2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206875

Coded Caching Design for Dynamic Networks with Reduced Subpacketizations

Xianzhang Wu†, Minquan Cheng‡, Li Chen§, Congduan Li†
† School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China
‡ Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin 541004, China
§ School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
Email: wuxzh7@mail2.sysu.edu.cn, chengqinshi@hotmail.com, chenli55@mail.sysu.edu.cn, licongd@mail.sysu.edu.cn

***Abstract*—Coded caching is an effective technique to reduce the data transmission load by exploiting the cache contents across the network. However, most coded caching schemes are designed for static networks that consist of only a placement phase and a delivery phase among a constant number of users. In practice, a network maybe dynamic with multiple rounds of placement and delivery phases, and the number of users may vary. In such dynamic networks, a conventional coded caching scheme may lead to the undesired content updates at the users' cache, which is caused by the newly joining users. This paper proposes a centralized coded caching scheme for dynamic networks that can support multiple rounds and accommodate the newly joining users during this process. It prevents cache contents of the existing users from being updated. It is shown that the proposed scheme can yield a reduced subpacketization level and achieve a good rate-memory tradeoff.**

***Index Terms*—Coded caching, dynamic networks, subpacketization level**

I. Introduction

The dramatic increase in the use of smart devices leads to an unprecedented growth in internet traffic. This generates a tremendous burden on smoothing the data transmission over the network, especially during the peak hours. Coded caching has been introduced as an effective technique to alleviate the network pressure by exploiting the network cache contents. The original coded caching network [1] consists of a central server which has access to a library of N files of the same size. It provides service to K users over an error free shared link. Each user has a cache memory with a size of M files. A coded caching scheme normally consists of two phases, the placement phase and the delivery phase. In the placement phase, the server sends the properly designed contents to each user's cache without any knowledge of the later demands. In the delivery phase, the server is informed with the users' demands. It broadcasts the coded packets to the users so that each user can reconstruct its desired file with the assistance of their own cached contents. The worst case broadcasting load normalized by the size of file is defined as the transmission rate R, i.e., the minimum number of files that must be communicated so that any possible demands can be satisfied.

Maddah-Ali and Niesen [1] have proposed a coded caching scheme that is realized by the combinatorial uncoded cache placement phase and the network coded delivery phase, which is referred as the MN scheme. It achieves an optimal transmission rate under the constraints of uncoded placement and $K \leq N$ [2]. Following the seminal work of [1], many later researches have focused on improving the transmission rate and the subpacketization level. E.g., if a file is requested by several users, the delivery phase design for improving the transmission rate of the MN scheme has been studied in [3]. The theoretical lower bounds on the transmission rate have been derived in [4], [5], which also characterize the optimal performance of a coded caching network. Moreover, generalizations of the MN scheme to other networks have been examined in [6]–[8]. The above mentioned schemes yield a subpacketization level (i.e., the number of smaller parts each file should be split into) that grows exponentially with the number of users, which makes them impractical for large networks. This problem has been addressed by several work using methodologies of placement delivery array (PDA) [9]–[14], projective geometry [15] and Ruzsa-Szemerédi graphs [16], respectively. However, they usually trade it with the transmission rate. Overall, it is still challenging to design a coded caching scheme that can achieve a good rate-memory tradeoff at a low subpacketization level.

Note that most existing work only considers the coded caching design for static networks that consist of only a placement phase and a delivery phase. In practice, the coded caching network would need to support multiple rounds of placement and delivery phases, during which the new users may join the network and the server does not have prior knowledge of the new users in the forthcoming rounds. In such dynamic networks, a conventional coded caching scheme may lead to the network with frequent updates at the users' cache contents, which is caused by the newly joining users. This may result in a waste of network resource for storage, especially in the case of few newly joining users. Therefore, it is important to design a coded caching scheme for dynamic networks that can not only yield a large coding gain but also require the minimal cache content updates. Intuitively, more users joining the network will lead to a greater coding gain. This implies that the existing users and the new users should be jointly considered in the design.

This paper proposes a novel cache placement and content delivery design for dynamic networks based on the combinatorial approach. In order to avoid resource consumption caused by the unnecessary cache content updates, a concatenating method is introduced in the placement phase to enable the cache contents of the existing users remain unchanged. Based on the placement contents, the coded messages can be designed to generate more multicast opportunities between the existing users and the new users. Our analysis shows that the proposed coded caching scheme can yield a smaller subpacketization level with a similar transmission rate over the existing scheme of [17]. It also generalizes the network model of [17], making it suited to the scenarios that can support more rounds.

II. Network Model and Problem Formulation

This section presents the dynamic coded caching model and its problem formulation. First, we introduce the key notations.

Notations: Let calligraphic symbols and bolded lower-case letters denote sets and vectors, respectively. Symbol $\oplus$ denotes the exclusive-or (XOR) operation. Let $\mathbb{Z}_q$ denote the ring of integers modulo q, and the $\mathbb{Z}_q^n$ further denote a set of vectors with elements obtained by the n-fold Cartesian product of $\mathbb{Z}_q$, i.e., $\mathbb{Z}_q^n = \{\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) \mid (x_0, x_1, \ldots, x_{n-1}) \in \mathbb{Z}_q \times \mathbb{Z}_q \times \cdots \times \mathbb{Z}_q\}$. We use $|\cdot|$ to denote the cardinality of a set. Let $\mathbb{N}^+$ denote the set of positive integers. The set of consecutive integers is denoted as $[x : y] = \{x, x+1, \ldots, y\}$. For a length-$m$ vector $\mathbf{a}$, let $\mathbf{a}|_i$ denote the ith element of $\mathbf{a}$, where $i \in [0 : m-1]$. Finally, the vectors in examples are written as strings, e.g., (1, 1, 1, 1) is written as 1111.

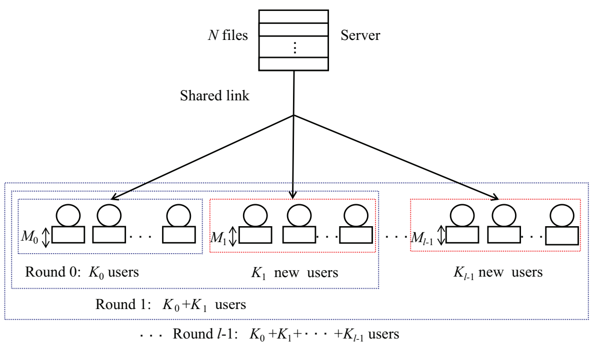

Fig. 1: A $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching network.

Based on the MN coded caching network, we consider a network model that contains l rounds, where each round consists of a placement phase and a delivery phase. It is illustrated as in Fig. 1. A server containing N files of the same size is connected to $\sum_{i=0}^{j} K_i$ users through an error free shared link at round j, where $\sum_{i=0}^{j} K_i \leq N$ and $j \in [0 : l-1]$. The N files are denoted as $\mathcal{W} = \{W_0, W_1, \ldots, W_{N-1}\}$. Let $\mathcal{K}_0$ denote the set of initial users in the network, and $\mathcal{K}_j$ further denote the set of new users at round j, where $j \in [1 : l-1]$. Each user in $\mathcal{K}_j$ is equipped with a dedicated cache with a size of M_j files, where $|\mathcal{K}_j| = K_j$ and $M_j < N$. Note that the cache sizes M_0, M_1, …, M_{l-1} are not necessary to be the same. For clarity, this model is referred as $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching network. It characterizes the dynamic networks at any round j, during which K_j new users join the network and a new coded caching of the existing $\sum_{i=0}^{j-1} K_i$ users is performed. Therefore, if $K_1 = K_2 = \cdots = K_{l-1} = 0$, the dynamic coded caching network dissolves into a static network of K_0 users.

It is assumed that in the current round, there is no prior knowledge of the forthcoming users. That says information of the number of new users and their cache sizes are not available. Moreover, the existing users are assumed to be stationary. They will be involved in the next round with the same placement strategy. This enables their cache contents remain unchanged in the forthcoming rounds. Hence, at the current round, the placement phase should be further partitioned into two subphases, one for the existing users and the other for new users. Since the cache contents remain unchanged for the existing users, the server would need to further partition the packets that are utilized by the existing users, and design the cache placement for the new users. Therefore, the challenge lies in how to design the cache placement for the new users and the multicast messages to ensure a small transmission rate.

III. Dynamic Coded Caching Scheme

This section introduces the dynamic coded caching scheme that can support multiple rounds with new users joining the network gradually.

A. Design Motivation

Given a $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic network, a trivial way of realizing it at round j is to implement a $(K_i; M_i; N)$ coded caching scheme for the users of $\mathcal{K}_i$ separately, where $i \in [0 : j]$. This is illustrated by the following example, namely as the grouping scheme.

Example 1 (Grouping Scheme). Consider a $(K_0, K_1; M_0, M_1; N)$ dynamic coded caching network within two rounds, where $\mathcal{K}_0 = \{0, 1, 2, 3\}, \mathcal{K}_1 = \{4, 5\}, M_0 = M_1 = 3$ and $N = 6$. At round 1, the users in $\mathcal{K}_0$ and $\mathcal{K}_1$ perform the $(K_0; M_0; N)$ and $(K_1; M_1; N)$ coded caching schemes, respectively.

- *Placement Phase*: The users in $\mathcal{K}_0$ perform a $(4; 3; 6)$ coded caching scheme. Each file in the server is divided into four packets of the same size, i.e., $W_n = \{W_{n,0}, W_{n,1}, W_{n,2}, W_{n,3}\}$, where $n \in [0 : 5]$. The contents cached by each user in $\mathcal{K}_0$ are $\mathcal{Z}_0 = \{W_{n,0}, W_{n,1}\}$, $\mathcal{Z}_1 = \{W_{n,2}, W_{n,3}\}$, $\mathcal{Z}_2 = \{W_{n,0}, W_{n,2}\}$ and $\mathcal{Z}_3 = \{W_{n,1}, W_{n,3}\}$, where $n \in [0 : 5]$. Similarly, two new users in $\mathcal{K}_1$ perform another $(2; 3; 6)$ coded caching scheme. Each file in the server is divided into two packets of the same size, i.e., $W_n = \{W'_{n,0}, W'_{n,1}\}$, where $n \in [0 : 5]$. Each user in $\mathcal{K}_1$ caches the following packets, $\mathcal{Z}_4 = \{W'_{n,0}\}$ and $\mathcal{Z}_5 = \{W'_{n,1}\}$, where $n \in [0 : 5]$.
- *Delivery Phase*: Let us assume that users 0, 1, 2, 3, 4 and 5 request files W_0, W_1, W_2, W_3, W_4 and W_5, respectively. The messages sent by the server are composed of two parts. The first part is $W_{0,2} \oplus W_{2,1}$, $W_{1,0} \oplus W_{2,3}$, $W_{0,3} \oplus W_{3,0}$ and $W_{1,1} \oplus W_{3,2}$, which is generated by the $(4; 3; 6)$ coded caching scheme. The second part that is generated by the $(2; 3; 6)$ coded caching scheme would be $W'_{4,1} \oplus W'_{5,0}$. Each user can then reconstruct its desired file and the transmission rate is $R(4; 3; 6) + R(2; 3; 6) = 1 + \frac{1}{2} = \frac{3}{2}$.

From *Example 1*, it can be seen that the cache contents of users in $\mathcal{K}_0$ do not need to be updated. However, the multicast opportunities between the users in $\mathcal{K}_0$ and $\mathcal{K}_1$ will be lost. This results in a larger transmission rate. In order to create more multicast opportunities among all the users in each round, the existing users and the new users should be jointly considered in the design of content delivery.

B. New Design

Unlike the grouping method, our new design is to concatenate the cache placement of the existing users and the new users so that the coding gains can be enlarged. This is realized through the combinatorial design of the placement and delivery phases. We first introduce details of the combinatorial cache placement and the content delivery design as the follows.

1) Placement Phase: We focus on the cache placement for the $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching network at round j, where $j \in [0 : l-1]$ and $l \in \mathbb{N}^+$. Let $K_i = m_i p_i$ and $M_i = \frac{N z_i}{p_i}$, where m_i, p_i and z_i are positive integers such that $p_i > z_i \geq 1$ and $\lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor = \cdots = \lfloor \frac{p_j-1}{p_j-z_j} \rfloor$ for $i \in [0:j]$. For clarity, the users in $\mathcal{K}_i$ are denoted as $\mathcal{K}_i = \{(i, g_i, v_i) \mid (g_i, v_i) \in [0 : m_i - 1] \times [0 : p_i - 1]\}$. Each file in the server is partitioned into $\alpha p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j}$ packets of the same size, i.e.,

$$W_n = \Big\{ W^{(\beta)}_{n,\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j} \mid (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, \beta \in [0 : \alpha - 1] \Big\},$$

where $n \in [0 : N-1]$ and $\alpha = \lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor = \cdots = \lfloor \frac{p_j-1}{p_j-z_j} \rfloor$. This implies that the subpacketization level of the scheme at round j is $F_j = \alpha p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j}$. The contents cached by user (i, g_i, v_i) are denoted as

$$\begin{aligned} \mathcal{Z}_{(i,g_i,v_i)} = \Big\{ & W^{(\beta)}_{n,\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j} \mid (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \\ & \cdots \times \mathbb{Z}_{p_j}^{m_j}, \mathbf{a}_i|_{g_i} \in \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}, \\ & \beta \in [0 : \alpha - 1], n \in [0 : N-1] \Big\}, \end{aligned} \tag{1}$$

where $(i, g_i, v_i) \in \mathcal{K}_i$ and the computations are performed under modulo p_i. Therefore, each user in $\mathcal{K}_i$ caches a total of $\frac{N F_j z_i}{p_i}$ packets, which requires a cache memory of $\frac{N F_j z_i}{p_i} = M_i F_j$ packets. Moreover, with this concatenating placement strategy, it can be seen that the contents cached by the users in round $j-1$ do not need to be updated at round j.

2) Delivery Phase: The content delivery at round j can be further described. Suppose that user $(i, g_i, v_i) \in \mathcal{K}_i$ requests file $W_{d_{(i,g_i,v_i)}}$ for $i \in [0 : j]$, where $d_{(i,g_i,v_i)} \in [0, N-1]$. Define a mapping ϕ_i from $\mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j} \times \mathcal{K}_i \times [0 : \alpha - 1]$ to $\mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j} \times [0 : p_i - z_i - 1]$ as

$$\begin{aligned} \phi_i(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, i, g_i, v_i, \beta) = (& \mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{i-1}, a_{i,0}, a_{i,1}, \\ & \ldots, a_{i,g_i-1}, v_i - \beta(p_i - z_i), a_{i,g_i+1} \ldots, a_{i,m_i-1}, \\ & \mathbf{a}_{i+1}, \ldots, \mathbf{a}_j, \mathbf{a}_i|_{g_i} - v_i - 1), \end{aligned} \tag{2}$$

where $\mathbf{a}_i|_{g_i} \notin \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}$. Based on (2), for each $\mathbf{b} \in \mathcal{B} = \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j} \times [0 : \max\{p_0 - z_0 - 1, p_1 - z_1 - 1, \ldots, p_j - z_j - 1\}]$, the server broadcasts the following coded packet to all the users at round j.

$$\bigoplus_{\substack{\phi_i(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,i,g_i,v_i,\beta)=\mathbf{b}, \beta \in [0:\alpha-1], \\ (\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, \\ (g_i,v_i) \in [0:m_i-1] \times [0:p_i-1], i \in [0:j]}} W^{(\beta)}_{d_{(i,g_i,v_i)},\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j}. \tag{3}$$

Since $|\mathcal{B}| = p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j} \max\{p_0 - z_0, p_1 - z_1, \ldots, p_j - z_j\}$, the total number of packets sent over the shared link is $R_j F_j = p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j} \max\{p_0 - z_0, p_1 - z_1, \ldots, p_j - z_j\}$.

The following example illustrates the above two phases.

Example 2. Consider the same dynamic network as in *Example 1*, i.e., $p_0 = p_1 = 2$, $z_0 = z_1 = 1$, $m_0 = 2$ and $m_1 = 1$. Based on the above placement and delivery design, at round 1, the network performs two phases as the follows.

- *Placement Phase*: Each packet $W^{(0)}_{n,\mathbf{a}_0}$ used at round 0 is further partitioned into two packets, i.e.,

$$\begin{aligned} W_n &= \Big\{ W^{(0)}_{n,\mathbf{a}_0} \mid \mathbf{a}_0 \in \mathbb{Z}_2^2 \Big\} = \Big\{ W^{(0)}_{n,\mathbf{a}_0,\mathbf{a}_1} \mid (\mathbf{a}_0, \mathbf{a}_1) \in \mathbb{Z}_2^3 \Big\} \\ &= \Big\{ W^{(0)}_{n,00,0}, W^{(0)}_{n,00,1}, W^{(0)}_{n,01,0}, W^{(0)}_{n,01,1}, W^{(0)}_{n,10,0}, W^{(0)}_{n,10,1}, \\ &\qquad W^{(0)}_{n,11,0}, W^{(0)}_{n,11,1} \Big\}, \end{aligned}$$

where $n \in [0 : 5]$. Based on (1), the contents cached by the users of $\mathcal{K}_0 = \{000, 001, 010, 011\}$ remain the same as before, i.e.,

$$\mathcal{Z}_{000} = \Big\{ W^{(0)}_{n,00,0}, W^{(0)}_{n,00,1}, W^{(0)}_{n,01,0}, W^{(0)}_{n,01,1} \Big\},$$
$$\mathcal{Z}_{001} = \Big\{ W^{(0)}_{n,10,0}, W^{(0)}_{n,10,1}, W^{(0)}_{n,11,0}, W^{(0)}_{n,11,1} \Big\},$$
$$\mathcal{Z}_{010} = \Big\{ W^{(0)}_{n,00,0}, W^{(0)}_{n,00,1}, W^{(0)}_{n,10,0}, W^{(0)}_{n,10,1} \Big\},$$
$$\mathcal{Z}_{011} = \Big\{ W^{(0)}_{n,01,0}, W^{(0)}_{n,01,1}, W^{(0)}_{n,11,0}, W^{(0)}_{n,11,1} \Big\},$$

where $n \in [0 : 5]$. The contents cached by two new users in $\mathcal{K}_1 = \{100, 101\}$ are

$$\mathcal{Z}_{100} = \Big\{ W^{(0)}_{n,00,0}, W^{(0)}_{n,01,0}, W^{(0)}_{n,10,0}, W^{(0)}_{n,11,0} \Big\},$$
$$\mathcal{Z}_{101} = \Big\{ W^{(0)}_{n,00,1}, W^{(0)}_{n,01,1}, W^{(0)}_{n,10,1}, W^{(0)}_{n,11,1} \Big\},$$

where again $n \in [0 : 5]$.

- *Delivery Phase*: Let us assume that users 000, 001, 010, 011, 100 and 101 request files W_0, W_1, W_2, W_3, W_4 and W_5, respectively. Based on (3), the messages sent by the server are

$$W^{(0)}_{0,10,0} \oplus W^{(0)}_{2,01,0} \oplus W^{(0)}_{4,00,1},\ W^{(0)}_{0,10,1} \oplus W^{(0)}_{2,01,1} \oplus W^{(0)}_{5,00,0},$$
$$W^{(0)}_{0,11,0} \oplus W^{(0)}_{3,00,0} \oplus W^{(0)}_{4,01,1},\ W^{(0)}_{0,11,1} \oplus W^{(0)}_{3,00,1} \oplus W^{(0)}_{5,01,0},$$
$$W^{(0)}_{1,00,0} \oplus W^{(0)}_{2,11,0} \oplus W^{(0)}_{4,10,1},\ W^{(0)}_{1,00,1} \oplus W^{(0)}_{2,11,1} \oplus W^{(0)}_{5,10,0},$$
$$W^{(0)}_{1,01,0} \oplus W^{(0)}_{3,10,0} \oplus W^{(0)}_{4,11,1},\ W^{(0)}_{1,01,1} \oplus W^{(0)}_{3,10,1} \oplus W^{(0)}_{5,11,0}.$$

Each user can then reconstruct its desired file. E.g., user 000 requires W_0 and it has cached $W^{(0)}_{0,00,0}, W^{(0)}_{0,00,1}, W^{(0)}_{0,01,0}$ and

$W^{(0)}_{0,01,1}$. It can obtain $W^{(0)}_{0,10,0}$, $W^{(0)}_{0,10,1}$,$W^{(0)}_{0,11,0}$ and $W^{(0)}_{0,11,1}$ with the following received coded packets

$$W^{(0)}_{0,10,0} \oplus W^{(0)}_{2,01,0} \oplus W^{(0)}_{4,00,1},\ W^{(0)}_{0,10,1} \oplus W^{(0)}_{2,01,1} \oplus W^{(0)}_{5,00,0},$$

$$W^{(0)}_{0,11,0} \oplus W^{(0)}_{3,00,0} \oplus W^{(0)}_{4,01,1},\ W^{(0)}_{0,11,1} \oplus W^{(0)}_{3,00,1} \oplus W^{(0)}_{5,01,0},$$

where $W^{(0)}_{2,01,0}$, $W^{(0)}_{4,00,1}$, $W^{(0)}_{2,01,1}$, $W^{(0)}_{5,00,0}$, $W^{(0)}_{3,00,0}$, $W^{(0)}_{4,01,1}$, $W^{(0)}_{3,00,1}$ and $W^{(0)}_{5,01,0}$ have been cached. Hence, the transmission rate is $R(4,2;3,3;6) = \frac{8}{8} = 1$, which is smaller than that of $\frac{3}{2}$ in *Example 1*.

IV. Performance Characterization

This section further analyses performance of the proposed dynamic coded caching scheme.

A. The Main Result

The following *Theorem 1* characterizes the proposed dynamic coded caching scheme. Due to the space limit, its detailed proof is omitted.

Theorem 1. Given any $p_i, z_i, m_i, l \in \mathbb{N}^+$ with $p_i > z_i \geq 1$ and $\lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor = \cdots = \lfloor \frac{p_{l-1}-1}{p_{l-1}-z_{l-l}} \rfloor = \alpha$ for $i \in [0 : l-1]$, there exists a $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching scheme with $K_i = m_i p_i$ and $M_i = \frac{N z_i}{p_i}$. At round j, the transmission rate of $R_j = \frac{\max\{p_0-z_0, p_1-z_1, \ldots, p_j-z_j\}}{\alpha}$ can be achieved with a subpacketization level of $F_j = \alpha p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j}$, where $j \in [0 : l-1]$.

It should be pointed out that the proposed combinatorial construction approach is also applicable to any positive integers p_i and z_i, where $p_i > z_i \geq 1$. In particular, when parameters p_i and z_i satisfy the constraint of *Theorem 1*, the proposed construction can achieve a better subpacketization level.

B. Performance Analysis

We further compare our proposed scheme with the scheme of [17] and the MN grouping scheme (i.e., the MN scheme with grouping). Note that the scheme of [17] can support for two rounds with order optimal transmission rate. Its features are further characterized in the following lemma.

Lemma 2 [17]. Given any $K_i, M_i, N \in \mathbb{N}^+$ with $M_i < N$ and $t_i = \frac{K_i M_i}{N} \in [1 : K_i - 1]$, where $i \in [0 : 1]$, there exists a $(K_0, K_1; M_0, M_1; N)$ dynamic coded caching scheme with a transmission rate of

$$R'_1 \leq \begin{cases} \frac{(K_0-t_0)(t_0t_1+t_0+1)}{t_0t_1(t_0+1)} + \frac{K_1-t_1}{t_1(t_0+1)}, \text{if } M_0 \leq M_1, \\ \frac{(K_1-t_1)(t_0t_1+t_1+1)}{t_0t_1(t_1+1)} + \frac{K_0-t_0}{t_0(t_1+1)}, \text{if } M_0 > M_1, \end{cases}$$

and a subpacketization level of $F'_1 = t_0 t_1 \binom{K_0}{t_0}\binom{K_1}{t_1}$ at round 1.

Given any $p_i, z_i, m_i \in \mathbb{N}^+$ and $l = 2$ with $\lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor$, $\frac{z_0}{p_0} > \frac{z_1}{p_1}$ and $z_i < p_i$ for $i \in [0 : 1]$, the round 1 subpacketization level and transmission rate of the scheme in *Theorem 1* can be written as

$$F_1 = \alpha p_0^{m_0} p_1^{m_1},\ R_1 = \frac{\max\{p_0 - z_0, p_1 - z_1\}}{\alpha},$$

where $\alpha = \lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor$. Further by letting $K_i = m_i p_i$ and $\frac{M_i}{N} = \frac{z_i}{p_i}$ in *Lemma 2*, the subpacketization level and transmission rate of the scheme in *Lemma 2* can be written as

$$F'_1 = m_0 m_1 z_0 z_1 \binom{m_0 p_0}{m_0 z_0}\binom{m_1 p_1}{m_1 z_1},$$

$$R'_1 \leq \frac{(p_1 - z_1)(m_0 m_1 z_0 z_1 + m_1 z_1 + 1)}{m_0 z_0 z_1 (m_1 z_1 + 1)} + \frac{p_0 - z_0}{z_0(m_1 z_1 + 1)}.$$

Based on Stirling's Formula $m! = \sqrt{2\pi m}(\frac{m}{e})^m$ $(m \to \infty)$, with $m_0, m_1 \to \infty$, we have

$$\begin{aligned} F'_1 &= m_0 m_1 z_0 z_1 \binom{m_0 p_0}{m_0 z_0}\binom{m_1 p_1}{m_1 z_1} \\ &\approx m_0 m_1 z_0 z_1 \sqrt{\frac{p_0}{2\pi z_0 m_0 (p_0 - z_0)}} \left(\frac{p_0^{p_0}}{z_0^{z_0}(p_0 - z_0)^{p_0 - z_0}}\right)^{m_0} \\ &\quad \sqrt{\frac{p_1}{2\pi z_1 m_1 (p_1 - z_1)}} \left(\frac{p_1^{p_1}}{z_1^{z_1}(p_1 - z_1)^{p_1 - z_1}}\right)^{m_1}. \end{aligned}$$

Hence, the subpacketization level ratio between the schemes in *Theorem 1* and *Lemma 2* can be written as

$$\begin{aligned} \frac{F_1}{F'_1} &= \sqrt{\frac{2\pi z_0 m_0 (p_0 - z_0)}{p_0}} \left(\frac{p_0^{p_0}}{p_0 z_0^{z_0}(p_0 - z_0)^{p_0 - z_0}}\right)^{-m_0} \\ &\quad \frac{\alpha}{m_0 m_1 z_0 z_1} \sqrt{\frac{2\pi z_1 m_1 (p_1 - z_1)}{p_1}} \left(\frac{p_1^{p_1}}{p_1 z_1^{z_1}(p_1 - z_1)^{p_1 - z_1}}\right)^{-m_1}. \end{aligned} \tag{4}$$

Note that for $i \in [0 : 1]$, based on the binomial expansion, we have

$$\begin{aligned} p_i^{p_i} &= (p_i - z_i)^{p_i} + \binom{p_i}{1}(p_i - z_i)^{p_i - 1} z_i + \cdots + z_i^{p_i} \\ &\geq \binom{p_i}{z_i - 1}(p_i - z_i)^{p_i - z_i + 1} z_i^{z_i - 1} + \binom{p_i}{z_i}(p_i - z_i)^{p_i - z_i} z_i^{z_i} \\ &\quad + \binom{p_i}{z_i + 1}(p_i - z_i)^{p_i - z_i - 1} z_i^{z_i + 1} \\ &= \binom{p_i}{z_i}(p_i - z_i)^{p_i - z_i} z_i^{z_i} + \binom{p_i}{z_i}(p_i - z_i)^{p_i - z_i} z_i^{z_i} \\ &\quad \left(\frac{p_i - z_i}{p_i - z_i + 1} + \frac{z_i}{z_i + 1}\right) \\ &\geq 2\binom{p_i}{z_i}(p_i - z_i)^{p_i - z_i} z_i^{z_i}. \end{aligned} \tag{5}$$

Substituting (5) into (4) yields

$$\begin{aligned} \frac{F_1}{F'_1} &\leq \sqrt{\frac{2\pi z_0 m_0 (p_0 - z_0)}{p_0}} \sqrt{\frac{2\pi z_1 m_1 (p_1 - z_1)}{p_1}} \\ &\quad \frac{\alpha}{m_0 m_1 z_0 z_1} \left(\frac{2\binom{p_0}{z_0}}{p_0}\right)^{-m_0} \left(\frac{2\binom{p_1}{z_1}}{p_1}\right)^{-m_1} \\ &= \frac{\alpha}{z_0 z_1} \sqrt{\frac{2\pi z_0 (p_0 - z_0)}{m_0 p_0}} \sqrt{\frac{2\pi z_1 (p_1 - z_1)}{m_1 p_1}} \left(\frac{2\binom{p_0}{z_0}}{p_0}\right)^{-m_0} \\ &\quad \left(\frac{2\binom{p_1}{z_1}}{p_1}\right)^{-m_1} = \mathcal{O}\left((m_0 m_1)^{-\frac{1}{2}} \eta^{-(m_0 + m_1)}\right), \end{aligned}$$

where $\eta \geq 2$. Meanwhile, without loss of generality, it is assumed that $p_0 - z_0 < p_1 - z_1$. The transmission rate ratio between the schemes in *Lemma 2* and *Theorem 1* is

$$\frac{R_1'}{R_1} \leq \alpha\left(\frac{m_1 z_1(m_0 z_0+1)+1}{m_0 z_0 z_1(m_1 z_1+1)} + \frac{p_0 - z_0}{z_0(m_1 z_1+1)(p_1 - z_1)}\right) \approx \frac{\alpha}{z_1},$$

where $m_0, m_1 \to \infty$.

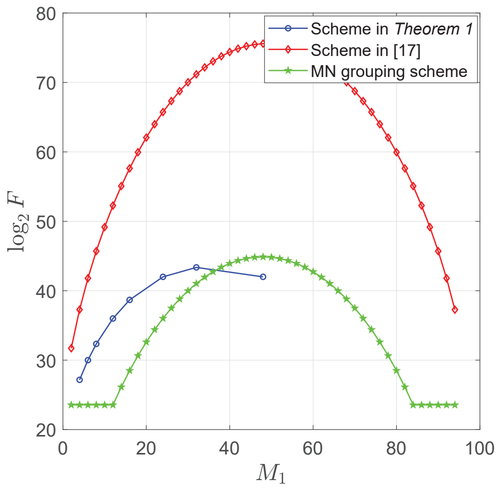

Fig. 2: Subpacketization level comparison, where $K_0 = K_1 = 48, M_0 = 12$ and $N = 96$.

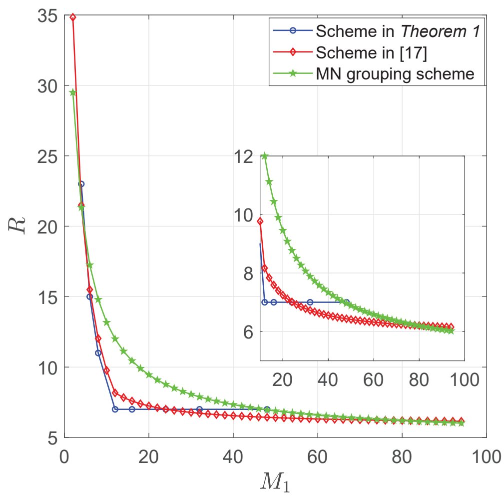

Fig. 3: Transmission rate comparison, where $K_0 = K_1 = 48, M_0 = 12$ and $N = 96$.

The above analysis shows that when the parameters p_0, p_1, z_0 and z_1 are fixed and $m_0, m_1 \to \infty$, our proposed scheme is able to substantially reduce the subpacketization level of the scheme in [17] when the network operates within two rounds. Meanwhile, the transmission rate of the proposed scheme is at least $\frac{z_1}{\alpha}$ of that of the scheme in [17]. In the following we numerically compare performance of the schemes in *Theorem 1*, [17] and the MN grouping scheme. Let $p_0 = 8, m_0 = 6, z_0 = z_1 = 1$ and $(m_1, p_1) \in \{(2,24),(3,16),(4,12),(6,8),(8,6),(12,4),(16,3),(24,2)\}$ for the proposed scheme; Let $K_0 = K_1 = 48, t_0 = 6$ and $t_1 \in [1:47]$ for the scheme of [17] and the MN grouping scheme. Figs. 2 and 3 compare the subpacketization level F and transmission rate R of the three schemes. For the subpacketization level of the MN grouping scheme, we choose a larger one in the two groups as a comparison benchmark. It can be seen that with a similar transmission rate, our proposed scheme yields a far smaller subpacketization level than that of [17]. When comparing with the MN grouping scheme, our proposed scheme yields a smaller transmission rate, but at the cost of a slightly increased subpacketization level. Meanwhile, for some cache sizes that are greater than 37, our proposed scheme has advantages in both the subpacketization level and the transmission rate.

V. Conclusion

This paper has proposed the design of coded caching for dynamic networks that can support multiple rounds with new users joining in each round. In order to minimize the cache content updates in the placement phase and the amount of transmissions in the delivery phase, a new dynamic coded caching scheme has been proposed through the combinatorial design. Our analytical and numerical results have both shown that the proposed scheme yields a reduced subpacketization level and achieves a good rate-memory tradeoff.

VI. Acknowledgement

This work is sponsored by the National Natural Science Foundation of China (NSFC) with project IDs 62071498, 62271514, 62061004 and U21A20474, the Science, Technology and Innovation Commission of Shenzhen Municipality with project IDs JCYJ20210324120002007, JCYJ20190807155617099, ZDSYS20210623091807023, and the Guangxi Natural Science Foundation with project ID DA035087.

References

[1] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856-2867, May 2014.
[2] K. Wan, D. Tuninetti, and P. Piantanida, “On the optimality of uncoded cache placement,” in Proc. *IEEE Inf. Theory Workshop (ITW)*, Cambridge, U.K., Sep. 2016, pp. 161-165.
[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “The exact rate-memory tradeoff for caching with uncoded prefetching,” *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281-1296, Feb. 2018.
[4] H. Ghasemi and A. Ramamoorthy, “Improved lower bounds for coded caching," in Proc. *IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 1696-1700.
[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “Characterizing the rate-memory tradeoff in cache networks within a factor of 2,” *IEEE Trans. Inf. Theory* , vol. 65, no. 1, pp. 647-663, Jan. 2019.
[6] M. Ji, A. Tulino, J. Llorca, and G. Caire, “Caching in combination networks,”in Proc. *49th Asilomar Conf. Signals, Syst. Comput. (ACSSC)*, Nov. 2015, pp. 1269-1273.
[7] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of caching in wireless D2D networks,” *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849-869, Feb. 2016.
[8] J. Zhang, X. Lin, and X. Wang, “Coded caching under arbitrary popularity distributions,” *IEEE Trans. Infor. Theory*, vol. 64, no. 1, pp. 349-366, Jan. 2018.

[9] M. Cheng, J. Jiang, Q. Yan, and X. Tang, "Constructions of coded caching schemes with flexible memory size," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4166-4176, Jun. 2019.

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

[11] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Commun. Lett.*, vol.22, no. 2, pp. 236-239, Feb. 2018.

[12] X. Wu, M. Cheng, C. Li, and L. Chen, "Design of placement delivery arrays for coded caching with small subpacketizations and flexible memory sizes," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7089-7104, Nov. 2022.

[13] X. Wu, M. Cheng, L. Chen, C. Li and Z. Shi, "Design of coded caching schemes with linear subpacketizations based on injective arc coloring of regular digraphs," *IEEE Trans. Commun.*, doi: 10.1109/T-COMM.2023.3252031.

[14] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755-5766, Aug. 2018.

[15] H. H. S. Chittoor, P. Krishnan, K. V. S. Sree, and B. Mamillapalli, "Subexponential and linear subpacketization coded caching via projective geometry," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6193-6222, Sep. 2021.

[16] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa-Szeméredi graphs," in Proc. *IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1237-1241.

[17] Q. Zhang, L. Zheng, M. Cheng, and Q. Chen, "On the dynamic centralized coded caching design," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2118-2128, Apr. 2020.