

# A Layered Caching Architecture for the Interference Channel

Jad Hachem

University of California, Los Angeles  
Email: jadhachem@ucla.edu

Urs Niesen

Qualcomm NJ Research Center  
Email: urs.niesen@ieee.org

Suhas Diggavi

University of California, Los Angeles  
Email: suhas@ee.ucla.edu

**Abstract**—Recent work has studied the benefits of caching in the interference channel, particularly by placing caches at the transmitters. In this paper, we study the two-user Gaussian interference channel in which caches are placed at both the transmitters and the receivers. We propose a separation strategy that divides the physical and network layers. While a natural separation approach might be to abstract the physical layer into several *independent* bit pipes at the network layer, we argue that this is inefficient. Instead, the separation approach we propose exposes *interacting* bit pipes at the network layer, so that the receivers observe related (yet not identical) quantities. We find the optimal strategy within this layered architecture, and we compute the degrees-of-freedom it achieves. Finally, we show that separation is optimal in regimes where the receiver caches are large.

## I. INTRODUCTION

Traditional communication networks are connection centric, i.e., they establish a reliable connection between two fixed network nodes. However, instead of a connection to a specific destination node, modern network applications often require a connection to a specific piece of content. Consequently, network architectures are shifting from being connection centric to being content centric. These content-centric architectures make heavy use of in-network caching and, in order to do so, redesign the protocol stack from the network layer upwards [1].

Recent work in the information theory literature indicates that the availability of in-network caching can also benefit the physical layer. This information-theoretic approach to caching was introduced in the context of the noiseless broadcast channel in [2], where it was shown that significant performance gains can be obtained using cache memories at the *receivers*. The setting was extended to the interference channel in [3], which presented an achievable scheme showing performance gains using cache memories at the *transmitters*. The achievable scheme from [3] uses the cache memories to create many virtual transmitters and improves transmission rate by performing elaborate interference alignment between those virtual transmitters.

In this paper we continue the study of cache-aided interference channels, but we allow for caches at both the *transmitters and receivers* as shown in Fig. 1. Furthermore, we propose a simpler, layered communication architecture, separating the problem into a physical layer and a network layer as shown in

This work was supported in part by NSF grant #1423271.

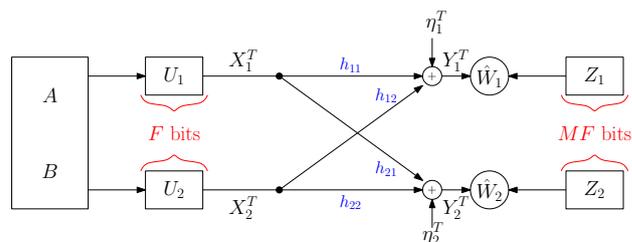


Fig. 1. The caching problem over the interference channel. The server holds files  $A$  and  $B$ , of size  $F$  bits each, and caches parts of them in four memories  $U_1, U_2, Z_1$ , and  $Z_2$ . The two users (circles) request files  $W_1, W_2 \in \{A, B\}$ , and aim to recover them using the output of the interference channel and their respective caches.

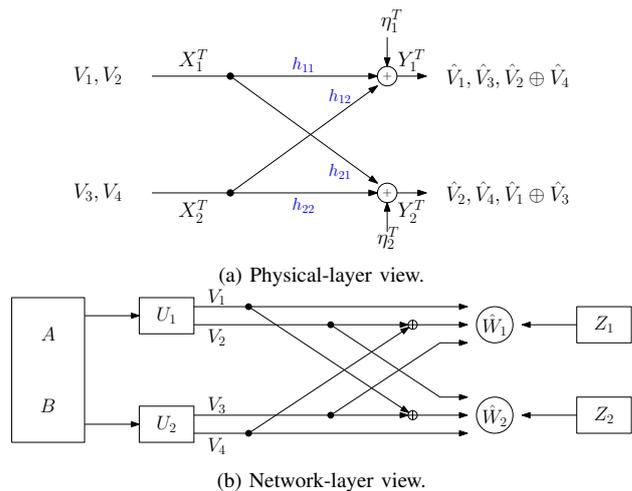


Fig. 2. Physical and network layers of the system in Fig. 1 under the proposed separation architecture.

Fig. 2. In other words, we propose a redesign of the protocol stack from the network layer downwards.

There are two seemingly natural network-layer abstractions for this problem. The first treats the physical layer as a standard interference channel and transforms it into two noninteracting error-free bit pipes. The second treats the physical layer as an X-channel and transforms it into four noninteracting error-free bit pipes. We argue that both of these abstractions are deficient. Instead a more appropriate abstraction needs to expose some of the algebraic structure of the underlying physical layer to the network layer. More

precisely, we propose a network-layer abstraction consisting of four *interacting* error-free bit pipes as illustrated in Fig. 2b.

We derive optimal communication schemes for this layered architecture. An interesting feature of these schemes is that they require coding during both the content placement and delivery phases (whereas the caching schemes studied in the prior literature utilize coding for only one or the other). For the regime of large cache sizes, it turns out that the layered architecture itself is fundamental, i.e., that the separation of the communication problem into the two proposed layers is without loss of optimality.

*Related work:* The information-theoretic framework for coded caching was introduced in [2] in the context of the deterministic broadcast channel. This has been extended to online caching systems [4], heterogeneous cache sizes [5], unequal file sizes [6], and improved converse arguments [7], [8]. Content caching and delivery in device-to-device networks, multi-server topologies, and heterogeneous wireless networks have been studied in [9], [10], [11], [12], [13]. This framework has also been extended to hierarchical (tree) topologies in [14]. More recently, it has been extended to interference channels in [3], where only transmit caches were considered and several interesting schemes were developed.

The paper is organized as follows. Section II provides the formal problem setting and introduces the proposed layered communication architecture. Section III presents a complete performance characterization for this architecture. A detailed description of the network-layer processing together with optimality proofs are given in the extended version of this paper [15].

## II. PROBLEM SETTING AND LAYERED ARCHITECTURE

We study a system in which a server delivers files to two users across a Gaussian interference channel with the help of caches at all network nodes, described in Section II-A. We propose a layered communication architecture consisting of a physical and a network layer. We introduce the physical layer in Section II-B and the network layer in Section II-C.

In this paper, we restrict the number of files to just two. We also fix the size of the *transmitter* caches to the smallest size required for normal operation of the system. This allows us to study the transmission rate of the files as a function of the receiver cache memory, without worrying about additional complexities arising from larger transmitter caches and a greater number of files. In fact, the results turn out to be rather complex even in this simplified setting. Extensions to this setup are a work in progress.

### A. The caching problem

Consider the setup in Fig. 1. A server has two files  $A$  and  $B$  of size  $F$  bits each. The server is connected to two transmitters. These in turn are connected to two receivers, referred to as users, through a Gaussian interference channel.

Communication occurs in two phases. In the *placement phase*, the server pushes file information to four caches denoted by  $U_1$ ,  $U_2$ ,  $Z_1$ , and  $Z_2$ . Caches  $U_1$  and  $U_2$  are at

transmitters 1 and 2, respectively, and can each store up to  $F$  bits. Caches  $Z_1$  and  $Z_2$  are at receivers 1 and 2, respectively, and can each store up to  $MF$  bits. The parameter  $M \geq 0$  is called the (normalized) *memory size*.

In the subsequent *delivery phase* each user requests one of the files (possibly the same). Formally, users 1 and 2 request files  $W_1, W_2 \in \{A, B\}$ , respectively, from the server. Each transmitter  $i$  then sends a length- $T$  sequence  $X_i^T$  through the interference channel. This message  $X_i^T$  can depend only on the transmitter's cache content  $U_i$ . In other words, the server itself does not participate in the delivery phase.

We impose a power constraint

$$\frac{1}{T} \sum_{t=1}^T X_{i,t}^2 \leq P$$

on the transmitted sequence  $X_i^T$ . Each receiver  $i$  observes the channel output

$$Y_i^T = h_{i1}X_1^T + h_{i2}X_2^T + \eta_i^T$$

of the interference channel, where  $\eta_{i,t} \sim \mathcal{N}(0, 1)$  is iid additive Gaussian noise. The receiver combines the channel output  $Y_i^T$  with the cache content  $Z_i$  to decode the requested file  $\hat{W}_i$ .

We define the *rate* of the system as  $R = F/T$ . Our goal is to characterize the trade-off between the rate  $R$  and the receiver cache memory  $M$  under the power constraint  $P$ . Formally, we say that a tuple  $(R, M, P)$  is achievable if there exists a strategy with rate  $R$ , receiver cache memory  $M$ , and power constraint  $P$  such that

$$\Pr \left\{ (\hat{W}_1, \hat{W}_2) \neq (W_1, W_2) \right\} \rightarrow 0 \text{ as } T \rightarrow \infty$$

for all possible demands  $(W_1, W_2) \in \{A, B\}^2$ . We define the optimal rate function as

$$R^*(M, P) = \sup \{ R : (R, M, P) \text{ is achievable} \}.$$

We are particularly interested in the high-SNR regime and focus on the optimal degrees of freedom (DoF)

$$d^*(M) = \lim_{P \rightarrow \infty} \frac{R^*(M, P)}{\frac{1}{2} \log P}$$

of the system. It will be convenient to work with the inverse-DoF  $1/d^*(M)$ , since it is a convex function of  $M$  [3, Lemma 1].

### B. Physical-layer view

We next describe the physical-layer view of the caching problem. There are several possible strategies of how to perform the layer separation, each leading to a different physical-layer view. We start by describing the advantages and disadvantages of some of them.

A natural strategy might be the complete separation of the physical and network layers, abstracting the physical channel into parallel error-free bit pipes. This can be achieved by treating the physical layer as a standard interference channel (IC) or a standard X-channel (XC). The IC abstraction gives

the network layer two independent bit pipes, each of them providing a DoF of  $1/2$  for a sum DoF of 1. The XC abstraction can do slightly better by creating four bit pipes of DoF  $1/3$  each for a sum DoF of  $4/3$ . However, by “relaxing” the separation, we can provide to the network layer the same four bit pipes of the XC, but with two additional linear combinations of these bit pipes. This can improve the performance of the system as soon as caches are available at the receivers. For example, if each receiver cache can store up to four fifths of a file, then we show below that a sum DoF of  $10/3$  can be achieved, compared with  $20/9$  for the XC and  $5/3$  for the IC for the same memory value.

The physical-layer view of the caching problem adopted in this paper is therefore the Gaussian interference channel together with an X-channel message set, i.e., four messages  $V_1, \dots, V_4$ , one to be sent from each transmitter to each receiver as shown in Fig. 2a. The physical layer applies real interference alignment [16] in order to, loosely speaking, allow recovery of the following quantities:

- Receiver 1 recovers  $V_1, V_3$ , and  $V_2 + V_4$ ;
- Receiver 2 recovers  $V_2, V_4$ , and  $V_1 + V_3$ .

More formally, real interference alignment is a modulation scheme that uses a one-dimensional lattice to create channel inputs  $G_i^T$  corresponding to message  $V_i$ . Each transmitter then creates the channel inputs

$$\begin{aligned} X_1^T &= h_{22}G_1^T + h_{12}G_2^T; \\ X_2^T &= h_{21}G_3^T + h_{11}G_4^T. \end{aligned}$$

At the channel output, the following signals will be received:

$$\begin{aligned} Y_1^T &= h_{11}h_{22}G_1^T + h_{12}h_{21}G_3^T + h_{11}h_{12}(G_2^T + G_4^T) + \eta_1^T; \\ Y_2^T &= h_{12}h_{21}G_2^T + h_{11}h_{22}G_4^T + h_{21}h_{22}(G_1^T + G_3^T) + \eta_2^T. \end{aligned}$$

Using the lattice structure of the  $G_i^T$ 's, user 1 can demodulate  $G_1^T, G_3^T$ , and  $G_2^T + G_4^T$ , while user 2 can demodulate  $G_2^T, G_4^T$ , and  $G_1^T + G_3^T$ . Using a linear block code over this modulated channel as proposed in [17], we can ensure that user 1 can decode  $V_1, V_3$ , and  $V_2 \oplus V_4$ , while user 2 can decode  $V_2, V_4$ , and  $V_1 \oplus V_3$ . The addition  $\oplus$  here is over some finite field. For the purposes of this paper, we can assume that this field is  $\text{GF}_2$ . If  $V_i \in [2^{R_i^T}]$ , then the rate

$$R' = \frac{1}{3} \cdot \frac{1}{2} \log P + o(\log P), \quad (1)$$

corresponding to a DoF of  $1/3$ , is achievable [16].

### C. Network-layer view

The network-layer abstraction replaces the noisy interference channel in Fig. 1 with the noiseless channel in Fig. 2b. The transmitters send four messages  $V_1, \dots, V_4$ , each of size  $cF$  bits, such that  $V_1$  and  $V_2$  are sent by transmitter 1 and depend only on  $U_1$ , and  $V_3$  and  $V_4$  are sent by transmitter 2 and depend only on  $U_2$ . The messages go through the channel and the users receive the following outputs (the symbol “ $\oplus$ ” denotes bitwise XOR):

- User 1 receives  $V_1, V_3$ , and  $V_2 \oplus V_4$ ;

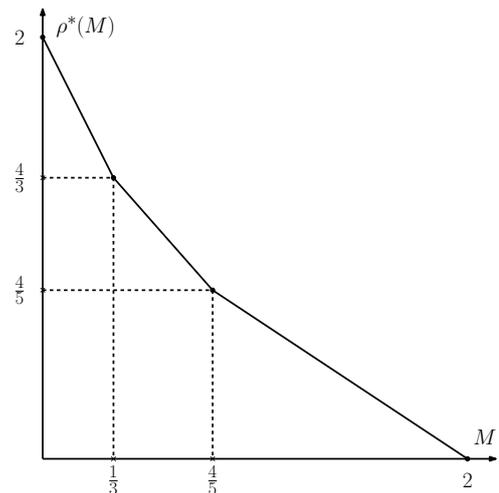


Fig. 3. Optimal trade-off between sum-network load  $\rho$  and receiver memory size  $M$  at the network layer.

- User 2 receives  $V_2, V_4$ , and  $V_1 \oplus V_3$ .

The quantity  $c$  is called the (normalized) *network load*. The sum of all the network loads  $\rho = 4c$  is called the *sum network load*.

A pair  $(M, \rho)$  is said to be achievable if a strategy with receiver memory  $M$  and sum network load  $\rho$  can deliver any requested files to the two users with high probability as the file size  $F \rightarrow \infty$ . For a cache memory  $M$ , we call the smallest achievable sum network load  $\rho^*(M)$ .

## III. PERFORMANCE ANALYSIS

We start with a complete characterization of the network layer trade-off in Section III-A. We then translate this to the end-to-end system in Section III-B, giving an achievability result for the original interference channel. We also show that in the high-memory regime, our layered architecture is end-to-end optimal.

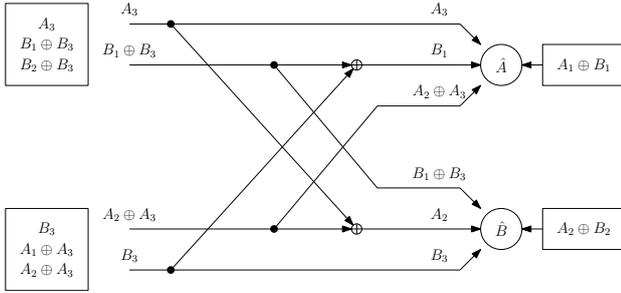
### A. Network-layer performance analysis

The following theorem, visualized in Fig. 3, gives a full characterization of the optimal sum network load  $\rho^*(M)$  as a function of receiver cache memory  $M$ .

**Theorem 1.** *At the network layer, the optimal trade-off between  $\rho$  and  $M$  is:*

$$\rho^*(M) = \max \left\{ 2 - 2M, \frac{12}{7} - \frac{8}{7}M, \frac{4}{3} - \frac{2}{3}M, 0 \right\}. \quad (2)$$

Proving this theorem requires the usual two steps: finding a scheme that achieves the right-hand-side in (2), and proving matching lower bounds. In the lower bounds, a non-cut-set inequality is needed to show optimality, as cut-set bounds (see [18, Theorem 15.10.1]) alone are insufficient. The details of both the achievability and the lower bounds are given in [15], but we will here give a brief overview of the ideas involved.


 Fig. 4. Achievable strategy for  $M = 1/3$  when the demands are  $(A, B)$ .

*Overview of the achievability:* The expression in (2) is a piece-wise linear function, with the following  $(M, \rho)$  corner points:  $(0, 2)$ ,  $(1/3, 4/3)$ ,  $(4/5, 4/5)$ , and  $(2, 0)$ . By time and memory sharing, the function  $\rho^*(M)$  is a convex function of  $M$ . Hence, it suffices to prove that the above corner points are achievable. We will here give a high-level overview of the scheme for the two most interesting points:  $(1/3, 4/3)$  and  $(4/5, 4/5)$ . We will consider only the demand pair  $(A, B)$  in the delivery phase, as the others are similar.

The strategy for point  $(M, \rho) = (1/3, 4/3)$  is illustrated in Fig. 4. If  $M = 1/3$ , then each receiver cache can store the equivalent of one third of a file. We split each file into three parts:  $A = (A_1, A_2, A_3)$  and  $B = (B_1, B_2, B_3)$ , and store  $Z_1 = (A_1 \oplus B_1)$  and  $Z_2 = (A_2 \oplus B_2)$  at the receivers. The transmitter caches store  $U_1 = (A_3, B_1 \oplus B_3, B_2 \oplus B_3)$  and  $U_2 = (B_3, A_1 \oplus A_3, A_2 \oplus A_3)$ . Notice that the contents of  $U_1$  and  $U_2$  are independent, and that each has a size of  $F$  bits.

Suppose now that user 1 requests  $A$  and user 2 requests  $B$ . Then the transmitters send the messages

$$\begin{aligned} V_1 &= A_3; & V_2 &= B_1 \oplus B_3; \\ V_3 &= A_2 \oplus A_3; & V_4 &= B_3. \end{aligned}$$

Each  $V_i$  has a size of  $F/3$  bits, so  $c = 1/3$  and  $\rho = 4c = 4/3$ .

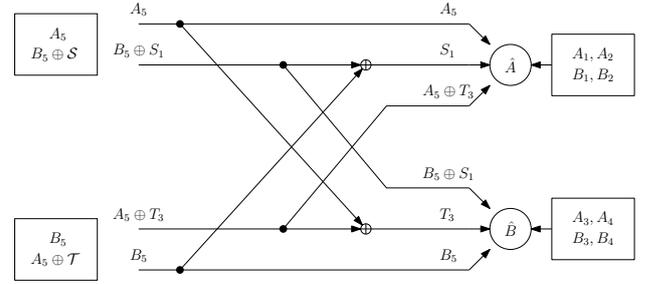
User 1 receives

$$\begin{aligned} (V_1, V_2 \oplus V_4, V_3) &= (A_3, (B_1 \oplus B_3) \oplus B_3, A_2 \oplus A_3) \\ &= (A_3, B_1, A_2 \oplus A_3). \end{aligned}$$

The user can recover  $A_2$  from  $A_2 \oplus A_3$  and  $A_3$ , and decode  $A_1$  from  $B_1$  and the cache content  $A_1 \oplus B_1$ . Thus, user 1 has completely recovered file  $A$ . User 2 applies a similar approach to decode file  $B$ .

Note that the transmitters take advantage of the contents of the receiver caches to send a reduced amount of information to the users. Specifically, they need to communicate  $A_3$  to user 1,  $B_3$  to user 2, and both  $A_2$  and  $B_1$  to both users. Using a similar strategy for the other demands, we see that the point  $(M, \rho) = (1/3, 4/3)$  is achievable.

Let us now consider point  $(M, \rho) = (4/5, 4/5)$ , whose strategy is visualized in Fig. 5. When  $M = 4/5$ , each receiver can store the equivalent of four fifths of a file. We thus divide each file into five equal parts,  $A = (A_1, \dots, A_5)$  and  $B = (B_1, \dots, B_5)$ , and place  $Z_1 = (A_1, A_2, B_1, B_2)$  and  $Z_2 = (A_3, A_4, B_3, B_4)$  in the receiver caches. The


 Fig. 5. Achievable strategy for  $M = 4/5$  when the demands are  $(A, B)$ . We write  $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$  and  $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$ , where  $S_1 = B_2 \oplus A_4$  and  $T_3 = B_1 \oplus A_3$  (the others are not used for demands  $(A, B)$ ).

transmitter caches, which have a capacity of one file each, store  $U_1 = (A_5, B_5 \oplus S)$  and  $U_2 = (B_5, A_5 \oplus T)$ , where  $\mathcal{S}$  and  $\mathcal{T}$  each consist of four linear combinations

$$\begin{aligned} \mathcal{S} &= \{S_i\}_{i=1}^4 = \{B_2 \oplus A_4, A_1 \oplus B_3, B_1 \oplus B_3, B_2 \oplus B_4\}, \\ \mathcal{T} &= \{T_i\}_{i=1}^4 = \{A_1 \oplus A_3, A_2 \oplus A_4, B_1 \oplus A_3, A_2 \oplus B_4\}, \end{aligned}$$

of parts of  $A$  and  $B$ .

Assume the users request files  $A$  and  $B$ , respectively. The transmitters then send the following messages:

$$\begin{aligned} V_1 &= A_5; \\ V_2 &= B_5 \oplus S_1 = B_5 \oplus (B_2 \oplus A_4); \\ V_3 &= A_5 \oplus T_3 = A_5 \oplus (B_1 \oplus A_3); \\ V_4 &= B_5. \end{aligned}$$

Notice that the size of each  $V_i$  is  $F/5$  bits, so that  $c = 1/5$  and  $\rho = 4/5$ .

User 1 receives

$$\begin{aligned} (V_1, V_2 \oplus V_4, V_3) &= (A_5, B_5 \oplus B_5 \oplus S_1, A_5 \oplus T_3) \\ &= (A_5, S_1, A_5 \oplus T_3). \end{aligned}$$

Recall that user 1's cache already contains  $Z_1 = (A_1, A_2, B_1, B_2)$ . This gives it the first two parts of  $A$ , and it receives the fifth part  $A_5$  from the channel. Furthermore, using  $A_5$  and  $A_5 \oplus T_3$  allows it to decode  $T_3 = (B_1 \oplus A_3)$ , and using  $B_1$  from its cache it can recover  $A_3$ . Finally, it can combine  $B_2$  from its cache with  $S_1 = (B_2 \oplus A_4)$  to decode the last part,  $A_4$ . User 1 has therefore completely recovered file  $A$ , and in a similar manner user 2 can recover file  $B$ . By transmitting similar linear combinations for the other demands, we can show that the point  $(M, \rho) = (4/5, 4/5)$  is achieved.

*Overview of the converse:* For the outer bounds, we must show that any achievable pair  $(M, \rho)$  must satisfy the following inequalities:

$$\begin{aligned} \rho &\geq 2 - 2M; \\ \rho &\geq \frac{12}{7} - \frac{8}{7}M; \\ \rho &\geq \frac{4}{3} - \frac{2}{3}M. \end{aligned}$$

The first and third inequalities can be proved using cut-set bounds. We will here focus on the second inequality, which

requires a non-cut-set argument. For convenience, we will write it in terms of  $c$  instead of  $\rho$ , and rearrange it as

$$7c + 2M \geq 3.$$

Informally, the proof proceeds as follows. Consider the three outputs of the channel observed by user 1, and suppose they result from user 1 requesting file  $A$  (user 2's request is irrelevant). Call these outputs collectively  $\mathbf{Y}^A$ . Then, these outputs combined with cache  $Z_1$  should allow user 1 to decode file  $A$ . In parallel, consider the four inputs of the channel when user 1 requests file  $B$  and user 2 requests file  $A$ . Collectively call these inputs  $\mathbf{V}^{BA}$ . Combining these inputs with cache  $Z_2$  should allow user 2 to also decode file  $A$ .

So far, user 1 has used  $\mathbf{Y}^A$  with its cache to decode  $A$ , and user 2 has combined  $\mathbf{V}^{BA}$  with its cache to also decode  $A$ . These decodings have occurred separately from each other. Let us now combine everything together (i.e.,  $\mathbf{V}^{BA}$ ,  $\mathbf{Y}^A$ , and the two caches  $Z_1$  and  $Z_2$ ). Then, user 1 should decode the remaining file  $B$  using  $\mathbf{V}^{BA}$  and its cache  $Z_1$ .

In summary, we have argued that four input messages  $\mathbf{V}^{BA}$  and three output messages  $\mathbf{Y}^A$ , each of which has a size of  $cF$  bits, as well as two caches  $Z_1$  and  $Z_2$ , each of which has a size of  $MF$  bits, should contain enough information to decode three files ( $A$  twice and  $B$  once) of size  $F$  bits each. This can be mathematically expressed as

$$(4 + 3)cF + 2MF \geq 3F,$$

thus proving the inequality. The complete formal proof can be found in [15].

### B. End-to-end performance analysis

Applying the physical-layer processing as described in Section II-B, the  $R'T$  message bits at the physical layer correspond to the  $cF$  message bits in the network layer, i.e.,  $R'T = cF$ . Since the files  $A$  and  $B$  have a size of  $F = RT$  bits, this implies that  $R' = cR$ . Therefore, using (1), the following rate is achievable:

$$R = \frac{R'}{c} = \frac{1}{3c} \cdot \frac{1}{2} \log P + o(\log P).$$

Equivalently, we can achieve the inverse-DoF of  $\frac{1}{d(M)} = 3c = \frac{3}{4}\rho$ . To get the largest DoF possible within this strategy, we want to achieve the smallest possible  $\rho$ , which leads to the following lemma.

**Lemma 1.** *Using the proposed separation architecture, the following end-to-end inverse-DoF is achievable:*

$$1/d(M) = (3/4)\rho^*(M),$$

where  $\rho^*(M)$  is the optimal trade-off between sum-network-layer and cache memory at the network layer.

A direct combination of Lemma 1 and Theorem 1 leads to the following result.

**Theorem 2.** *The following inverse-DoF is achievable:*

$$\frac{1}{d(M)} = \max \left\{ \frac{3}{2} - \frac{3}{2}M, \frac{9}{7} - \frac{6}{7}M, 1 - \frac{1}{2}M, 0 \right\}.$$

The optimality of  $\rho^*(M)$  in the network layer implies that our strategy is optimal among all separation-based approaches with the physical-layer processing as proposed here. In fact, it provides a net improvement over the natural layering strategies: up to a factor of  $3/2$  and a factor of 2 improvement over the X-channel and the interference channel layering schemes, respectively. In addition, we can show that it is optimal over *all possible strategies* for large receiver cache memory ( $M \geq 4/5$ ); see [15] for more details. Thus, in this regime, the proposed separation into a physical layer and a network layer is without loss of optimality. We are currently working on extending these results to smaller memories.

### REFERENCES

- [1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012.
- [2] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *Information Theory, IEEE Transactions on*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [3] —, "Cache-aided interference channels," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, June 2015, pp. 809–813.
- [4] R. Pedarsani, M. Maddah-Ali, and U. Niesen, "Online coded caching," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 1878–1883.
- [5] S. Wang, W. Li, X. Tian, and H. Liu, "Coded caching with heterogeneous cache sizes," *arXiv:1504.01123v3 [cs.IT]*, Aug. 2015.
- [6] J. Zhang, X. Lin, C.-C. Wang, and X. Wang, "Coded caching for files with distinct file sizes," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, June 2015, pp. 1686–1690.
- [7] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, June 2015, pp. 1696–1700.
- [8] A. Sengupta, R. Tandon, and T. Clancy, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," in *Information Theory (ISIT), 2015 IEEE International Symposium on*, June 2015, pp. 1691–1695.
- [9] M. Ji, G. Caire, and A. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *Selected Areas in Communications, IEEE Journal on*, vol. 34, no. 1, pp. 176–189, Jan 2016.
- [10] S. P. Shariatpanahi, A. S. Motahari, and B. H. Khalaj, "Multi-server coded caching," *arXiv:1503.00265v1 [cs.IT]*, Mar. 2015.
- [11] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1107–1115.
- [12] J. Hachem, N. Karamchandani, and S. Diggavi, "Multi-level coded caching," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Jun. 2014.
- [13] —, "Content caching and delivery over heterogeneous wireless networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*, April 2015, pp. 756–764.
- [14] N. Karamchandani, U. Niesen, M. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, June 2014, pp. 2142–2146.
- [15] J. Hachem, U. Niesen, and S. Diggavi, "A layered caching architecture for the interference channel," *arXiv:1605.01668v1 [cs.IT]*, May 2016.
- [16] A. Motahari, S. Oveis-Gharan, M.-A. Maddah-Ali, and A. Khandani, "Real interference alignment: Exploiting the potential of single antenna systems," *Information Theory, IEEE Transactions on*, vol. 60, no. 8, pp. 4799–4810, Aug 2014.
- [17] U. Niesen and P. Whiting, "The degrees of freedom of compute-and-forward," *Information Theory, IEEE Transactions on*, vol. 58, no. 8, pp. 5214–5232, Aug 2012.
- [18] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.