Coded Caching for Combination Networks with Cache-Aided Relays

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 a two-hop cache-aided network, where a layer of relay nodes connects a server and a set of end users, i.e., a combination network. We consider the case where both the relay nodes and the end users have caching capabilities. We provide upper and lower bounds which are applicable to any combination network, noting that previous work had focused on models where the relays do not have caches as well as schemes that were suitable for a special class of combination networks. Utilizing maximum distance separable (MDS) codes, we jointly optimize the placement and the delivery phases, demonstrating the impact of cache memories in alleviating the delivery load over the two hop communications. Moreover, we show how cooperation between the relay nodes and the end users can effectively replace the server during the delivery phase whenever the total memory at each end user and its connected relay nodes is sufficient to store the database.**

I. Introduction

Caching is considered as a promising technique in the soon to arrive 5G systems. By utilizing the storage capabilities of the network nodes, caching helps network congestion and large end-to-end delays. During low traffic hours, network resources are exploited to place functions of data contents, from the library of files at the server, in the cache memories of the network nodes. This is known as the *cache placement phase*. The cached contents decrease the needed transmission rates during the *delivery phase* which takes place in peak traffic hours, when the users request content [1].

Following the pioneering work of reference [1] in which a single server is connected to a set of end users via a multicast link and the gain that can be achieved by caching quantified, references [2]–[5] have studied cache-aided two-hop networks. In particular, references [3] and [4] have 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. The cache placement policy in these references is decentralized [6], i.e., the users randomly cache a fraction of bits of each file. Recently, reference [5] has considered a class of networks that satisfies the *resolvability property*, where h is divisible by r. Considering centralized coded caching [1], reference [5] proposed a novel achievability scheme which outperforms the ones in [3] and [4]. The scheme, however relies heavily on the resolvability property.

In this paper, we investigate general cache-aided combination networks. We start by developing a centralized coded caching scheme that is applicable to any combination network, resolvable or not, utilizing maximum distance separable (MDS) codes and jointly optimizing the cache placement and delivery phases. In particular, we encode each file using an (h, r) MDS code, and each relay node acts as a virtual server with a library formed by one of the 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 delivery phase, each user is able to decode its requested file from r of its encoded symbols. We show the achievability of the same rate-memory tuples, achieved in [5], for any combination network. Then, we extend our scheme to the case where the relay nodes are equipped with cache memories. We show that if the accumulated memory size of each user and its connected relay nodes is sufficient to store the whole database files, then the server can remain silent during the delivery phase and all users' requests can be satisfied utilizing the cache memories of the relay nodes and end users. We derive genie-aided cut-set lower bounds on the delivery load. Moreover, we provide numerical results that show the performance of our proposed schemes. It is worth mentioning that a combination network with cache memories at the relay nodes differs also from the hierarchical network model in reference [2] where the server is connected to the relay nodes via a shared multicast link and the end users are divided into equal size groups such that each group of users is connected to only one relay node via a multicast link.

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. More specifically, each end user is connected to a distinct set of r relay nodes, $r < h$. Such class of networks is known as combination networks [3]. The number of end users is $K = \binom{h}{r}$. In such networks, each relay node is connected to $L = \binom{h-1}{r-1} = \frac{rK}{h}$ end users. Similar to references [3]–[5],

978-1-5090-4096-4/17/$31.00 ©2017 IEEE

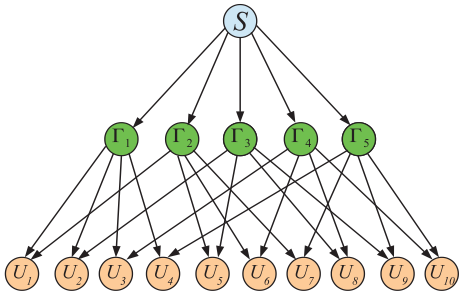

Fig. 1: A combination network with $K=10$, $h=5$ and $r=2$.

all network links are assumed to be noiseless and unicast. We define $\mathcal{R}=\{\Gamma_1,..,\Gamma_h\}$ to denote the set of relay nodes, and $\mathcal{U}=\{U_1,..,U_K\}$ to represent the set of all end users in the network. We denote the set of end users connected to Γ_i by $\mathcal{N}(\Gamma_i)$, i.e., $|\mathcal{N}(\Gamma_i)|=L$ for $i=1,..,h$, and the set relay nodes connected to user i by $\mathcal{N}(U_i)$, i.e., $|\mathcal{N}(U_i)|=r$.

We define the following function which returns the relative order of user k with respect to the neighbors of relay node Γ_i. The function $Index(,):(i,k)\rightarrow\{1,..,L\}$, where $i\in\{1,..,h\}$ and $k\in\mathcal{N}(\Gamma_i)$, is defined as a function that orders the end users connected to each relay in ascending manner.

For example, in the network in Fig. 1, we have $\mathcal{N}(\Gamma_2)=\{1,5,6,7\}$, $\mathcal{N}(\Gamma_4)=\{3,6,8,10\}$ and

$$Index(2,1)=1,\ Index(2,5)=2,\ Index(2,7)=4,$$
$$Index(4,3)=1,\ Index(4,6)=2,\ Index(4,8)=3.$$

B. Caching Model

The server S has a database of N files, $W_1,..,W_N$, each with size F bits. Each relay node is equipped with a cache memory of size M_1F bits, while each end user has a cache memory of size M_2F bits, i.e., M_1 and M_2 represent the normalized memory sizes. The system operates over two phases. Here, we consider the case where the total number of files is no less than the number of end users, i.e., $N\geq K$.

1) Cache Placement Phase: The server allocates functions of its database in the relay nodes' and 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 relay node Γ_j are given by*

$$V_j=\lambda_j(W_1,W_2,..,W_N),\tag{1}$$

where $\lambda_j:[2^F]^N\rightarrow[2^F]^{M_1}$, i.e., $H(V_j)\leq M_1F$. The contents of the cache memory at user k are given by

$$Z_k=\phi_k(W_1,W_2,..,W_N),\tag{2}$$

where $\phi_k:[2^F]^N\rightarrow[2^F]^{M_2}$, i.e., $H(Z_k)\leq M_2F$. ■

2) Delivery Phase: During peak traffic, each user requests a randomly selected file [1]. 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 utilizes its received signal and its cached contents to transmit unicast signals to each of its connected end users. From the r 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, $\mathbf{d}$, into the transmitted signal by the server to Γ_i is represented by the encoding function*

$$X_{i,\boldsymbol{d}}=\psi_i(W_1,..,W_N,\boldsymbol{d}),\qquad i=1,2,..,h,\tag{3}$$

where $\psi_i:[2^F]^N\times\{1,..N\}^K\rightarrow[2^F]^{R_1}$, 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_i to user $k\in\mathcal{N}(\Gamma_i)$, is given by the encoding function

$$Y_{i,\boldsymbol{d},k}=\varphi_k(X_{i,\boldsymbol{d}},V_i,\boldsymbol{d}),\tag{4}$$

where $\varphi_k:[2^F]^{R_1}\times[2^F]^{M_1}\times\{1,..N\}^K\rightarrow[2^F]^{R_2}$, and R_2 is the normalized rate of the transmitted signal from the relay node to a connected end user. In addition, user k has a decoding function to recover its requested file, given by

$$\hat{W}_k=\mu_k(Z_k,\boldsymbol{d},\{Y_{i,\boldsymbol{d},k}:i\in\mathcal{N}(U_k)\}),\tag{5}$$

where $\mu_k:[2^F]^{M_2}\times\{1,..N\}^K\times[2^F]^{rR_2}\rightarrow[2^F]$. ■

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

$$\max_{\boldsymbol{d},k}P(\hat{W}_{d_k}\neq W_{d_k})<\epsilon.\tag{6}$$

Definition 3. *The rate-memory tuple (R_1,R_2,M_1,M_2) is said to be achievable, if for $F\rightarrow\infty$, there exists a set of caching functions, λ_i's and ϕ_i's, encoding functions, ψ_i's and φ_i's, and decoding functions, μ_k's, such that for any $\epsilon>0$ condition (6) is satisfied.* ■

III. Main Results

Theorem 1. *The normalized transmission rates without cache memories at the relay nodes, i.e., $M_1=0$, for $M_2=\frac{tN}{L}$, $L=\frac{rK}{h}$ and $t\in\{0,1,..,L\}$, are upper bounded by*

$$R_1\leq\frac{L}{r}\left(1-\frac{M_2}{N}\right)\frac{1}{1+\frac{LM_2}{N}},\quad R_2\leq\frac{1}{r}\left(1-\frac{M_2}{N}\right).\tag{7}$$

Also, the convex envelope of these points is achievable.

Theorem 2. *The normalized transmission rates, for $0<M_1\leq\frac{N}{r}$, $M_2=\frac{(t_1-t_2)M_1r}{L}+\frac{t_2N}{L}$, $L=\frac{rK}{h}$, and $t_1,t_2\in\{0,1,..,L\}$, are upper bounded by*

$$R_1\leq\frac{L-t_2}{r(t_2+1)}\left(1-\frac{M_1r}{N}\right),\quad R_2\leq\frac{1}{r}\left(1-\frac{M_2}{N}\right).\tag{8}$$

Also, the convex envelope of these points is achievable.

Theorem 3. *The normalized transmission rates, for $0<M_2+rM_1\leq N$, are lower bounded by*

$$R_1\geq\max_{l\in\{r,..,h\}}\ \max_{s\in\{1,..,\min(N,\binom{l}{r})\}}\frac{1}{l}\left(s-\frac{sM_2+lM_1}{\lfloor N/s\rfloor}\right),\tag{9}$$

$$R_2\geq\frac{1}{r}\left(1-\frac{M_2}{N}\right).\tag{10}$$

Remark 1. One can see, from (7), that the upper bound on the normalized rate R_1 is formed by the multiplication of three terms, like the case in [1]. The first term $\frac{L}{r}$ is due to the fact that each relay node is connected to L end users, each of which is connected to r relay nodes. Thus, each relay node is responsible for $\frac{1}{r}$ of the load on a server that is connected to L end users. The second term $(1-\frac{M_2}{N})$ represents the local caching gain at each end user. Finally, the term $\frac{1}{1+\frac{LM_2}{N}}$ represents the global caching gain of the proposed scheme. On the other hand, the normalized rate R_2 is upper bounded by the local caching gain divided by the number of relay nodes connected to the end user, which is optimal from (10). ■

IV. No Cache Memories at The Relay Nodes

Here, we focus on combination networks with no cache memories at the relay nodes, i.e., $M_1 = 0$ and our caching model reduces to the ones in references [3]–[5]. We now prove that the upper bound derived in [5] for resolvable networks, is in fact achievable for any combination network.

The main idea behind our proposed scheme is that each file is encoded using an (h, r) maximum distance separable (MDS) code [7]. Then, each relay node acts as a virtual server for one of the resulting encoded symbols. Since, each end user is connected to r different relay nodes, by the end of the delivery phase, it will be able to obtain r different encoded symbols that can be used to recover its requested file.

A. Cache Placement Phase

As a first step, the server divides each file into r equal-size subfiles. Then, it encodes them using an (h, r) maximum distance separable (MDS) code [7]. We denote by f_n^i the resulting encoded symbols, where n is the file index and $i = 1, 2, .., h$. The size of each encoded symbol, f_n^i, is F/r bits, and any r encoded symbols are sufficient to reconstruct the whole file.

For $M_2{=}\frac{tN}{L}$, and $t\in\{0,1,..,L\}$, each encoded symbol is divided into $\binom{L}{t}$ disjoint pieces each of which is denoted by $f^i_{n,\mathcal{T}}$, where $\mathcal{T}\subseteq\{1,..,L\}$, and $|\mathcal{T}|{=}t$. The size of each piece is $\frac{F}{r\binom{L}{t}}$ bits. 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, the cached contents at user k is given by

$$Z_k=\left\{f^j_{n,\mathcal{T}}: k{\in}\mathcal{N}(\Gamma_j),\ Index(j,k)\in\mathcal{T},\ \forall n\right\}. \quad (11)$$

At the end of the cache placement phase, each user stores $r\binom{L-1}{t-1}$ pieces each of size $\frac{F}{r\binom{L}{t}}$ bits. Therefore, the accumulated number of bits in its cache memory is given by

$$r\binom{L-1}{t-1}\frac{F}{r\binom{L}{t}}=\frac{tN}{L}F=M_2F \text{ bits}, \quad (12)$$

i.e., the memory capacity constraint is satisfied and $t{=}\frac{M_2L}{N}$.

B. 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 $\mathcal{S}\subseteq\{1,..,L\}$, where $|\mathcal{S}|=t+1$. For each choice of the set $\mathcal{S}$, the server transmits to the relay node Γ_j, the signal

$$X^{\mathcal{S}}_{j,\boldsymbol{d}}=\bigoplus_{\{i:i\in\mathcal{N}(\Gamma_j),\ Index(j,i)\in\mathcal{S}\}} f^j_{d_i,\mathcal{S}\setminus\{Index(j,i)\}}. \quad (13)$$

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

$$X_{j,\boldsymbol{d}}=\bigcup_{\mathcal{S}\subseteq\{1,..,L\}:|\mathcal{S}|=t+1}\{X^{\mathcal{S}}_{j,\boldsymbol{d}}\}. \quad (14)$$

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

$$Y_{j,\boldsymbol{d},i}=\bigcup_{\mathcal{S}\subseteq\{1,..,L\}:|\mathcal{S}|=t+1,Index(j,i)\in\mathcal{S}}\{X^{\mathcal{S}}_{j,\boldsymbol{d}}\}. \quad (15)$$

User i can recover the following set of pieces from the signals received from Γ_j, utilizing its cached contents

$$\left\{f^j_{d_i,\mathcal{T}}:\mathcal{T}\subseteq\{1,..,L\}\setminus\{Index(j,i)\},|\mathcal{T}|=t\right\}.$$

Adding these pieces to the ones preallocated in its cache memory, i.e., $f^j_{d_i,\mathcal{T}}$ with $Index(j,i)\in\mathcal{T}$, user i can recover the encoded symbol $f^j_{d_i}$. Since, user i receives signals from r different relay nodes, it obtains the encoded symbols $f^j_{d_i}$, $\forall j\in\mathcal{N}(U_i)$, thus user i is able to reconstruct W_{d_i}.

Now, we calculate the transmission rates resulting from this scheme. First, observe that each relay node is responsible for $\binom{L}{t+1}$ transmissions, each of length $\frac{F}{r\binom{L}{t}}$, thus the transmission rate in bits from the server to a relay node is

$$R_1F=\frac{\binom{L}{t+1}}{r\binom{L}{t}}F=\frac{L-t}{r(t+1)}F=\frac{L(N-M_2)}{r(LM_2+N)}F. \quad (16)$$

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

$$R_2F=\frac{\binom{L-1}{t}}{r\binom{L}{t}}F=\frac{L-t}{rL}F=\frac{1}{r}\left(1-\frac{M_2}{N}\right)F. \quad (17)$$

Therefore, we obtain the upper bound on the normalized rates as stated in Theorem 1. Note that if M_2 is not in the form of $\frac{tN}{L}$, we apply memory sharing as in [1] for achievability.

Remark 2. The achievable rates in Theorem 1 are the same as the ones in [5] which in [5] have been shown to be achievable for a special class of combination networks where r divides h, i.e., resolvable networks. It has been shown in [5] that, for resolvable networks, the achievable rates in [5] are lower than the ones in [3] and [4]. Thus, our proposed scheme also always achieves lower rates compared to the ones in [3] and [4]. ■

C. Example

In this subsection, we explain our proposed scheme by an example. In particular, we consider the network depicted in Fig. 1, where $N = 10$, $M_1 = 0$ and $M_2 = \frac{15}{2}$. Clearly, this network is not resolvable, and $t = 3$.

1) Cache Placement Phase: Each file, W_n, is divided into 2 subfiles. Then, the server encodes them using an $(5, 2)$ MDS code. We denote the resulting encoded symbols by f^j_n, where n is the file index, i.e., $n = 1, .., 10$, and $j = 1, .., 5$.

User i	Z_i
1	$f^1_{n,123}, f^1_{n,124}, f^1_{n,134}, f^2_{n,123}, f^2_{n,124}, f^2_{n,134} : \forall n$
2	$f^1_{n,123}, f^1_{n,124}, f^1_{n,234}, f^3_{n,123}, f^3_{n,124}, f^3_{n,134} : \forall n$
3	$f^1_{n,123}, f^1_{n,134}, f^1_{n,234}, f^4_{n,123}, f^4_{n,124}, f^4_{n,134} : \forall n$
4	$f^1_{n,124}, f^1_{n,134}, f^1_{n,234}, f^5_{n,123}, f^5_{n,124}, f^5_{n,134} : \forall n$
5	$f^2_{n,123}, f^2_{n,124}, f^2_{n,234}, f^3_{n,123}, f^3_{n,124}, f^3_{n,234} : \forall n$
6	$f^2_{n,123}, f^2_{n,134}, f^2_{n,234}, f^4_{n,123}, f^4_{n,124}, f^4_{n,234} : \forall n$
7	$f^2_{n,124}, f^2_{n,134}, f^2_{n,234}, f^5_{n,123}, f^5_{n,124}, f^5_{n,234} : \forall n$
8	$f^3_{n,123}, f^3_{n,134}, f^3_{n,234}, f^4_{n,123}, f^4_{n,134}, f^4_{n,234} : \forall n$
9	$f^3_{n,124}, f^3_{n,134}, f^3_{n,234}, f^5_{n,123}, f^5_{n,134}, f^5_{n,234} : \forall n$
10	$f^4_{n,124}, f^4_{n,134}, f^4_{n,234}, f^5_{n,124}, f^5_{n,134}, f^5_{n,234} : \forall n$

TABLE I: The cached contents for $K{=}N{=}10$, $M_1{=}0$ and $M_2{=}\frac{15}{2}$.

Furthermore, we divide each encoded symbol into 4 pieces each of size $\frac{F}{8}$ bits, and denoted by $f^j_{n,\mathcal{T}}$, where $\mathcal{T}\subseteq\{1,..,4\}$ and $|\mathcal{T}|=3$. The contents of the cache memories at the end users are given in Table I. Observe that each user stores 6 pieces of the encoded symbols of each file, i.e., $\frac{3}{4}F$ bits, which satisfies the memory constraint.

2) Coded Delivery Phase: Assume that user k requests the file W_k. Then, the server transmits the following signals

$$X_{1,\boldsymbol{d}} = \left\{ f^1_{4,123} \oplus f^1_{3,124} \oplus f^1_{2,134} \oplus f^1_{1,234} \right\},$$
$$X_{2,\boldsymbol{d}} = \left\{ f^2_{7,123} \oplus f^2_{6,124} \oplus f^2_{5,134} \oplus f^2_{1,234} \right\},$$
$$X_{3,\boldsymbol{d}} = \left\{ f^3_{9,123} \oplus f^3_{8,124} \oplus f^3_{5,134} \oplus f^3_{2,234} \right\},$$
$$X_{4,\boldsymbol{d}} = \left\{ f^4_{10,123} \oplus f^4_{8,124} \oplus f^4_{6,134} \oplus f^4_{4,234} \right\},$$
$$X_{5,\boldsymbol{d}} = \left\{ f^5_{10,123} \oplus f^5_{9,124} \oplus f^5_{7,134} \oplus f^5_{4,234} \right\}.$$

Then, each relay node forwards its received signal to the set of connected users, i.e., $Y_{i,\boldsymbol{d},k} = X_{i,\boldsymbol{d}}$, $\forall k \in \mathcal{N}(\Gamma_i)$. The size of each transmitted signal is equal to the size of a piece of the encoded symbols, i.e., $\frac{F}{8}$. Thus, $R_1 = R_2 = \frac{1}{8}$. Now, utilizing its memory, user 1 can extract the pieces $f^1_{1,234}$ and $f^2_{1,234}$ from the signals received from relay nodes Γ_1 and Γ_2, respectively. Therefore, user 1 reconstructs f^1_1 and f^2_1, and decodes its requested file W_1. Similarly, user 2 reconstructs f^1_2 and f^3_2, then decodes W_2, and so on for the remaining users.

V. Combination Networks with Cache-Aided Relays

In this section, we investigate combination networks where the relays have caching capabilities, i.e., $M_1 > 0$.

First, one can observe, from the description of the scheme in Section IV, that when $M_1 \geq \frac{N}{r}$, the users' requests can be satisfied without the participation of the main server during the delivery phase, i.e., $R_1 = 0$, by allocating the pieces $f^j_{n,\mathcal{T}}$, $\forall n, \mathcal{T}$ in the cache memory of relay node Γ_j. Then, during the delivery phase, after announcing the demand vector, Γ_j transmits the pieces $f^j_{d_k,\mathcal{T}}$ to user k, $k \in \mathcal{N}(\Gamma_j)$ whenever $Index(j,k) \notin \mathcal{T}$. In the following, we extended our proposed scheme to the case where $0 < M_1 \leq \frac{N}{r}$.

A. Cache Placement Phase

Again, as a first step, the server divides each file into r equal-size subfiles. Then, it encodes them using an (h, r) maximum distance separable (MDS) code. We denote by f^i_n the resulting encoded symbol, where n is the file index and $i = 1, 2, .., h$. Then, the server divides each encoded symbol into two parts, $f^{i,1}_n$ and $f^{i,2}_n$ with sizes $\frac{M_1 F}{N}$ bits, and $(\frac{1}{r} - \frac{M_1}{N})F$ bits, respectively. We define the achievability for $M_2 = \frac{(t_1-t_2)M_1 r}{L} + \frac{t_2 N}{L}$, and $t_1, t_2 \in \{0, 1, .., L\}$, and the convex envelope is achievable by memory sharing.

First, the server places $f^{j,1}_n$, $\forall n$ in the cache memory of relay node Γ_j. Then, user k, with $k \in \mathcal{N}(\Gamma_j)$, caches a random fraction of $\frac{t_1}{L}$ bits from $f^{j,1}_n$, $\forall n$, which we denote by $f^{j,1}_{n,k}$. On the other hand, $f^{i,2}_n$ is divided into $\binom{L}{t_2}$ disjoint pieces each of which is denoted by $f^{i,2}_{n,\mathcal{T}_2}$, where $\mathcal{T}_2 \subseteq \{1, .., L\}$ and $|\mathcal{T}_2| = t_2$. The size of each piece is $\frac{\frac{1}{r} - \frac{M_1}{N}}{\binom{L}{t_2}} F$ bits. The server allocates the pieces $f^{j,2}_{n,\mathcal{T}}$, $\forall n$ in the cache memory of user k if $k \in \mathcal{N}(\Gamma_j)$ and $Index(j,k) \in \mathcal{T}$. Therefore, the cached contents at the relay nodes and end users are given by

$$V_k = \left\{ f^{k,1}_n : \ \forall n, \right\}, \tag{18}$$

$$Z_k = \left\{ f^{j,1}_{n,k}, f^{j,2}_{n,\mathcal{T}_2} : j \in \mathcal{N}(U_k), Index(j,k) \in \mathcal{T}_2, \forall n \right\}. \tag{19}$$

Clearly, the number of the accumulated bits in the cache memory of each relay node equals to $M_1 F$ bits. The number of the accumulated bits in the cache memory of each end user is given by

$$Nr\frac{M_1}{N}\frac{t_1}{L}F + Nr\frac{(\frac{1}{r} - \frac{M_1}{N})}{\binom{L}{t_2}}F\binom{L-1}{t_2-1} = \frac{M_1 t_1 r}{L}F + \frac{(N - M_1 r)t_2}{L}F = M_2 F, \tag{20}$$

i.e., these allocations satisfy the memory capacity constraints.

B. Coded Delivery Phase

After announcing the demand vector to all network users, the server and the relay nodes start to serve these requests.

For each relay Γ_j, at each transmission instance, we consider $\mathcal{S} \subseteq \{1, .., L\}$, where $|\mathcal{S}| = t_2 + 1$. For each choice of the set $\mathcal{S}$, the server transmits to the relay node Γ_j, the signal

$$X^{\mathcal{S}}_{j,\boldsymbol{d}} = \bigoplus_{\{i : i \in \mathcal{N}(\Gamma_j),\ Index(j,i) \in \mathcal{S}\}} f^{j,2}_{d_i, \mathcal{S}\setminus\{Index(j,i)\}}. \tag{21}$$

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

$$X_{j,\boldsymbol{d}} = \bigcup_{\mathcal{S}\subseteq\{1,..,L\} : |\mathcal{S}| = t_2+1} \left\{ X^{\mathcal{S}}_{j,\boldsymbol{d}} \right\}. \tag{22}$$

Then, Γ_j forwards the signal $X^{\mathcal{S}}_{j,\boldsymbol{d}}$ to user i whenever $Index(j,i) \in \mathcal{S}$. In addition, Γ_j transmits missing bits from $f^{j,1}_{d_i}$ to user i, $i \in \mathcal{N}(\Gamma_j)$. Therefore, the transmitted signal from Γ_j to user i can be expressed as

$$Y_{j,\boldsymbol{d},i} = \left\{ f^{j,1}_{d_i} \setminus f^{j,1}_{d_i,i} \right\} \bigcup_{\mathcal{S}\subseteq\{1,..,L\} : Index(j,i) \in \mathcal{S}} \left\{ X^{\mathcal{S}}_{j,\boldsymbol{d}} \right\}. \tag{23}$$

From its received signals and cached contents, user i obtains the encoded symbols $f^j_{d_i}$, $\forall j \in \mathcal{N}(U_i)$. Thus, user i can successfully reconstruct W_{d_i}.

In the following, we calculate the transmission rates resulting from this scheme. First, we observe that the server transmits $\binom{L}{t_2+1}$ sub-signals to each relay node, each of which has length equal to $\frac{\frac{1}{r}-\frac{M_1}{N}}{\binom{L}{t_2}}F$ bits, thus the transmission rate in bits from the server to each relay node is

$$R_1F = \binom{L}{t_2+1}\frac{\frac{1}{r}-\frac{M_1}{N}}{\binom{L}{t_2}}F = \frac{(L-t_2)(\frac{1}{r}-\frac{M_1}{N})}{t_2+1}F. \quad (24)$$

During the second hop, each relay node forwards $\binom{L-1}{t_2}$ from its received sub-signals to each of its connected end users. Additionally, it sends $(1-\frac{t_1}{L})\frac{M_1}{N}F$ bits, from its cache memory to each of its connected end users. Thus, the transmission rate in bits from each relay to each of its connected end users is

$$\begin{aligned} R_2F &= \binom{L-1}{t_2}\frac{(\frac{1}{r}-\frac{M_1}{N})}{\binom{L}{t_2}}F + (1-\frac{t_1}{L})\frac{M_1}{N}F \\ &= \frac{1}{r}\Big(1-\frac{t_2}{L}-\frac{(t_1-t_2)M_1r}{NL}\Big)F = \frac{1}{r}\Big(1-\frac{M_2}{N}\Big)F. \quad (25) \end{aligned}$$

Therefore, we can obtain the upper bound on the normalized transmission rates as stated in Theorem 2. Note that if M_2 is not in the form of $M_2 = \frac{(t_1-t_2)M_1r}{L} + \frac{t_2N}{L}$, with $t_1, t_2 \in \{0, 1, .., L\}$, we apply memory sharing as in [1] and [2] to achieve the convex envelope of the defined points.

Remark 3. Observe that the caching capabilities of the relay nodes help decrease the transmission load only during the first hop, i.e., R_1. The transmission load over the second hop, i.e., R_2, depends only on the size of end users' caches, as it is always equal to the local caching gain divided by the number of relay nodes connected to each end user, r. ■

Remark 4. It can be seen from (24) that when $t_2 = L$, i.e., $M_2 \geq N - M_1r$, we can achieve $R_1 = 0$. In other words, whenever $M_2 + M_1r \geq N$, i.e., the total memory at each end user and its connected relay nodes is sufficient to store the whole file library, the server is not required to transmit during the delivery phase. Also, note that whenever we have $M_2 < N - M_1r$, to minimize the transmission load, t_1 is set to zero, as the transmission rate over the first hop is minimized by maximizing t_2. ■

The derivation of the genie-aided cut-set lower bounds in Theorem 3 is based on the ideas in [1] and [3]. To derive a lower bound on R_1, we consider the cuts that contain a set of l relay nodes, $l \geq r$, and s end users out of the $\binom{l}{s}$ users who are exclusively connected to these relay nodes. The remaining users are served by a genie. For the lower bound on R_2, we consider the cut that contains only one of the end users. The details are omitted due to space limitation. In Fig. 2, we plot our lower and upper bounds for different sizes of the relay nodes cache memories. It can be observed that the normalized rates are decreasing functions of the memory capacities and whenever $M_2+rM_1 \geq 60$, $R_1 = 0$ while $R_2 = 0$ when $M_2 \geq 60$. Also, the gap between lower and upper bound decreases as M_2 increases.

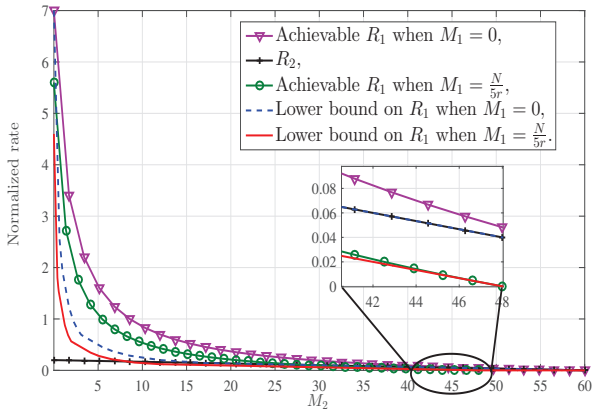

Fig. 2: Comparison between the lower and upper bounds for $N = 60$, $K = 56$, $h = 8$ and $r = 5$.

VI. Conclusions

In this work, we have investigated cache-aided combination networks, where both the relay nodes and end users are equipped with cache memories. By utilizing MDS coding and jointly optimizing both cache placement and delivery phases, we have proposed a new achievability scheme that satisfies the users' requests without the participation of the main server whenever the total memory at each end user and its connected relay nodes is sufficient to store the whole database. Our scheme generalizes the previous treatments that consider special cases, i.e., networks that do not employ caches at the relays and those that belong to a special class: of combination networks, i.e., resolvable networks.

In this work, we have provided lower and upper bounds. Current effort is underway to analyze the gap between the lower and the upper bounds. Future directions include utilizing the proposed techniques to extend the schemes in [8] to general combination networks under secrecy requirements.

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Info. Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

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

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

[4] M. Ji, M. F. Wong, A. M. Tulino, J. Llorca, G. Caire, M. Effros, and M. Langberg, "On the fundamental limits of caching in combination networks," in *16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2015.

[5] L. Tang and A. Ramamoorthy, "Coded caching for networks with the resolvability property," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[6] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. on Networking*, vol. 23, no. 4, pp. 1029–1040, 2015.

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

[8] A. A. Zewail and A. Yener, "Coded caching for resolvable networks with security requirements," in *the 3rd Workshop on Physical-Layer Methods for Wireless Security, CNS*, 2016.