1 - M_2, \\ &b_3 = F_1 + F_2 - M_2, \quad b_5 = b_6 = F_1 + 2F_2 - M_1 - M_2. \end{aligned} \tag{c1.ab}$$

Using our previous notation and the definition of a_1 to b_6, the above statement can be summarized as $\{(\text{c1})\} \cup \mathcal{G} \Rightarrow \{(\text{c1.ab})\}$. We now further divide this case into two sub-cases. **Case 1.1:** We assume

$$M_1 + M_2 \leq F_2 \tag{c1.1}$$

and **Case 1.2:** We assume

$$M_1 + M_2 > F_2. \tag{c1.2}$$

Cases 1.1 and 1.2 further divide the solid lower-left triangle of Fig. 1 by a dotted line. In the following we focus on Case 1.1, the left sub-triangle.

Case 1.1: We consider the following 5 subcases.

Case 1.1.1 (A1) is tight. i.e., $\overline{(\text{A1})}$ holds. Under conditions $\mathcal{G}$, (c1) and (c1.1), we can prove the following relationship

$$\begin{aligned} &\{\overline{(\text{A1})}, (\text{B1}), (\text{B2}), (\text{B3})\} \overset{\mathcal{G},(\text{c1}),(\text{c1.1})}{\Longrightarrow} \\ &\{(\text{A2}), (\text{A3}), (\text{A4}), (\text{B4}), (\text{B5}), (\text{B6})\}. \end{aligned} \tag{15}$$

The above relationship is derived by first noting $\{(\text{c1})\} \cup \mathcal{G} \Rightarrow \{(\text{c1.ab})\}$ and by the following intermediate steps

$$\{\overline{(\text{A1})}, (\text{B1})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A2}) \tag{16}$$

$$\{\overline{(\text{A1})}, (\text{B2})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A3}) \tag{17}$$

$$\{\overline{(\text{A1})}, (\text{B3})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A4}) \tag{18}$$

$$\{\overline{(\text{A1})}, (\text{B1}), (\text{B2})\} \overset{(\text{c1.ab}),(\text{G1}),(\text{c1})}{\Longrightarrow} (\text{B4}) \tag{19}$$

$$\{\overline{(\text{A1})}, (\text{B1}), (\text{B3})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{B5}) \tag{20}$$

$$\{\overline{(\text{A1})}, (\text{B2}), (\text{B3})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{B6}). \tag{21}$$

Each intermediate step can be verified by straightforward algebraic operations. For example, part of (c1.ab) ensures that $a_1 = F_1 - M_2$, $a_2 = F_1 + F_2 - M_1 - M_2$, and $b_1 = 2F_1 + F_2 - M_1 - M_2$. Under these a_1, a_2, and b_1 values, $\overline{(\text{A1})}$, (A2), and (B1) become

$$R_{(1,1)} = F_1 - M_2 \tag{22}$$

$$R_{(1,2)} \geq F_1 + F_2 - M_1 - M_2 \tag{23}$$

$$R_{(1,1)} + R_{(1,2)} \geq 2F_1 + F_2 - M_1 - M_2 \tag{24}$$

Subtracting $\overline{(\text{A1})}$ (i.e., (22)) from (B1) (i.e., (24)), we have $R_{(1,2)} \geq F_1 + F_2 - M_1$ which implies (A2) (i.e., (23)) under condition (G1). We thus prove the intermediate step (16).

Similarly, (c1.ab) implies that (B2), and (B4) become

$$R_{(1,1)} + R_{(2,1)} \geq 2F_1 + F_2 - M_1 - M_2 \tag{25}$$

$$R_{(1,2)} + R_{(2,1)} \geq 2F_1 + F_2 - M_1 - M_2 \tag{26}$$

Adding up (B1) and (B2) (i.e., (24) and (25)) and subtracting $\overline{(\text{A1})}$ (i.e., (22)) twice, we have $R_{(1,2)} + R_{(2,1)} \geq 2F_1 + 2F_2 - 2M_1$ which implies (B4) (i.e., (26)), provided both (G1) and (c1) hold simultaneously. We have thus proven the intermediate step (19). Since the proofs of other intermediate steps (17), (18), (20), and (21) are very similar and straightforward, we omit their details.

By (15), the four *tight* linear inequalities in Case-1.1.1 can only be (A1) (thus $\overline{(\text{A1})}$), (B1), (B2), and (B3). Solving these equations, the corresponding corner point is *Vertex 1* $(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$ listed in Table III.

Case 1.1.2: (A2) is tight. i.e., $\overline{(\text{A2})}$ holds. We can then prove the following relationship

$$\begin{aligned} &\{\overline{(\text{A2})}, (\text{A3}), (\text{B1}), (\text{B5})\} \overset{\mathcal{G},(\text{c1}),(\text{c1.1})}{\Longrightarrow} \\ &\{(\text{A1}), (\text{A4}), (\text{B2}), (\text{B3}), (\text{B4}), (\text{B6})\}. \end{aligned} \tag{27}$$

The above relationship is derived by $\{(\text{c1})\} \cup \mathcal{G} \Rightarrow \{(\text{c1.ab})\}$ and by the following intermediate steps

$$\{\overline{(\text{A2})}, (\text{B1})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A1}) \tag{28}$$

$$\{\overline{(\text{A2})}, (\text{A3})\} \overset{(\text{c1.ab}),(\text{c1.1})}{\Longrightarrow} (\text{B4}) \tag{29}$$

$$\{\overline{(\text{A2})}, (\text{B5})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A4}) \tag{30}$$

$$\{\overline{(\text{A2})}, (\text{A3}), (\text{B1})\} \overset{(\text{c1.ab})}{\Longrightarrow} (\text{B2}) \tag{31}$$

$$\{\overline{(\text{A2})}, (\text{B1}), (\text{B5})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{B3}) \tag{32}$$

$$\{\overline{(\text{A2})}, (\text{A3}), (\text{B5})\} \overset{(\text{c1.ab})}{\Longrightarrow} (\text{B6}). \tag{33}$$

We omit the detailed proofs of the intermediate steps as they are extremely similar to the two examples discussed in the proof of Case 1.1.1.

By (27), the four tight linear inequalities in Case-1.1.2 can only be (A2) (thus $\overline{(\text{A2})}$), (A3), (B1), and (B5). Solving these equations, the corresponding corner point is *Vertex 2* $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ listed in Table III.

Case 1.1.3: (A3) is tight. i.e., $\overline{(\text{A3})}$ holds. We can then prove the following relationship

$$\begin{aligned} &\{\overline{(\text{A3})}, (\text{A2}), (\text{B2}), (\text{B6})\} \overset{\mathcal{G},(\text{c1}),(\text{c1.1})}{\Longrightarrow} \\ &\{(\text{A1}), (\text{A4}), (\text{B1}), (\text{B3}), (\text{B4}), (\text{B5})\} \end{aligned} \tag{34}$$

by the following straightforward intermediate steps

$$\{\overline{(\text{A3})}, (\text{B2})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A1}) \tag{35}$$

$$\{\overline{(\text{A3})}, (\text{A2})\} \overset{(\text{c1.ab}),(\text{c1.1})}{\Longrightarrow} (\text{B4}) \tag{36}$$

$$\{\overline{(\text{A3})}, (\text{B6})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{A4}) \tag{37}$$

$$\{\overline{(\text{A3})}, (\text{A2}), (\text{B2})\} \overset{(\text{c1.ab})}{\Longrightarrow} (\text{B1}) \tag{38}$$

$$\{\overline{(\text{A3})}, (\text{B2}), (\text{B6})\} \overset{(\text{c1.ab}),(\text{G1})}{\Longrightarrow} (\text{B3}) \tag{39}$$

$$\{\overline{(\text{A3})}, (\text{A2}), (\text{B6})\} \overset{(\text{c1.ab})}{\Longrightarrow} (\text{B5}). \tag{40}$$

By (34), the four tight linear inequalities in Case-1.1.3 can only be (A3) (thus $\overline{(\text{A3})}$), (A2), (B2), and (B6). Solving these equations, the corresponding corner point is Vertex 2 $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ listed in Table III.

Case 1.1.4: (A4) is tight. i.e., $\overline{(\text{A4})}$ holds. We can then prove the following relationship

$$\begin{aligned} &\{\overline{(\text{A4})}, (\text{B3}), (\text{B5}), (\text{B6})\} \overset{\mathcal{G},(\text{c1}),(\text{c1.1})}{\Longrightarrow} \\ &\{(\text{A1}), (\text{A2}), (\text{A3}), (\text{B1}), (\text{B2}), (\text{B4})\} \end{aligned} \tag{41}$$

by the following straightforward intermediate steps

$$\{\overline{(\mathrm{A4})}, (\mathrm{B3})\} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Longrightarrow} (\mathrm{A1}) \tag{42}$$

$$\{\overline{(\mathrm{A4})}, (\mathrm{B5})\} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Longrightarrow} (\mathrm{A2}) \tag{43}$$

$$\{\overline{(\mathrm{A4})}, (\mathrm{B6})\} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Longrightarrow} (\mathrm{A3}) \tag{44}$$

$$\{\overline{(\mathrm{A4})}, (\mathrm{B3}), (\mathrm{B5})\} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Longrightarrow} (\mathrm{B1}) \tag{45}$$

$$\{\overline{(\mathrm{A4})}, (\mathrm{B3}), (\mathrm{B6})\} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Longrightarrow} (\mathrm{B2}) \tag{46}$$

$$\{\overline{(\mathrm{A4})}, (\mathrm{B5}), (\mathrm{B6})\} \overset{(\mathrm{c1.ab}),(\mathrm{G1}),(\mathrm{c1})}{\Longrightarrow} (\mathrm{B4}). \tag{47}$$

By (41), the four tight linear inequalities in Case-1.1.4 can only be (A4) (thus $\overline{(\mathrm{A4})}$), (B3), (B5), and (B6). Solving these equations, the corresponding corner point is *Vertex 3* $(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$ listed in Table III.

Case 1.1.5: None of (A1) to (A4) is tight. Recall that in all the discussion of Case 1.1 and its subcases, we assume $\mathcal{G}$, (c1), (c1.1), and (c1.ab). Since

$$\{(\mathrm{A2}), (\mathrm{A3})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.1})}{\Longrightarrow} (\mathrm{B4}), \tag{48}$$

any corner point that is *loose* for all 4 inequalities (A1) to (A4) (and thus being loose for (A2) and (A3)) must also be loose for (B4). Therefore the corner point must be decided by 4 out of the 5 remaining inequalities (B1), (B2), (B3), (B5), and (B6). By (c1.ab), the inequalities corresponding to (B1), (B2), (B5), and (B6) become

$$R_{(1,1)} + R_{(1,2)} \geq b_1 = 2F_1 + F_2 - M_1 - M_2 \tag{49}$$

$$R_{(1,1)} + R_{(2,1)} \geq b_2 = 2F_1 + F_2 - M_1 - M_2 \tag{50}$$

$$R_{(1,2)} + R_{(2,2)} \geq b_5 = F_1 + 2F_2 - M_1 - M_2 \tag{51}$$

$$R_{(2,1)} + R_{(2,2)} \geq b_6 = F_1 + 2F_2 - M_1 - M_2 \tag{52}$$

We observe that the tight versions (equalities) of these inequalities are linearly dependent. As a result, any three of them being tight implies the fourth one is also tight. We hereby say that these 4 inequalities are *co-dependent*.

Since (B1), (B2), (B5), and (B6) are co-dependent and since a corner point requires 4 tight linearly *independent* inequalities, all 5 inequalities (B1), (B2), (B3), (B5), and (B6) must be tight simultaneously and jointly they yield exactly one corner point *Vertex 4* $(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$ listed in Table III.

The proof of Case 1.1.5 is completed by further proving that Vertex 4 is a legitimate corner point that satisfies (A1) to (A4) as well. The detailed verification steps are

$$\text{Vertex 4} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Rightarrow} (\mathrm{A1}); \quad \text{Vertex 4} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Rightarrow} (\mathrm{A2}); \tag{53}$$

$$\text{Vertex 4} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Rightarrow} (\mathrm{A3}); \quad \text{Vertex 4} \overset{(\mathrm{c1.ab}),(\mathrm{G1})}{\Rightarrow} (\mathrm{A4}). \tag{54}$$

Case 1.2: In this case we assume both (c1) and (c1.2) are true. This sub-case is the right sub-triangle above the dotted line in the solid lower-left triangle (Case 1) of Fig. 1. We now consider the following 6 subcases of Case 1.2.

Case 1.2.1: (A1) is tight, i.e., $\overline{(\mathrm{A1})}$ holds. Since the intermediate steps of Case 1.1.1, i.e., (16)-(21) does not require condition (c1.2), the statement in (15) holds even if we swap out (c1.1) by (c1.2). The rest of the analysis is verbatim to Case 1.1.1 and the corner point is also Vertex 1.

Case 1.2.2: (A2) is tight. i.e., $\overline{(\mathrm{A2})}$ holds. Note that we cannot reuse the derivation in Case 1.1.2 since (29) requires (c1.1) being true but in this case we only have (c1.2). That said, we can still prove the following relationship

$$\begin{aligned}&\{\overline{(\mathrm{A2})}, (\mathrm{B1}), (\mathrm{B4}), (\mathrm{B5})\} \overset{\mathcal{G},(\mathrm{c1}),(\mathrm{c1.2})}{\Longrightarrow} \\ &\{(\mathrm{A1}), (\mathrm{A3}), (\mathrm{A4}), (\mathrm{B2}), (\mathrm{B3}), (\mathrm{B6})\}\end{aligned} \tag{55}$$

by reusing (28), (30), (32), and the following straightforward intermediate steps

$$\{\overline{(\mathrm{A2})}, (\mathrm{B4})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.2})}{\Longrightarrow} (\mathrm{A3}) \tag{56}$$

$$\{\overline{(\mathrm{A2})}, (\mathrm{B1}), (\mathrm{B4})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.2})}{\Longrightarrow} (\mathrm{B2}) \tag{57}$$

$$\{\overline{(\mathrm{A2})}, (\mathrm{B4}), (\mathrm{B5})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.2})}{\Longrightarrow} (\mathrm{B6}). \tag{58}$$

By (55), the four tight linear inequalities in Case-1.2.2 can only be (A2) (thus $\overline{(\mathrm{A2})}$), (B1), (B4), and (B5). Solving these equations, the corresponding corner point is Vertex 5 $(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$ listed in Table III.

Case 1.2.3: (A3) is tight. i.e., $\overline{(\mathrm{A3})}$ holds. We can then prove the following relationship

$$\begin{aligned}&\{\overline{(\mathrm{A3})}, (\mathrm{B2}), (\mathrm{B4}), (\mathrm{B6})\} \overset{\mathcal{G},(\mathrm{c1}),(\mathrm{c1.2})}{\Longrightarrow} \\ &\{(\mathrm{A1}), (\mathrm{A2}), (\mathrm{A4}), (\mathrm{B1}), (\mathrm{B3}), (\mathrm{B5})\}\end{aligned} \tag{59}$$

by reusing (35), (37), (39), and the following straightforward intermediate steps

$$\{\overline{(\mathrm{A3})}, (\mathrm{B4})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.2})}{\Longrightarrow} (\mathrm{A2}) \tag{60}$$

$$\{\overline{(\mathrm{A3})}, (\mathrm{B2}), (\mathrm{B4})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.2})}{\Longrightarrow} (\mathrm{B1}) \tag{61}$$

$$\{\overline{(\mathrm{A3})}, (\mathrm{B4}), (\mathrm{B6})\} \overset{(\mathrm{c1.ab}),(\mathrm{c1.2})}{\Longrightarrow} (\mathrm{B5}). \tag{62}$$

By (59), the four tight linear inequalities in Case-1.2.3 can only be (A3) (thus $\overline{(\mathrm{A3})}$), (B2), (B4), and (B6). Solving these equations, the corresponding corner point is Vertex 6 $(F_1, F_1, F_1 + F_2 - M_1 - M_2, F_2)$ listed in Table III.

Case 1.2.4: (A4) is tight, i.e., $\overline{(\mathrm{A4})}$ holds. Since the intermediate steps of Case 1.1.4, i.e., (42)-(47) does not require condition (c1.2), the statement in (41) holds even if we swap out (c1.1) by (c1.2). The rest of the analysis is verbatim to Case 1.1.4 and the corner point is also Vertex 3.

Case 1.2.5: None of (A1) to (A4) is tight, but (B4) is tight, i.e., $\overline{(\mathrm{B4})}$ holds. We can prove

$$\{(\mathrm{B1}), (\mathrm{B2}), (\mathrm{B5}), (\mathrm{B6}), \overline{(\mathrm{B4})}\} \overset{\mathcal{G},(\mathrm{c1}),(\mathrm{c1.2})}{\Rightarrow} \{(\mathrm{B3})\} \tag{63}$$

using the following intermediate step

$$\{(\mathrm{B1}), (\mathrm{B6}), \overline{(\mathrm{B4})}\} \overset{(\mathrm{c1.ab}),(\mathrm{c1})}{\Rightarrow} \{(\mathrm{B3})\}. \tag{64}$$

The statement (63) implies that when none of (A1) to (A4) is tight but $\overline{(\mathrm{B4})}$ holds, the corner point is decided solely by the inequalities (B1), (B2), (B5), and (B6) and we do not need to check whether (B3) is tight or not.

Recall that under the a_1 to b_6 values in (c1.ab), the inequalities corresponding to (B1), (B2), (B5), and (B6) are *co-dependent* as shown in Case 1.1.5. Therefore, the statement (63) further implies that the corner point must be tight for all 5 inequalities (B1), (B2), (B5), (B6), $\overline{(\mathrm{B4})}$. Solving these 5 joint equations (four of them are codependent), we obtain the corner point *Vertex 7* $(F_1+\frac{F_2-M_1-M_2}{2}, F_1+\frac{F_2-M_1-M_2}{2}, F_1+\frac{F_2-M_1-M_2}{2}, F_2+\frac{F_2-M_1-M_2}{2})$ listed in Table III.

The proof of Case 1.2.5 is completed by further proving that Vertex 7 is a legitimate corner point that satisfies (A1) to (A4) as well. The detailed verification steps are

$$\text{Vertex 7} \overset{\text{(c1.ab),(G1),(c1)}}{\Rightarrow} \text{(A1)}; \quad \text{Vertex 7} \overset{\text{(c1.ab),(c1.2)}}{\Rightarrow} \text{(A2)}; \quad (65)$$

$$\text{Vertex 7} \overset{\text{(c1.ab),(c1.2)}}{\Rightarrow} \text{(A3)}; \quad \text{Vertex 7} \overset{\text{(c1.ab),(G1),(c1)}}{\Rightarrow} \text{(A4)}. \quad (66)$$

Case 1.2.6: None of (A1) to (A4) is tight, nor is (B4). If we retrace the proof of Case 1.1.5, we notice that (48) ensures that when in Case 1.1.5, we always have (B4) being loose. Since (48) holds only under (c1.1), ineq. (B4) can be tight or loose in Case 1.2. That is why in Case 1.2.5, we discussed the case when (B4) is tight and in this case we assume (B4) is loose. Since the arguments in Case 1.1.5 after (48) no longer uses the condition (c1.1), we can use the same argument verbatim and prove that the corner point in Case 1.2.6 is the Vertex 4 $(F_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_2-\frac{M_2}{2})$ listed in Table III.

Appendix B
Proof of Proposition 3

In the following, we will prove that each of the 7 vertices in Case 1 can be achieved by space sharing among the 7 basic achievable schemes listed in Table II. The space sharing schemes with the 7 basic achievable schemes for the rest 21 vertices (Vertices 8-28) are provided in [33].

Vertex 1: As summarized in Fig. 1, Vertex-1 rate vector $(F_1-M_2, F_1+F_2-M_1, F_1+F_2-M_1, F_2)$ is the corner point for 4 out of 11 sub-regions. Table IV describes an achievable scheme that attains Vertex-1 rate vector as long as the following Applicable Range (AR) holds

$$\text{[AR]:} \quad M_2 \leq M_1 \leq \min(F_1, F_2+M_2), \qquad (67)$$

which is the union of the 4 desired sub-regions.

Each row of Table IV describes one basic scheme and the last row describes the total (combined) effect after space sharing. Each basic scheme takes parts of files 1 and 2 and stores coded data in parts of memories 1 and 2. The columns of f_1, f_2, m_1, and m_2 correspond to the amount of files 1 and 2 and memories 1 and 2 of each basic scheme. The columns of $R_{(1,1)}$, $R_{(1,2)}$, $R_{(2,1)}$, and $R_{(2,2)}$ correspond to the rates contributed by each basic scheme under each request pattern. The intersection of the "Total" row and the four rate columns thus represents the achievable rate vector of the overall scheme.

To ensure that Table IV indeed describes a legitimate scheme, one needs to verify the following three conditions:

1) *All the file sizes and the memory sizes are non-negative.* For example, 1.1.Cov uses $F_1-M_1+M_2$ of file 1 and $F_2-M_1+M_2$ of file 2 to encode. The AR (67) ensures that both sub-file sizes are non-negative.
2) *For any given row, the assigned subfile sizes and the assigned memory sizes satisfy the required condition of the basic scheme listed in Table II.* For example, as summarized in Table II the 1.1.Cov scheme requires that $\max(m_1,m_2)\leq f_1$. As a result, in the row of 1.1.Cov in Table IV we must satisfy $\max(M_2,M_2)\leq F_1-M_1+M_2$, which is ensured by the AR (67).
3) *For any given row, the delivery rate vector is computed correctly according to Table II.* For example, as summarized in Table II the 1.1.Cov scheme achieves $R_{(1,1)} = f_1-\min(m_1,m_2)$. As a result, correct rate computation in the row of 1.1.Cov of Table IV requires the equality $F_1-M_1 = (F_1-M_1+M_2)-\min(M_2,M_2)$ to hold, which is ensured by[6] the AR (67).

Verifying these three statements is very straightforward and we thus omit the details here.

Remark: The main reason that the Vertex-1 scheme in Table IV is accompanied by an AR condition (67) is to ensure that the above three conditions about file/memory sizes and rate computation are properly met.

Vertex 2: As summarized in Fig. 1, Vertex-2 rate vector $(F_1, F_1+F_2-M_1-M_2, F_1+F_2-M_1-M_2, F_2)$ is the corner point of 1 out of 11 sub-regions. Table V describes the corresponding space-sharing scheme that attains Vertex-2 rate vector. One can easily verify that the applicable range listed in Table V, i.e. $M_1+M_2\leq\min(F_1,F_2)$, completely covers the desired sub-region. Furthermore, the applicable range ensures that the three conditions on the file/memory sizes and rate computation are met, also see the discussion of Vertex 1. Since the verification step is straightforward, we omit the details. The proof of Vertex-2 achievability is thus complete.

Remark: As will be seen later, each of our schemes, described in the corresponding table, is associated with an applicable range. To prove the legitimacy of the scheme, we always have to check (i) the applicable range covers the sub-regions of the corresponding corner point; and (ii) the applicable range ensures that the three conditions on the file/memory sizes and rate computation are met. Since checking (i) and (ii) can all be verified very easily, we will not repeatedly emphasize these important verification steps in the sequel. Instead we only describe the schemes and the corresponding applicable ranges.

Vertex 3: Per Fig. 1, Vertex-3 rate vector $(F_1, F_1+F_2-M_1, F_1+F_2-M_1, F_2-M_2)$ is the corner point of 2 out of 11 sub-regions. Table VI describes an achievable scheme that attains Vertex-3 rate vector and its applicable range.

Vertex 4: Per Fig. 1, Vertex-4 rate vector $(F_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_1+F_2-M_1-\frac{M_2}{2}, F_2-\frac{M_2}{2})$ is the corner point of 2 out of 11 sub-regions. Table VII describes an achievable scheme that attains Vertex-4 rate vector and its applicable range.

Vertex 5: Per Fig. 1, Vertex-5 rate vector $(F_1, F_1+F_2-M_1-M_2, F_1, F_2)$ is the corner point of 1 out of 11 sub-regions. Table VIII describes an achievable scheme that attains Vertex-5 rate vector and its applicable range.

Vertex 6: Per Fig. 1, Vertex-6 rate vector $(F_1, F_1, F_1+F_2-M_1-M_2, F_2)$ is the corner point of 1 out of 11 sub-regions.

[6] In this example, it is trivially true regardless whether the AR condition (67) holds or not.

TABLE IV
VERTEX 1 $(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$ WITH APPLICABLE RANGE: $M_2 \le M_1 \le \min(F_1, F_2 + M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$	0	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$
1.1.Cov	$F_1{-}M_1{+}M_2$	$F_2{-}M_1{+}M_2$	M_2	M_2	$F_1{-}M_1$	$F_1{+}F_2{-}2M_1{+}M_2$	$F_1{+}F_2{-}2M_1{+}M_2$	$F_2{-}M_1{+}M_2$
Total	F_1	F_2	M_1	M_2	$F_1{-}M_2$	$F_1{+}F_2{-}M_1$	$F_1{+}F_2{-}M_1$	F_2

TABLE V
VERTEX 2 $(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$ WITH APPLICABLE RANGE: $M_1 + M_2 \le \min(F_1, F_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	M_1	M_1	M_1	0	M_1	M_1	M_1	M_1
Emp.Mix	M_2	M_2	0	M_2	M_2	M_2	M_2	M_2
1.1.Cov	$F_1{-}M_1{-}M_2$	$F_2{-}M_1{-}M_2$	0	0	$F_1{-}M_1{-}M_2$	$F_1{+}F_2{-}2M_1{-}2M_2$	$F_1{+}F_2{-}2M_1{-}2M_2$	$F_2{-}M_1{-}M_2$
Total	F_1	F_2	M_1	M_2	F_1	$F_1{+}F_2{-}M_1{-}M_2$	$F_1{+}F_2{-}M_1{-}M_2$	F_2

TABLE VI
VERTEX 3 $(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$ WITH APPLICABLE RANGE: $M_2 \le M_1 \le \min(F_1 + M_2, F_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$	0	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$
2.2.Cov	$F_1{-}M_1{+}M_2$	$F_2{-}M_1{+}M_2$	M_2	M_2	$F_1{-}M_1{+}M_2$	$F_1{+}F_2{-}2M_1{+}M_2$	$F_1{+}F_2{-}2M_1{+}M_2$	$F_2{-}M_1$
Total	F_1	F_2	M_1	M_2	F_1	$F_1{+}F_2{-}M_1$	$F_1{+}F_2{-}M_1$	$F_2{-}M_2$

TABLE VII
VERTEX 4 $(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$ WITH APPLICABLE RANGE: $M_2 \le M_1 \le \min(F_1, F_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$	0	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$	$M_1{-}M_2$
Ha.Fi	M_2	M_2	M_2	M_2	$M_2/2$	$M_2/2$	$M_2/2$	$M_2/2$
1.1.Cov	$F_1{-}M_1$	$F_2{-}M_1$	0	0	$F_1{-}M_1$	$F_1{+}F_2{-}2M_1$	$F_1{+}F_2{-}2M_1$	$F_2{-}M_1$
Total	F_1	F_2	M_1	M_2	$F_1{-}\frac{1}{2}M_2$	$F_1{+}F_2{-}M_1{-}\frac{1}{2}M_2$	$F_1{+}F_2{-}M_1{-}\frac{1}{2}M_2$	$F_2{-}\frac{1}{2}M_2$

Table IX describes an achievable scheme that attains Vertex-6 rate vector and its applicable range.

Vertex 7: Per Fig. 1, Vertex-7 rate vector $(F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_1 + \frac{F_2-M_1-M_2}{2}, F_2 + \frac{F_2-M_1-M_2}{2})$ is the corner point of 1 out of 11 sub-regions. Table X describes an achievable scheme that attains Vertex-7 rate vector and its applicable range.

APPENDIX C
PROOF OF COROLLARY 2

The uniform average rate capacity $\tilde{R} = \frac{1}{4}(R_{(1,1)} + R_{(1,2)} + R_{(2,1)} + R_{(2,2)})$ is a special case of the average rate with popularity $p_{\vec{d}} = 0.25$ for all $\vec{d} \in \{1, 2\}$. In the following, we use Proposition 1 to derive the closed-form expression of the uniform average rate capacity as in Fig. 6.

We combine (B1) and (B6) to obtain

$$\tilde{R} \ge \frac{1}{4}\max\{2F_1 + 2F_2 - 2M_2, 3F_1 + 2F_2 - M_1 - 2M_2, \\ 2F_1 + 3F_2 - M_1 - 2M_2, 3F_1 + 3F_2 - 2M_1 - 2M_2\} \tag{68}$$

and combine (B2) and (B5) to obtain

$$\tilde{R} \ge \frac{1}{4}\max\{2F_1 + 2F_2 - 2M_1, 3F_1 + 2F_2 - 2M_1 - M_2, \\ 2F_1 + 3F_2 - 2M_1 - M_2, 3F_1 + 3F_2 - 2M_1 - 2M_2\}. \tag{69}$$

Therefore, the uniform average rate $\tilde{R}$ must simultaneously satisfy (68) and (69), which can be expanded as a set of 7 inequalities[7] as follows.

$$\tilde{R} \ge \frac{F_1 + F_2 - M_1}{2} \tag{P1}$$

$$\tilde{R} \ge \frac{F_1 + F_2 - M_2}{2} \tag{P2}$$

$$\tilde{R} \ge \frac{3F_1 + 2F_2 - 2M_1 - M_2}{4} \tag{P3}$$

$$\tilde{R} \ge \frac{2F_1 + 3F_2 - 2M_1 - M_2}{4} \tag{P4}$$

$$\tilde{R} \ge \frac{3F_1 + 2F_2 - M_1 - 2M_2}{4} \tag{P5}$$

$$\tilde{R} \ge \frac{2F_1 + 3F_2 - M_1 - 2M_2}{4} \tag{P6}$$

$$\tilde{R} \ge \frac{3F_1 + 3F_2 - 2M_1 - 2M_2}{4}. \tag{P7}$$

Now we show that the set of 7 inequalities is sufficient to describe the uniform average rate capacity. Note that the joint 7 inequalities (P1) to (P7) are symmetric with respect to the file and user indices; therefore without loss of generality, we assume $F_1 \ge F_2$ and $M_1 \ge M_2$. Under this assumption, (P1), (P3), (P4) and (P6) are always loose and the remaining 3 inequalities (P2), (P5), and (P7) jointly form the lower-triangular region ($M_1 \ge M_2$) in Fig. 6.

[7] When assuming $F_1 \ge F_2$, (P4) and (P6) are always loose. The remaining 5 inequalities are indeed the 5 facets described in Fig. 6.

TABLE VIII
VERTEX 5 $(F_1, F_1+F_2-M_1-M_2, F_1, F_2)$ WITH APPLICABLE RANGE: $\max(M_1, M_2) \le F_2 \le \min(F_1, M_1+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Emp.Mix	F_2-M_1	F_2-M_1	0	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1
1.2.Cov	F_1-2F_2 $+M_1+M_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	F_1-2F_2 $+M_1+M_2$	F_1-F_2	F_1-2F_2 $+M_1+M_2$	$M_1+M_2-F_2$
Total	F_1	F_2	M_1	M_2	F_1	$F_1+F_2-M_1-M_2$	F_1	F_2

TABLE IX
VERTEX 6 $(F_1, F_1, F_1+F_2-M_1-M_2, F_2)$ WITH APPLICABLE RANGE: $\max(M_1, M_2) \le F_2 \le \min(F_1, M_1+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Emp.Mix	F_2-M_1	F_2-M_1	0	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1
2.1.Cov	F_1-2F_2 $+M_1+M_2$	M_1+M_2 $-F_2$	M_1+M_2 $-F_2$	M_1+M_2 $-F_2$	F_1-2F_2 $+M_1+M_2$	F_1-2F_2 $+M_1+M_2$	F_1-F_2	M_1+M_2 $-F_2$
Total	F_1	F_2	M_1	M_2	F_1	F_1	$F_1+F_2-M_1-M_2$	F_2

TABLE X
VERTEX 7 $(F_1+\frac{1}{2}(F_2-M_1-M_2), F_1+\frac{1}{2}(F_2-M_1-M_2), F_1+\frac{1}{2}(F_2-M_1-M_2), F_2+\frac{1}{2}(F_2-M_1-M_2))$ WITH APPLICABLE RANGE: $\max(M_1, M_2) \le F_2 \le \min(F_1, M_1+M_2)$.

Scheme	f_1	f_2	m_1	m_2	$R_{(1,1)}$	$R_{(1,2)}$	$R_{(2,1)}$	$R_{(2,2)}$
Mix.Emp	F_2-M_2	F_2-M_2	F_2-M_2	0	F_2-M_2	F_2-M_2	F_2-M_2	F_2-M_2
Emp.Mix	F_2-M_1	F_2-M_1	0	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1	F_2-M_1
Ha.Fi	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$M_1+M_2-F_2$	$\frac{M_1+M_2-F_2}{2}$	$\frac{M_1+M_2-F_2}{2}$	$\frac{M_1+M_2-F_2}{2}$	$\frac{M_1+M_2-F_2}{2}$
1.1.Cov	F_1-F_2	0	0	0	F_1-F_2	F_1-F_2	F_1-F_2	0
Total	F_1	F_2	M_1	M_2	F_1+ $\frac{F_2-M_1-M_2}{2}$	F_1+ $\frac{F_2-M_1-M_2}{2}$	F_1+ $\frac{F_2-M_1-M_2}{2}$	F_2+ $\frac{F_2-M_1-M_2}{2}$

We now provide the proof of the achievability part. If we describe each corner point of the triangular region of $M_1 \ge M_2$ by the corresponding tuple $(M_1, M_2, \bar{R})$, then there are 7 vertices in the region, and they are

$$\begin{aligned} &v_1 = (0, 0, \frac{3F_1+3F_2}{4}),\ v_2 = (F_2, 0, \frac{3F_1+F_2}{4}),\\ &v_3 = (F_2, F_2, \frac{3F_1-F_2}{4}),\ v_4 = (F_1, 0, \frac{F_1+F_2}{2}),\\ &v_5 = (F_1, F_1, \frac{F_2}{2}),\ v_6 = (F_1+F_2, 0, \frac{F_1+F_2}{2}),\\ &\text{and } v_7 = (F_1+F_2, F_1+F_2, 0). \end{aligned} \tag{70}$$

The vertices v_1 and v_7 can be achieved by some trivial schemes. The rest of them, v_2, v_3, v_4, v_5, and v_6, can be achieved by Vertices 1, 4, 8, 23, and 14 described in Table III.

APPENDIX D
RE-DERIVATION OF WORST-CASE RATE CAPACITY IN [17]

Corollary 3. *The 2-user/2-file zero-error worst-case capacity is characterized by the following 9 inequalities:*

$$R^* \ge \frac{F_1+F_2-M_1}{2} \tag{Q1}$$

$$R^* \ge \frac{F_1+F_2-M_2}{2} \tag{Q2}$$

$$R^* \ge F_1 - M_1 \tag{Q3}$$

$$R^* \ge F_1 - M_2 \tag{Q4}$$

$$R^* \ge F_2 - M_1 \tag{Q5}$$

$$R^* \ge F_2 - M_2 \tag{Q6}$$

$$R^* \ge \frac{2F_1+F_2-M_1-M_2}{2} \tag{Q7}$$

$$R^* \ge \frac{F_1+2F_2-M_1-M_2}{2} \tag{Q8}$$

$$R^* \ge F_1+F_2-M_1-M_2. \tag{Q9}$$

If we further assume $F_1 \ge F_2$, then (Q5), (Q6), *and* (Q8) *are always loose. The corresponding worst-case capacity R^* is described by Fig. 7, which consists of 11 vertices and the 6 planes* (Q1), (Q2), (Q3), (Q4), (Q7), *and* (Q9).

Recall that the worst-case rate objective (4) is convex and the PRCR is characterized by 28 linear constraints (O-1) to (IV-6). The problem of minimizing R^* subject to PRCR is therefore a convex optimization problem. We solve the convex optimization problem by first converting to the equivalent linear programming problem as follows.

$$\begin{aligned} \min_{\vec{R}}\ & R^* \\ \text{s.t. } & R^* \ge R_{(1,1)} && (71)\\ & R^* \ge R_{(1,2)} && (72)\\ & R^* \ge R_{(2,1)} && (73)\\ & R^* \ge R_{(2,2)} && (74)\\ & \text{(A1) to (B6).} \end{aligned}$$

By plugging in (A1) to (B6), the linear programming problem

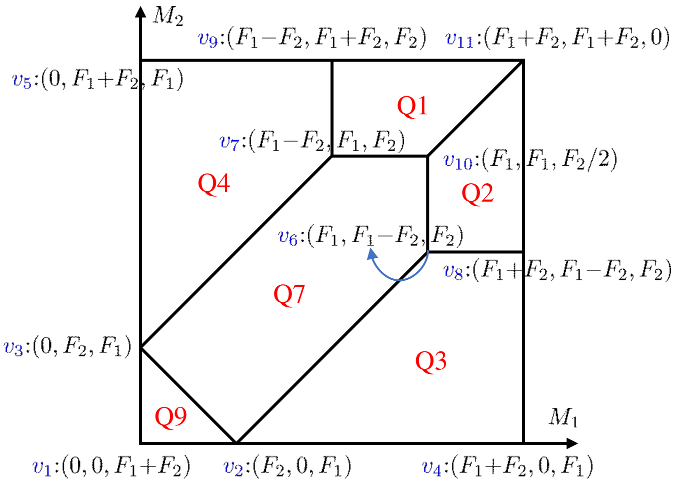

M_2
v_9:(F_1-F_2, F_1+F_2, F_2)
v_{11}:$(F_1+F_2, F_1+F_2, 0)$
v_5:$(0, F_1+F_2, F_1)$
Q1
v_7:(F_1-F_2, F_1, F_2)
v_{10}:$(F_1, F_1, F_2/2)$
Q4
Q2
v_6:(F_1, F_1-F_2, F_2)
v_8:(F_1+F_2, F_1-F_2, F_2)
Q7
v_3:$(0, F_2, F_1)$
Q3
Q9
M_1
v_1:$(0, 0, F_1+F_2)$
v_2:$(F_2, 0, F_1)$
v_4:$(F_1+F_2, 0, F_1)$

Fig. 7. The capacity of worst-case rate under the assumption $F_1 \geq F_2$. Each corner point is labeled by a tuple (M_1, M_2, R^*), where (M_1, M_2) describe the location and the third coordinate describe the corresponding exact worst-case rate capacity R^*.

can be further converted to the following equivalent form

$$\begin{aligned} &\min_{\vec{R}} R^* \\ &\text{s.t. } R^* \geq a_i, \quad i \in \{1,2,3,4\} \qquad (75) \\ &\qquad 2R^* \geq b_j, \quad j \in \{1,2,3,4,5,6\}. \qquad (76) \end{aligned}$$

Note that the values of a_1 to b_6 are calculated by max operations and we can expand each $\geq$ inequality with max operation on the right-hand side to multiple inequalities. If we expand the max operation in (75), we will have (Q3) to (Q6), and if we expand the max operation in (76), we will have (Q1), (Q2), (Q7) to (Q9). The converse proof is thus complete.

For the achievability part, without loss of generality, we only consider the corner points in the region $F_1 \geq F_2$ and $M_1 \geq M_2$, and there are 7 vertices v_1, v_2, v_4, v_6, v_8, v_{10}, and v_{11} as shown in Fig. 7. The vertices v_1 and v_{11} can be achieved by some trivial schemes. The rest of them, v_2, v_4, v_6, v_8, and v_{10}, can be achieved by Vertices 1, 14, 1, 14, and 23 described in Table III.

Since the worst-case objective (4) is not a linear function of rates $R_{\vec{d}}$, $\vec{d} \in [N]^K$, the facets of the worst-case rate region may not match the facets of PRCR. For example, it can be observed that the edge (v_6, v_8) in Fig. 7 is a new edge that did not appear in Fig. 1.

Appendix E
Proofs of 3-user/3-file Examples

When $(N, K) = (3, 3)$, the five families of linear inequalities (Instances 0-4) described in Section III-B could be extended to include over 933 lower bounds on the PRCR polytope: 27 Instance-0 inequalities, 81 Instance-1 inequalities, 24 Instance-2 inequalities, 729 Instance-3 inequalities, and over 72 Instance-4 inequalities. For the sake of simplicity, the following proofs only list a minimal subset of them, appropriately labeled with their instance number (Inst. 0-4).

TABLE XI
Transmitted messages $X_{\vec{d}}$ for each possible demand instance $\vec{d}$. These achievable rates match the obtained lower bounds in Example 3, hence they correspond to the per-request capacity.

Request $\vec{d}$	Transmitted messages $X_{\vec{d}}$	Rate $R_{\vec{d}}$
(1,2,3)	$\emptyset$	0
(1,3,2)	$(u_3,\ v_1 \oplus u_1,\ v_2 \oplus u_2)$	3
(2,1,3)	$(v_2,\ w_1 \oplus v_1)$	2
(2,3,1)	$(u_2,\ u_3,\ v_2,\ v_1 \oplus u_1,\ v_1 \oplus w_1)$	5
(3,1,2)	$(u_2,\ u_3,\ v_2,\ v_1 \oplus u_1,\ u_1 \oplus w_1)$	5
(3,2,1)	$(u_2,\ u_3,\ w_1 \oplus u_1)$	3

A. Proof of Example 3

For the given $(M_1, M_2, M_3, F_1, F_2, F_3)$ and $p_{\vec{d}}$ values in this example, with the aid of a computer we find only 6 inequalities among those in the extended Instances 0-4 are needed/active when solving the LP problem of minimizing the rate $\bar{R}$:

$$\min \sum_{\vec{d}} p_{\vec{d}} R_{\vec{d}}$$

s.t.

$$\begin{aligned} &R_{(1,2,3)} \geq 0 && \text{(Inst. 0)} \\ &R_{(1,2,3)} + R_{(1,3,2)} + M_2 \geq F_2 + F_3 && \text{(Inst. 3)} \\ &R_{(1,2,3)} + R_{(2,1,3)} + M_1 \geq F_1 + F_2 && \text{(Inst. 3)} \\ &R_{(1,2,3)} + R_{(3,2,1)} + M_1 \geq F_1 + F_3 && \text{(Inst. 3)} \\ &R_{(1,2,3)} + R_{(2,3,1)} + M_1 + M_2 \geq F_1 + 2F_2 + F_3 && \text{(Inst. 4)} \\ &R_{(1,2,3)} + R_{(3,1,2)} + M_1 + M_3 \geq F_1 + F_2 + 2F_3. && \text{(Inst. 4)} \end{aligned}$$

Such solution is given by the vertex

$$\begin{aligned} &R_{(1,2,3)} = 0, \quad &&R_{(2,1,3)} = 2, \quad &&R_{(3,1,2)} = 5, \\ &R_{(1,3,2)} = 3, \quad &&R_{(2,3,1)} = 5, \quad &&R_{(3,2,1)} = 3, \end{aligned}$$

resulting in $\bar{R} = \frac{25}{16}$. This is an information-theoretical lower bound for the achievable average rate.

We now proceed to show that this lower bound can be achieved and is therefore tight. Divide each file into subfiles of unit size, $w_2 = (v_1, v_2)$ and $w_3 = (u_1, u_2, u_3)$, and let each user cache the corresponding file:

$$Z_1 = \{w_1\}, \quad Z_2 = \{v_1, v_2\}, \quad Z_3 = \{u_1, u_2, u_3\}.$$

It can be shown that the transmitted messages $X_{\vec{d}}$ in Table XI allow the server to fulfill all requests and yield an average rate of $\bar{R} = \frac{25}{16}$. Since the average rate with this scheme matches the above lower bound, we can conclude that $\bar{R} = \frac{25}{16}$ is the minimum achievable average-rate of this system.

Furthermore, the above scheme corresponds to a simple extension of the basic coded caching schemes described in Section III-A, which we can call 1.2.3.Cov: In the placement phase, for $k = 1, 2, 3$, user k's strategy is "to cover as much file k as possible". In the delivery phase, their demands can be fulfilled by the transmission messages in ways similar to the a.b.Cov schemes for 2-user/2-file. When at least one of the users demands the same file that it is already caching (i.e., demand vectors $(1, i, j)$, $(i, 2, j)$ or $(i, j, 3)$) the transmitted messages in Table XI can be one-to-one mapped to our 2-user/2-file setting for demand (i, j) in a.b.Cov schemes, where a and b are the corresponding user indexes.

When all three users request a file different from the one that they are caching (i.e., demands $(2,3,1)$ and $(3,1,2)$) their demands can be fulfilled by concatenating the transmission messages for demands in two a.b.Cov schemes. Take the demand $(2,3,1)$ for example, we first focus on users 1 and 2 to transmit the message $(v_2,\, w_1\oplus v_1)$ for demand $(2,1)$ in 1.2.Cov scheme despite user 2 actually demands file 3 instead of file 1. This provides user 1 with the desired file 2 but user 2 receives the undesired file 1; then we focus on users 2 and 3 transmitting the message $(u_2,\, u_3,\, v_1\oplus w_1)$ for demand $(3,1)$ in 1.3.Cov scheme as if user 2 already stored file 1. Finally, both users 2 and 3 get their desired files. Similarly, the transmission message for demand $(3,1,2)$ can be obtained by concatenating the message for demand $(2,1)$ in 1.2.Cov and the message for demand $(3,2)$ in 2.3.Cov.

B. Proof of Example 4

This proof follows a parallel argument to that in Appendix E-A, solving the LP resulting from the inequalities in Section III-B (only the relevant ones are listed) to obtain a lower bound and then showing that the bound is achievable. Noted that in this example, the naïve extension of Instance 0-4 is not sufficient to obtain a tight lower bound, therefore we further introduce Instance 4'. The LP problem

$$\min \sum_{\vec{d}} p_{\vec{d}} R_{\vec{d}}$$

s.t.

$$\begin{aligned}
&R_{(3,2,1)} + M_1 + M_2 \geq F_3 + F_2 && \text{(Inst. 2)}\\
&R_{(1,2,3)} + R_{(2,3,1)} + M_1 + M_2 \geq F_1 + 2F_2 + F_3 && \text{(Inst. 4)}\\
&R_{(1,3,2)} + R_{(3,2,1)} + M_1 + M_2 \geq F_1 + F_2 + 2F_3 && \text{(Inst. 4)}\\
&R_{(2,1,3)} + R_{(3,2,1)} + M_1 + M_2 \geq F_1 + 2F_2 + F_3 && \text{(Inst. 4)}\\
&R_{(2,3,1)} + R_{(3,1,2)} + M_1 + M_2 \geq F_1 + F_2 + 2F_3 && \text{(Inst. 4)}\\
&R_{(3,1,2)} + R_{(3,2,1)} + 2M_1 + M_2 \geq F_1 + F_2 + 2F_3 && \text{(Inst. 4')}
\end{aligned}$$

yields

$$\begin{aligned}
&R_{(1,2,3)} = 2, \quad && R_{(2,1,3)} = 3, \quad && R_{(3,1,2)} = 3,\\
&R_{(1,3,2)} = 4, && R_{(2,3,1)} = 3, && R_{(3,2,1)} = 2,
\end{aligned}$$

with $\bar{R} = \frac{5}{2}$. The last inequality is labeled Instance 4' because it does not correspond to a strict application of Instance 4 in Section III-B. Its derivation follows a similar reasoning, but they are not strictly identical. Specifically

$$\begin{aligned}
&R_{(3,1,2)} + R_{(3,2,1)} + 2M_1 + M_2 && (77)\\
&\geq H(X_{(3,1,2)}) + 2H(Z_1) + H(X_{(3,2,1)}) + H(Z_2) && (78)\\
&\geq H(X_{(3,1,2)}, Z_1) + H(X_{(3,2,1)}, Z_1, Z_2) && (79)\\
&\geq H(X_{(3,1,2)}, Z_1, W_3) + H(X_{(3,2,1)}, Z_1, Z_2, W_2, W_3) && (80)\\
&\geq H(X_{(3,1,2)}, X_{(3,2,1)}, Z_1, Z_2, W_2, W_3) + H(W_3) && (81)\\
&\geq H(X_{(3,1,2)}, X_{(3,2,1)}, Z_1, Z_2, W_2, W_3, W_1) + H(W_3) && (82)\\
&= H(W_1, W_2, W_3) + H(W_3) = F_1 + F_2 + 2F_3 && (83)
\end{aligned}$$

The per-request rates above yield an information-theoretical lower bound for the average rate $\bar{R} \geq \frac{5}{2}$. This bound can be achieved by dividing the files into unit-length subfiles, $w_2 = (v_1, v_2)$ and $w_3 = (u_1, u_2, u_3)$, to be stored in the caches as follows.

TABLE XII
TRANSMITTED MESSAGES $X_{\vec{d}}$ FOR EACH POSSIBLE DEMAND INSTANCE $\vec{d}$. THESE ACHIEVABLE RATES MATCH THE OBTAINED LOWER BOUNDS IN EXAMPLE 4, HENCE THEY CORRESPOND TO THE PER-REQUEST CAPACITY.

Request $\vec{d}$	Transmitted messages $X_{\vec{d}}$	Rate $R_{\vec{d}}$
(1,2,3)	$(u_2,\, w_1 \oplus u_1)$	2
(1,3,2)	$(w_1,\, v_2,\, u_1,\, u_3)$	4
(2,1,3)	$(v_2,\, u_1 \oplus v_1,\, w_1 \oplus u_2)$	3
(2,3,1)	$(v_2,\, u_3,\, u_1 \oplus v_1)$	3
(3,1,2)	$(u_2,\, u_3,\, w_1 \oplus v_2)$	3
(3,2,1)	$(u_2,\, u_3)$	2

$$Z_1 = \{u_1\} \quad Z_2 = \{v_1,\ v_2 \oplus u_2\} \quad Z_3 = \{w_1,\ v_1,\ u_3\}.$$

The transmitted messages $X_{\vec{d}}$ in Table XII allow the server to fulfill all requests and yield an average rate of $\bar{R} = \frac{5}{2}$, identical to the above lower bound. Hence, this is the minimum achievable average-rate of the system.

It is unclear whether this scheme can be decomposed in terms of the basic coded caching schemes described in Section III-A. All efforts to confirm or deny this hypothesis have been unsuccessful.

C. Proof of Example 5

Without solving an LP problem as in the proofs of Example 3 and 4, we can directly combine the following relevant Instance 4 inequalities to obtain a tight lower bound:

$$\begin{aligned}
&R_{(1,2,3)} + R_{(2,3,1)} + M_1 + M_2 \geq F_1 + 2F_2 + F_3 && \text{(Inst. 4)}\\
&R_{(1,3,2)} + R_{(3,2,1)} + M_1 + M_2 \geq F_1 + F_2 + 2F_3 && \text{(Inst. 4)}\\
&R_{(2,1,3)} + R_{(1,3,2)} + M_1 + M_2 \geq 2F_1 + F_2 + F_3 && \text{(Inst. 4)}\\
&R_{(2,3,1)} + R_{(3,1,2)} + M_1 + M_2 \geq F_1 + F_2 + 2F_3 && \text{(Inst. 4)}\\
&R_{(3,1,2)} + R_{(1,2,3)} + M_1 + M_2 \geq 2F_1 + F_2 + F_3 && \text{(Inst. 4)}\\
&R_{(3,2,1)} + R_{(2,1,3)} + M_1 + M_2 \geq F_1 + 2F_2 + F_3 && \text{(Inst. 4)}
\end{aligned}$$

to yield

$$\begin{aligned}
&2(R_{(1,2,3)}{+}R_{(1,3,2)}{+}R_{(2,1,3)}{+}R_{(2,3,1)}{+}R_{(3,1,2)}{+}R_{(3,2,1)})\\
&\quad + 6M_1 + 6M_2 \geq 8(F_1 + F_2 + F_3)
\end{aligned}$$

or equivalently $\bar{R} \geq 1$. The average rate can be achieved by Example 5 in [3]. That is, dividing the three files as $w_1 = (t_1, t_2, t_3)$ $w_2 = (u_1, u_2, u_3)$ and $w_3 = (v_1, v_2, v_3)$, to be stored in the caches as follows.

$$Z_1 = \{t_1, u_1, v_1\} \quad Z_2 = \{t_2, u_2, v_2\} \quad Z_3 = \{t_3, u_3, v_3\}.$$

The transmitted messages $X_{\vec{d}}$ in Table XIII allow the server to fulfill all requests and yield an average rate of $\bar{R} = 1$, identical to the above lower bound. Hence, this is the minimum achievable average-rate of the system. This scheme can be regarded as a simple extension of Ha.Fi scheme in Section III-B, which we can call 1/3.Fi: In the placement phase, each file is divided equally into three subfiles of size $\frac{1}{3}$ instead of half and each user stores one subfile of each file. In the delivery phase, coded subfiles are transmitted to benefit multiple users.

TABLE XIII

TRANSMITTED MESSAGES $X_{\vec{d}}$ FOR EACH POSSIBLE DEMAND INSTANCE $\vec{d}$. THESE ACHIEVABLE RATES MATCH THE OBTAINED LOWER BOUNDS IN EXAMPLE 5, HENCE THEY CORRESPOND TO THE PER-REQUEST CAPACITY.

Request $\vec{d}$	Transmitted messages $X_{\vec{d}}$	Rate $R_{\vec{d}}$
(1,2,3)	$(t_2 \oplus u_1, t_3 \oplus v_1, u_3 \oplus v_2)$	1
(1,3,2)	$(t_2 \oplus u_1, t_3 \oplus v_1, u_2 \oplus v_3)$	1
(2,1,3)	$(t_1 \oplus u_2, t_3 \oplus v_1, u_3 \oplus v_2)$	1
(2,3,1)	$(t_1 \oplus u_2, t_2 \oplus v_1, u_3 \oplus v_3)$	1
(3,1,2)	$(t_1 \oplus u_1, t_3 \oplus v_2, u_2 \oplus v_3)$	1
(3,2,1)	$(t_1 \oplus u_1, t_2 \oplus v_2, u_3 \oplus v_3)$	1

REFERENCES

[1] C. Chang and C. Wang, “Coded caching with full heterogeneity: Exact capacity of the two-user/two-file case,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2019, pp. 6–10.

[2] J. Wang, “A survey of web caching schemes for the internet,” *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 5, pp. 36–46, Oct. 1999. [Online]. Available: http://doi.acm.org/10.1145/505696.505701

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

[4] C. Tian, “Symmetry, demand types and outer bounds in caching systems,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jul. 2016, pp. 825–829.

[5] D. Cao, D. Zhang, P. Chen, N. Liu, W. Kang, and D. Gündüz, “Coded caching with asymmetric cache sizes and link qualities: The two-user case,” *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6112–6126, Sep. 2019.

[6] C.-H. Chang and C.-C. Wang, “Coded caching with heterogeneous file demand sets — the insufficiency of selfish coded caching,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2019, pp. 1–5.

[7] C.-H. Chang, C.-C. Wang, and B. Peleato, “On coded caching for two users with overlapping demand sets,” in *Proc. IEEE Int. Conf. on Communications (ICC)*, 2020, pp. 1–6.

[8] 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 2015.

[9] Q. Yan, M. Cheng, X. Tang, and Q. Chen, “On the placement delivery array design for centralized coded caching scheme,” *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.

[10] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “Characterizing the rate-memory tradeoff in cache networks within a factor of 2,” *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan 2019.

[11] 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.

[12] J. Zhang, X. Lin, and X. Wang, “Coded caching under arbitrary popularity distributions,” *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 349–366, Jan. 2018.

[13] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, “Order-optimal rate of caching and coded multicasting with random demands,” *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017.

[14] T. Luo, V. Aggarwal, and B. Peleato, “Coded caching with distributed storage,” *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7742–7755, 2019.

[15] K. Wan, M. Cheng, M. Kobayashi, and G. Caire, “On the optimal load-memory tradeoff of coded caching for location-based content,” *arXiv preprint arXiv:2109.06016*, 2021.

[16] H. Ghasemi and A. Ramamoorthy, “Asynchronous coded caching,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, June 2017, pp. 2438–2442.

[17] C. Li, “On rate region of caching problems with non-uniform file and cache sizes,” *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 238–241, Feb 2017.

[18] A. M. Daniel and W. Yu, “Optimization of heterogeneous coded caching,” *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1893–1919, 2020.

[19] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis., “Finite-length analysis of caching-aided coded multicasting,” *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct 2016.

[20] E. Ozfatura and D. Guenduez, “Uncoded caching and cross-level coded delivery for non-uniform file popularity,” in *Proc. IEEE Int. Conf. on Communications (ICC)*, May 2018, pp. 1–6.

[21] J. Hachem, N. Karamchandani, and S. N. Diggavi, “Coded caching for multi-level popularity and access,” *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3108–3141, May 2017.

[22] P. Quinton, S. Sahraei, and M. Gastpar, “A novel centralized strategy for coded caching with non-uniform demands,” *arXiv:1801.10563*, Jan. 2018.

[23] C. Zhang, S. Wang, V. Aggarwal, and B. Peleato, “Coded caching with heterogeneous user profiles,” *arXiv preprint arXiv:2201.10646*, 2022.

[24] S. Sahraei, P. Quinton, and M. Gastpar, “The optimal memory-rate trade-off for the non-uniform centralized caching problem with two files under uncoded placement,” *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 7756–7770, Dec. 2019.

[25] A. M. Ibrahim, A. A. Zewail, and A. Yener, “Coded caching for heterogeneous systems: An optimization perspective,” *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5321–5335, Aug 2019.

[26] J. Zhang, X. Lin, C. Wang, and X. Wang, “Coded caching for files with distinct file sizes,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, June 2015, pp. 1686–1690.

[27] L. Zheng, Q. Chen, Q. Yan, and X. Tang, “Decentralized coded caching scheme with heterogenous file sizes,” *IEEE Trans. Veh. Technol.*, pp. 1–1, 2019.

[28] J. Zhang, X. Lin, and C. Wang, “Closing the gap for coded caching with distinct file sizes,” in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, July 2019, pp. 687–691.

[29] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.

[30] H. Ghasemi and A. Ramamoorthy, “Improved lower bounds for coded caching,” *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4388–4413, July 2017.

[31] C. Wang, S. Saeedi Bidokhti, and M. Wigger, “Improved converses and gap results for coded caching,” *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7051–7062, Nov 2018.

[32] A. M. Ibrahim, A. A. Zewail, and A. Yener, “Optimization of heterogeneous caching systems with rate limited links,” in *Proc. IEEE Int. Conf. on Communications (ICC)*, May 2017, pp. 1–6.

[33] https://engineering.purdue.edu/~chihw/pub_pdf/ISIT2019_micro_cap_proof.pdf.

Chih-Hua Chang Chih-Hua Chang received the B.S. degree in Electrical Engineering and the M.S. degree in Communication Engineering from National Taiwan University, Taipei, Taiwan, in 2010 and 2012, and the Ph.D. degree in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 2021. From 2014 to 2015, he was a Research Assistant with the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei, Taiwan. From May to August 2020, he worked in Facebook's Network Infrastructure Group as a PhD Intern. He is currently a Software Engineer at Google. His research interests are in wireless communications, information theory, and networking.

Borja Peleato Borja Peleato received the B.S. degrees in Telecommunications and Mathematics from the Universitat Politecnica de Catalunya, Barcelona, Spain, in 2007, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University, Stanford, CA, USA, in 2009 and 2013, respectively. He was a visiting student at the Massachusetts Institute of Technology in 2006 and a Senior Flash Channel Architect with Proton Digital Systems in 2013. From 2014 to 2020, he was an Assistant Professor in the Electrical and Computer Engineering Department at Purdue University, West Lafayette, IN, USA. In 2020, he was awarded a CONEX-Marie Curie Fellowship and joined the Signal Theory and Communications group at the Universidad Carlos III de Madrid, Leganes, Spain, where he is currently a CAM Atraccion de Talento Fellow.

His research interests include wireless communications, information theory, convex optimization and nonvolatile storage.

Chih-Chun Wang Chih-Chun Wang is a Professor of the School of Electrical and Computer Engineering of Purdue University. He received the B.E. degree in E.E. from National Taiwan University, Taipei, Taiwan in 1999, the M.S. degree in E.E., the Ph.D. degree in E.E. from Princeton University in 2002 and 2005, respectively. He worked in Comtrend Corporation, Taipei, Taiwan, as a design engineer in 2000 and spent the summer of 2004 with Flarion Technologies, New Jersey. In 2005, he held a post-doctoral researcher position in the Department of Electrical Engineering of Princeton University. He joined Purdue University in 2006, and became a Professor in 2017. He is currently a senior member of IEEE and served as an associate editor of IEEE Transactions on Information Theory during 2014 to 2017. He served as the technical co-chair of the 2017 IEEE Information Theory Workshop. His current research interests are in the latency minimization of 5G wireless networks and the corresponding protocol design, information theory, network coding, and cyber-physical systems. Other research interests of his fall in the general areas of networking, optimal control, information theory, detection theory, and coding theory.

Dr. Wang received the National Science Foundation Faculty Early Career Development (CAREER) Award in 2009.