Computer Networks and Communications
http://ojs.wiserpub.com/index.php/CNC/

UNIVERSAL WISER PUBLISHER

Research Article

Coded Caching in Combinatorial Multi-Access Networks with Private Caches

Dhruv Pratap Singh[1], Anjana A. Mahesh[2], B. Sundar Rajan[1*]

[1] Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India
[2] Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Telangana 502285, India
[*] Correspondence: bsrajan@iisc.ac.in

Received: 28 January 2026; **Revised:** 9 March 2026; **Accepted:** 11 March 2026; **Published:** 19 March 2026

Abstract: Content caching at the network edge plays a critical role in reducing server load and improving scalability in large-scale content delivery networks. We consider a cache-aided multi-access network in which each user is equipped with a private cache and connects to a distinct subset of access caches. A central server stores the content library and populates both private and access caches using uncoded placement. For this setting, we establish upper and lower bounds on the optimal worst-case rate under uncoded placement, leveraging the Maddah-Ali-Niesen (MAN) schemes for dedicated and combinatorial multi-access caching, and also derive a cut-set lower bound under general placement. We then propose an extension scheme that integrates the principles of the MAN schemes for dedicated and multi-access caching, generalizing them to the considered setting, at the cost of high subpacketization complexity. To address this limitation, we introduce an improved coded caching scheme for a fixed private cache size, which achieves significantly lower subpacketization while maintaining comparable rates. Under the placement strategy of the improved scheme, we further derive an index-coding-based lower bound on the rate. Numerical comparisons demonstrate that the improved scheme reduces rate and subpacketization, with performance approaching theoretical lower bounds as the multi-access connectivity increases. In addition, we present a general placement policy applicable to arbitrary private cache sizes. Finally, we show that the proposed scheme is order-optimal in certain regimes and prove its optimality when the number of access caches equals four.

Keywords: coded caching, multi-access networks, edge caching, cache-aided networks, information-theoretic analysis, content delivery

1. Introduction

The rapid growth of Internet traffic, driven by video streaming, social media, and other content-centric applications, has created major challenges for low-latency and bandwidth-efficient delivery in next-generation wireless networks. Edge caching has emerged as a key approach to alleviate congestion, improve spectral efficiency, and reduce access delays by storing popular content closer to users. In particular, coded caching, introduced by Maddah-Ali and Niesen (MAN) [1], exploits coding opportunities across cached content to reduce transmissions during peak demand periods.

Coded caching operates in two phases. In the *placement phase*, caches are pre-filled with coded [2–4] or uncoded [5, 6] file segments without knowledge of future requests. In the *delivery phase*, the server broadcasts coded transmissions

DOI: https://doi.org/10.37256/cnc.4120269691

to simultaneously serve multiple users. The key design goal is to jointly design placement and delivery strategies that minimize the required number of transmissions. The MAN scheme, proposed for the dedicated caching scenario where each of the K users has a dedicated cache of size $M \leq N$ files, is optimal under uncoded placement when $N \geq K$ [7].

Subsequent works extended coded caching to decentralized placement [8], hierarchical networks [9], secure and private delivery [10, 11] and mobile edge caching [12], and many more [13–19]. Caching has also been extensively studied in the context of information-centric wireless networks and infrastructure-supported systems, where content-centric communication and user mobility play a key role in network performance. Prior works have analyzed the impact of caching on throughput, delay, and scalability in large-scale wireless networks, including scenarios with inhomogeneous node distributions and correlated mobility patterns [20, 21]. These studies highlight the importance of caching as a fundamental architectural component for modern wireless systems and motivate further investigation into caching strategies under diverse access and network constraints. This work focuses on a centralized coded caching setting, which enables the characterization of fundamental rate-memory trade-offs under structured combinatorial multi-access constraints, complementing recent studies on decentralized coded caching [22].

Multi-layer cache networks are particularly relevant in wireless systems, where edge servers, base stations, and access points can store content to serve users cooperatively. Practical implementations of coded caching in wireless systems have also been explored, such as the design and evaluation of index-coded content delivery at the WiFi edge in [23], which demonstrated measurable performance gains in real-world deployments. More recent works, including [24], have developed new delivery algorithms for decentralized multi-access coded caching systems. These works highlight coded caching as a fundamental network service for content delivery across heterogeneous and distributed network infrastructures. Caching is widely recognized as a key enabling technology for next-generation wireless networks, including emerging 6G systems [25], due to its potential to reduce latency and alleviate backhaul congestion. As wireless networks evolve toward dense and heterogeneous architectures with distributed edge resources, users may access content through multiple cache-enabled nodes with overlapping coverage. This motivates the study of caching models with structured multi-access constraints. In this context, coded caching offers a principled framework for exploiting cache memory to reduce delivery load. Understanding fundamental rate-memory trade-offs under such access structures is therefore of particular relevance.

Existing models usually assume either (i) dedicated user caches [1], (ii) shared caches at network nodes [26, 27], or (iii) multi-access caches where users connect to multiple nodes but have no private storage [28–31]. Hybrid models with one private and one shared cache per user have also been studied [32–34]. However, these models do not capture scenarios in which users simultaneously rely on local storage while accessing multiple shared cache nodes, a setting common in wireless edge networks with dense and heterogeneous connectivity. To the best of our knowledge, no prior work considers a system where each user maintains a private cache while connecting to multiple shared access caches.

Motivated by this gap, we study the Combinatorial Multi-Access with Private caches (CMAP) model. A server with N files connects to K users and Λ access caches via an error-free shared link. Each user connects to a distinct r-subset of the Λ access caches and also has a private cache. Both caches are populated using uncoded placement, while delivery jointly exploits both storage levels to minimize transmissions. This two-level structure models practical wireless deployments, simultaneously improving spectral efficiency, reducing backhaul load, and enabling low-latency delivery in multi-access edge-assisted networks.

While practical wireless systems involve heterogeneous channel conditions and physical-layer constraints, many existing caching studies incorporate these aspects through system- and optimization-based formulations tailored to specific deployment scenarios [35–38]. In contrast, the error-free shared-link abstraction adopted here serves a complementary purpose: it isolates the impact of cache memory and access structure on coded caching gains, enabling the characterization of fundamental rate-memory limits under structured multi-access constraints. Such information-theoretic characterizations provide analytical benchmarks that are independent of physical-layer assumptions and can guide the design and evaluation of more realistic caching architectures.

1.1 *Our contributions*

From a network perspective, this work characterizes the fundamental limits and design trade-offs of cache-aided multi-access networks with both private and shared storage. The main contributions are summarized as follows:

• We introduce the CMAP network model for cache-aided multi-access networks, generalizing both the dedicated caching network [1] and the combinatorial multi-access caching network [30].

• We derive upper and lower bounds on the optimal worst-case rate under uncoded placement by leveraging the schemes in [1] and [30], and present a cut-set lower bound for general placement [39], providing fundamental performance benchmarks for CMAP networks.

• We propose an extension scheme that generalizes the schemes in [1] and [30] to the CMAP setting for arbitrary private cache memory values, providing a natural performance benchmark. For a specific private cache size, we design an improved centralized coded caching scheme based on a novel placement policy, achieving substantially reduced subpacketization while offering rates comparable to those of the extension scheme. We also derive an index-coding-based lower bound on the worst-case rate under the placement policy of the improved scheme.

• Numerical results show that the improved scheme achieves lower subpacketization and better rate performance than the extension scheme, approaching the lower bounds in certain memory regimes.

• We discuss a general placement strategy for arbitrary private cache sizes and establish order-optimality for a certain class of parameters, and exact optimality for all memory pairs when $\Lambda = 4$.

Organization: Section 2 presents the system model and preliminaries. Main results are given in Section 3, with proofs in Section 4. Section 5 provides numerical comparisons. Section 6 discusses general placement and optimality results. Section 7 concludes the paper.

Notation: $[N] = \{1, \cdots, N\}$, $[a,b] = \{a, \cdots, b\}$ for $a, b \in \mathbb{Z}^+$. $|A|$ is the cardinality of A, $\mathbb{F}_q$ denotes the finite field with q elements, $\lceil a \rceil$ and $\lfloor a \rfloor$ denote ceiling and floor, $\binom{n}{k}$ is the binomial coefficient (0 if $n < k$), and $\oplus$ denotes bitwise XOR.

2. System model

Consider the system model shown in Figure 1. The access caches can be interpreted as distributed edge servers or access points, while the private cache represents user-side storage. A central server stores N files $\{W_1, \ldots, W_N\}$, each of size B bits, and connects to K users through an error-free broadcast link, where $N \geq K$. The system has Λ access caches, each with storage capacity $M_a \leq N$ files. Every distinct r-subset of these caches serves one user via error-free, infinite-capacity wireless links, resulting in $K = \binom{\Lambda}{r}$ users. Each user is indexed by the access caches they connect to and also has a private cache of size $M_p \leq N$. The memory pair (M_a, M_p) satisfies $rM_a + M_p < N$. The case $K = \binom{\Lambda}{r}$ corresponds to the full combinatorial instance. Systems with fewer users can be handled by padding the instance with dummy users, while systems with more users can be addressed by grouping users with identical access patterns and applying the proposed scheme via repetition or time-sharing. This model is referred to as the $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system. The system operates in two phases:

Placement phase: The server populates both private and access caches under their memory constraints. Each file is partitioned into subfiles, which are further divided into mini-subfiles. The private cache of user $\mathscr{U}$ stores a subset of mini-subfiles, while access cache a stores subfiles directly. The contents of access cache a and private cache of user $\mathscr{U}$ are denoted by Z_a and $Z^p_{\mathscr{U}}$, respectively. The total content available to user $\mathscr{U}$ from its access caches and private cache is denoted by $\mathscr{Z}_{\mathscr{U}}$, and the number of mini-subfiles per file defines the subpacketization level.

Delivery phase: Each user $\mathscr{U}$ requests one file, forming the demand vector $\mathbf{d} = (d_{\mathscr{U}} : \mathscr{U} \subseteq [\Lambda], |\mathscr{U}| = r)$. After $\mathbf{d}$ is revealed, the server broadcasts minimum number of coded transmissions of mini-subfiles (via bitwise XOR) to satisfy all demands. Each transmission has the size of one mini-subfile. The rate R is defined as the total number of bits transmitted, normalized by the file size, or equivalently, the number of transmissions normalized by the subpacketization level. The worst-case rate corresponds to the case where all users request distinct files, resulting in maximum number of transmissions.

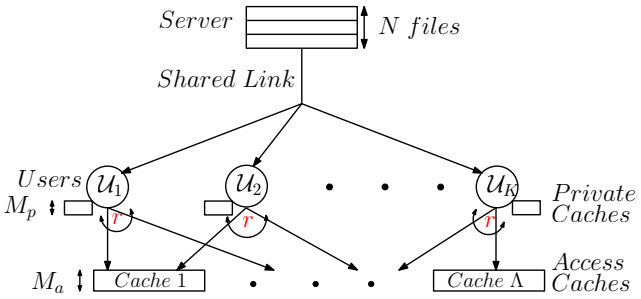

Figure 1. The $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system

Definition 1 In the $(\Lambda, r, M_a, M_p, N)$-CMAP setting, a triplet (M_a, M_p, R) is achievable if there exists a coded caching scheme attaining rate R for the given memory pair (M_a, M_p) with sufficiently large file size. The optimal worst-case rate is defined as

$$R^*(M_a, M_p) = \inf\{R : (M_a, M_p, R) \text{ is achievable } \}.$$

Here, the worst-case rate denotes the maximum delivery load over all possible user request patterns, providing a guaranteed performance bound independent of demand statistics. The objective is to design joint placement and delivery policies such that $R^*(M_a, M_p)$ is achieved.

2.1 *MAN scheme*

The MAN scheme [1] considers a dedicated caching network with K users and a central server storing N files. Each user connects to a distinct cache of size $M \leq N$. The delivery phase begins after demand vector $\mathbf{d} = (d_1, \ldots, d_K)$ is known, where d_k is the index of the file requested by the k^{th} user.

Placement Phase: Each file is split into $\binom{K}{t}$ subfiles, $W_n = \{W_{n,\mathcal{T}} : \mathcal{T} \subseteq [K], |\mathcal{T}| = t\}$, where $t = \frac{KM}{N} \in \mathbb{Z}^+$. Cache k stores $Z_k = \{W_{n,\mathcal{T}} : k \in \mathcal{T}, |\mathcal{T}| = t, \forall n \in [N]\}$.

Delivery Phase: For each subset $\mathcal{S} \subseteq [K], |\mathcal{S}| = t+1$, the server transmits $T_{\mathcal{S}} = \bigoplus_{s \in \mathcal{S}} W_{d_s, \mathcal{S}\setminus s}$.

Rate: With $\binom{K}{t}$ subfiles per file and one transmission per $(t+1)$-subset, the rate, proven optimal under uncoded placement for $N \geq K$ [6], is $R_D^*(M) = \frac{\binom{K}{t+1}}{\binom{K}{t}}$.

2.2 *MAN scheme for Combinatorial Multi-Access Coded Caching (CMACC) network*

Consider a combinatorial multi-access setting with N files, Λ caches, each of memory $M \leq N$ files, and $K = \binom{\Lambda}{r}$ users, each accessing a distinct r-subset of the Λ caches.

Placement Phase: Each file is divided into $\binom{\Lambda}{t}$ subfiles, $W_n = \{W_{n,\mathcal{T}} : \mathcal{T} \subset [\Lambda], |\mathcal{T}| = t\}$, where $t = \frac{\Lambda M}{N} \in \mathbb{Z}^+$. Cache λ stores $Z_\lambda = \{W_{n,\mathcal{T}} : \lambda \in \mathcal{T}, |\mathcal{T}| = t, \forall n \in [N]\}$.

Delivery Phase: For a demand vector $\mathbf{d} = (d_{\mathcal{U}} : \mathcal{U} \subseteq [\Lambda], |\mathcal{U}| = r)$, the server transmits $T_{\mathcal{S}} = \oplus_{\mathcal{U} \subseteq \mathcal{S}, |\mathcal{U}|=r} W_{d_{\mathcal{U}}, \mathcal{S}\setminus\mathcal{U}}$, $\forall \mathcal{S} \subseteq [\Lambda], |\mathcal{S}| = (t+r)$.

Rate: The scheme [30], shown optimal under uncoded placement for $N \geq K$ [31], achieves $R^*_{CMACC} = \frac{\binom{\Lambda}{t+r}}{\binom{\Lambda}{t}}$ for subpacketization level $\binom{\Lambda}{t}$.

2.3 *Index coding preliminaries*

The Index Coding Problem (ICP) with side information [40, 41] involves a source with n messages $x_1, \ldots, x_n \in \mathbb{F}_q$ broadcasting to K receivers $\{R_i\}_{i=1}^K$. Receiver R_i knows $\{x_j : j \in \mathcal{X}_i\}$, where $\mathcal{X}_i \subseteq [n]$ is the index set of messages

belonging to the side information of receiver R_i and demands $x_{f(i)}$, where $f:[K]\to[n]$, and $f(i)\notin\mathscr{X}_i$. For an ICP $\mathscr{I}$, define $\mathscr{Y}_i=[n]\setminus\{f(i)\cup\mathscr{X}_i\}$ for each R_i and $\mathscr{J}(\mathscr{I})=\cup_{i\in[K]}\{\{f(i)\}\cup Y_i:Y_i\subseteq\mathscr{Y}_i\}$. A subset $H\subseteq[n]$ is a generalized independent set in $\mathscr{I}$ if all its subsets belong to $\mathscr{J}(\mathscr{I})$. The maximal such set, referred as maximal generalized independent set, has size $\alpha(\mathscr{I})$, called the generalized independence number [42]. It lower bounds the number of scalar linear transmissions required to solve $\mathscr{I}$ [43]. In coded caching, for a given demand vector, each user acts as a receiver, with demanded subfiles as messages and cached subfiles as side information. Thus, the delivery phase can be modeled as an ICP and $\alpha(\mathscr{I})$ provides a lower bound on the number of transmissions needed to satisfy all demands.

3. Main results

This section develops fundamental bounds and achievable schemes for the CMAP coded caching model under uncoded placement. We first benchmark the optimal worst-case rate by relating CMAP to classical centralized coded caching architectures, thereby providing reference performance limits. We then derive a cut-set-based converse bound that applies to general system parameters. Building on these bounds, we propose a general extension scheme that unifies the MAN and CMACC frameworks, followed by an improved low-subpacketization scheme for a practically relevant private-cache regime. Finally, we derive an index-coding-based lower bound that enables a principled comparison between achievable rates and fundamental limits. Together, these results clarify both the performance potential and inherent limitations of CMAP systems. From a network design perspective, these results characterize the fundamental trade-offs between cache memory, rate, and subpacketization in cache-aided multi-access networks. Theorem 1 establishes the upper and lower bounds on the optimal worst-case rate under uncoded placement for the CMAP network defined in Section 2. Theorem 2 provides a cut-set-based lower bound on the optimal worst-case rate under general placement. Theorem 3 introduces an achievability scheme that extends the MAN [1] and CMACC [30] schemes to the CMAP setting. Subsequently, Theorem 4 proposes an improved coded caching scheme for the specific case $M_p=\frac{N}{K}$, achieving substantially lower subpacketization. Under the placement policy described later in Section 4.2, Theorem 5 establishes a lower bound on the required number of transmissions, followed by Corollary 1, which provides the corresponding lower bound on the optimal worst-case rate, under uncoded placement.

Theorem 1 For a (Λ,r,M_a,M_p,N)-CMAP coded caching system, the optimal worst-case rate $R^*_{UC}(M_a,M_p)$ under uncoded placement is bounded as:

$$R^*_D(rM_a+M_p)\le R^*_{UC}(M_a,M_p)\le R^*_{CMACC}(M_a+\tfrac{M_p}{r}),$$

where, $R^*_D(rM_a+M_p)$ is the rate achieved by MAN scheme [1] by letting $M=rM_a+M_p$ and $R^*_{CMACC}(M_a+\frac{M_p}{r})$ is the rate achieved by MAN scheme for CMACC network [30] by letting $M=M_a+\frac{M_p}{r}$.

Proof. Consider a (Λ,r,M_a,M_p,N)-CMAP coded caching system. Each user in this system connects to r access caches, each of which is capable of storing M_a files. In addition to this, the user also has a private cache of storage capacity M_p files. Hence, the total memory accessed by each user is rM_a+M_p. For a fair comparison, the total memory accessed by each user is kept the same in all the three settings under consideration, namely the CMAP coded caching setting, the combinatorial multi-access network, and the dedicated caching network. We will calculate the size of the caches in the combinatorial multi-access network first, followed by the calculation of the size of the cache memories in the dedicated caching network. In the multi-access coded caching network, each user connects to r caches. If every cache is of size M_{CMACC}, the total memory accessed by the user will be rM_{CMACC}. Since the total memory accessed by a user in the CMAP setting is rM_a+M_p, we have $M_{CMACC}=M_a+\frac{M_p}{r}$. In the dedicated caching network, each user connects to a cache of size M_D which implies $M_D=rM_a+M_p$. We will now prove the inequality given above.

Let Z^* and D^* be the placement and delivery policy that results in $R^*_{UC}(M_a,M_p)$. Here, the contents accessible to each user $\mathscr{U}$ can be written as $\mathscr{Z}_{\mathscr{U}}=\{\{\cup_{i\in\mathscr{U}}Z_i\}\cup Z^p_{\mathscr{U}}\}$. In a dedicated cache network with $K=\binom{\Lambda}{r}$ users, each having a cache of size rM_a+M_p files, it is possible to follow a placement such that the contents available to user $k\in[K]$ is the

same as $\{\{\cup_{i\in\mathscr{U}} Z_i\}\cup Z^p_{\mathscr{U}}\}$ where $\mathscr{U}$ is the k^{th} user, when the user-index sets are arranged lexicographically. Thus, we can conclude that by following the delivery policy D^* in the dedicated caching network, we achieve a rate of $R^*_{UC}(M_a,M_p)$. Hence, we have $R^*_D(rM_a+M_p)\leq R^*_{UC}(M_a,M_p)$.

Consider a combinatorial multi-access coded caching network with Λ caches, each of memory $M=M_a+\frac{M_p}{r}$ files, achieving the rate $R^*_{CMACC}(M)$. The contents of a cache $i\in[\Lambda]$ can be written as $Z_i=Z_{a_i}\cup Z_{p_i}$, where $|Z_{a_i}|=M_a$, and, $|Z_{p_i}|=\frac{M_p}{r}$. For the CMAP setting, if we populate the i^{th} access cache as $Z_i=Z_{a_i}, i\in[\Lambda]$, and the content of the private cache of user $\mathscr{U}$ as $Z^p_{\mathscr{U}}=\cup_{i\in\mathscr{U}}Z_{p_i}$, following the delivery policy of [30], we obtain a rate of $R^*_{CMACC}(M_a+\frac{M_p}{r})$. Hence, $R^*_{UC}(M_a,M_p)\leq R^*_{CMACC}(M_a+\frac{M_p}{r})$. □

Theorem 1 characterizes the bounds on the optimal rate under uncoded placement and places the CMAP coded caching system between two well-understood centralized caching architectures. The lower bound corresponds to a single-cache MAN system with effective cache size rM_a+M_p, while the upper bound corresponds to a CMACC system with effective access-cache memory $M_a+\frac{M_p}{r}$. These bounds provide clear performance benchmarks and reveal how private cache memory contributes differently to the achievable caching gain at the two extremes. Next, we derive a general lower bound that applies to any placement strategy.

Theorem 2 (Cut-Set Bound) For a (Λ,r,M_a,M_p,N)-CMAP coded caching system, the worst-case rate is lower bounded as

$$R^*(M_a,M_p)\geq \max_{s\in\{1,2,\cdots,K\}}\left(s-\frac{qM_a+sM_p}{\left\lfloor\frac{N}{s}\right\rfloor}\right),$$

where $q=\min(\Lambda+r-1,\Lambda)$.

Proof. For $s\in\{1,2,\cdots,K\}$, consider the first s users, given that the user-index sets are arranged in a lexicographic manner. These s users will connect to the first $q=\min(s+r-1,\Lambda)$ access caches. For the demand vector where the first s users request the files $W_1,W_2,\cdots,W_s$, respectively, and the remaining $K-s$ users demand arbitrary files, let the server make the transmission T_1. The first s users decode the files $W_1,W_2,\cdots,W_s$ using T_1, along with the cache contents of the first q access caches and their private caches. Similarly, for the demand vector where the first s users request the files $W_{s+1},W_{s+2},\cdots,W_{2s}$, and the remaining $K-s$ users make arbitrary demands, let the server make the transmission T_2. Using the transmission T_2, the contents of the first q access caches and the contents of their respective private caches, the first s users are able to decode the files $W_{s+1},W_{s+2}\cdots,W_{2s}$. Continuing in this manner, the first s users will be able to decode the files $W_{(\left\lfloor\frac{N}{s}\right\rfloor-1)s+1},W_{(\left\lfloor\frac{N}{s}\right\rfloor-1)s+2},\cdots,W_{\left\lfloor\frac{N}{s}\right\rfloor s}$ using the contents of the first q access caches, the contents of their respective private caches and the transmission $T_{\left\lfloor\frac{N}{s}\right\rfloor}$. The server has transmitted $\left\lfloor\frac{N}{s}\right\rfloor R^*(M_a,M_p)B$ bits, the first s users have access to qM_aB+sM_pB bits, and, using these transmissions and the cache contents, the first s users have been able to decode $s\left\lfloor\frac{N}{s}\right\rfloor B$ bits. Therefore, we have,

$$\left\lfloor\frac{N}{s}\right\rfloor R^*(M_a,M_p)B+qM_aB+sM_pB\geq s\left\lfloor\frac{N}{s}\right\rfloor B,$$

$$\implies R^*(M_a,M_p)\geq s-\frac{qM_a+sM_p}{\left\lfloor\frac{N}{s}\right\rfloor}.$$

Maximizing over all $s\in\{1,2,\cdots,K\}$, we have,

$$R^*(M_a,M_p)\geq \max_{s\in\{1,2,\cdots,K\}}\left(s-\frac{qM_a+sM_p}{\left\lfloor\frac{N}{s}\right\rfloor}\right).$$

□

The cut-set bound in Theorem 2 establishes a fundamental performance limitation that applies irrespective of the delivery strategy. It serves as a converse benchmark for the CMAP model and is used to assess the tightness of achievable schemes developed later in the section.

We now present an achievability scheme extending the delivery strategies of [1] and [30] to the CMAP setting.

Theorem 3 (Extension Scheme) For a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching setting, a worst-case rate

$$R_{ext} = (1-\gamma)\frac{\binom{\Lambda}{t_{ext}^C+r}}{\binom{\Lambda}{t_{ext}^C}} + \gamma\frac{\binom{\binom{\Lambda}{r}}{1+t_{ext}^D}}{\binom{\binom{\Lambda}{r}}{t_{ext}^D}}, \tag{1}$$

is achievable for the subpacketization $F_{ext} = \binom{\Lambda}{t_{ext}^C}\binom{\binom{\Lambda}{r}}{t_{ext}^D}$, where $t_{ext}^C = \frac{\Lambda M_a}{N(1-\gamma)} \in \mathbb{Z}^+$ and $t_{ext}^D = \frac{\binom{\Lambda}{r}M_p}{N\gamma} \in \mathbb{Z}^+$.

Proof. Section 4.1 gives a scheme achieving this rate. □

The extension scheme in Theorem 3 unifies the MAN and CMACC frameworks by jointly exploiting private and access cache memories. While this scheme illustrates the full range of coded caching gains achievable in CMAP systems, it incurs high subpacketization, motivating the development of more structured schemes with reduced complexity. Theorem 3 defines the rate for integer values of t_{ext}^C and t_{ext}^D. For general t_{ext}^C and t_{ext}^D, the lower convex envelope of these points is achievable via memory sharing (Section 4.1). Although this scheme unifies the MAN and CMACC frameworks, it incurs high subpacketization. To mitigate this, we design an improved scheme for $M_p = \frac{N}{K}$ with significantly reduced subpacketization.

Theorem 4 (Improved Scheme) For a $(\Lambda, r, M_a, M_p = \frac{N}{\binom{\Lambda}{r}}, N)$-CMAP coded caching setting, a worst-case rate

$$R_{imp} = \frac{\binom{\Lambda-r-t_a}{r}}{\binom{t_a+r}{t_a}} + \sum_{i=1}^{r-1}\frac{\binom{r}{i}\binom{\Lambda-t_a-r}{r-i}}{2\binom{t_a+i}{i}}, \tag{2}$$

is achievable for the subpacketization $F_{imp} = \binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}$, where $t_a = \frac{\Lambda M_a}{N}$ and $t_a \in [0, \Lambda]$.

Proof. Section 4.2 gives a scheme achieving this rate. □

Theorem 4 focuses on the practically relevant regime $M_p = \frac{N}{K}$ and presents an improved scheme with significantly reduced subpacketization. This scheme retains substantial coded caching gains while enabling efficient numerical evaluation and comparison with lower bounds. For non-integer t_a, the lower convex envelope is achievable via memory sharing (Section 4.2). We now derive a lower bound on the number of transmissions for the same setting.

Theorem 5 (α-Bound) For a $(\Lambda, r, M_a, M_p = \frac{N}{K}, N)$-CMAP coded caching setting, and the placement policy described in Section 4.2, the number of transmissions T required to satisfy the demands of all the users is lower bounded as

$$T \geq \binom{\Lambda-t_a}{r}\binom{\Lambda-r+1}{t_a+1} - \binom{\Lambda-r+2}{t_a+2} + \frac{\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-2\right)\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-1\right)}{2}, \tag{3}$$

where $t_a = \frac{\Lambda M_a}{N}$ and $t_a \in [0, \Lambda]$.

Proof. The proof is provided in Section 4.4. □

Theorems 4 and 5 are both defined for $M_p = \frac{N}{K}$ and general $M_a = \frac{Nt}{\Lambda}$ with $t_a \in [0, \Lambda]$. The α-bound is derived by modeling the delivery phase as an ICP, where each mini-subfile is a message and cached content forms side information. This yields a subpacketization level F_{imp} determined by (Λ, r, t_a), and the lower bound on T implies a corresponding lower bound on the rate, stated next.

Corollary 1 For a $(\Lambda, r, M_a, M_p = \frac{N}{K}, N)$-CMAP coded caching system, the optimal worst-case rate $R^*_{M_a,M_p}$ is lower bounded as

$$R^*_{M_a,M_p} \geq \frac{\binom{\Lambda-r+1}{t_a+1}}{\binom{\Lambda}{t_a}} - \frac{(\Lambda-r+2)(\Lambda-r+1)}{(t_a+2)(t_a+1)\binom{\Lambda}{r}} + \frac{\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-2\right)\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-1\right)}{2\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}}, \tag{4}$$

where $t_a = \frac{\Lambda M_a}{N}$ and $t_a \in [0, \Lambda]$.

Theorem 5 derives an index-coding-based lower bound on the number of transmissions required for the same cache placement structure as in Theorem 4. Corollary 1 translates this bound into a lower bound on the achievable rate, enabling a direct comparison with the improved scheme and providing insight into its optimality gap as shown in Section 5.

4. Achievability schemes and lower bound

In this section, we describe the placement and delivery policies of the extension scheme achieving the rate in Theorem 3. We then present the placement and delivery procedures of the improved scheme achieving the rate in Theorem 4. Finally, we derive an index-coding-based lower bound on the number of transmissions in the delivery phase, corresponding to Theorem 5. The results for the improved scheme and the lower bound are derived for the case $M_p = \frac{N}{K}$.

4.1 *Achievability scheme 1: extension scheme*

In this subsection, we describe a natural extension of existing schemes [1, 30] to the CMAP setting. We begin with a simple illustrative example, followed by a general description.

Example 1 Consider a $(\Lambda = 5, r = 2, M_a = 2, M_p = 4, N = 10)$-CMAP coded caching system. Each file W_n is divided into two parts, W^a_n and W^p_n, used to populate the access and private caches, respectively. From [30], the fraction of each file available to the user via the access caches is $\frac{M'}{N} = \frac{1}{\binom{\Lambda}{\frac{\Lambda M_a}{N}}}\left(\sum_{n=1}^{r}(-1)^{n+1}\binom{r}{n}\binom{\Lambda-n}{\frac{\Lambda M_a}{N}-n}\right) = 0.4$, yielding $M' = 4$. Here, M' represents the effective amount of distinct content available to a user through the access caches, after accounting for overlaps in the cached contents. Hence, we have $|W^a_n| = \frac{M'}{M_p+M'}B = 0.5B$ and $|W^p_n| = \frac{M_p}{M_p+M'} = 0.5B$, where we denote $\gamma = \frac{M_p}{M_p+M'}$.

Employing the CMACC placement [30], each file-part W^a_n is divided into $\binom{\Lambda}{t^C_{ext}} = 10$ subfiles, where $t^C_{ext} = \frac{\Lambda M_a}{N(1-\gamma)} = 2$ as $W^a_n = \{W^a_{n,12}, W^a_{n,13}, W^a_{n,14}, W^a_{n,15}, W^a_{n,23}, W^a_{n,24}, W^a_{n,25}, W^a_{n,34}, W^a_{n,35}, W^a_{n,45}\}, \forall n \in [N]$. Access cache j stores $Z^a_i = \{W^a_{n,T} : i \in T, T \subset [\Lambda], |T| = 2, \forall n \in [N]\}$, totalling the equivalent of two full files and satisfying the memory constraint.

For the private caches, the MAN placement [1] is applied to the file-parts W^p_n. With $K = \binom{5}{2} = 10$ and $t^D_{ext} = \frac{KM_p}{N\gamma} = 8$, each W^p_n is split into $\binom{K}{t^D_{ext}} = 45$ subfiles, as $W^p_n = \{W^p_{n,T} : T \subseteq [K], |T| = 8\}$. Observe that, since the users are ordered lexicographically, user 12 is the first, user 13 is the second, and so on, up to user 45, who is the tenth. The private caches in the system are populated according to the placement scheme given in [1] as $Z^p_i = \{W^p_{n,T} : i \in T, T \subset [K], |T| = 8, \forall n \in [N]\}$. Thus, each private cache stores 180 sub-parts. As each file is first divided into two and then further divided into 45 sub-parts, 180 sub-parts are equivalent to four files, satisfying its memory constraint. We now explain how the demands of the users are satisfied. For a given demand vector $\mathbf{d}$, the server first makes coded transmissions corresponding to the access-cache file-parts W^a_n, following the CMACC-based delivery procedure in [30]. These transmissions exploit the contents stored in the access caches of the system. Next, the server makes coded transmissions corresponding to the private-cache file-parts W^p_n, following the MAN delivery scheme in [1], using the contents stored in the private caches of the users. For a demand vector $\mathbf{d} = (d_{\mathscr{U}} : \mathscr{U} \subset [\Lambda], |\mathscr{U}| = 2)$, the server makes the following transmission for the sub-parts of $W^a_n, \forall n \in [N]$:

1) $W^a_{d_{12},34} \oplus W^a_{d_{13},24} \oplus W^a_{d_{14},23} \oplus W^a_{d_{23},14} \oplus W^a_{d_{24},13} \oplus W^a_{d_{34},12}$

2) $W^a_{d_{12},35} \oplus W^a_{d_{13},25} \oplus W^a_{d_{15},23} \oplus W^a_{d_{23},15} \oplus W^a_{d_{25},13} \oplus W^a_{d_{35},12}$

3) $W^a_{d_{12},45} \oplus W^a_{d_{14},25} \oplus W^a_{d_{15},24} \oplus W^a_{d_{24},15} \oplus W^a_{d_{25},15} \oplus W^a_{d_{45},12}$

4) $W^a_{d_{13},45} \oplus W^a_{d_{14},35} \oplus W^a_{d_{15},34} \oplus W^a_{d_{34},15} \oplus W^a_{d_{35},14} \oplus W^a_{d_{45},13}$

5) $W^a_{d_{23},45} \oplus W^a_{d_{24},35} \oplus W^a_{d_{25},34} \oplus W^a_{d_{34},25} \oplus W^a_{d_{35},24} \oplus W^a_{d_{45},23}$

Consider the transmission $W^a_{d_{12},34} \oplus W^a_{d_{13},24} \oplus W^a_{d_{14},23} \oplus W^a_{d_{23},14} \oplus W^a_{d_{24},13} \oplus W^a_{d_{34},12}$. Observe that the user 12 has access to all the subfiles in the transmission except $W^a_{d_{12},34}$ as the remaining subfiles are stored in the access caches it connects to. Thus, user 12 is able to decode the transmission and obtain its desired subfile. The same decoding logic applies to all other users involved in the transmission.

Since the server employs the MAN delivery scheme [1] for the private-cache file-parts W^p_n, the users are indexed according to a numerical ordering, rather than by the access-cache subsets they are connected to. Thus, the users are ordered as: $12 \to 1$, $13 \to 2$, $14 \to 3$, $15 \to 4$, $23 \to 5$, $24 \to 6$, $25 \to 7$, $34 \to 8$, $35 \to 9$, $45 \to A$, the demand vector $\mathbf{d}$ can be written as $\mathbf{d} = (d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_A)$. Hence, the following transmissions based on the delivery scheme proposed in [1] for the sub-parts of $W^p_n, \forall n \in [N]$ as:

1) $W^p_{d_1,23456789} \oplus W^p_{d_2,13456789} \oplus W^p_{d_3,12456789} \oplus W^p_{d_4,12356789} \oplus W^p_{d_5,12346789} \oplus W^p_{d_6,12345789} \oplus W^p_{d_7,12345689} \oplus W^p_{d_8,12345679} \oplus W^p_{d_9,12345678}$

2) $W^p_{d_1,2345678A} \oplus W^p_{d_2,1345678A} \oplus W^p_{d_3,1245678A} \oplus W^p_{d_4,1235678A} \oplus W^p_{d_5,1234678A} \oplus W^p_{d_6,1234578A} \oplus W^p_{d_7,1234568A} \oplus W^p_{d_8,1234567A} \oplus W^p_{d_A,12345678}$

3) $W^p_{d_1,2345679A} \oplus W^p_{d_2,1345679A} \oplus W^p_{d_3,1245679A} \oplus W^p_{d_4,1235679A} \oplus W^p_{d_5,1234679A} \oplus W^p_{d_6,1234579A} \oplus W^p_{d_7,1234569A} \oplus W^p_{d_9,1234567A} \oplus W^p_{d_A,12345679}$

4) $W^p_{d_1,2345689A} \oplus W^p_{d_2,1345689A} \oplus W^p_{d_3,1245689A} \oplus W^p_{d_4,1235689A} \oplus W^p_{d_5,1234689A} \oplus W^p_{d_6,1234589A} \oplus W^p_{d_8,1234569A} \oplus W^p_{d_9,1234568A} \oplus W^p_{d_A,12345689}$

5) $W^p_{d_1,2345789A} \oplus W^p_{d_2,1345789A} \oplus W^p_{d_3,1245789A} \oplus W^p_{d_4,1235789A} \oplus W^p_{d_5,1234789A} \oplus W^p_{d_7,1234589A} \oplus W^p_{d_8,1234579A} \oplus W^p_{d_9,1234578A} \oplus W^p_{d_A,12345789}$

6) $W^p_{d_1,2346789A} \oplus W^p_{d_2,1346789A} \oplus W^p_{d_3,1246789A} \oplus W^p_{d_4,1236789A} \oplus W^p_{d_6,1234789A} \oplus W^p_{d_7,1234689A} \oplus W^p_{d_8,1234679A} \oplus W^p_{d_9,1234678A} \oplus W^p_{d_A,12346789}$

7) $W^p_{d_1,2356789A} \oplus W^p_{d_2,1356789A} \oplus W^p_{d_3,1256789A} \oplus W^p_{d_5,1236789A} \oplus W^p_{d_6,1235789A} \oplus W^p_{d_7,1235689A} \oplus W^p_{d_8,1235679A} \oplus W^p_{d_9,1235678A} \oplus W^p_{d_A,12356789}$

8) $W^p_{d_1,2456789A} \oplus W^p_{d_2,1456789A} \oplus W^p_{d_4,1256789A} \oplus W^p_{d_5,1246789A} \oplus W^p_{d_6,1245789A} \oplus W^p_{d_7,1245689A} \oplus W^p_{d_8,1245679A} \oplus W^p_{d_9,1245678A} \oplus W^p_{d_A,12456789}$

9) $W^p_{d_1,3456789A} \oplus W^p_{d_3,1456789A} \oplus W^p_{d_4,1356789A} \oplus W^p_{d_5,1346789A} \oplus W^p_{d_6,1345789A} \oplus W^p_{d_7,1345689A} \oplus W^p_{d_8,1345679A} \oplus W^p_{d_9,1345678A} \oplus W^p_{d_A,13456789}$

10) $W^p_{d_2,3456789A} \oplus W^p_{d_3,2456789A} \oplus W^p_{d_4,2356789A} \oplus W^p_{d_5,2346789A} \oplus W^p_{d_6,2345789A} \oplus W^p_{d_7,2345689A} \oplus W^p_{d_8,2345679A} \oplus W^p_{d_9,2345678A} \oplus W^p_{d_A,23456789}$

Consider the transmission $W^p_{d_1,23456789} \oplus W^p_{d_2,13456789} \oplus W^p_{d_3,12456789} \oplus W^p_{d_4,12356789} \oplus W^p_{d_5,12346789} \oplus W^p_{d_6,12345789} \oplus W^p_{d_7,12345689} \oplus W^p_{d_8,12345679} \oplus W^p_{d_9,12345678}$. The user 1 (user 12) has access to all the subfiles except the subfile $W^p_{d_1,23456789}$. This follows from the placement policy of the private caches of the users, where user i stores the subfile $W^p_{n,T} : i \in T$ in its private cache. Hence, user 1 is able to decode the transmission and obtain its desired subfile. The same logic holds for all other users involved.

The server makes five transmissions of $\frac{0.5B}{10}$ bits and ten transmissions of $\frac{0.5B}{45}$ bits. Thus, total bits transmitted are $\frac{5B}{20} + \frac{10B}{90} = \frac{13B}{36}$, leading to a rate of $R_{ext} = 0.3611$ and a subpacketization of $F_{ext} = 10 + 45 = 55$.

We now give the general description of the placement and delivery policy of the extension scheme.

Placement Policy: Each file W_n is divided into file-parts, W^p_n and W^a_n, in proportion to the file content available to the user via its private cache and the access cache it connects to, respectively. From [30], the fraction of each file available to the user via the access caches is $\frac{M'}{N} = \frac{1}{\binom{\Lambda}{\frac{\Lambda M_a}{N}}} \left(\sum_{n=1}^{r} (-1)^{n+1} \binom{r}{n} \binom{\Lambda - n}{\frac{\Lambda M_a}{N} - n} \right)$. Thus, each file is split as

$W_n = \{W_n^p, W_n^a : |W_n^p| = \gamma B, |W_n^a| = (1-\gamma)B\}$, where $\gamma = \frac{M_p}{M_p+M'}$. The file-parts W_n^p and W_n^a populate the private caches and the access caches using placement policy of [1] and [30], respectively. As user-index sets are arranged lexicographically, the private cache of user $i, i \in [K]$ stores:

$$Z_i^p = \{W_{n,T}^p : i \in T, T \subseteq [K], |T| = t_{ext}^D, \forall n \in [N]\}, \tag{5}$$

where $t_{ext}^D = \frac{KM_p}{N\gamma}$. Each private cache is populated by $\frac{\binom{K-1}{t_{ext}^D-1}}{\binom{K}{t_{ext}^D}}\gamma NB = M_pB$ bits, satisfying its memory constraint. The contents of the access cache $j \in [\Lambda]$ are given as:

$$Z_j = \{W_{n,T}^a : j \in T, T \subseteq [\Lambda], |T| = t_{ext}^C, \forall n \in [N]\}, \tag{6}$$

where $t_{ext}^C = \frac{\Lambda M_a}{N(1-\gamma)}$, resulting in $\frac{\binom{\Lambda-1}{t_{ext}^C-1}}{\binom{\Lambda}{t_{ext}^C}}(1-\gamma)N = M_a$ files, meeting the memory constraint.

Delivery Phase: For a given demand vector $\mathbf{d} = (d_U : U \subset [\Lambda], |U| = r)$, the server employs the MAN delivery strategy [1] and the CMACC delivery strategy [30] to deliver the requested sub-parts of the file-parts W_n^p and W_n^a, respectively. Hence, the server transmits

$$Y_{S^a} = \bigoplus_{U \subset S^a, |U|=r} W_{d_U, S^a \setminus U}^a, S^a \subseteq [\Lambda], |S^a| = r + t_{ext}^C, \tag{7}$$

to deliver the requested sub-parts of the file-part W_n^a and

$$Y_{S^p} = \bigoplus_{i \in S^p} W_{d_i, S^p \setminus \{i\}}^p, S^p \subseteq [K], |S^p| = 1 + t_{ext}^D, \tag{8}$$

where i denotes the i^{th} user when the user-index sets are arranged lexicographically, to deliver the requested sub-parts of the file-part W_n^p.

We now prove the decodability of the extension scheme.

Decodability: For transmissions of type Y_{S^p}, each user i has all subfiles $W_{d_j, S^p \setminus j}^p, j \neq i$, since Z_i^p stores all $W_{n,T}^p$ such that $i \in T$. Hence, user i can decode $W_{d_i, S^p \setminus i}$.

For transmissions of type Y_{S^a}, user U can recover all subfiles $W_{d_{U'}, S^a \setminus U'}^a$ with $U' \neq U$ via its connected access caches, which store all $W_{n,T}^a$ satisfying $U \cap T \neq \emptyset$. Thus, every user can decode its desired subfile $W_{d_U, S^a \setminus U}^a$.

Since each transmission is decodable, all the users are able to obtain their desired files.

Performance of the scheme: The server makes $\binom{\Lambda}{t_{ext}^C+r}$ transmissions of $\frac{(1-\gamma)}{\binom{\Lambda}{t_{ext}^C}}B$ bits and $\binom{K}{t_{ext}^D+1}$ transmissions of $\frac{\gamma}{\binom{K}{t_{ext}^D}}B$ bits. Thus, the rate is

$$R_{ext} = (1-\gamma)\frac{\binom{\Lambda}{t_{ext}^C+r}}{\binom{\Lambda}{t_{ext}^C}} + \gamma\frac{\binom{K}{1+t_{ext}^D}}{\binom{K}{t_{ext}^D}} \tag{9}$$

with a subpacketization level of

$$F_{ext} = \binom{\Lambda}{t_{ext}^{C}} + \binom{K}{t_{ext}^{D}}. \tag{10}$$

We explain memory sharing for this scheme in Section 4.3.

We now present the improved coded caching scheme, defined for $M_p = \frac{N}{K}$, which achieves a significantly lower subpacketization level compared to the extension scheme.

4.2 *Achievability scheme 2: improved scheme*

Before describing the general placement and delivery policy of the improved scheme, we define three functions-$flip$, $swap_o$, and $swap_{no}$-which are used in constructing the server transmissions, as detailed in Algorithm 1. We then provide examples illustrating the operation of these functions.

Definition 2 For a mini-subfile $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$, the function *flip* is defined as $flip(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}) = W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'} \oplus W_{d_{\mathscr{U}'},\mathscr{S},\mathscr{U}}$. Further, we have $flip(\oplus_{i=1}^{n} W_{d_{\mathscr{U}_i},\mathscr{S}_i,\mathscr{U}_i'}) = \oplus_{i=1}^{n}(flip(W_{d_{\mathscr{U}_i},\mathscr{S}_i,\mathscr{U}_i'}))$.

Example 2 For a mini-subfile $W_{d_{12},3,14}$, we have $flip(W_{d_{12},3,14}) = W_{d_{12},3,14} \oplus W_{d_{14},3,12}$.

Definition 3 For a mini-subfile $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$ such that $\mathscr{U}$ and $\mathscr{U}'$ overlap, i.e., $I = \mathscr{U} \cap \mathscr{U}' \neq \emptyset$, the function $swap_o$ is defined as $swap_o\left(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}, i\right) = \oplus_{\widetilde{\mathscr{U}} \subseteq I, |\widetilde{\mathscr{U}}|=i, \widetilde{\mathscr{S}} \subseteq \mathscr{S}, |\widetilde{\mathscr{S}}|=i} W_{d_{\{\mathscr{U} \cup \widetilde{\mathscr{S}}\} \setminus \widetilde{\mathscr{U}}, \{\mathscr{S} \cup \widetilde{\mathscr{U}}\} \setminus \widetilde{\mathscr{S}}, \{\mathscr{U}' \cup \widetilde{\mathscr{S}}\} \setminus \widetilde{\mathscr{U}}}}$.

Example 3 Consider a mini-subfile $W_{d_{123},45,126}$ for which $swap_o(W_{d_{123},45,126}, 1) = W_{d_{234},15,246} \oplus W_{d_{134},25,146} \oplus W_{d_{235},14,256} \oplus W_{d_{135},24,156}$. Observe that all 1-subsets of the intersection set $\mathscr{U} \cap \mathscr{U}' = \{1,2\}$ have been swapped with all possible 1-subsets of the subfile-index set $\{4,5\}$.

For the mini-subfile $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$ such that there is no-overlap between $\mathscr{U}$ and $\mathscr{U}'$, we define the function $swap_{no}$ as follows.

Definition 4 For a mini-subfile $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$ such that $\mathscr{U} \cap \mathscr{U}' = \emptyset$, the function $swap_{no}$ is defined as $swap_{no}(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}, i) = \oplus_{\widetilde{\mathscr{U}} \subseteq \mathscr{U}, |\widetilde{\mathscr{U}}|=i, \widetilde{\mathscr{S}} \subseteq \mathscr{S}, |\widetilde{\mathscr{S}}|=i} W_{d_{\{\mathscr{U} \cup \widetilde{\mathscr{S}}\} \setminus \widetilde{\mathscr{U}}, \{\mathscr{S} \cup \widetilde{\mathscr{U}}\} \setminus \widetilde{\mathscr{S}}, \mathscr{U}'}}$.

Example 4 For a mini-subfile $W_{d_{123},45,678}$, we have $swap_{no}(W_{d_{123},45,678}, 2) = W_{d_{345},12,678} \oplus W_{d_{245},13,678} \oplus W_{d_{145},23,678}$, which is obtained by swapping every 2-subset of the user-index set $\{1,2,3\}$ with the sub-file index set $\{4,5\}$.

We now provide the following example which illustrates the main idea behind the improved scheme. From now on, the user-index set, the subfile-index set, and the mini-subfile-index set will be compactly written without the set notation.

Example 5 Consider a $(\Lambda = 5, r = 2, M_a = 2, M_p = 1, N = 10)$-CMAP coded caching system. The server has a library of $N = 10$ files and connects to $K = 10$ users, each equipped with a private cache of size $M_p = 1$ file. There are $\Lambda = 5$ access caches, each storing $M_a = 2$ files, and every user accesses a unique subset of $r = 2$ access caches. For this setup, $t_a = \frac{\Lambda M_a}{N} = 1$. Each file W_n is split into $\binom{\Lambda}{t_a} = 5$ subfiles as $W_n = \{W_{n,1}, W_{n,2}, W_{n,3}, W_{n,4}, W_{n,5}\}, \forall n \in [10]$. The contents of the access caches are $Z_i = \{W_{n,i}, \forall n \in [10]\}, i \in [5]$, satisfying its memory constraint since each cache stores $\frac{10}{5} = 2$ files.

Each user can obtain subfiles in its connected access caches and thus lacks $\binom{\Lambda - t_a}{r} = 6$ subfiles of every file. These missing subfiles are further divided into $\binom{\Lambda - t_a}{r} = 6$ mini-subfiles as $W_{n,\mathscr{S}} = \{W_{n,\mathscr{S},\mathscr{U}'} : \mathscr{U}' \subseteq [\Lambda] \setminus \mathscr{S}, |\mathscr{U}'| = r\}$. For instance, $W_{n,1} = \{W_{n,1,23}, W_{n,1,24}, W_{n,1,25}, W_{n,1,34}, W_{n,1,35}, W_{n,1,45}\}$, resulting in subpacketization $F_{imp} = 30$. The private cache of user $\mathscr{U}$ stores $Z_{\mathscr{U}}^{p} = \{W_{n,\mathscr{S},\mathscr{U}} : \mathscr{S} \subset [\Lambda] \setminus \mathscr{U}, |\mathscr{S}| = t_a, \forall n \in [N]\}$, holding $\frac{30}{30} = 1$ file in total, meeting its capacity.

We now explain how transmissions are constructed. The server makes two types of transmissions: for mini-subfiles with $I \neq \emptyset$ and with $I = \emptyset$. We first address how the server constructs transmissions for the mini-subfiles with $I \neq \emptyset$. For the demand vector $\mathbf{d} = (d_{\mathscr{U}} : \mathscr{U} \subset [\Lambda], |\mathscr{U}| = r)$, consider the mini-subfile $W_{d_{12},3,14}$ demanded by user 12. Since $I = \{12 \cap 14\} = \{1\}$, the server uses the function $swap_o(W_{d_{12},3,14}, 1)$ to obtain the mini-subfile $W_{d_{23},1,34}$. Thereafter, the server uses the function $flip(W_{d_{12},3,14} \oplus W_{d_{23},1,34})$ to generate the transmission $W_{d_{12},3,14} \oplus W_{d_{23},1,34} \oplus W_{d_{14},3,12} \oplus W_{d_{34},1,23}$. Observe that user 12 has the mini-subfiles $W_{d_{23},1,34}$ and $W_{d_{34},1,23}$ due to access cache 1 and the mini-subfile $W_{d_{14},3,12}$ from its private cache. Hence, user 12 is able to decode the transmission and obtain its desired mini-subfile $W_{d_{12},3,14}$. The same holds for all the users in the above transmission.

We now focus here on the case where $I = \emptyset$.

Consider the mini-subfile $W_{d_{12},3,45}$ requested by user 12. Since $I = \{12 \cap 45\} = \emptyset$, the server uses the function $swap_{no}(W_{d_{12},3,45}, 1)$ to obtain $W_{d_{13},2,45} \oplus W_{d_{23},1,45}$. Finally, the server performs XOR of $W_{d_{12},3,45}$ and $W_{d_{13},2,45} \oplus W_{d_{23},1,45}$ to generate the transmission $W_{d_{12},3,45} \oplus W_{d_{13},2,45} \oplus W_{d_{23},1,45}$. Note that the mini-subfiles $W_{d_{13},2,45}$ and $W_{d_{23},1,45}$ are available to user 12 from the access caches 1 and 2, respectively. Hence, user 12 is able to decode this transmission and obtain its desired files. Same can be said for the other users in the transmission.

The server makes the following transmissions for $I \neq \emptyset$:

1) $W_{d_{12},3,14} \oplus W_{d_{23},1,34} \oplus W_{d_{34},1,23} \oplus W_{d_{14},3,12}$
2) $W_{d_{12},3,15} \oplus W_{d_{23},1,35} \oplus W_{d_{35},1,23} \oplus W_{d_{15},3,12}$
3) $W_{d_{12},3,24} \oplus W_{d_{13},2,34} \oplus W_{d_{34},2,13} \oplus W_{d_{24},3,12}$
4) $W_{d_{12},3,25} \oplus W_{d_{13},2,35} \oplus W_{d_{35},2,13} \oplus W_{d_{25},3,12}$
5) $W_{d_{12},4,13} \oplus W_{d_{24},1,34} \oplus W_{d_{34},1,24} \oplus W_{d_{13},4,12}$
6) $W_{d_{12},4,15} \oplus W_{d_{24},1,45} \oplus W_{d_{45},1,24} \oplus W_{d_{15},4,12}$
7) $W_{d_{12},4,23} \oplus W_{d_{14},2,34} \oplus W_{d_{34},2,14} \oplus W_{d_{23},4,12}$
8) $W_{d_{12},4,25} \oplus W_{d_{14},2,45} \oplus W_{d_{45},2,14} \oplus W_{d_{25},4,12}$
9) $W_{d_{12},5,13} \oplus W_{d_{25},1,35} \oplus W_{d_{35},1,25} \oplus W_{d_{13},5,12}$
10) $W_{d_{12},5,14} \oplus W_{d_{25},1,45} \oplus W_{d_{45},1,25} \oplus W_{d_{14},5,12}$
11) $W_{d_{12},5,23} \oplus W_{d_{15},2,35} \oplus W_{d_{35},2,15} \oplus W_{d_{23},5,12}$
12) $W_{d_{12},5,24} \oplus W_{d_{15},2,45} \oplus W_{d_{45},2,15} \oplus W_{d_{24},5,12}$
13) $W_{d_{13},2,14} \oplus W_{d_{23},1,24} \oplus W_{d_{24},1,23} \oplus W_{d_{14},2,13}$
14) $W_{d_{13},2,15} \oplus W_{d_{23},1,25} \oplus W_{d_{25},1,23} \oplus W_{d_{15},2,13}$
15) $W_{d_{13},4,15} \oplus W_{d_{34},1,45} \oplus W_{d_{45},1,34} \oplus W_{d_{15},4,13}$
16) $W_{d_{13},4,23} \oplus W_{d_{14},3,24} \oplus W_{d_{24},3,14} \oplus W_{d_{23},4,13}$
17) $W_{d_{13},4,35} \oplus W_{d_{14},3,45} \oplus W_{d_{45},3,14} \oplus W_{d_{35},4,13}$
18) $W_{d_{13},5,14} \oplus W_{d_{35},1,45} \oplus W_{d_{45},1,35} \oplus W_{d_{14},5,13}$
19) $W_{d_{13},5,23} \oplus W_{d_{15},3,25} \oplus W_{d_{25},3,15} \oplus W_{d_{23},5,13}$
20) $W_{d_{13},5,34} \oplus W_{d_{15},3,45} \oplus W_{d_{45},3,15} \oplus W_{d_{34},5,13}$
21) $W_{d_{14},2,15} \oplus W_{d_{24},1,25} \oplus W_{d_{25},1,24} \oplus W_{d_{15},2,14}$
22) $W_{d_{14},3,15} \oplus W_{d_{34},1,35} \oplus W_{d_{35},1,34} \oplus W_{d_{15},3,14}$
23) $W_{d_{14},5,24} \oplus W_{d_{15},4,25} \oplus W_{d_{25},4,15} \oplus W_{d_{24},5,14}$
24) $W_{d_{14},5,34} \oplus W_{d_{15},4,35} \oplus W_{d_{35},4,15} \oplus W_{d_{34},5,14}$
25) $W_{d_{23},4,25} \oplus W_{d_{34},2,45} \oplus W_{d_{45},2,34} \oplus W_{d_{25},4,23}$
26) $W_{d_{23},4,35} \oplus W_{d_{24},3,45} \oplus W_{d_{45},3,24} \oplus W_{d_{35},4,23}$
27) $W_{d_{23},5,24} \oplus W_{d_{35},2,45} \oplus W_{d_{45},2,35} \oplus W_{d_{24},5,23}$
28) $W_{d_{23},5,34} \oplus W_{d_{25},3,45} \oplus W_{d_{45},3,25} \oplus W_{d_{34},5,23}$
29) $W_{d_{24},3,25} \oplus W_{d_{34},2,35} \oplus W_{d_{35},2,34} \oplus W_{d_{25},3,24}$, and,
30) $W_{d_{24},5,34} \oplus W_{d_{25},4,35} \oplus W_{d_{35},4,25} \oplus W_{d_{34},5,24}$

and the following transmissions for $I = \emptyset$:

1) $W_{d_{12},3,45} \oplus W_{d_{13},2,45} \oplus W_{d_{23},1,45}$
2) $W_{d_{12},4,35} \oplus W_{d_{14},2,35} \oplus W_{d_{24},1,35}$
3) $W_{d_{12},5,34} \oplus W_{d_{15},2,34} \oplus W_{d_{25},1,34}$
4) $W_{d_{13},4,25} \oplus W_{d_{14},3,25} \oplus W_{d_{34},1,25}$
5) $W_{d_{13},5,24} \oplus W_{d_{15},3,24} \oplus W_{d_{35},1,24}$
6) $W_{d_{14},5,23} \oplus W_{d_{15},4,23} \oplus W_{d_{45},1,23}$
7) $W_{d_{23},4,15} \oplus W_{d_{24},3,15} \oplus W_{d_{34},2,15}$
8) $W_{d_{23},5,14} \oplus W_{d_{25},3,14} \oplus W_{d_{35},2,14}$
9) $W_{d_{24},5,13} \oplus W_{d_{25},4,13} \oplus W_{d_{45},2,13}$, and,
10) $W_{d_{34},5,12} \oplus W_{d_{35},4,12} \oplus W_{d_{45},3,12}$

As each mini-subfile occurs once across all transmissions and each transmission is decodable by the users in that transmission, the server satisfies the demands of all the users. As the server makes 40 transmissions, the rate $R_{imp} = \frac{40}{30} = 1.33$.

Remark 1 When the user-index set $\mathscr{U}$ and mini-subfile-index set $\mathscr{U}'$ of mini-subfile $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$ overlap, the users in the transmission share a greater portion of cached content compared to the no-overlap case. This increased overlap enhances multicasting opportunities, reducing the overall rate of the improved scheme. This effect can be observed in Example 5, where transmissions with $I \neq \emptyset$ involve coded combinations of four mini-subfiles, whereas those with $I = \emptyset$ involve combinations of only three mini-subfiles.

We now give the general description of the placement and delivery phase of the improved scheme.

Placement Phase: First, we describe the placement policy of the access caches. Each file is split into $\binom{\Lambda}{t_a}$ non-overlapping subfiles of equal size as $W_n = \{W_{n,\mathscr{T}} : \mathscr{T} \subseteq [\Lambda], |\mathscr{T}| = t_a\}, \forall n \in [N]$, where $t_a = \frac{\Lambda M_a}{N}$ is the access cache memory replication factor, and the access cache $a \in [\Lambda]$ stores:

$$Z_a = \{W_{n,\mathscr{T}} : a \in \mathscr{T}, \mathscr{T} \subseteq [\Lambda], |\mathscr{T}| = t_a, \forall n \in [N]\}. \tag{11}$$

Each access cache is populated by $\frac{N\binom{\Lambda-1}{t_a-1}}{\binom{\Lambda}{t_a}} = M_a$ files, satisfying the memory constraint.

Now, we describe the placement strategy of the private caches. Each user $\mathscr{U}$ stores the parts of the subfiles unavailable through its connected access caches. Since every subfile is requested by $\binom{\Lambda-t_a}{r}$ users, each subfile $W_{n,\mathscr{T}}$ is further divided into $\binom{\Lambda-t_a}{r}$ as $W_{n,\mathscr{T}} = \{W_{n,\mathscr{T},\mathscr{U}} : \mathscr{U} \subseteq [\Lambda] \setminus \mathscr{S}, |\mathscr{U}| = r\}$. The private cache of user $\mathscr{U}$ stores

$$Z^p_{\mathscr{U}} = \{W_{n,\mathscr{T},\mathscr{U}} : \mathscr{T} \subseteq [\Lambda] \setminus \mathscr{U}, |\mathscr{T}| = t_a, \forall n \in [N]\}. \tag{12}$$

Each private cache stores $\frac{N\binom{\Lambda-r}{t_a}}{\binom{\Lambda-t_a}{r}\binom{\Lambda}{t_a}} = \frac{N}{\binom{\Lambda}{r}} = M_p$ files, satisfying the memory constraint. By construction, there is no overlap between the contents of a user's private cache and those of the access caches it connects to.

Delivery Phase: For a given demand vector $\mathbf{d}$, the server broadcasts the set of transmissions T generated by Algorithm 1. We now describe the working of Algorithm 1. The proposed delivery algorithm systematically constructs coded multicast transmissions by exploiting the combinatorial structure of the CMAP system. For each user $\mathscr{U}$, the algorithm maintains a demand set $\mathscr{D}_{\mathscr{U}}$ containing mini-subfiles that are not available through the user's caches. At each iteration, the algorithm selects one unsatisfied demand tuple and forms a coded transmission by combining it with other compatible mini-subfiles using the prescribed *swap* and $flip$ operations. Each coded transmission simultaneously serves multiple users. Once a transmission is formed, all mini-subfiles included in it are removed from the corresponding demand sets. This process is repeated until all demand sets are exhausted, ensuring that every requested mini-subfile is delivered exactly once.

Given $\mathbf{d}$ and the placement policy in Section 4.2, Algorithm 1 constructs all transmissions required to satisfy the K users. For each user $\mathscr{U}$, the algorithm defines a demand set $\mathscr{D}_{\mathscr{U}}$ containing indices of all mini-subfiles required by that user. For instance, in Example 5, the demand sets of users $12, 14, 23$, and 34 as $\mathscr{D}_{12} = \{(3,14),(3,24),\cdots,(5,34)\}, \mathscr{D}_{14} = \{(2,13),\cdots,(3,12),\cdots,(5,34)\}, \mathscr{D}_{23} = \{(1,24),\cdots,(1,34),\cdots,(5,34)\}$, and $\mathscr{D}_{34} = \{(1,23),(1,24),\cdots,(5,24)\}$.

The algorithm selects a user $\mathscr{U}$ and an element $(\mathscr{S},\mathscr{U}')$ from $\mathscr{D}_{\mathscr{U}}$, corresponding to the mini-subfile $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$. It computes the intersection $I = \mathscr{U} \cap \mathscr{U}'$. Depending on whether I is empty or not, the transmission is constructed as in Line 8 or Line 10, respectively. For example, in Example 5, choosing user 12 and the element $(3,14)$ yields $I = \{1\}$, resulting in the transmission $W_{d_{12},3,14} \oplus W_{d_{23},1,34} \oplus W_{d_{14},3,12} \oplus W_{d_{34},1,23}$. After each transmission, the corresponding mini-subfile indices are removed from all relevant demand sets. For example, after the transmission, the demand sets of users $12, 14, 23$ and 34 becomes $\mathscr{D}_{12} = \{(3,24),\cdots,(5,34)\}, \mathscr{D}_{14} = \{(2,13),(2,45)\cdots,(3,15),\cdots,(5,34)\}, \mathscr{D}_{23} = \{(1,24),\cdots,(1,25),(1,35)\cdots,(5,34)\}$, and $\mathscr{D}_{34} = \{(1,24),\cdots,(5,24)\}$. The algorithm continues until all $\mathscr{D}_{\mathscr{U}}$ are empty.

Algorithm 1 Algorithm for generating transmission during delivery phase

Input: $\mathbf{d} = (d_{\mathscr{U}} : \mathscr{U} \subseteq [\Lambda], |\mathscr{U}| = r)$, $\mathscr{Z} = (\mathscr{Z}_{\mathscr{U}} : \mathscr{U} \subseteq [\Lambda], |\mathscr{U}| = r)$.

Output: The set of transmissions T.

Initialize $T = \emptyset$.

For each user $\mathscr{U}$, define the user-demand set $\mathscr{D}_{\mathscr{U}} = \{(\mathscr{S}, \mathscr{U}') : \mathscr{S} \subseteq [\Lambda] \setminus \mathscr{U}, |\mathscr{S}| = t_a,\ \mathscr{U}' \subseteq [\Lambda], |\mathscr{U}'| = r, \mathscr{U}' \neq \mathscr{U}, \mathscr{S} \cap \mathscr{U}' = \emptyset\}$.

for $\mathscr{U} \subseteq [\Lambda], |\mathscr{U}| = r$ **do**

  while $\mathscr{D}_{\mathscr{U}} \neq \emptyset$ **do**

    Select an element $(\mathscr{S}, \mathscr{U}')$ from $\mathscr{D}_{\mathscr{U}}$.

    For $(\mathscr{S}, \mathscr{U}')$, define $I = \mathscr{U} \cap \mathscr{U}'$.

    if $|I| > 0$ **then**

$$T' = flip\left(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'} \oplus \bigoplus_{i=1}^{min(|I|,t_a)} swap_o(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}, i)\right).$$

    else

$$T' = W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'} \oplus \bigoplus_{i=1}^{min(r,t_a)} swap_{no}(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}, i).$$

    end if

    $T \leftarrow T \cup T'$.

    Let $S_c = \{(\mathscr{U}, \mathscr{S}, \mathscr{U}') : W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$ is a mini-subfile in $T'\}$. For each $(\hat{\mathscr{U}}, \hat{\mathscr{S}}, \hat{\mathscr{U}}') \in S_c$, do $\mathscr{D}_{\hat{\mathscr{U}}} \leftarrow \mathscr{D}_{\hat{\mathscr{U}}} \setminus (\hat{\mathscr{S}}, \hat{\mathscr{U}}')$

  end while

end for

Time-Complexity Analysis of Algorithm 1: For a fixed user $\mathscr{U}$, the size of the demand set is $|\mathscr{D}_{\mathscr{U}}| = \binom{\Lambda-r}{t_a}\left(\binom{\Lambda}{r} - 1\right) = O\left(\binom{\Lambda-r}{t_a}\binom{\Lambda}{r}\right)$. Hence, the total number of demand tuples across all users is $\sum_{\mathscr{U}} |\mathscr{D}_{\mathscr{U}}| = \left(\binom{\Lambda-r}{t_a}^2\binom{\Lambda}{r}\right)$, as the number of users is $K = \binom{\Lambda}{r}$. Each demand tuple is processed or removed exactly once. The operations performed in each iteration consist of three components. First, computing the intersection between the user sets requires $O(r)$ time, since each user is connected to r access caches. Second, forming a coded multicast transmission involves combining at most $\min(t_a, r) + 1$ mini-subfiles using XOR operations, which requires $O(\min(t_a, r))$ time. Therefore, the overall time complexity of the algorithm is $O\left(\min(t_a, r)\binom{\Lambda}{r}^2\binom{\Lambda-r}{t_a}\right)$.

Decodability: Algorithm 1 generates two types of transmissions based on whether $I = \emptyset$ or $I \neq \emptyset$. For $|I| = 0$, each transmission is of the form $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'} \oplus \bigoplus_{i=1}^{\min(r,t_a)} swap_{no}(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}, i)$. Since $swap_{no}$ exchanges elements of $\mathscr{U}$ and $\mathscr{S}$, all resulting mini-subfiles have non-empty intersection between their subfile-index and user-index sets. By (11), each user $\mathscr{U}$ has access to all such mini-subfiles from the access caches it connects to. Hence, $\mathscr{U}$ can decode its desired mini-subfile using the XOR of available mini-subfiles. For $|I| > 0$, each transmission is of the form $flip(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'} \oplus \bigoplus_{i=1}^{\min(|I|,t_a)} swap_o(W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}, i))$. The function $swap_o$ exchanges elements of $\mathscr{U}$ and $\mathscr{S}$, so all mini-subfiles except $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$ and $W_{d_{\mathscr{U}'},\mathscr{S},\mathscr{U}}$ are available from access caches. The latter is present in the private cache of $\mathscr{U}$, enabling it to decode $W_{d_{\mathscr{U}},\mathscr{S},\mathscr{U}'}$. Since both transmission types are decodable and Algorithm 1 runs until all $\mathscr{D}_{\mathscr{U}}$ are empty, all users recover their requested files.

Performance of Algorithm 1: For $|I| = 0$, each transmission contains $\sum_{j=0}^{\min(r,t_a)}\binom{r}{j}\binom{t_a}{j} = \binom{t_a+r}{t_a}$ mini-subfiles, obtained by swapping j elements of $\mathscr{U}$ and $\mathscr{S}$. Hence, each transmission is a coded combination of $\binom{t_a+r}{r}$ mini-subfiles. For $|I| = i > 0$, each transmission has $2\sum_{j=1}^{\min(i,t_a)}\binom{t_a}{j}\binom{i}{j} = 2\binom{t_a+i}{i}$ mini-subfiles. The factor of 2 accounts for the additional mini-subfiles generated by the $flip$ function. As there are $\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}\binom{\Lambda-r-t_a}{r}$ mini-subfiles with $|I| = 0$, and $\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}\binom{r}{i}\binom{\Lambda-r-t_a}{r-i}$ mini-subfiles with $|I| = i$, and each file is divided into $F_{imp} = \binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}$ mini-subfiles, the rate is:

$$R_{imp} = \frac{\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}\binom{\Lambda-r-t_a}{r}}{\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}\binom{t_a+r}{r}} + \sum_{i=1}^{r-1} \frac{\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}\binom{r}{i}\binom{\Lambda-r-t_a}{r-i}}{2\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}\binom{t_a+i}{i}} \tag{13}$$

$$= \frac{\binom{\Lambda-r-t_a}{r}}{\binom{t_a+r}{t_a}} + \sum_{i=1}^{r-1} \frac{\binom{r}{i}\binom{\Lambda-t_a-r}{r-i}}{2\binom{t_a+i}{i}}. \tag{14}$$

Remark 2 It can be seen that for the case where $|I| = 0$, the coding gain, defined as the total number of users benefiting from each transmission, is $\binom{t_a+r}{r}$ and when $|I| = i$, the coding gain is $2\binom{t_a+i}{i}$. Hence, as the access cache memory replication factor t_a increases, the coding gain for both types of transmissions increases, while as the access degree r increases, the coding gain of the transmissions of $|I| = 0$ case increases.

Memory sharing for the improved CMAP coded caching scheme is described in Section 4.3.

Remark 3 Consider a $(\Lambda, r, M_a, M_p = 0, N)$-CMAP coded caching system where users have no private cache memory and rely solely on the access caches they connect to. This setting is identical to the CMACC network studied in [30]. In this case, each mini-subfile takes the form $W_{d_{\mathscr{U}},\mathscr{S},\emptyset}$ since $M_p = 0$. The cache contents accessible to user $\mathscr{U}$ are $\mathscr{Z}_{\mathscr{U}} = \{W_{d_{\mathscr{U}},\mathscr{S},\emptyset} : \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = t_a, \mathscr{U} \cap \mathscr{S} \neq \emptyset\}$. The corresponding subpacketization is $F_{imp} = \binom{\Lambda}{t_a}$, which equals that of the MAN scheme for the CMACC network [30]. Since $I = \mathscr{U} \cap \emptyset = \emptyset$ for every mini-subfile, all transmissions correspond to the $|I| = 0$ case and are of the form $W_{d_{\mathscr{U}},\mathscr{S},\emptyset} \oplus \bigoplus_{i=1}^{min(r,t_a)} swap_{no}(W_{d_{\mathscr{U}},\mathscr{S},\emptyset}, i)$. Each transmission thus combines $\binom{t_a+r}{r}$ mini-subfiles. A transmission is generated for every pair $(\mathscr{U}, \mathscr{S})$ such that $\mathscr{U} \cap \mathscr{S} = \emptyset$, i.e., for each subset of $[\Lambda]$ of size $t_a + r$. Therefore, the achievable rate is $R_{imp} = \frac{\binom{\Lambda}{t_a+r}}{\binom{\Lambda}{t_a}}$, which matches the rate of the MAN scheme for the CMACC network [30].

Hence, the improved scheme specializes to the scheme present in [30], when private caches have no memory.

Note that while the CMAP setting is an extension of the settings considered in the MAN scheme [1] and the CMACC scheme [30], the improved scheme itself is not an extension.

4.3 *Memory sharing*

We define $t_a = \frac{\Lambda M_a}{N}$ and $t_p = \frac{KM_p}{N}$, where M_a and M_p are used to denote the memory of the access and private cache respectively, for this section. We now explain how memory sharing is done.

Case 1: $t_a \notin \mathbb{Z}^+$ and $t_p \in \mathbb{Z}^+$. Since $t_a \notin \mathbb{Z}^+$, we define $M_{a,1} = \frac{N\lceil t_a \rceil}{\Lambda}$ and $M_{a,2} = \frac{N\lfloor t_a \rfloor}{\Lambda}$. Using these, we can express $M_a = \delta_1 M_{a,1} + (1-\delta_1) M_{a,2}$, for $0 \leq \delta_1 \leq 1$.

The file contents being stored in access caches are divided into δ_1 and $1-\delta_1$ fractions and placed in the access caches using the placement strategies defined for the scheme.

Extension scheme: Each file-part W_n^a is split as W_n^{a,δ_1} and $W_n^{a,1-\delta_1}$ of sizes $\delta_1(1-\gamma)B$ and $(1-\delta_1)(1-\gamma)B$ bits. Next, W_n^{a,δ_1} and $W_n^{a,1-\delta_1}$ are divided as $W_n^{a,\delta_1} = \{W_{n,T_1}^{a,\delta_1} : T_1 \subset [\Lambda], |T_1| = \lceil t_1 \rceil\}$ and $W_n^{a,1-\delta_1} = \{W_{n,T_2}^{a,1-\delta_1} : T_2 \subset [\Lambda], |T_2| = \lfloor t_1 \rfloor\}$, respectively. Each access cache is populated using W_{n,T_1}^{a,δ_1} and $W_{n,T_2}^{a,1-\delta_1}$ using the placement given by (6) for $\lceil t_a \rceil$ and $\lfloor t_a \rfloor$, respectively. Hence, each access cache stores $N\frac{\binom{\Lambda-1}{\lceil t_a \rceil - 1}}{\binom{\Lambda}{\lceil t_a \rceil}}(\delta_1(1-\gamma))B + N\frac{\binom{\Lambda-1}{\lfloor t_a \rfloor - 1}}{\binom{\Lambda}{\lfloor t_a \rfloor}}((1-\delta_1)(1-\gamma))B = M_a B$, satisfying its memory constraint. Thus, we have $R_{ext} = (1-\gamma)R^a + \gamma\frac{\binom{K}{1+t_p}}{\binom{K}{t_p}}$ with $R^a = \delta_1 \frac{\binom{\Lambda}{\lceil t_a \rceil + r}}{\binom{\Lambda}{\lceil t_a \rceil}} + (1-\delta_1)\frac{\binom{\Lambda}{\lfloor t_a \rfloor + r}}{\binom{\Lambda}{\lfloor t_a \rfloor}}$.

Improved scheme: The file W_n is split into $W_n^{\delta_1}$ and $W_n^{(1-\delta_1)}$ of $\delta_1 B$ and $(1-\delta_1)B$ bits, respectively. Next, $W_n^{\delta_1}$ and $W_n^{(1-\delta_1)}$ are further broken down into subfiles as $W_n^{\delta_1} = \{W_{n,\mathscr{S}}^{\delta_1} : \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = \lceil t_a \rceil\}$ and $W_n^{(1-\delta_1)} = \{W_{n,\mathscr{S}}^{(1-\delta_1)} : \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = \lfloor t_a \rfloor\}$, respectively. The access caches are filled with subfiles $W_{n,\mathscr{S}}^{\delta_1}$ and $W_{n,\mathscr{S}}^{(1-\delta_1)}$ as described in (11) for $\lceil t_a \rceil$ and $\lfloor t_a \rfloor$, respectively. Thus, every access cache stores $\frac{N\delta_1\binom{\Lambda-1}{\lceil t_a \rceil - 1}B}{\binom{\Lambda}{\lceil t_a \rceil}} + \frac{N(1-\delta_1)\binom{\Lambda-1}{\lfloor t_a \rfloor - 1}B}{\binom{\Lambda}{\lfloor t_a \rfloor}} = M_a B$ bits, satisfying its memory constraint.

The private caches of the users will be populated with the mini-subfiles of $W_{n,\mathscr{S}}^{\delta_1}$ and $W_{n,\mathscr{S}}^{(1-\delta_1)}$ as described in (12) for $\lceil t_a \rceil$ and $\lfloor t_a \rfloor$, respectively. Every private cache stores $\frac{\delta_1 N\binom{\Lambda-r}{\lceil t_a \rceil}}{\binom{\Lambda-\lceil t_a \rceil}{r}\binom{\Lambda}{\lceil t_a \rceil}} + \frac{(1-\delta_1)N\binom{\Lambda-r}{\lfloor t_a \rfloor}}{\binom{\Lambda-\lfloor t_a \rfloor}{r}\binom{\Lambda}{\lfloor t_a \rfloor}} = M_p$ files, satisfying its memory constraint. The rate is $R_{M_a} = \delta_1 R_{M_{a,1}} + (1-\delta_1) R_{M_{a,2}}$.

Case 2: $t_a \in \mathbb{Z}^+$ and $t_p \notin \mathbb{Z}^+$. Since $t_p \notin \mathbb{Z}^+$, we define $M_{p,1} = \frac{N\lceil t_p \rceil}{K}$ and $M_{p,2} = \frac{N\lfloor t_p \rfloor}{K}$. Hence, M_p can be expressed as $M_p = \delta_2 M_{p,1} + (1-\delta_2) M_{p,2}$, for $0 \le \delta_2 \le 1$.

Similar to *Case 1*, the file contents being stored in the private caches are divided into δ_2 and $1-\delta_2$ fraction. These file contents are then placed in the private caches using the placement strategy of the scheme.

Extension scheme: Each file-part W_n^p is split into two components of size $\delta_2 \gamma B$ bits and $(1-\delta_2)\gamma B$ bits, denoted as W_n^{p,δ_2} and $W_n^{p,1-\delta_2}$, respectively. Next, W_n^{p,δ_2} is divided as $W_n^{p,\delta_2} = \{W_{n,T_3}^{p,\delta_2} : T_3 \subset [K], |T_3| = \lceil t_p \rceil\}$ and $W_n^{p,1-\delta_2}$ is divided as $W_n^{p,1-\delta_2} = \{W_{n,T_4}^{p,1-\delta_2} : T_4 \subset [K], |T_4| = \lfloor t_p \rfloor\}$.

Each private cache is then filled using W_{n,T_3}^{p,δ_2} and $W_{n,T_4}^{p,1-\delta_2}$ using the placement given by (5) for $\lceil t_p \rceil$ and $\lfloor t_p \rfloor$, respectively. Hence, each private cache stores $N\frac{\binom{K-1}{\lceil t_p \rceil - 1}}{\binom{K}{\lceil t_p \rceil}}\delta_2\gamma B + N\frac{\binom{K-1}{\lfloor t_p \rfloor - 1}}{\binom{K}{\lfloor t_p \rfloor}}(1-\delta_2)\gamma B = M_p B$, satisfying its memory constraint. Hence, we have $R^p = \delta_2 \frac{\binom{K}{\lceil t_p \rceil + 1}}{\binom{K}{\lceil t_p \rceil}} + (1-\delta_2)\frac{\binom{K}{\lfloor t_p \rfloor + 1}}{\binom{K}{\lfloor t_p \rfloor}}$, which gives $R_{ext} = (1-\gamma)\frac{\binom{\Lambda}{t_a+r}}{\binom{\Lambda}{t_a}} + \gamma R^p$.

Case 3: $t_a \notin \mathbb{Z}^+$ and $t_p \notin \mathbb{Z}^+$. The memory point (M_a, M_p) is achieved via a convex combination as $(M_a, M_p) = \delta_1(1-\delta_2)(M_{a,1}, M_{p,2}) + \delta_2(1-\delta_1)(M_{a,2}, M_{p,1}) + \delta_1\delta_2(M_{a,1}, M_{p,1}) + (1-\delta_1)(1-\delta_2)(M_{a,2}, M_{p,2})$. Therefore, the rate R_{ext} is obtained via the same convex combination as $R_{ext} = \delta_1(1-\delta_2)R_{ext}^{(M_{a,1},M_{p,2})} + \delta_2(1-\delta_1)R_{ext}^{(M_{a,2},M_{p,1})} + \delta_1\delta_2 R_{ext}^{(M_{a,1},M_{p,1})} + (1-\delta_1)(1-\delta_2)R_{ext}^{(M_{a,2},M_{p,2})}$, where $R_{ext}^{M_1,M_2} = (1-\gamma)\frac{\binom{\Lambda}{t_1+r}}{\binom{\Lambda}{t_1}} + \gamma\frac{\binom{K}{t_1+t_2}}{\binom{K}{t_2}}$, with $t_1 = \frac{\Lambda M_1}{N}$ and $t_2 = \frac{KM_2}{N}$.

4.4 α-bound

The proof of Theorem 5 formulates the delivery phase as an ICP $\mathscr{I}$, similar to [43]. We derive a lower bound on $\alpha(\mathscr{I})$, which directly lower bounds the number of transmissions required in the delivery phase.

Let $\mathscr{U}_i$ denote the i^{th} user, $i \in \left[\binom{\Lambda}{r}\right]$, and $\mathscr{U}_j^{\mathscr{S}}$ denote the j^{th} user requesting subfile $\mathscr{S}$, $j \in \left[\binom{\Lambda-t_a}{r}\right]$, both indexed lexicographically. We construct the set $B(\mathbf{d}) = B_1(\mathbf{d}) \cup B_2(\mathbf{d})$, whose elements are messages of the ICP $\mathscr{I}$ such that the set of their indices forms a generalized independent set, where:

$$B_1(\mathbf{d}) = \bigcup_{i=1}^{m'-1} \bigcup_{k=i+1}^{\binom{\Lambda-t_a}{r}} \left\{ W_{d_{\mathscr{U}_i},\mathscr{S},\mathscr{U}_k^{\mathscr{S}}} : \mathscr{S} \subseteq [r+i,\Lambda], |S| = t_a \right\}$$

and,

$$B_2(\mathbf{d}) = \bigcup_{m=m'}^{\binom{\Lambda-t_a}{r}} \bigcup_{k=m+1}^{\binom{\Lambda-t_a}{r}} \left\{ W_{d_{\mathscr{U}_m^{\mathscr{S}'}},\mathscr{S}',U_k^{\mathscr{S}'}} \right\},$$

where $\mathscr{S}' = [\Lambda - t_a + 1, \Lambda]$ and $m' = \Lambda - r - t_a + 2$. Let $H(\mathbf{d})$ denote the set of indices of messages in $B(\mathbf{d})$.

Claim: $H(\mathbf{d})$ forms a generalized independent set.

Each message in $B(\mathbf{d})$ is demanded by a unique receiver; thus all singleton subsets of $H(\mathbf{d})$ lie in $\mathscr{J}(\mathscr{I})$. Consider any subset $C = \{W_{d_{\mathscr{U}_{i_1}},\mathscr{S}_{j_1},\mathscr{U}'_{l_1}}, \ldots, W_{d_{\mathscr{U}_{i_c}},\mathscr{S}_{j_c},\mathscr{U}'_{l_c}}\} \subseteq B(\mathbf{d})$, with $i_1 \le i_2 \le \cdots \le i_c$. The receiver demanding $W_{d_{\mathscr{U}_{i_1}},\mathscr{S}_{j_1},\mathscr{U}'_{l_1}}$ has no other message from C as side information. Hence, all indices in C lie in $\mathscr{J}(\mathscr{I})$, implying every subset of $H(\mathbf{d})$ lies in $\mathscr{J}(\mathscr{I})$. Thus, $H(\mathbf{d})$ is a generalized independent set and $\alpha \ge |H(\mathbf{d})| = |B(\mathbf{d})|$. We now compute $|B(\mathbf{d})|$. Since $B_1(\mathbf{d})$ and $B_2(\mathbf{d})$ are disjoint, we have $|B(\mathbf{d})| = |B_1(\mathbf{d})| + |B_2(\mathbf{d})|$. The number of subfiles corresponding to

user $\mathscr{U}_i$ in $B_1(\mathbf{d})$ is $\binom{\Lambda-r-i+1}{t_a}$ and the number of mini-subfiles corresponding to these subfiles is $\binom{\Lambda-t_a}{r}-i$. Thus, $|B_1(\mathbf{d})|=\sum_{i=0}^{\Lambda-r-t_a}\binom{\Lambda-r-i}{t_a}\left(\binom{\Lambda-t_a}{r}-1\right)-\sum_{i=0}^{\Lambda-r-t_a} i\binom{\Lambda-r-i}{t_a}$. Simplifying using Hockey-Stick identity, we have $|B_1(\mathbf{d})|=\binom{\Lambda-t_a}{r}\binom{\Lambda-r+1}{t_a+1}-\binom{\Lambda-r+2}{t_a+2}$. We now consider $B_2(\mathbf{d})$. For a user $\mathscr{U}_m^{\mathscr{S}'}$, there are $\binom{\Lambda-t_a}{r}-m$ mini-subfiles of the subfile $\mathscr{S}'$ in $B_2(\mathbf{d})$ which implies $|B_2(\mathbf{d})|=\sum_{m=\Lambda-r-t_a+2}^{\binom{\Lambda-t_a}{r}}\left(\binom{\Lambda-t_a}{r}-m\right)=\frac{\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-2\right)\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-1\right)}{2}$. Combining, we have

$$\alpha\geq\binom{\Lambda-t_a}{r}\binom{\Lambda-r+1}{t_a+1}-\binom{\Lambda-r+2}{t_a+2}+\frac{\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-2\right)\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-1\right)}{2}.$$

Since α is the cardinality of the maximal generalized independent set, we have $\alpha\geq|H(\mathbf{d})|=|B(\mathbf{d})|$. The theorem is proved since α lower bounds T.

5. Numerical comparison

In this section, we compare the rate R_{ext} of the extension scheme in Theorem 3 and the rate R_{imp} of the proposed scheme in Theorem 4 with the bounds in Theorem 1 and Theorem 2, and Corollary 1, for a CMAP system with $\Lambda=6$ access caches, N files, and $K=\binom{\Lambda}{r}$ users, where $N=K$ and $M_p=\frac{N}{K}=1$. The two schemes are also compared in terms of subpacketization.

We denote the normalized lower bound on the rate from Corollary 1 as $LB(Corr.\ 1)=\frac{\binom{\Lambda-r+1}{t_a+1}}{\binom{\Lambda}{t_a}}-\frac{(\Lambda-r+2)(\Lambda-r+1)}{(t_a+2)(t_a+1)}{\binom{\Lambda}{r}}+\frac{\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-2\right)\left(\binom{\Lambda-t_a}{r}-\Lambda+r+t_a-1\right)}{2\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}}$. This expression corresponds to the lower bound on the rate obtained by normalizing the lower bound on the number of transmissions (Theorem 5) by the subpacketization level, ensuring consistency when comparing with the rate expressions plotted in the figures. We provide numerical plots of the rate R_{imp} and R_{ext} for different values of the access degree r and the access cache memory replication factor t_a for a system with $\Lambda=6$ access caches, N files, and $K=\binom{\Lambda}{r}$ users such that $N=K$, and $M_p=\frac{N}{K}=1$. The three sets of plots in Figure 2 correspond to $r=2,3$, and $r=4$ with t_a taking values in $[\Lambda]$.

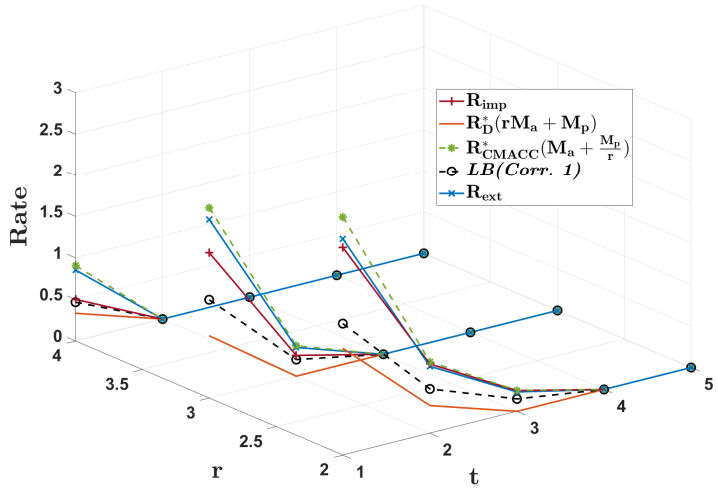

Figure 2. Rate vs. r and t_a for $M_p=\frac{N}{K}$

In Figure 3 to Figure 6, the rate R_{imp}, R_{ext}, the rate of the MAN scheme [1] such that each cache has a memory of $rM_a + M_p$, the rate of the MAN scheme for CMACC network [30] such that each cache has a storage capacity of $M_a + \frac{M_p}{r}$, the lower bound in Theorem 2, and $LB(Corr.\ 1)$ have been plotted. Further, these figures illustrate Remark 2.

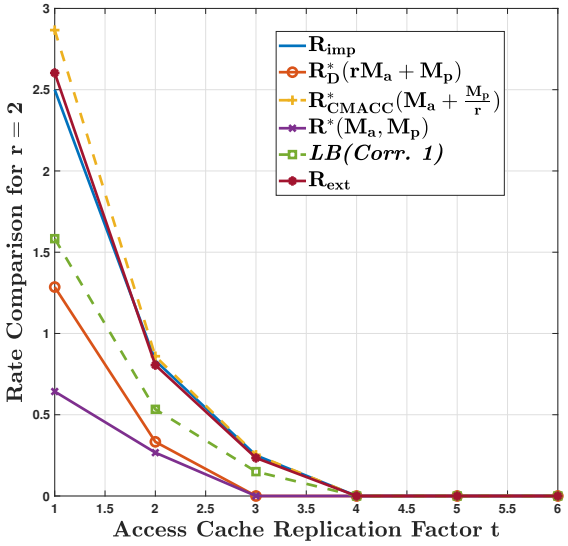

Figure 3. Rate vs. t_a for $r = 2$ and $M_p = \frac{N}{K}$

Figures 3 and 4 show that for fixed r, the rate R_{imp} moves closer to $LB(Corr.\ 1)$ as t_a increases, consistent with Remark 2. Figure 5 and 6, which plot R_{imp} versus r for $M_a = \frac{N}{6}$ and $M_a = \frac{N}{3}$, respectively, show that R_{imp} also approaches the lower bound as r increases for fixed t_a.

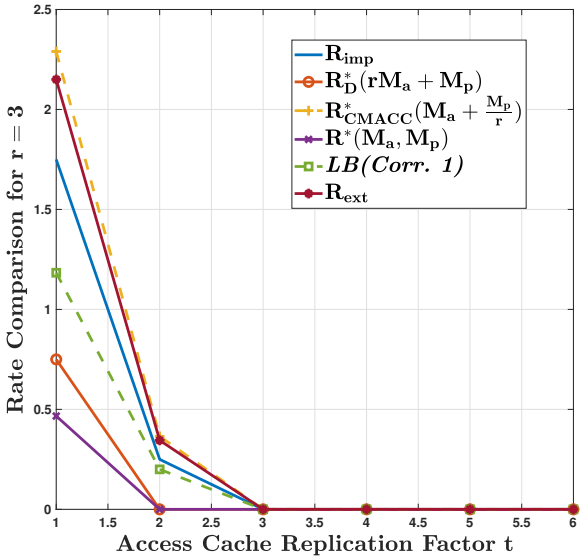

Figure 4. Rate vs. t_a for $r = 3$ and $M_p = \frac{N}{K}$

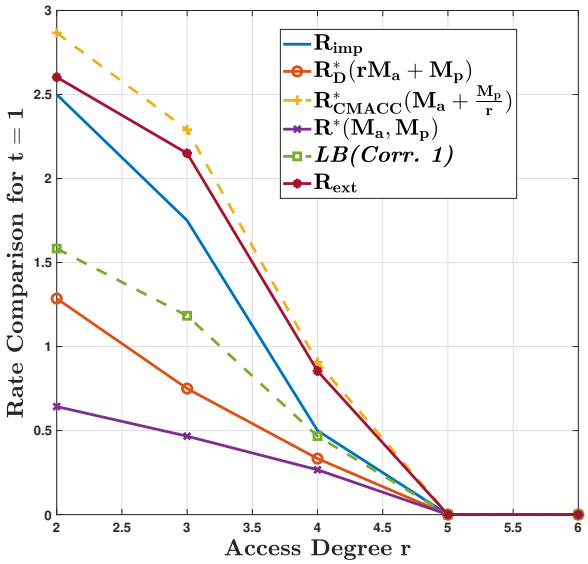

Figure 5. Rate vs. r for $t_a = 1$ and $M_p = \frac{N}{K}$

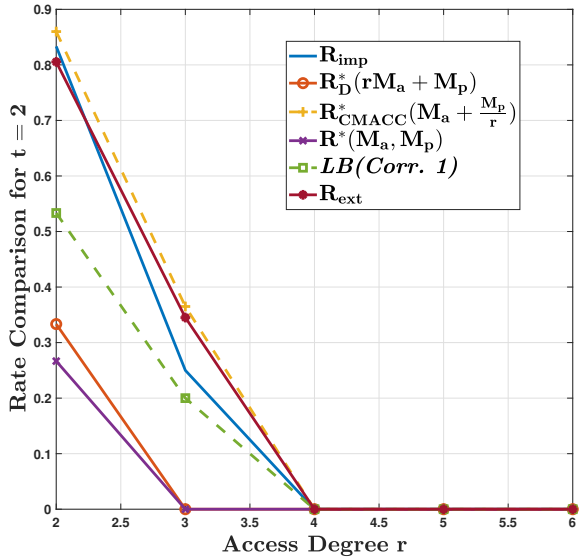

Figure 6. Rate vs. r for $t_a = 2$ and $M_p = \frac{N}{K}$

Observe that for $r = 2$ in Figure 3, the extension scheme attains a slightly lower rate than the improved scheme for $t_a \in [2,4]$. But this improvement in rate comes with substantially increased subpacketization as shown in Table 1. For the other cases (Figure 4 - Figure 6), the improved scheme outperforms the extension scheme in both rate and subpacketization.

For the CMAP coded caching system considered in this section, we have $\gamma = \frac{M_p}{M_p + M'} = \frac{1}{1+M'}$, where M' is the fraction of each file available to the user via the access caches. As t_a increases, so does M_a, which leads to an increase in M', leading to a decrease in γ. Since $M_p = 1$, we have $t^D_{ext} = \frac{1}{\gamma}$, that is, $t^D_{ext} = 1 + M'$. We know from [1] that $t^D_{ext} = \frac{KM^D_{ext}}{N}$ and since $N = K$, it follows that $M^D_{ext} = 1 + M'$. Similarly, we have $t^C_{ext} = \frac{t}{1-\gamma}$, leading to $t^C_{ext} = t_a\left(1 + \frac{1}{M'}\right)$. Using $t^C_{ext} = \frac{\Lambda M^C_{ext}}{N}$ from [30], we have $M^C_{ext} = M_a\left(1 + \frac{1}{M'}\right)$. Thus, as t_a increases, both M^D_{ext} and M^C_{ext} increase due to the growth in M_a, leading to a reduction in the rate.

Table 1. Subpacketization levels of the improved scheme and the extension scheme for selected values of r and t_a in a CMAP system with $\Lambda = 6$, $N = K = 15$, and $M_p = \frac{N}{K}$

t_a	$r=2$		$r=3$	
	F_{ext}	F_{im-p}	F_{ext}	F_{imp}
$t_a = 1$	5,012	60	167,970	60
$t_a = 2$	3,019	90	1,160	60
$t_a = 3$	124	60	—	—

6. Optimality analysis

In this section, we first present a novel placement strategy for the general CMAP coded caching system that accommodates multiple values of the private cache memory M_p. This placement strategy applies to arbitrary system parameters Λ, r, M_a, and M_p, and we also describe how it operates when either $t_a = \frac{\Lambda M_a}{N}$ or $t_p = \frac{KM_p}{N}$ is not an integer. We then specialize to the case of $\Lambda = 4$ access caches and examine various configurations of r, M_a, and M_p. For each such configuration, we describe the corresponding server transmissions and prove the optimality of the resulting scheme. Finally, we prove the order-optimality of the scheme in the regime where $\Lambda = r + t_a + 1$.

We now describe the novel placement strategy.

Placement Policy: Each file is split into $\binom{\Lambda}{t_a}$ non-overlapping, equal-sized subfiles $W_n = \{W_{n,\mathscr{S}} : \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = t_a, \forall n \in [N]\}$, where $t_a = \frac{\Lambda M_a}{N}$, is the access cache replication factor. The contents of the access cache i is

$$Z_i = \{W_{n,\mathscr{S}} : i \in \mathscr{S}, \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = t_a, \forall n \in [N]\}, \tag{15}$$

for $i \in [\Lambda]$. Each access cache stores $\frac{N\binom{\Lambda-1}{t_a-1}}{\binom{\Lambda}{t_a}} = M_a$ files, satisfying its memory constraint.

We will now describe how the private cache placement. Each subfile is further split into $\binom{\binom{\Lambda-t_a}{r}}{t_p}$ mini-subfiles as $W_{n,S} = \{W_{n,S,T_{i,i\in[t_p]}} : T_i \subset [\Lambda] \setminus S, |T_i| = r, \forall i \in [t_p]\}$, where $t_p = \frac{KM_p}{N} \in \mathbb{Z}^+$ is the private cache memory replication factor. $W_{n,\mathscr{S},\mathscr{T}_{i,u\in[t_p]}}$ denotes the mini-subfile of subfile $\mathscr{S}$ of file n present in the private caches of users $\mathscr{T}_{i,i\in[t_p]}$. The private cache of user $\mathscr{U}$ stores

$$Z^p_{\mathscr{U}} = \{W_{n,\mathscr{S},\mathscr{T}_{i,i\in[t_p]}} : \mathscr{S} \subseteq [\Lambda] \setminus \mathscr{U}, \mathscr{T}_i \subseteq [\Lambda] \setminus \mathscr{S}, \forall i \in [t_p],$$

$$|\{i \in [t_p] : \mathscr{T}_i = \mathscr{U}\}| = 1, |\mathscr{S}| = t_a, |\mathscr{T}_i| = r, \forall n \in [N]\}. \tag{16}$$

The private cache of each user stores $\frac{N\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-t_a}{r}-1}{t_p-1}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}} = M_p$ files, satisfying its memory constraint. Notably, for $t_p = 1$, this placement reduces to the one discussed in Section 4.2We will now consider the case where $t_a \notin \mathbb{Z}^+$ or $t_p \notin \mathbb{Z}^+$.

Case 1: $t_a \notin \mathbb{Z}^+, t_p \in \mathbb{Z}^+$: Consider M_a such that $t_a = \frac{\Lambda M_a}{N} \notin \mathbb{Z}^+$. Let $M_1 = \frac{\lceil t_a \rceil N}{\Lambda}$ and $M_2 = \frac{\lfloor t_a \rfloor N}{\Lambda}$. Since $M = \frac{t_a N}{\Lambda}$, we know that $M_2 \leq M_a \leq M_1$. Hence, $M_a = \alpha_1 M_1 + (1-\alpha_1)M_2$, for some $0 \leq \alpha_1 \leq 1$. The file W_n is split into $W_n^{\alpha_1}$ and $W_n^{(1-\alpha_1)}$ of sizes $\alpha_1 B$ and $(1-\alpha_1)B$ bits, respectively, $\forall n \in [N]$. These parts are further divided as $W_n^{\alpha_1} = \{W^{\alpha_1}_{n,\mathscr{S}} : \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = \lceil t_a \rceil\}$ and $W_n^{(1-\alpha_1)} = \{W^{(1-\alpha_1)}_{n,\mathscr{S}} : \mathscr{S} \subseteq [\Lambda], |\mathscr{S}| = \lfloor t_a \rfloor\}$. The access caches are filled with subfiles $W^{\alpha_1}_{n,\mathscr{S}}$ for $t = \lceil t_a \rceil$ and with subfiles $W^{(1-\alpha_1)}_{n,\mathscr{S}}$ for $t = \lfloor t_a \rfloor$, as described in (11). Thus, every access cache stores $N\alpha_1\binom{\Lambda-1}{\lceil t_a \rceil - 1}B + N(1-\alpha_1)\binom{\Lambda-1}{\lfloor t_a \rfloor - 1}B$

bits, which is equivalent to $N\alpha_1 \frac{\binom{\Lambda-1}{\lceil t_a\rceil-1}}{\binom{\Lambda}{\lceil t_a\rceil}} + N(1-\alpha_1)\frac{\binom{\Lambda-1}{\lfloor t_a\rfloor-1}}{\binom{\Lambda}{\lfloor t_a\rfloor}} = M_a$ files, satisfying its memory constraint. The private caches of the users will be populated with the mini-subfiles of $W^{\alpha}_{n,\mathscr{S}}$ and $W^{(1-\alpha)}_{n,\mathscr{S}}$ as described in (16) for $\lceil t_a\rceil$ and $\lfloor t_a\rfloor$. Every private cache stores $\frac{\alpha_1 N\binom{\Lambda-r}{\lceil t_a\rceil}\binom{\binom{\Lambda-\lceil t_a\rceil}{r}-1}{t_p-1}}{\binom{\Lambda}{\lceil t_a\rceil}\binom{\binom{\Lambda-\lceil ta\rceil}{r}}{t_p}} + \frac{(1-\alpha_1)N\binom{\Lambda-r}{\lfloor t_a\rfloor}\binom{\binom{\Lambda-\lfloor t_a\rfloor}{r}-1}{t_p-1}}{\binom{\Lambda}{\lfloor t_a\rfloor}\binom{\binom{\Lambda-\lfloor ta\rfloor}{r}}{t_p}} = M_p$ files, satisfying its memory constraint.

Case 2: $t_a \in \mathbb{Z}^+, t_p \notin \mathbb{Z}^+$: Let $M_3 = \frac{\lceil t_p\rceil N}{K}$ and $M_4 = \frac{\lfloor t_p\rfloor N}{K}$. Since $M_p = \frac{t_p N}{K}$, we know that $M_p = \alpha_2 M_3 + (1-\alpha_2)M_4$, for some $0 \le \alpha_2 \le 1$. The subfile $W_{n,\mathscr{S}}$ is split into $W^{\alpha_2}_{n,\mathscr{S}}$ and $W^{(1-\alpha_2)}_{n,\mathscr{S}}$ of sizes $\alpha_2 B_s$ and $(1-\alpha_2)B_s$ bits, respectively, where B_s is the size of a subfile. These are further divded as $W^{\alpha_2}_{n,\mathscr{S}} = \{W^{\alpha_2}_{n,\mathscr{S},\mathscr{T}_1,\mathscr{T}_2,\cdots,\mathscr{T}_{\lceil t_p\rceil}} : \mathscr{T}_i \subseteq [\Lambda]\setminus S, |\mathscr{T}_i| = r, \forall i \in [\lceil t_p\rceil]\}$ and $W^{(1-\alpha_2)}_{n,\mathscr{S}} = \{W^{(1-\alpha_2)}_{n,\mathscr{S},\mathscr{T}_1,\mathscr{T}_2,\cdots,\mathscr{T}_{\lfloor t_p\rfloor}} : \mathscr{T}_i \subseteq [\Lambda]\setminus S, |\mathscr{T}_i| = r, \forall i \in [\lfloor t_p\rfloor]\}$, respectively. The private caches of the users are filled with the mini-subfiles as described in (16), with each private cache storing $N\alpha_2 \frac{\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-ta}{r}-1}{\lceil t_p\rceil-1}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-ta}{r}}{\lceil t_p\rceil}} + N(1-\alpha_2)\frac{N\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-ta}{r}-1}{\lfloor t_p\rfloor-1}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-ta}{r}}{\lfloor t_p\rfloor}} = M_p$ files.

We now examine all non-trivial combinations of M_a, M_p, and r, where every user can access only a part of the library. From the access caches, a user obtains $N\left(\binom{\Lambda}{t_a} - \binom{\Lambda-r}{t_a}\right)$, each of which yields all associated mini-subfiles. Thus, the total number of mini-subfiles obtained from the access caches are $N\left(\binom{\Lambda}{t_a} - \binom{\Lambda-r}{t_a}\right)\binom{\binom{\Lambda-ta}{r}}{t_p}$. Users receive $N\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-ta}{r}-1}{t_p-1}$ mini-subfiles from its private cache. Hence, the total number of mini-subfiles accessible to a user is $N\left(\left(\binom{\Lambda}{t_a} - \binom{\Lambda-r}{t_a}\right)\binom{\binom{\Lambda-ta}{r}}{t_p} + \binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-ta}{r}-1}{t_p-1}\right)$. As each file is broken into $\binom{\Lambda}{t_a}\binom{\binom{\Lambda-ta}{r}}{t_p}$ mini-subfiles, every user has access to $N\left(1 - \frac{\binom{\binom{\Lambda-ta}{r}-t_p}{\binom{\Lambda}{r}}}{}\right)$ files.

Hence, for $\Lambda = 4$, the only non-trivial (r, M_a, M_p) triplets are $(2, \frac{N}{4}, \frac{N}{6}) = (r=2, t_a=1, t_p=1)$ and $(2, \frac{N}{4}, \frac{N}{3}) = (r=2, t_a=1, t_p=2)$. For all the other points, the users have access to the entire library. We discuss optimality for both these cases.

6.1 *Optimality for* ($\Lambda = 4$, $r = 2$, $t_a = 1$, $t_p = 1$)

We establish the optimality of the improved scheme with respect to the worst-case rate achieved, under the assumptions that $N \ge K = \binom{\Lambda}{r}$, uncoded placement, and a regular delivery scheme. We begin this section by defining regular delivery schemes for coded caching systems.

Definition 5 A delivery scheme for a coded caching system is said to be g-regular if each transmission is a coded combination of g mini-subfiles.

We now show that,for the given setting, it is impossible to construct a valid transmission by combining five mini-subfiles. In a valid transmission, each user must have access to all the mini-subfiles in the transmission except the one it requests. Our goal is to construct a transmission involving five users that satisfies this constraint.

Consider the mini-subfile $W_{d_{12,3,14}}$. The users 13, 23, and 34 have access to this mini-subfile via access cache Z_3, while user 14 stores it in its private cache. Hence these users can participate in the transmission.

For user 13, in order for user 12 to decode the transmission, it must have access to the mini-subfile corresponding to user 13. Therefore, that mini-subfile must either be stored in access cache Z_2 or in the private cache of user 12. i.e., it must be of the form $W_{d_{13,2,\mathscr{T}_1}}$ or $W_{d_{13,\mathscr{S}_1,12}}$, where $\mathscr{T}_1 \neq 12$, and $2 \notin \mathscr{S}_1$.

Next, consider the user 23. For the transmission to be decodable by users 12 and 13, the mini-subfile of user 23 must be of the form $W_{d_{23,1,\mathscr{T}_2}}$, where $\mathscr{T}_2 \neq \{12,13\}$.

For user 34, to ensure decodability by users 12, 13 and 23, its mini-subfile must be either $W_{d_{34,1,23}}$ or $W_{d_{34,2,13}}$.

Finally, consider the user 14. For the transmissions to be decodable by users 12, 13, 23 and 34, the mini-subfile for the user 14 must be of the form $W_{d_{14,2,\mathscr{T}_3}}$ or $W_{d_{14,3,\mathscr{T}_4}}$.

We will now try to contruct a transmission with the mini-subfiles $W_{d_{13},2,\mathscr{T}_1}$, $W_{d_{23},1,\mathscr{T}_2}$ and $W_{d_{34},1,23}$ and separately consider $W_{d_{14},2,\mathscr{T}_3}$ and $W_{d_{14},3,\mathscr{T}_4}$.

Case 1: Mini-subfile $W_{d_{14},2,\mathscr{T}_3}$: The transmission is

$$W_{d_{12},3,14} \oplus W_{d_{13},2,\mathscr{T}_1} \oplus W_{d_{23},1,\mathscr{T}_2} \oplus W_{d_{34},1,23} \oplus W_{d_{14},2,\mathscr{T}_3}.$$

For user 13 to decode this transmission, $\mathscr{T}_3$ must be equal to 13, giving $W_{d_{14},2,13}$. Thus, the transmission becomes

$$W_{d_{12},3,14} \oplus W_{d_{13},2,\mathscr{T}_1} \oplus W_{d_{23},1,\mathscr{T}_2} \oplus W_{d_{34},1,23} \oplus W_{d_{14},2,13}.$$

However, user 34 cannot decode since it does not have access to $W_{d_{14},2,13}$. Thus, user 34 must be excluded, yielding a valid transmission involving four mini-subfiles.

Case 2: Mini-subfile $W_{d_{14},3,\mathscr{T}_4}$: The transmission is

$$W_{d_{12},3,14} \oplus W_{d_{13},2,\mathscr{T}_1} \oplus W_{d_{23},1,\mathscr{T}_2} \oplus W_{d_{34},1,23} \oplus W_{d_{14},3,\mathscr{T}_4}.$$

For user 34 to decode, we have $\mathscr{T}_1 = 34$ and $\mathscr{T}_2 = 34$, giving

$$W_{d_{12},3,14} \oplus W_{d_{13},2,34} \oplus W_{d_{23},1,34} \oplus W_{d_{34},1,23} \oplus W_{d_{14},3,\mathscr{T}_4}.$$

However, user 14 lacks $W_{d_{13},2,34}$ and cannot decode the transmission. Hence, user 14 must be excluded, again resulting in a valid transmission with four mini-subfiles. By exhaustively examining all possible five-mini-subfile combinations, it can be verified that no valid transmission involving five mini-subfiles can be constructed. Therefore, the maximum number of mini-subfiles per valid transmission is four. For the given CMAP system, the improved scheme achieves this bound, with each transmission being a coded combination of four mini-subfiles, thereby proving its optimality.

6.2 *Optimality for* ($\Lambda = 4$, $r = 2$, $t_a = 1$, $t_p = 2$)

Consider a CMAP coded caching system with a central server storing six files $W_1, W_2, \ldots, W_6$ and $\Lambda = 4$ access caches, each of capacity $M_a = 1.5$ files. There are $K = 6$ users, each equipped with a private cache of size $M_p = 2$ files, where every user connects to a unique subset of $r = 2$ access caches. Since $t_a = 1$, each file W_n is split into $\binom{\Lambda}{t_a} = 4$ subfiles as $W_n = \{W_{n,1}, W_{n,2}, W_{n,3}, W_{n,4}\}$ for $n \in [6]$. The server populates access caches as:

$$Z_1 = \{W_{n,1}, \forall n \in [6]\},$$

$$Z_2 = \{W_{n,2}, \forall n \in [6]\},$$

$$Z_3 = \{W_{n,3}, \forall n \in [6]\}, \text{ and,}$$

$$Z_4 = \{W_{n,4}, \forall n \in [6]\}.$$

Each access cache stores $\frac{6}{4} = 1.5$ files, satisfying its memory constraint. Every subfile is divided into $\binom{\binom{\Lambda-t_a}{r}}{t_p} = 3$ mini-subfiles, where $t_p = \frac{KM_p}{N} = 2$. Since each file is split into four subfiles, the subpacketization level is $F = 12$. The server populates the private caches of the users with mini-subfiles of the subfiles unavailable to the user via the access caches as

$$Z_{12}^p = \{W_{n,3,12,14}, W_{n,3,12,24}, W_{n,4,12,13}, W_{n,4,12,23}, \forall n \in [6]\},$$

$$Z_{13}^p = \{W_{n,2,13,14}, W_{n,2,13,34}, W_{n,4,12,13}, W_{n,4,13,23}, \forall n \in [6]\},$$

$$Z_{14}^p = \{W_{n,2,13,14}, W_{n,2,14,34}, W_{n,3,12,14}, W_{n,3,14,24}, \forall n \in [6]\},$$

$$Z_{23}^p = \{W_{n,1,23,24}, W_{n,1,23,34}, W_{n,4,12,23}, W_{n,4,13,23}, \forall n \in [6]\},$$

$$Z_{24}^p = \{W_{n,1,23,24}, W_{n,1,24,34}, W_{n,3,12,24}, W_{n,3,14,24}, \forall n \in [6]\},$$

$$Z_{34}^p = \{W_{n,1,23,34}, W_{n,1,24,34}, W_{n,2,13,34}, W_{n,2,14,34}, \forall n \in [6]\}.$$

Each private cache stores $\frac{24}{12} = 2$ files, satisfying its memory constraint. The transmissions made by the server for the demand vector $\mathbf{d} = (d_{\mathscr{U}} : \mathscr{U} \subseteq [\Lambda], |\mathscr{U}| = r)$ are:

1. $W_{d_{12},3,14,24} + W_{d_{13},2,14,34} + W_{d_{14},3,12,24} + W_{d_{23},1,24,34} + W_{d_{24},3,12,14} + W_{d_{34},1,23,24}$.
2. $W_{d_{12},4,13,23} + W_{d_{13},4,13,23} + W_{d_{14},2,13,34} + W_{d_{23},4,12,13} + W_{d_{24},1,23,34} + W_{d_{34},2,13,14}$.

As both transmissions are decodable, the rate $R = \frac{2}{12} = \frac{1}{6}$.

6.2.1 *Proof of optimality*

Consider the cut-set bound derived in Theorem 2. For $s = 1$, we have $R^*(M_a, M_p) \geq 1 - \frac{rM_a + M_p}{N}$. For the case discussed in this section, we have $R^*(1.5, 2) \geq 1 - \frac{2\frac{N}{4} + \frac{N}{3}}{N} = 1 - \frac{1}{2} - \frac{1}{3} = \frac{1}{6}$. The scheme is optimal for this case since the cut-set bound is achieved.

6.3 *Order-optimality analysis for the $\Lambda = r + t_a + 1$ regime*

We now compare the rate achieved by the improved scheme with the lower bound on the optimal worst-case rate under uncoded placement, as derived in Corollary 1, for the case $\Lambda = t_a + r + 1$, where $t_a = \frac{\Lambda M_a}{N}$. Specifically, we upper bound $\frac{R_{imp}}{R^* M_a, M_p}$. Since $R^*_{M_a,M_p} \geq LB(Corr.\ 1)$, it follows that $\frac{R_{imp}}{R^*_{M_a,M_p}} \leq \frac{R_{imp}}{LB(Corr.\ 1)}$. Thus, an upper bound on $\frac{R_{imp}}{LB(Corr.\ 1)}$ also bounds $\frac{R_{imp}}{R^*_{M_a,M_p}}$. From Theorem 4, we have

$$R_{imp} = \frac{\binom{1}{r}}{\binom{t_a+r}{r}} + \sum_{i=1}^{r-1} \frac{\binom{r}{i}\binom{1}{r-i}}{2\binom{t_a+i}{t_a}} \tag{17}$$

$$= \frac{r}{2\binom{t_a+r-1}{t_a}}. \tag{18}$$

Equation (17) follows by substituting $\Lambda = t_a + r + 1$ in the rate expression given in Theorem 4. Since $r \geq 2$, $\binom{1}{r} = 0$, and only the term $i = r-1$ in the summation is nonzero, yielding (18). We now specialize $LB(Corr.\ 1)$ for the case $\Lambda = t_a + r + 1$ as follows

$$LB(Corr.\ 1) = \frac{t_a+2}{\binom{t_a+r+1}{t_a}} - \frac{t_a+3}{(t_a+1)\binom{t_a+r+1}{r}} + \frac{(r-2)(r-1)}{2\binom{t_a+r+1}{t_a}(r+1)} \tag{19}$$

$$= \frac{2r(t_a+2)-2+(r-1)(r-2)}{2(t_a+r+1)\binom{t_a+r}{t_a}}. \tag{20}$$

We obtain Equation (19) by substituting $\Lambda = t_a + r + 1$ into the expression for the $LB(Corr.\ 1))$ given in Corollary 1. Equation (20) is then derived by applying standard identities of the binomial coefficient, namely $\binom{n}{k} = \binom{n}{n-k}$ and $\binom{n+1}{k+1} = \frac{n+1}{k+1}\binom{n}{k}$. Hence, we have:

$$\frac{R_{imp}}{LB(Corr.\ 1)} = \frac{r(t_a+r+1)\binom{t_a+r}{t_a}}{(2r(t_a+2)-2+(r-1)(r-2))\binom{t_a+r-1}{t_a}}$$

$$= \frac{\Lambda(\Lambda-1)}{r(t_a+\Lambda)} \leq \frac{\Lambda(\Lambda-1)}{2(2\Lambda-3)} \leq 0.3\Lambda \tag{21}$$

Equation (21) follows from $\frac{\binom{n}{k}}{\binom{n-1}{k}} = \frac{n}{n-k}$ and $\Lambda = t_a + r + 1$. To obtain the inequality, we set $r = 2$ to minimize the denominator and thus obtain the maximum upper bound. Replacing $\frac{\Lambda-1}{2\Lambda-3}$ with its maximum value 0.6 yields the final expression.

The bound shows that $\frac{R_{imp}}{R^*_{M_a,M_p}} \leq 0.3\Lambda$, implying that the achievable rate of the improved scheme increases at most linearly with the number of access caches Λ. Since the number of users is $K = \binom{\Lambda}{r}$, which grows combinatorially with Λ, the bound scales much slower than the system size. This demonstrates the scalability and order-optimality of the improved scheme in CMAP systems with large Λ.

7. Conclusions

We studied the Combinatorial Multi-Access with Private caches (CMAP) coded caching system, which captures cache-aided networks in which users simultaneously rely on local storage and access multiple shared cache nodes. This setting generalizes existing coded caching models with either private or shared caches, and introduces additional combinatorial coupling between the two storage layers.

We first proposed an extension scheme that generalizes the MAN schemes in [1] and [30] to the CMAP setting, and established upper and lower bounds on the optimal worst-case rate under uncoded placement. We then designed an improved coded caching scheme for a fixed private cache size and derived an index-coding-based lower bound on the required number of transmissions under its placement policy. Numerical results demonstrated that the improved scheme significantly reduces subpacketization while achieving competitive rate performance, and approaches the derived lower bounds in regimes with large total accessible memory per user.

Furthermore, we characterized the fundamental performance limits of the CMAP system by establishing order-optimality for the regime $\Lambda = t_a + r + 1$, and exact optimality for all memory pairs when $\Lambda = 4$. These results provide the first systematic information-theoretic characterization of two-level multi-access caching networks with both private and shared storage, and offer fundamental design insights for future cache-aided network architectures.

Acknowledgment

This work was supported partly by the Science and Engineering Research Board (SERB) of the Department of Science and Technology (DST), Government of India, through J.C Bose National Fellowship to Prof. B. Sundar Rajan. In the preparation of this work, the authors used ChatGPT for language editing and improving clarity of presentation. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication. All technical analysis, results, and conclusions were developed independently by the authors without the use of AI tools.

Conflicts of interest

The authors declare no competing financial interest.

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856-2867, 2014. https://doi.org/10.1109/TIT.2014.2306938.

[2] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental limits of caching: improved bounds for users with small buffers," *IET Communications*, vol. 10, no. 17, pp. 2315-2318, 2016. https://doi.org/10.1049/iet-com.2015.1205.

[3] J. Gómez-Vilardebó, "Fundamental limits of caching: improved bounds with coded prefetching," 2016. [Online]. Available: http://arxiv.org/abs/1612.09071 [Accessed Jan. 17, 2026].

[4] M. Mohammadi Amiri and D. Gündüz, "Fundamental limits of coded caching: improved delivery rate-cache capacity tradeoff," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 806-815, 2017. https://doi.org/10.1109/TCOMM.2016.2638841.

[5] K. Wan, D. Tuninetti, and P. Piantanida, "On caching with more users than files," In Proc. 2016 IEEE International Symposium on Information Theory, Barcelona, Spain, Jul. 10-15, 2016, pp. 135-139. https://doi.org/10.1109/ISIT.2016.7541276.

[6] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281-1296, 2018. https://doi.org/10.1109/TIT.2017.2785237.

[7] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," In Proc. 2016 IEEE Information Theory Workshop, Cambridge, U.K., Sep. 11-14, 2016, pp. 161-165. https://doi.org/10.1109/ITW.2016.7606816.

[8] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029-1040, 2015. https://doi.org/10.1109/TNET.2014.2317316.

[9] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212-3229, 2016. https://doi.org/10.1109/TIT.2016.2557804.

[10] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 355-370, 2015. https://doi.org/10.1109/TIFS.2014.2375553.

[11] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. M. Prabhakaran, "Private coded caching," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 685-694, 2018. https://doi.org/10.1109/TIFS.2017.2765503.

[12] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2525-2553, 2019. https://doi.org/10.1109/COMST.2019.2908280.

[13] A. Pal and K. Kant, "A neighborhood aware caching and interest dissemination scheme for content centric networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3900-3917, 2021. https://doi.org/10.1109/TNSM.2021.3079326.

[14] Z. Chen, Z. Zhou, and C. Chen, "Code caching-assisted computation offloading and resource allocation for multi-user mobile edge computing," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4517-4530, 2021. https://doi.org/10.1109/TNSM.2021.3103533.

[15] L. Chhangte, N. Karamchandani, D. Manjunath, and E. Viterbo, "Towards a distributed caching service at the WiFi edge using Wi-Cache," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4489-4502, 2021. https://doi.org/10.1109/TNSM.2021.3105496.

[16] Q. Fan, X. Li, J. Li, Q. He, K. Wang, and J. Wen, "PA-cache: evolving learning-based popularity-aware content caching in edge networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1746-1757, 2021. https://doi.org/10.1109/TNSM.2021.3053645.

[17] S. Imai, K. Leibnitz, and M. Murata, "Statistical approximation of efficient caching mechanisms for one-timers," *IEEE Transactions on Network and Service Management*, vol. 12, no. 4, pp. 595-604, 2015. https://doi.org/10.1109/TNSM.2015.2501837.

[18] S. Qiu, Q. Fan, X. Li, X. Zhang, G. Min, and Y. Lyu, "OA-cache: oracle approximation-based cache replacement at the network edge," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3177-3189, 2023. https://doi.org/10.1109/TNSM.2023.3239664.

[19] N. Kamiyama, Y. Nakano, and K. Shiomoto, "Cache replacement based on distance to origin servers," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 848-859, 2016. https://doi.org/10.1109/TNSM.2016.2600240.

[20] K. Zheng, Y. Cui, X. Liu, X. Wang, X. Jiang, and J. Tian, "Asymptotic analysis of inhomogeneous information-centric wireless networks with infrastructure support," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5245-5259, 2018. https://doi.org/10.1109/TVT.2018.2810312.

[21] X. Liu, K. Zheng, J. Zhao, X.-Y. Liu, X. Wang, and X. Di, "Information-centric networks with correlated mobility," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4256-4270, 2017. https://doi.org/10.1109/TVT.2016.2602373.

[22] W. Xing, L. Liu, Y. Zhou, and J. Shi, "A low-complexity user-preference-aware decentralized coded caching based on user pairing," *IEEE Internet of Things Journal*, vol. 11, no. 20, pp. 33120-33132, 2024. https://doi.org/10.1109/JIOT.2024.3425637.

[23] L. Chhangte, N. Karamchandani, D. Manjunath, and E. Viterbo, "On index coded video delivery at the WiFi edge: performance and system design," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1442-1457, 2022. https://doi.org/10.1109/TNSM.2021.3138833.

[24] M. Dutta, A. Thomas, and B. S. Rajan, "Novel delivery algorithms for decentralized multi-access coded caching systems," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 502-515, 2025. https://doi.org/10.1109/TNSM.2025.3629715.

[25] Y. Zhou, L. Liu, L. Wang, N. Hui, X. Cui, J. Wu, et al., "Service-aware 6G: an intelligent and open network based on the convergence of communication, computing and caching," *Digital Communications and Networks*, vol. 6, no. 3, pp. 253-260, 2020. https://doi.org/10.1016/j.dcan.2020.05.003.

[26] E. Parrinello, A. Ünsal, and P. Elia, "Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2252-2268, 2020. https://doi.org/10.1109/TIT.2019.2955384.

[27] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded placement for systems with shared caches," In Proc. IEEE International Conference on Communications, Shanghai, China, May 20-24, 2019, pp. 1-6. https://doi.org/10.1109/ICC.2019.8761409.

[28] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3108-3141, 2017. https://doi.org/10.1109/TIT.2017.2664817.

[29] B. Serbetci, E. Parrinello, and P. Elia, "Multi-access coded caching: gains beyond cache-redundancy," In Proc. IEEE Information Theory Workshop, Visby, Sweden, Aug. 25-28, 2019. https://doi.org/10.1109/ITW44776.2019.8989128.

[30] P. N. Muralidhar, D. Katyal, and B. S. Rajan, "Maddah-Ali-Niesen scheme for multi-access coded caching," In Proc. IEEE Information Theory Workshop, Kanazawa, Japan, Oct. 17-21, 2021, pp. 1-6. https://doi.org/10.1109/ITW48936.2021.9611394.

[31] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching," *IEEE Transactions on Information Theory*, vol. 69, no. 2, pp. 1037-1056, 2023. https://doi.org/10.1109/TIT.2022.3193723.

[32] S. S. Meel and B. S. Rajan, "Secretive coded caching with shared caches," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2849-2853, 2021. https://doi.org/10.1109/LCOMM.2021.3094405.

[33] E. Peter, K. K. K. Namboodiri, and B. S. Rajan, "A secretive coded caching for shared cache systems using placement delivery arrays," In Proc. IEEE International Symposium on Information Theory, Espoo, Finland, Jun. 26-Jul. 1, 2022, pp. 1402-1407. https://doi.org/10.1109/ISIT50566.2022.9834824.

[34] E. Peter, K. K. Krishnan Namboodiri, and B. S. Rajan, "Coded caching with shared caches and private caches," *IEEE Transactions on Communications*, vol. 72, no. 8, pp. 4857-4872, 2024. https://doi.org/10.1109/TCOMM.2024.3383112.

[35] Y. Chen, M. Wen, E. Basar, Y.-C. Wu, L. Wang, and W. Liu, "Exploiting reconfigurable intelligent surfaces in edge caching: joint hybrid beamforming and content placement optimization," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7799-7812, 2021. https://doi.org/10.1109/TWC.2021.3087912.

[36] Y. Li, M. Xia, and Y.-C. Wu, "Caching at base stations with multi-cluster multicast wireless backhaul via accelerated first-order algorithms," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2920-2933, 2020. https://doi.org/10.1109/TWC.2020.2969149.

[37] J. Zhang, X. Hu, Z. Ning, E. C.-H, Ngai, L. Zhou, J. Wei, et al., "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4283-4294, 2019. https://doi.org/10.1109/JIOT.2018.2875917.

[38] C. Zhang, J. Liu, F. Chen, Y. Cui, E. C.-H. Nhai, Y. Hu, "Dependency- and similarity-aware caching for HTTP adaptive streaming," *Multimedia Tools and Applications*, vol. 77, pp. 1453-1474, 2018. https://doi.org/10.1007/s11042-016-4308-z.

[39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[40] Y. Birk and T. Kol, "Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2825-2830, 2006. https://doi.org/10.1109/TIT.2006.874540.

[41] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479-1494, 2011. https://doi.org/10.1109/TIT.2010.2103753.

[42] S. H. Dau, V. Skachek, and Y. M. Chee, "Error correction for index coding with side information," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1517-1531, 2013. https://doi.org/10.1109/TIT.2012.2227674.

[43] N. S. Karat, A. Thomas, and B. S. Rajan, "Error correction in coded caching with symmetric batch prefetching," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5264-5274, 2019. https://doi.org/10.1109/TCOMM.2019.2916556.