Fifty-Fifth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 3-6, 2017

Coded Caching with Multiple File Requests

Yi-Peng Wei Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland College Park, MD 20742
ypwei@umd.edu *ulukus@umd.edu*

***Abstract*—We study a two-phase caching network consisting of one server with N files connected to K users through an error-free shared link. Each user has a cache memory which can store M files in the placement phase. In the delivery phase, each user requests L files, and the server transmits the messages accordingly. Using the message sent by the server combined with the cache memory, each user reconstructs the L files they requested. In this work, we focus on the case $L > 1$, i.e., the case of multiple file requests. We adopt the symmetric batch caching scheme and propose a general delivery scheme. To prove the optimality of the proposed general delivery scheme, we apply two converse techniques. The first converse technique is for general coding schemes and is obtained through virtual user construction. The second converse technique is for vector linear coding schemes and is obtained using an interference alignment point of view. With these two converse techniques, we characterize either the unconstrained optimal coding rate, or optimum linear coding rate, with symmetric batch caching for certain cases.**

I. Introduction

Consider a two-phase caching network [1] consisting of one server with N files connected to K users through an error-free shared link. Each user has their local cache memory which can store M files. The two phases are the placement phase and the delivery phase. In the placement phase, the network traffic load is low. Each user can access the whole N files in the server and fill their cache memory in advance. In the delivery phase, the network traffic load is high. Each user requests L files from the server, and the server delivers messages through the error-free shared link to K users. The request of each user is unknown a priori in the placement phase. Each user reconstructs the L files they requested by the messages sent from the server and the side information stored in their cache memory. The objective is to minimize the traffic load in the delivery phase due to the high traffic load in this phase.

The two-phase caching network is first studied in [1], with the assumption that in the delivery phase, each user requests one file, i.e., $L = 1$. Reference [1] proposes *symmetric batch caching* for the placement phase. Combined with the coded multicasting in the delivery phase, reference [1] shows that global caching gain can be obtained. Using a cut-set bound analysis, order optimality of rate-memory trade off is shown for the worst-case file requests in [1]. *Independent and identical caching* is proposed in [2] to account for the decentralized nature of practical caching networks. Both symmetric batch caching and independent and identical caching are uncoded cache placement schemes [3], [4], i.e., each user stores a subset of the bits of the original files. For uncoded cache placement, with $N \geq K$, by using index coding converse bound [5], reference [3] shows the optimality of rate-memory trade off for the worst-case file requests. In addition, reference [4] shows the optimality for arbitrary N, K and M not only for the worst-case file requests but also for the average case. For coded placement, order optimality results can be found in [6] and references therein.

In the delivery phase, each user can request more than one file, i.e., $L > 1$. This case is first studied in [7] which adopts symmetric batch caching as in [1]. For the delivery phase, [7] treats each different file request as a different index coding problem, and generalizes the achievability scheme for multiple unicast index coding in [8] to group casting index coding. Reference [7] shows the order optimality for the worst-case file request with a multiplicative constant 18 based on a cut-set bound analysis. Then, reference [9] shows the order optimality for the worst-case file request with multiplicative constant 11 by improving the converse bound through Han's inequality. Reference [9] adopts symmetric batch caching as in [1] and applies the delivery scheme in [1] L times.

In this work, we also adopt symmetric batch caching as in [1], and focus on exact optimality as opposed to order optimality. We propose a general delivery scheme for multiple file requests. To show the optimality of the delivery scheme, we use two converse techniques. The first technique is for general coding schemes and is inspired by the converse in [4]. The second technique is for vector linear coding schemes using an interference alignment point of view and is inspired by [10], [11]. With these two converse techniques, we determine either unconstrained optimal coding rate, or optimal linear coding rate with symmetric batch caching for certain cases. If LK different files are requested, we characterize the optimal coding rate. For $L = 2$ and $K = 3, 4$, when each subfile is cached only at one user (i.e., $t = 1$), we characterize either the unconstrained optimal coding rate, or the optimal linear coding rate.

II. System Model and Problem Setting

We consider a caching network (see Fig. 1) consisting of one server and K users. The server connects to the K users through an error-free shared link. The server has N files denoted by $W_1, W_2, \ldots, W_N$. Each file is of size F bits. Each user has a local cache memory Z_k of size MF bits for some real number $M \in [0, N]$. There are two phases in this network, a placement

This work was supported by NSF Grants CNS 13-14733, CCF 14-22111, and CNS 15-26608.

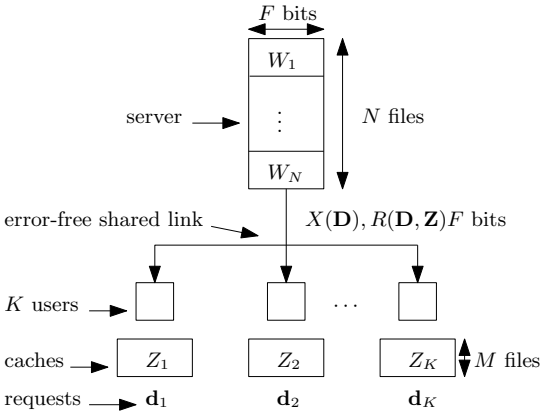

Fig. 1. Caching network.

phase and a delivery phase. In the placement phase, user k can access all the N files and fill its cache memory Z_k. Therefore, $Z_k = \phi_k(W_1, W_2, \ldots, W_N)$, where $\phi_k : \mathbb{F}_2^{NF} \to \mathbb{F}_2^{MF}$. In the delivery phase, each user requests L files out of the N files. Let us denote each user's request by an $L \times 1$ vector $\mathbf{d}_k = (d_{k,1}, \ldots, d_{k,L})^{\mathrm{T}}$, where $d_{k,i}$ is the index of the file, i.e., $d_{k,i} \in \{1, 2, \ldots, N\},\ 1 \le i \le L$. A request matrix $\mathbf{D}$ of size $L \times K$ is formed accordingly. The server outputs $X(\mathbf{D})$ of size RF bits through the error-free shared link to the K users, where R refers to the load of the network in the delivery phase. A rate R is said to be achievable if each user k can decode the L files it requested by utilizing $X(\mathbf{D})$ and Z_k.

Note that in the placement phase, which L files will be requested by each user is unknown in advance. Therefore, the cache memory Z_k is determined before knowing $\mathbf{D}$. In addition, we focus on the uncoded cache placement as in [4], which means that each user k chooses MF bits out of NF bits to fill its cache memory Z_k. An example of coded cache placement is provided in [1, Appendix]. We denote the minimum achievable rate in the delivery phase by $R^*(\mathbf{D}, \mathbf{Z})$, where $\mathbf{Z} = (Z_1, \ldots, Z_K)$. An average rate for a given cache placement $\mathbf{Z}$ is defined as $R^*(\mathbf{Z}) = \mathbb{E}_{\mathbf{D}}[R^*(\mathbf{D}, \mathbf{Z})]$, by assuming that each user chooses the L files equally likely from the N files. The minimum rate is defined as $R^* = \min_{\mathbf{Z}} R^*(\mathbf{Z})$.

If $L = 1$, reference [4] shows that *symmetric batch caching* originally proposed in [1] attains the minimum rate R^*. We summarize the symmetric batch caching here. Given each user has cache memory size $M = \frac{tN}{K}$, where $t \in \{1, \ldots, K\}$, for each file W_i, we partition the file into $\binom{K}{t}$ non-overlapping and equal-size subfiles, and denote $[K] = \{1, 2, \ldots, K\}$ and $\mathcal{T} = \{T : T \subset [K], |T| = t\}$, where $|\cdot|$ means the cardinality of a set. Note $|\mathcal{T}| = \binom{K}{t}$. We label the subfiles of W_i as $W_{i,T}$, where $T \in \mathcal{T}$. Equivalently, $W_i = \cup_{T \in \mathcal{T}} W_{i,T}$ and $W_{i,T} \cap W_{i,T'} = \emptyset$ if $T \neq T'$. In the placement phase, user k places the subfile $W_{i,T}$ into the cache memory Z_k if $k \in T$. Thus, user k gets $\binom{K-1}{t-1}$ subfiles of each file W_i. We denote the symmetric batch caching with parameter t as $\mathbf{Z}^t_{\mathrm{sym}}$.

In this work, we adopt the symmetric batch caching, $\mathbf{Z}^t_{\mathrm{sym}}$, and study $R^*(\mathbf{D}, \mathbf{Z}^t_{\mathrm{sym}})$. Also, we denote $R^*_l(\mathbf{D}, \mathbf{Z}^t_{\mathrm{sym}})$ as the minimum achievable rate confined to vector linear coding schemes. We propose a general delivery scheme. We characterize $R^*(\mathbf{D}, \mathbf{Z}^t_{\mathrm{sym}})$ if LK distinct files are requested by the users. For $L = 2$, $t = 1$, $K = 3, 4$, we characterize $R^*(\mathbf{D}, \mathbf{Z}^t_{\mathrm{sym}})$ for certain request matrices $\mathbf{D}$, while we characterize $R^*_l(\mathbf{D}, \mathbf{Z}^t_{\mathrm{sym}})$ for all other request matrices.

III. Proposed Delivery Scheme

To illustrate the delivery scheme, we introduce the following vector space representation. In the placement phase, we partition each file into $\binom{K}{t}$ subfiles. We view each subfile as a vector, and regard each subfile as the basis of the vector space. Since the subfiles are linearly independent, with all the subfiles, we have a $N\binom{K}{t}$-dimensional vector space. Note that in the delivery phase, we do not further sub-packetize the subfiles. Therefore, our delivery scheme can also be viewed as scalar linear coding. We consider the vector space over $\mathbb{F}_2$.

Let $\mathcal{S} = \{S : S \subset [K], |S| = t + 1\}$. For every $S \in \mathcal{S}$ and $1 \le l \le L$, a candidate delivery message is as follows:

$$Y_{S,l} = \bigoplus_{s \in S} W_{d_{s,l}, S \setminus \{s\}}. \tag{1}$$

Here, $d_{s,l}$ identifies the index of the lth file user s requests. Since $|S \setminus \{s\}| = t$, it identifies the subfile $W_{d_{s,l}, S\setminus\{s\}}$ owned by these t users. Note that among the $t+1$ terms on the right hand side of (1) only one term is unknown to each user s. By sending the candidate delivery message $Y_{S,l}$, each user s in S can decode a subfile of $W_{d_{s,l}}$ they requested.

For $L = 1$, the delivery scheme proposed in [1] is to go over all the sets in $\mathcal{S}$ and send out all the candidate delivery message given in (1). This results in $\binom{K}{t+1}$ transmissions. However, reference [4] shows that going over all the sets in $\mathcal{S}$ is unnecessary. Instead, [4] goes over $\mathcal{S}' = \{S : S \subset [K], S \cap \mathcal{U} \neq \emptyset, |S| = t + 1\}$, where $\mathcal{U}$ corresponds to the leaders defined in [4]. In [4], the number of transmissions is reduced to $\binom{K}{t+1} - \binom{K-|\mathcal{U}|}{t+1}$.

For $L > 1$, we know that sending out all the candidate delivery messages by going over all the sets in $\mathcal{S}$ and $1 \le l \le L$ is sufficient to satisfy all the requests. To reduce the traffic load, if the candidate delivery message can be obtained through a linear combination of the sent messages, then the server does not need to send this candidate delivery message. Since each candidate delivery message given in (1) has a vector representation, the necessary delivery messages consist of the messages in the maximal linearly independent subset of the candidate delivery messages formed by going over all the sets in $\mathcal{S}$ and $1 \le l \le L$.

We use an example to illustrate the proposed delivery scheme. Let $N = 4$, $K = 4$, $M = 1$, and $L = 2$. Then, $t = 1$. To simplify the notation, let A, B, C and D denote the four files. By applying symmetric batch caching, $\mathbf{Z}^1_{\mathrm{sym}}$, we have

$$Z_k = (A_k, B_k, C_k, D_k), \tag{2}$$

where $k = 1, 2, 3, 4$. Suppose the request matrix is as follows

$$\mathbf{D} = \begin{pmatrix} A & A & A & B \\ B & C & C & C \end{pmatrix}, \tag{3}$$