Cache-Aided Combination Networks with Asymmetric End Users

Ahmed A. Zewail and Aylin Yener

Wireless Communications and Networking Laboratory (WCAN)
The School of Electrical Engineering and Computer Science
The Pennsylvania State University, University Park, PA 16802.
zewail@psu.edu *yener@engr.psu.edu*

***Abstract*—We study combination networks where a layer of relay nodes connects a server to a set of end users with cache memories via unicast links. Unlike previous models on cache-aided combination networks where all users are connected to the same number of relay nodes, in this work, we consider two classes of end users, where users from the same class are connected to the same number of relay nodes. Using maximum distance separable (MDS) codes, we provide a coded caching scheme by jointly optimizing the cache placement and the delivery phase in order to minimize the delivery load over the two hops. The scheme is performed over two stages. In the first stage, we serve users from both classes, then during the second stage, we serve users from the class connected to a smaller number of relays. We extend the proposed scheme to networks with more than two classes of end users.**

I. Introduction

Caching [1], [2] is a promising solution to reduce network congestion during peak traffic hours. Coded caching [2] not only utilizes the users' cache memories in shifting some of the network traffic to off-peak hours, but also creates multicast opportunities that reduce the delivery load on the server. Following the pioneering work of reference [2] where the performance gain from coded caching is shown for a server connected to a set of end users via a multicast link, references [3]–[7] have studied cache-aided communication in two-hop networks, where a layer of relays connects the server to its end users. In particular, reference [4] has investigated a single-server symmetric layered network, known as a *combination network*, where the end users are equipped with cache memories. In such networks, the server is connected to a set of h relay nodes, which communicate to $\binom{h}{r}$ users, such that each user is connected to a distinct set of r relay nodes. In references, [5], [6], we have boosted the storage capabilities of the combination networks by adding cache memories at the relay nodes. We have proposed a coded caching scheme that decomposes the combination network into h virtual sub-networks such that the delivery load per relay node is optimal with respect to the cut-set bound.

Recently, caching models which capture the heterogeneity in content delivery networks has been considered. For example, references [8], [9] have considered designing the cache sizes of the end users, as well as the caching schemes, subject to a total memory budget constraint where each of the end users is connected to the server via a rate-limited-link with finite-capacity that differs from a user to another. Reference [10] has considered such a setup when the end users are connected to the server via a noisy broadcast channel. In this work, we consider heterogeneity in the connectivity of combination networks by considering different classes of end users, each of which is connected to a different number of relay nodes. More specifically, we start by considering two classes of end users such that each user from class 1 is connected to r_1 relay nodes while each user from class 2 is connected to r_2 relay nodes, $r_1 > r_2$. We develop a centralized coded caching scheme that utilizes maximum distance separable (MDS) codes and jointly optimizes the cache placement and delivery phases. In particular, we encode each file using an $(h+r_1-r_2, r_1)$ MDS code, and the caching scheme is performed over two stages. In the first stage, each relay node acts as a virtual server with a library formed by one of the first h encoded symbols of each file. From the contents of its cache memory and the received signal from each of its connected relay nodes, each end user can reconstruct one encoded symbol of its requested file. Thus, by the end of the first stage, each user from class 1 can decode its requested file from r_1 of its encoded symbols, while the users from class 2 have received r_2 encoded symbols from their requested files. During the second stage, we consider serving the users from class 2 only by providing the missing $r_1 - r_2$ encoded symbols that are needed to reconstruct their requested files.

We discuss the extensions of the proposed scheme for more than two classes of end users.

Notation: $\oplus$ refers to bitwise XOR operation, $|W|$ denotes size of W, and $[K] \triangleq \{1, \ldots, K\}$.

II. System Model

A. Network Connectivity

We consider a two-hop network, where the server, S, is connected to K end users via a set of h relay nodes. The end users are classified into two groups. Specifically, K_1 users belong to class 1; each of these users is connected to a distinct set of r_1 relay nodes, i.e., $K_1 = \binom{h}{r_1}$. The remaining K_2 users, $K = K_1 + K_2$, belong to class 2 and each of them is connected to a distinct set of r_2 relay nodes, i.e., $K_2 = \binom{h}{r_2}$.

978-1-5386-6528-2/19/$31.00 ©2019 IEEE

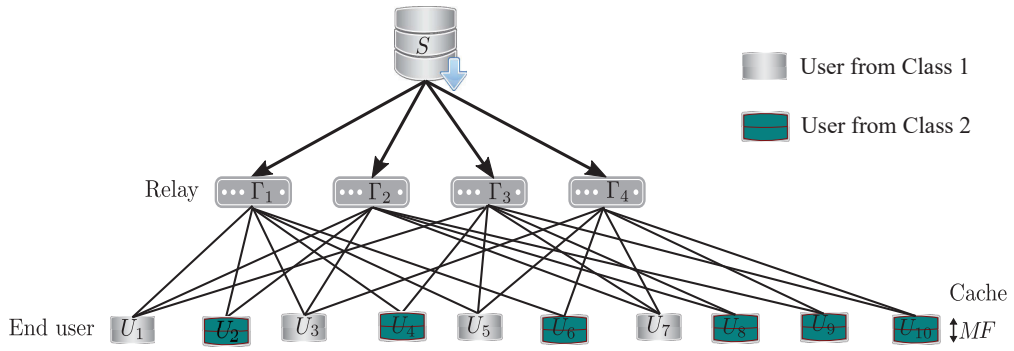

Fig. 1: An asymmetric combination network with two classes of end users where $K=10$, $h=4$, $r_1=3$ and $r_2=2$.

Thus, each relay node is connected to $L_1=\binom{h-1}{r_1-1}=\frac{r_1K_1}{h}$ and $L_2=\binom{h-1}{r_2-1}=\frac{r_2K_2}{h}$ users from class 1 and 2, respectively. Similar to [4], all network links are assumed to be noiseless and unicast. We define $\mathcal{R}=\{\Gamma_1,..,\Gamma_h\}$ as the set of relay nodes, and $\mathcal{U}=\{U_1,..,U_K\}$ as the set of all end users in the network. We denote the set of end users connected to Γ_j by $\mathcal{N}(\Gamma_j)$, i.e., $|\mathcal{N}(\Gamma_j)|=L_1+L_2$ for $j=1,..,h$, and the set relay nodes connected to user k from class i by $\mathcal{N}(U_k)$, i.e., $|\mathcal{N}(U_k)|=r_i$, $i=1,2$. Without loss of generality, we assume that $r_1\geq r_2$. We define the following function which returns the relative order of user k with respect to the neighbors of relay node Γ_j. The function $Index(,):(j,k)\rightarrow\{1,..,L_1+L_2\}$, where $j\in\{1,..,h\}$ and $k\in\mathcal{N}(\Gamma_j)$, is defined as a function that orders the end users connected to each relay in ascending order. For example, in Fig. 1, we have $\mathcal{N}(\Gamma_2)=\{1,2,3,7,8,9\}$, $\mathcal{N}(\Gamma_4)=\{3,5,6,7,9,10\}$ and

$$Index(2,1)=1,\ Index(2,2)=2,\ Index(2,9)=6,$$
$$Index(4,3)=1,\ Index(4,6)=3,\ Index(4,9)=5.$$

B. Caching Model

The server S has a database of N files, $W_1,..,W_N$, each with size F symbols over the field $\mathbb{F}_{2^q}$. Each end user has a cache memory of size MF symbols, i.e., M represents the normalized memory size. The system operates over two phases.

1) Cache Placement Phase: The server allocates functions of its database in the end users' cache memories. These allocations are designed, without the knowledge of the actual demands in the delivery phase, subject to the memory capacity constraints.

Definition 1. *(Cache Placement): The contents of the cache memory at user k are given by*

$$Z_k=\phi_k(W_1,W_2,..,W_N),\tag{1}$$

where $\phi_k:[2^F]^N\rightarrow[2^F]^M$, such that $H(Z_k)\leq MF$. ■

2) Delivery Phase: During peak traffic, each user requests a randomly selected file [2]. We define d_k to denote the index of the requested file by user k, i.e., $d_k\in\{1,2,..,N\}$, and $\boldsymbol{d}$ to represent the demand vector of all network users at any request instance. The server responds to the users' requests by transmitting signals to each of the relay nodes. Then, each relay node forwards its received signal to the set of intended end users. From its received signals and Z_k, user k should be able to reconstruct its requested file W_{d_k}.

Definition 2. *(Coded Delivery): The mapping from the database, and the demand vector, $\boldsymbol{d}$, into the transmitted signal by the server to Γ_j is represented by the encoding function*

$$X_{j,\boldsymbol{d}}=\psi_j(W_1,..,W_N,\boldsymbol{d}),\qquad j=1,2,..,h,\tag{2}$$

where $\psi_i:[2^F]^N\times[N]^K\rightarrow[2^F]^{R_1}$, and R_1 is the rate, normalized by the file size, F, of the transmitted signal from the server to each relay node. The transmitted signal from Γ_j to user $k\in\mathcal{N}(\Gamma_j)$, is given by the encoding function

$$Y_{j,\boldsymbol{d},k}=\varphi_k(X_{j,\boldsymbol{d}},\boldsymbol{d}),\tag{3}$$

where $\varphi_k:[2^F]^{R_1}\times[N]^K\rightarrow[2^F]^{R_{2,i}}$, and $R_{2,i}$ is the normalized rate of the transmitted signal from the relay node to a connected end user from class i. In addition, user k, from class i, has a decoding function to recover its requested file, given by

$$\hat{W}_k=\mu_k(Z_k,\boldsymbol{d},\{Y_{j,\boldsymbol{d},k}:j\in\mathcal{N}(U_k)\}),\tag{4}$$

where $\mu_k:[2^F]^{M_2}\times[N]^K\times[2^F]^{r_iR_{2,i}}\rightarrow[2^F]$, and $i=1,2$. ■

Each of the end users must be able to recover its requested file *reliably*, i.e., for any $\epsilon>0$,

$$\max_{\boldsymbol{d},k}P(\hat{W}_{d_k}\neq W_{d_k})<\epsilon.\tag{5}$$

Definition 3. *The rate-memory tuple $(R_1,R_{2,1},R_{2,2},M)$ is said to be achievable, if for $F\rightarrow\infty$, there exists a set of caching functions, $\{\phi_i\}$, encoding functions, $\{\psi_i\}$, $\{\varphi_i\}$, and decoding functions, $\{\mu_k\}$, such that for any $\epsilon>0$ (5) is satisfied.* ■

We focus on the case where the total number of files is no less than the number of end users, i.e., $N\geq K$.

III. The Proposed Coded Caching Scheme

The proposed scheme encodes each file using an $(h+r_1-r_2,r_1)$ maximum distance separable (MDS) code [11]. The scheme is performed over two stages. At the first stage, each relay node acts as a virtual server for one of the first h resulting encoded symbols. Since each user from class 1 is connected

to r_1 different relay nodes by the end of the first stage of the delivery phase, it will have obtained r_1 different encoded symbols that can be used to recover its requested file. By contrast, each of the users from the class 2 will have only obtained r_2 encoded symbols from its requested files, and would still need r_1-r_2 additional encoded symbols to recover the file. During the second stage, the remaining r_1-r_2 encoded symbols are delivered to users from class 2.

A. First Stage

1) Cache Placement Phase: As a first step, the server divides each file into r_1 equal-size subfiles. Then, it encodes them using an $(h+r_1-r_2, r_1)$ maximum distance separable (MDS) code [11]. We denote by f_n^j the resulting encoded symbols, where n is the file index and $j=1,2,..,h+r_1-r_2$. The size of each encoded symbol, f_n^j, is F/r_1 symbols, and any r_1 encoded symbols are sufficient to reconstruct the file.

For $M=\frac{tN}{L_1+L_2}$, and $t\in\{0,1,..,L_1+L_2\}$, each encoded symbol is divided into $\binom{L_1+L_2}{t}$ disjoint pieces each of which is denoted by $f^j_{n,\mathcal{T}}$, where $\mathcal{T}\subseteq[L_1+L_2]$, and $|\mathcal{T}|=t$. The size of each piece is $\frac{F}{r_1\binom{L_1+L_2}{t}}$ symbols. The server allocates the pieces $f^j_{n,\mathcal{T}}$, $\forall n$ in the cache memory of user k if $k\in\mathcal{N}(\Gamma_j)$ and $Index(j,k)\in\mathcal{T}$. Thus, we have

$$Z_k=\left\{f^j_{n,\mathcal{T}}: k\in\mathcal{N}(\Gamma_j),\ Index(j,k)\in\mathcal{T},\ \forall n\right\}. \quad (6)$$

At the end of the cache placement phase of the first stage, each user from class 1 stores $r_1\binom{L_1+L_2-1}{t-1}$ pieces each of size $\frac{F}{r_1\binom{L_1+L_2}{t}}$ symbols. Therefore, the accumulated number of symbols in its cache memory is given by

$$r_1N\binom{L_1+L_2-1}{t-1}\frac{F}{r_1\binom{L_1+L_2}{t}}=\frac{tN}{L_1+L_2}F=MF \text{ symbols}, \quad (7)$$

i.e., the memory capacity constraint is satisfied and $t=\frac{M(L_1+L_2)}{N}$. Each user from class 2 at this stage have stored $r_2N\binom{L_1+L_2-1}{t-1}$ pieces each of size $\frac{F}{r_1\binom{L_1+L_2}{t}}$ symbols. We define M_f to be normalized memory size of the users from class 2 at the end of the first stage, i.e.,

$$M_f=M-\frac{r_2\binom{L_1+L_2-1}{t-1}}{r_1\binom{L_1+L_2}{t}}=\frac{tN(r_1-r_2)}{r_1(L_1+L_2)}. \quad (8)$$

2) Coded Delivery Phase: At the beginning of the delivery phase, the demand vector, $\boldsymbol{d}$, is announced in the network as public information. For each relay Γ_j, at each transmission instance, we consider $\mathcal{S}\subseteq[L_1+L_2]$, where $|\mathcal{S}|=t+1$. For each choice of $\mathcal{S}$, the server transmits to the relay node Γ_j, the signal

$$X^{\mathcal{S},1}_{j,\boldsymbol{d}}=\oplus_{\{k:k\in\mathcal{N}(\Gamma_j),\ Index(j,k)\in\mathcal{S}\}}f^j_{d_k,\mathcal{S}\setminus\{Index(j,k)\}}. \quad (9)$$

In total, the server transmits to Γ_j, the following signal

$$X^1_{j,\boldsymbol{d}}=\bigcup_{\mathcal{S}\subseteq[L_1+L_2]:|\mathcal{S}|=t+1}\{X^{\mathcal{S},1}_{j,\boldsymbol{d}}\}. \quad (10)$$

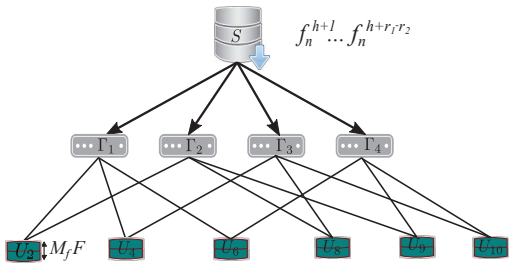

Fig. 2: An example of a reduced network during the second stage where $h=4$ and $r_2=2$.

Then, Γ_j forwards $X^{\mathcal{S}}_{j,\boldsymbol{d}}$ to user k if $Index(j,k)\in\mathcal{S}$, i.e.,

$$Y^1_{j,\boldsymbol{d},k}=\bigcup_{\mathcal{S}\subseteq[L_1+L_2]:|\mathcal{S}|=t+1,Index(j,k)\in\mathcal{S}}\{X^{\mathcal{S},1}_{j,\boldsymbol{d}}\}. \quad (11)$$

User k can recover the following set of pieces from the signals received from Γ_j, utilizing its cached contents $\left\{f^j_{d_i,\mathcal{T}}:\mathcal{T}\subseteq[L_1+L_2]\setminus\{Index(j,i)\},|\mathcal{T}|=t\right\}$. Adding these pieces to the cached ones, i.e., $f^j_{d_k,\mathcal{T}}$ with $Index(j,k)\in\mathcal{T}$, user k can recover the encoded symbol $f^j_{d_k}$. If user k belongs to class 1, i.e., it receives signals from r_1 different relay nodes, it obtains the encoded symbols $f^j_{d_k}$, $\forall j\in\mathcal{N}(U_k)$, thus user k is able to reconstruct W_{d_k}. If user k belongs to class 2 it obtains only r_2 encoded symbols from its requested file.

B. Second Stage

In the second stage, we focus on delivering the missing r_1-r_2 encoded symbols of the requested files by the users in class 2. After the first stage, we have a reduced network, where the server has a library of N files, each of them is formed by the concatenation of the encoded symbols $f_n^{h+1},...,f_n^{h+r_1-r_2}$, i.e., the size of each reduced file is $\frac{r_1-r_2}{r_1}F$ symbols. We illustrate the reduced network in Fig. 2. The server is connected to K_2 users, each of them is connected to r_2 relays and has a memory of size M_fF symbols. To describe our achievability, we define

$$t_1=\lceil\frac{L_2t}{L_1+L_2}\rceil,\ \text{and}\ t_2=\lfloor\frac{L_2t}{L_1+L_2}\rfloor. \quad (12)$$

The size of free cache memory at user k from class 2 can be expressed as

$$M_f=[\alpha t_1+(1-\alpha)t_2]\frac{N(r_1-r_2)}{r_1L_2}, \quad (13)$$

for some $\alpha\in[0,1]$. The scheme is described given the memory parameters t_1 and t_2 as follows. The concatenation of $f_n^{h+1},...,f_n^{h+r_1-r_2}$ is divided into two parts, $\hat{W}^1_n$ and $\hat{W}^2_n$, of sizes $\alpha\frac{r_1-r_2}{r_1}F$ symbols and $(1-\alpha)\frac{r_1-r_2}{r_1}F$ symbols, respectively.

1) Cache Placement Phase: The first part, $\hat{W}^1_n$, is divided into r_2 equal-size subfiles. Then, the server encodes them using an (h,r_2) maximum distance separable (MDS) code [11]. We denote by $g_n^{1,j}$ the resulting encoded symbols, where n is the file index and $j=1,2,..,h$. The size of each encoded

symbol, $g_n^{1,j}$, is $\alpha\frac{r_1-r_2}{r_2r_1}F$ symbols, and any r_2 encoded symbols are sufficient to reconstruct $\hat{W}_n^1$.

Each encoded symbol is divided into $\binom{L_2}{t_1}$ disjoint pieces each of which is denoted by $g_{n,\mathcal{T}_1}^{1,j}$, where $\mathcal{T}_1 \subseteq [L_2]$, and $|\mathcal{T}_1| = t_1$. The size of each piece is $\alpha\frac{r_1-r_2}{r_2r_1\binom{L_2}{t_1}}F$ symbols. The server allocates the pieces $g_{n,\mathcal{T}_1}^{1,j}$, $\forall n$ in the cache memory of user k from class 2 if $k \in \mathcal{N}(\Gamma_j)$ and $Index(j,k) \in \mathcal{T}$.

A similar allocation scheme is applied to $\hat{W}_n^2$ with parameter t_2 instead of t_1. Therefore, user k from class 2 caches $g_{n,\mathcal{T}_2}^{2,j}$, $\forall n$ if $k \in \mathcal{N}(\Gamma_j)$ and $Index(j,k) \in \mathcal{T}_2$. Therefore, by the end of the cache placement phase, the cached contents at user k from class 2 is given by

$$Z_k = \Big\{ f_{n,\mathcal{T}}^j, g_{n,\mathcal{T}_1}^{1,j}, g_{n,\mathcal{T}_2}^{2,j} : k \in \mathcal{N}(\Gamma_j), \\ Index(j,k) \in \mathcal{T}, \mathcal{T}_1, \mathcal{T}_2, \ \forall n \Big\}. \quad (14)$$

The accumulated symbols in the cache memory of each user from class 2 is given by

$$\frac{Nr_2\binom{L_1+L_2-1}{t-1}F}{r_1\binom{L_1+L_2}{t}} + \frac{\alpha N(r_1-r_2)r_2\binom{L_2-1}{t_1-1}F}{r_2r_1\binom{L_2}{t_1}} + \frac{(1-\alpha)N(r_1-r_2)r_2\binom{L_2-1}{t_2-1}F}{r_2r_1\binom{L_2}{t_2}} = \frac{r_2tNF}{r_1(L_1+L_2)} + M_fF = MF \text{ symbols}, \quad (15)$$

i.e., the memory capacity constraint is satisfied.

2) Coded Delivery Phase: For each relay Γ_j, we consider $\mathcal{S}_i \subseteq [L_2]$, where $|\mathcal{S}| = t_i + 1$, and $i = 1, 2$. For each choice of $\mathcal{S}_i$, the server transmits to Γ_j, the signal

$$\oplus_{\{k:k\in\mathcal{N}(\Gamma_j),\ Index(j,k)\in\mathcal{S}_i\}} g_{d_k,\mathcal{S}_i\setminus\{Index(j,k)\}}^{i,j}. \quad (16)$$

Then, Γ_j forwards its received signal to user k from class 2 if $Index(j,k) \in \mathcal{S}_i$. At the end of the second stage, user k from class 2 can recover the following set of pieces from the signals received from Γ_j, utilizing its cached contents

$$\Big\{ g_{d_k,\mathcal{T}}^{i,j} : \mathcal{T}_i \subseteq [L_2] \setminus \{Index(j,k)\}, |\mathcal{T}_i| = t_i, \ i = 1, 2 \Big\}.$$

Note that user k had cached $g_{d_k,\mathcal{T}_i}^{i,j}$ with $Index(j,k) \in \mathcal{T}_i$, thus user k can recover the encoded symbol $g_{d_k}^{i,j}$. Since, user k from class 2 receives signals from r_2 different relay nodes, it obtains the encoded symbols $g_{d_k}^{i,j}$, $\forall j \in \mathcal{N}(U_k)$, thus user k can reconstruct $f_{d_k}^{h+1}, ..., f_{d_k}^{h+r_1-r_2}$. Therefore, at the end of the delivery phase, user k from class 2 can decode its requested file from r_1 of its encoded symbols.

Note that we use the two-stage description to illustrate the idea of achievability, however, the cache placement procedures from the first and second stages are performed during the cache placement phase without the knowledge of the actual users' demands, while the coded delivery procedures are performed during the delivery phase after the users announce their requests.

C. Rate Calculation

Now, we calculate the transmission rates of this scheme.

1) First Stage: Since, each relay node is responsible for $\binom{L_1+L_2}{t+1}$ transmissions, each of length $\frac{F}{r_1\binom{L_1+L_2}{t}}$, thus

$$R_1^1F = \frac{\binom{L_1+L_2}{t+1}}{r_1\binom{L_1+L_2}{t}}F = \frac{L_1+L_2-t}{r_1(t+1)}F. \quad (17)$$

In addition, each relay node forwards $\binom{L_1+L_2-1}{t}$ from its received signals to each of its connected end users, thus

$$R_{2,i}^1F = \frac{\binom{L_1+L_2-1}{t}}{r_1\binom{L_1+L_2}{t}}F = \frac{L_1+L_2-t}{r_1(L_1+L_2)}F, \quad (18)$$

where $i = 1, 2$. Since, the users from class 1 are served only during the first stage, we get

$$R_{2,1} = R_{2,i}^1 = \frac{1}{r_1}\left(1 - \frac{M}{N}\right). \quad (19)$$

2) Second Stage: Each relay node is responsible for $\binom{L_2}{t_1+1}$ transmissions, each of length $\alpha\frac{r_1-r_2}{r_2r_1\binom{L_2}{t_1}}F$, and $\binom{L_2}{t_2+1}$ transmissions, each of length $(1-\alpha)\frac{r_1-r_2}{r_2r_1\binom{L_2}{t_2}}F$, thus we have

$$R_1^2F = \alpha\frac{(r_1-r_2)\binom{L_2}{t_1+1}}{r_2r_1\binom{L_2}{t_1}}F + (1-\alpha)\frac{(r_1-r_2)\binom{L_2}{t_2+1}}{r_2r_1\binom{L_2}{t_2}}F = \frac{r_1-r_2}{r_2r_1}\left(\alpha\frac{L_2-t_1}{t_1+1} + (1-\alpha)\frac{L_2-t_2}{t_2+1}\right)F. \quad (20)$$

During the second hop, each relay forwards $\binom{L_2-1}{t_1}$ and $\binom{L_2-1}{t_2}$ from its received signals to each of its connected end users from class 2, each of length equal to $\alpha\frac{r_1-r_2}{r_2r_1\binom{L_2}{t_1}}F$ and $(1-\alpha)\frac{r_1-r_2}{r_2r_1\binom{L_2}{t_2}}F$, respectively, thus

$$R_{2,2}^2F = \alpha\frac{(r_1-r_2)\binom{L_2-1}{t_1}}{r_2r_1\binom{L_2}{t_1}}F + (1-\alpha)\frac{(r_1-r_2)\binom{L_2-1}{t_2}}{r_2r_1\binom{L_2}{t_2}}F = \frac{r_1-r_2}{r_2r_1L_2}\left(\alpha(L_2-t_1) + (1-\alpha)(L_2-t_2)\right)F. \quad (21)$$

In total, $R_1 = R_1^1 + R_1^2$ and $R_{2,2} = R_{2,2}^1 + R_{2,2}^2$. Therefore, we obtain the upper bound on the normalized delivery rates as stated in the following theorem.

Theorem 1. *The normalized transmission rates for $M = \frac{tN}{L_1+L_2}$, and $t \in \{0, 1, .., L_1+L_2\}$, are upper bounded by*

$$R_1 \leq \frac{L_1+L_2-t}{r_1(t+1)} + \frac{r_1-r_2}{r_2r_1}\left(\alpha\frac{L_2-t_1}{t_1+1} + (1-\alpha)\frac{L_2-t_2}{t_2+1}\right), \quad (22)$$

$$R_{2,i} \leq \frac{1}{r_i}\left(1-\frac{M}{N}\right), \qquad i = 1, 2. \quad (23)$$

where $t_1 = \lceil\frac{L_2t}{L_1+L_2}\rceil$, $t_2 = \lfloor\frac{L_2t}{L_1+L_2}\rfloor$ and $\alpha \in [0, 1]$ is chosen such that $\frac{tL_2}{L_1+L_2} = \alpha t_1 + (1-\alpha)t_2$. The convex envelope of these points is achievable.

Note that if M is not in the form of $M = \frac{tN}{L_1+L_2}$, we apply memory sharing as in [2] for achievability.

IV. Discussion

A. Optimality over the Second Hop

Our achievable rate over the second hop is optimal, i.e., the total delivery load per relay is minimized. To see this, consider N request instance, such that user k from class i, requests file j at instance j, and $j = 1, ..., N$. Thus, we have the following constraint in order to satisfy the user's requests

$$H(W_1, .., W_N) = NF \leq Nr_i R_{2,i} F + MF. \tag{24}$$

Therefore, we can get the following bound on R_2

$$R_{2,i} \geq \frac{1}{r_i}\left(1 - \frac{M}{N}\right). \tag{25}$$

Note that the total delivery load of the network, i.e., $hR_1 + r_1K_1R_{2,1} + r_2K_2R_{2,2}$, is almost dominated by the delivery load over the second hop. Therefore, the optimality over the second hop is essential to minimize the overall delivery load.

B. More than Two Classes of End Users

The idea behind our proposed caching scheme can be extended to networks with more than two classes of end users. Suppose that there are three classes of end users such that each user from class i is connected to r_i relay nodes and $r_1 > r_2 > r_3$. The proposed scheme can be extended as follows. We start by encoding each file with an $(h+r_1-r_3, r_1)$ MDS code. At the first stage, each of the first h resulting encoded symbols will be delivered by considering $L_1+L_2+L_3$ end users connected to each relay. By the end of this stage, all the requests of users from class 1 are satisfied, while the users from class 2 and 3 recover r_2 and r_3 encoded symbols from their requested files, respectively. At the second stage, we form a reduced library by the concatenation of next $r_1 - r_2$ encoded symbols, that were not involved in the first stage, and encode each of them using an $(h + r_2 - r_3, r_2)$ MDS code. The scheme works as described before where there are $L_2 + L_3$ end users connected to each relay node. By the end of the second stage, the users from class 2 can decode their requested files. At the third stage, we focus on delivering the missing $r_2 - r_3$ encoded symbols from the second stage to the users from class 3 so that they can recover an additional set of $r_1 - r_2$ encoded symbols from their requested files. In addition, we need to deliver the remaining $r_2 - r_3$ resulting from the first encoding process, so that they can decode their requested files from their r_1 encoded symbols. The same procedure can be used where there are more than three classes of the end users in the network.

C. Numerical Results

In Fig. 3, we compare the achievable delivery load of our proposed scheme with the resulting load from treating the two classes of end users as two independent symmetric combination networks and applying the scheme from [5], [6]. It is evident that having multicast transmissions to users from different classes allows the proposed scheme to achieve lower delivery load over the first hop. Note that both schemes achieve the optimal load over the second hop.

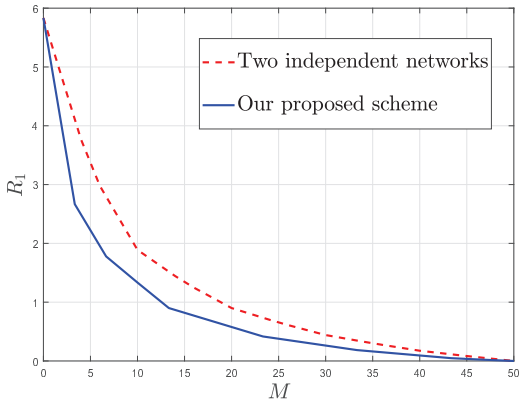

Fig. 3: The delivery load for $N = 50$, $h = 6$, $r_1 = 3$ and $r_2 = 2$.

V. Conclusions

In this work, we have investigated cache-aided combination networks with two classes of end users with different degrees of connectivity. By utilizing MDS coding and jointly optimizing both cache placement and delivery phases, we have proposed a new achievability scheme that serves the users requests' over two stages. During the first stage, we serve users from both classes, then during the second stage we serve only the users with weaker connectivity. We have shown that the proposed scheme is optimal over the second hop of communication and it can be generalized for network with more than two classes of end users.

Future directions include extending the proposed techniques to asymmetric combination networks with cache-aided relays, and with secrecy requirements [6], [12].

References

[1] M. A. Maddah-Ali and U. Niesen, "Coding for caching: fundamental limits and practical challenges," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 23–29, Aug. 2016.

[2] ——, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, Mar. 2014.

[3] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[4] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching in combination networks," in *49th Asilomar Conference on Signals, Systems and Computers*, 2015.

[5] A. A. Zewail and A. Yener, "Coded caching for combination networks with cache-aided relays," in *Proc. IEEE ISIT*, 2017.

[6] ——, "Combination networks with or without secrecy constraints: The impact of caching relays," *IEEE Journ. Sel. Areas in Commun.*, vol. 36, no. 6, pp. 1140–1152, Jun. 2018.

[7] K. Wan, M. Ji, P. Piantanida, and D. Tuninetti, "Combination networks with caches: Novel inner and outer bounds with uncoded cache placement," *arXiv:1701.06884*, 2017.

[8] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Optimization of heterogeneous caching systems with rate limited links," in *Proc. IEEE ICC*, 2017.

[9] ——, "Coded caching for heterogeneous systems: An optimization prespective," *arXiv:1810.08187*, 2018.

[10] S. S. Bidokhti, M. Wigger, and A. Yener, "Benefits of cache assignment on degraded broadcast channels," *arXiv:1702.08044*, 2017.

[11] S. Lin and D. J. Costello, *Error control coding*. Pearson Education India, 2004.

[12] A. A. Zewail and A. Yener, "Coded caching for resolvable networks with security requirements," in *Proc. IEEE CNS Workshops*, 2016.