

Wireless MapReduce Distributed Computing

Fan Li, Jinyuan Chen and Zhiying Wang

Abstract—Motivated by mobile edge computing and wireless data centers, we study a wireless distributed computing framework where the distributed nodes exchange information over a wireless interference network. Our framework follows the structure of MapReduce. This framework consists of Map, Shuffle, and Reduce phases, where Map and Reduce are computation phases and Shuffle is a data transmission phase. In our setting, we assume that the transmission is operated over a wireless interference network. We demonstrate that, by duplicating the computation work at a cluster of distributed nodes in the Map phase, one can reduce the amount of transmission load required for the Shuffle phase. In this work, we characterize the fundamental tradeoff between computation load and communication load, under the assumption of one-shot linear schemes. The proposed scheme is based on side information cancellation and zero-forcing, and we prove that it is optimal in terms of computation-communication tradeoff. The proposed scheme outperforms the naive TDMA scheme with single node transmission at a time, as well as the coded TDMA scheme that allows coding across data, in terms of the computation-communication tradeoff.

Index Terms—Distributed computing, MapReduce, wireless interference network, interference management.

I. INTRODUCTION

In recent years, wireless distributed computing technologies developed rapidly due to the advancements in wireless communications and devices. For example, interconnected autonomous vehicles can utilize distributed computing for collision avoidance and congestion management. For another example, distributed computing among smart phones and nearby fog nodes can implement augmented reality for gaming or entertainment. Other use cases of wireless distributed computing include wireless data centers [1], [2], cloud computing in wireless networks [3], edge computing and fog computing for mobile networks and Internet of Things (IoT) [4]–[8].

In this work, we study distributed computing based on the MapReduce framework over a wireless interference network. In MapReduce distributed computing (cf. [9], [10]), data is first split and processed (called *Map*) at the distributed nodes, and then the results are *shuffled* (called *Shuffle*), and processed again (called *Reduce*). As the amount of data and the number of nodes grow, the Shuffle phase could lead to a significant delay for the overall performance. In this work, we study a MapReduce-based wireless distributed computing

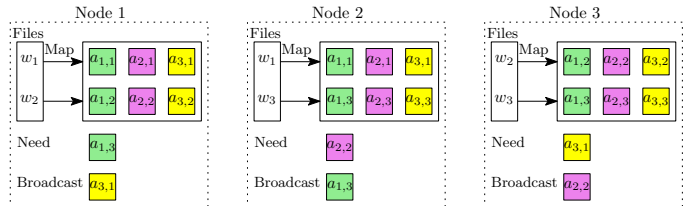


Fig. 1. An example of wireless distributed computing with $K = Q = N = 3$ and $r = 2$.

framework, where the Shuffle phase is operated over a wireless interference network, and explore the advantages of wireless communication to reduce the system latency.

We parameterize the MapReduce problem by N, K, r, Q , where N is the number of data files, K is the number of nodes, each file is duplicated at r nodes on average (called *computation load*), and Q is the number of Reduce functions. See Fig. 1 for an example. In this example, three distributed nodes ($K = 3$) seek to compute three Reduce functions ($Q = 3$) for three data files ($N = 3$), with each file stored at two nodes ($r = 2$). Every Map function takes one file as input, and outputs 3 intermediate values, one for each Reduce function. The intermediate value is denoted as $a_{q,n}$ for File n and Reduce function q . The Reduce function q takes $(a_{q,1}, a_{q,2}, a_{q,3})$ as inputs and produces the q -th final value. In the Map phase, every node computes 6 intermediate values for 2 files. For example, Node 1 computes 6 intermediate values, i.e., $\{a_{q,n} : q = 1, 2, 3, n = 1, 2\}$, for Files 1 and 2. In the Shuffle phase, some intermediate values are communicated in order to complete the computation in the Reduce phase. In the Reduce phase, assume that Node k computes the k -th Reduce function, for $k = 1, 2, 3$. In order to compute the first Reduce function, Node 1 needs input $(a_{1,1}, a_{1,2}, a_{1,3})$. While $a_{1,1}$ and $a_{1,2}$ are already cached locally, $a_{1,3}$ needs to be transmitted from a different node in the Shuffle phase. Similarly, Node 2 requires $a_{2,2}$ and Node 3 requires $a_{3,1}$ in the Shuffle phase.

In our setting, communication in the Shuffle phase takes place over a wireless interference channel. Assume that the channel state information is available to all nodes, and the communication is full-duplex. One possible application scenario is in data centers, where the environment (and hence the channel) is fixed for a long enough period, hence one may assume that channel state information is available at all users. Let the (non-interfered) transmission time of 1 intermediate value be 1 time unit, namely, a coded packet corresponding to $a_{q,n}$ is transmitted using 1 time unit, such that $a_{q,n}$ can be successfully decoded. In order to handle interference, we have the following possible solutions.

- If we use a naive *uncoded* time-division multiple access (TDMA) broadcast scheme, allowing only 1 node to

Fan Li and Jinyuan Chen are with Louisiana Tech University, Department of Electrical Engineering, Ruston, LA 71272, US (emails: fli005@latech.edu, jinyuan@latech.edu). Zhiying Wang is with University of California, Irvine, Center for Pervasive Communications and Computing (CPCC), Irvine, CA 92697, US (email:zhiying@uci.edu). This work was presented in part at the 2018 IEEE International Symposium on Information Theory. The work of Jinyuan Chen was partly supported by Louisiana Board of Regents Support Fund (BoRSF) Research Competitiveness Subprogram (RCS) under grant 32-4121-40336. Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

transmit 1 intermediate value at any time unit, we need 3 time units to transmit in total.

- We could also use a *coded* TDMA broadcast scheme (cf. [10]), allowing only 1 node to transmit 1 *coded* intermediate value at any time. For example, Node 3 can transmit a linear combination of the coded packets of $a_{1,3}$ and $a_{2,2}$. Through the cached intermediate values, Nodes 1 and 2 can respectively decode their desired information. Then Node 1 can transmit $a_{3,1}$ for Node 3. We need 2 time units in total.
- Alternatively, we can let 3 nodes transmit *at the same time*. Each node receives the superposition of the 3 transmitted symbols. However, the two undesired symbols can be canceled using cached intermediate values (side information). Thus the desired symbol is decoded. We need only 1 time unit.

In this paper we study the shuffle communication time units normalized by NQ , termed as *communication load*, which is a function of K and the computation load r . For practical purposes, we assume that the *one-shot linear* scheme is used, where each intermediate value is encoded into a coded packet, and the transmitted symbol is a linear combination of the coded packets in the cache, ensuring that the coded packet can be decoded at the intended receiver with a linear operation. We show that the optimal communication load is given as

$$\frac{1 - \frac{r}{K}}{\min\{K, 2r\}}, \quad r \in \{1, 2, \dots, K\}. \quad (1)$$

The significant improvement of our scheme compared to uncoded and coded TDMA schemes is depicted in Fig. 2. As shown in Fig. 2, considering the case of $r = 1$, namely, when there is no extra computation in the Map phase, the communication load of the proposed one-shot linear scheme is 50% lower than that of both uncoded TDMA and coded TDMA schemes. For the case of $r = 5$, the communication load of the proposed one-shot linear scheme is 90% lower than that of uncoded TDMA scheme and 50% lower than that of coded TDMA scheme.

The two key factors to obtain (1) are side information cancellation and zero-forcing. The role of side information has been demonstrated in the example of Fig. 1. If an intermediate value is stored in multiple nodes, then by simultaneously transmitting this intermediate value from these nodes, the corresponding signal may be zero-forced at some undesired receivers. It is similar to the interference cancellation in a MISO interference channel. In fact, we convert our problem to a MISO interference channel problem to obtain the converse.

The technical challenges of obtaining the optimal communication load of (1) lie in both the converse and the achievability. For the converse, our main task is to bound the maximum number of coded packets that can be transmitted simultaneously at the ℓ -th time unit, denoted by $|\mathcal{D}_\ell|$. When each file is replicated r times, referred to as *symmetric file replications*, we prove that $|\mathcal{D}_\ell|$ is upper bounded by a value that depends on the number of times each file is replicated, i.e., r . However, when different files are replicated with different numbers of times, referred to as *asymmetric file replications*, the problem becomes more challenging, because we have N

parameters, each corresponding to the replication number of one file. For this case, even though each $|\mathcal{D}_\ell|$ depends on the replication numbers of the particular files involved in time unit ℓ , we prove that the total number of required transmission time units is upper bounded by a value that depends on the average number of times the files are replicated (i.e., r). In fact, this proof combined with our achievability shows that asymmetric file replications cannot have a better communication load than symmetric ones.

For the achievability, we provide an explicit one-shot linear scheme, in which files are placed symmetrically, and the number of transmitted coded packets at each time unit attains the maximum of $|\mathcal{D}_\ell|$ from the converse. Note that the difficulty of the achievability lies in the case with $r < K/2$, where interference might not be eliminated completely if all nodes participate in transmission simultaneously. For this case, the proposed scheme guarantees that a subset of nodes can receive packets without interference at each time unit, by using side information cancellation and *partial* zero-forcing.

Related work: In [10], [11], coded distributed computing for MapReduce is introduced to utilize cache and broadcast to reduce communication delay. A lot of work appeared after that regarding communication in MapReduce distributed computing [12]–[24]. Specifically, [12] aims at reducing the overall computation and communication time. In [13] and [14], the number of computed Map intermediate results is also considered as one of the performance parameters. In [15], heterogeneous nodes with different communication constraints are considered. In the scheme of [10] the required minimum number of files, termed subpacketization, is exponential in K and r . The works in [16]–[18] focus on reducing the subpacketization. When the Reduce function depends on only a subset of the files, described by a graph, the problem is studied in [19]. When each node can broadcast only to a subset of the nodes, the problem is addressed in [20]. Linear aggregation of intermediate results at the Reduce stage is studied in [21]. Wireless MapReduce is studied in [22] where the distributed nodes must be connected through a wireless access point (or a relay), while in our paper the nodes can directly communicate with each other and the communication channel is a wireless *interference* channel. In our work, coding is used in a smart way for improving the performance of wireless distributed computing. In some other research directions, coding was used in different applications such as data shuffling [25]–[27], caching [28]–[33], and straggling distributed computing [34]–[44]. Another topic for distributed computing is federated learning, where communication efficiency is one of the main concerns [45]–[47].

The remainder of this work is organized as follows. Section II describes the system model. Section III provides the main results of this work. The converse proof is described in Sections V, while the achievability proof is described in Sections VI. Section IV provides the scheme examples. The work is concluded in Section VII.

Notation: Throughout this work, $[c_1 : c_2]$ denotes the set of integers from c_1 to c_2 , for some nonnegative integers $c_1 \leq c_2$. $|\bullet|$ denotes the magnitude of a scalar or the cardinality of a set. $o(\cdot)$ is the standard Landau notation, where $f(x) = o(g(x))$

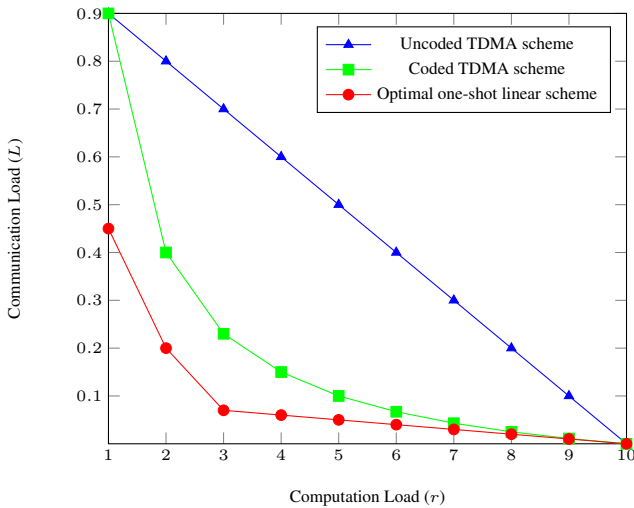


Fig. 2. Comparison on the communication load vs. computation load performance for uncoded TDMA scheme, coded TDMA, and the optimal one-shot linear scheme, given $K = 10$, $N = 2520$, and $Q = 360$.

implies that $\lim_{x \rightarrow \infty} f(x)/g(x) = 0$. \mathbb{C} and \mathbb{R} denote the sets of complex numbers and real numbers, respectively. \mathbb{F}_2^q denotes the set of q -tuples over the binary field. \mathbb{N}^+ denotes the set of positive natural numbers. Logarithms are in base 2. $\lceil c \rceil$ denotes the least integer that is no less than c , and $\lfloor c \rfloor$ denotes the greatest integer that is no larger than c . $s \sim \mathcal{CN}(0, \sigma^2)$ denotes that the random variable s has a circularly symmetric complex normal distribution with zero mean and σ^2 variance.

II. SYSTEM MODEL

We consider a wireless distributed computing system based on a MapReduce framework (cf. [9], [10]), where K nodes (servers) first compute Map functions to generate intermediate values for N input files, and then exchange (Shuffle) information over a wireless interference channel, and finally compute Q outputs (Reduce functions), for some $K, N, Q \in \mathbb{N}^+$, with $N \geq K$. The formal model is described as follows.

Map phase: Consider a total of N independent input files w_1, w_2, \dots, w_N . Let $\mathcal{M}_k \subseteq [1 : N]$ denote the indices of the files assigned at Node k , $k \in [1 : K]$. For each file w_n , $n \in \mathcal{M}_k$, after the Map function Node k generates Q intermediate values, i.e., $\{a_{q,n}\}_{q=1}^Q$, $a_{q,n} \in \mathbb{F}_2^B$, for some $B \in \mathbb{N}^+$. The *computation load* of the system is defined as the total number of map functions computed over K nodes, normalized by the total number of independent files, that is,

$$r \triangleq \frac{\sum_{k=1}^K |\mathcal{M}_k|}{N}. \quad (2)$$

Shuffle phase and the interference channel: In the Shuffle phase, distributed nodes exchange the intermediate values over a wireless interference channel, in order to compute Reduce functions. Let \mathcal{W}_k denote the indices of Reduce functions computed at Node k , $k \in [1 : K]$. Node k needs the set of intermediate values $\{a_{q,n} : q \in \mathcal{W}_k, n \in [1 : N]\}$. Note that after the Map phase, Node k already has

$$\mathcal{P}_k \triangleq \{a_{q,n} : q \in [1 : Q], n \in \mathcal{M}_k\} \quad (3)$$

for $k \in [1 : K]$. Therefore, it only requires

$$\mathcal{G}_k \triangleq \{a_{q,n} : q \in \mathcal{W}_k, n \in [1 : N], n \notin \mathcal{M}_k\}.$$

The communication over this interference channel at time t is modeled as

$$y_k(t) = \sum_{i=1}^K h_{k,i} x_i(t) + z_k(t), \quad k \in [1 : K], \quad (4)$$

where $y_k(t)$ denotes the received signal at Node k at time t ; $x_k(t)$ is the transmitted signal of Node k at time t subject to a power constraint $\mathbb{E}[|x_k(t)|^2] \leq P$, and $z_k(t) \sim \mathcal{CN}(0, 1)$ denotes the additive white Gaussian noise (AWGN). $h_{k,i} \in \mathbb{C}$ denotes the coefficient of the channel from Transmitter i to Receiver k , assumed to be fixed and known by all the nodes¹, for all $k, i \in [1 : K]$. We assume that all submatrices of the channel matrix consisting of all the channel coefficients are full rank. We also assume that the absolute value of each channel coefficient is bounded between a finite maximum value and a nonzero minimum value. We consider the *full-duplex* communication, where each node can receive and transmit signal at the same time.

In this phase, each node first employs a random Gaussian coding scheme (cf. [48]) to encode each of its generated intermediate values $a_{q,n} \in \mathbb{F}_2^B$ into a *coded packet* $\tilde{a}_{q,n} \in \mathbb{C}^\tau$, corresponding to τ channel uses (called a block), for some integer τ such that $B = \tau \log P + o(\tau \log P)$. The rate is $B/\tau \approx \log P$ bits/channel use, equivalent to one degree of freedom (DoF). The transmission of all the required coded packets takes place over a total of T blocks. In block ℓ , a subset of the required packets, denoted by \mathcal{D}_ℓ , is delivered to a subset of receivers whose indices are denoted by \mathcal{R}_ℓ , with each packet intended for one of the receivers, i.e., $|\mathcal{D}_\ell| = |\mathcal{R}_\ell|$, for $\mathcal{D}_\ell \cap \mathcal{D}_{\ell'} = \emptyset, \forall \ell, \ell' \in [1 : T], \ell \neq \ell'$.

Specifically, in block ℓ we consider the *one-shot linear scheme*. The signal transmitted by Node i , denoted by $\mathbf{x}_i[\ell] \in \mathbb{C}^\tau$, is a linear combination of the coded packets $\{\tilde{a}_{q,n} : \tilde{a}_{q,n} \in \mathcal{D}_\ell, n \in \mathcal{M}_i\}$ generated by Node i , that is,

$$\mathbf{x}_i[\ell] = \sum_{(q,n): \tilde{a}_{q,n} \in \mathcal{D}_\ell, n \in \mathcal{M}_i} \beta_{i,q,n} \tilde{a}_{q,n}, \quad (5)$$

where $\beta_{i,q,n}$ is the beamforming coefficient, for $\ell \in [1 : T]$ and $i \in [1 : K]$. Then, the received signal of Node k at block ℓ takes the following form

$$\mathbf{y}_k[\ell] = \sum_{i=1}^K h_{k,i} \mathbf{x}_i[\ell] + \mathbf{z}_k[\ell], \quad \ell \in [1 : T], \quad (6)$$

where $\mathbf{z}_k[\ell] \in \mathbb{C}^\tau$ denotes the noise vector at Receiver k (Node k) in block ℓ , for $k \in [1 : K]$. In terms of decoding, Node k utilizes its side information (the generated coded packets), i.e.,

$$\tilde{\mathcal{P}}_k \triangleq \{\tilde{a}_{q,n} : a_{q,n} \in \mathcal{P}_k\}$$

¹Although we assume that the channel coefficients are fixed, our result also holds for the setting with time varying channel coefficients. An example is provided in Section IV-C. For simplicity of presentation, we will derive our results for fixed channel coefficients.

(see (3)), to subtract the interference from $\mathbf{y}_k[\ell]$ using a linear function, denoted as,

$$\mathcal{L}_{k,\ell}(\mathbf{y}_k[\ell], \tilde{\mathcal{P}}_k). \quad (7)$$

The communication in block ℓ , $\ell \in [1 : T]$, is successful if there exist linear operations as in (5) and (7) to obtain

$$\mathcal{L}_{k,\ell}(\mathbf{y}_k[\ell], \tilde{\mathcal{P}}_k) = \tilde{\mathbf{a}}_{q,n} + \mathbf{z}_k[\ell] \quad (8)$$

for $\forall k \in \mathcal{R}_\ell$ and $\tilde{\mathbf{a}}_{q,n} \in \mathcal{D}_\ell \cap \{\tilde{\mathbf{a}}_{q',n'} : a_{q',n'} \in \mathcal{G}_k\}$. Because the channel in (8) is a point to point AWGN channel and its capacity is roughly $\log P$ bits/channel use, $a_{q,n}$ can be decoded with vanishing error probability as B increases [48]. Note that, in our setting we use the random Gaussian coding scheme to encode each of the intermediate values. In terms of decoding, the maximum likelihood (ML) decoding can be used. However, the complexity of the Gaussian coding and ML decoding is very high. To reduce the complexity, one could use the low-complexity encoding/decoding method, e.g., lattice-based encoding and decoding [49].

Reduce phase: Node k computes the Reduce function $b_q, q \in \mathcal{W}_k$, as a function of $(a_{q,1}, a_{q,2}, \dots, a_{q,N})$. In this work we consider a symmetric job assignment, that is, each node has Q/K number of output functions to compute, for $\frac{Q}{K} \in \mathbb{N}$. Specifically,

$$|\mathcal{W}_1| = |\mathcal{W}_2| = \dots = |\mathcal{W}_K| = Q/K, \quad (9)$$

and $\mathcal{W}_k \cap \mathcal{W}_j = \emptyset$ for any $k, j \in [1 : K]$, $k \neq j$.

We define the *communication load* of this wireless distributed computing system as

$$L \triangleq \frac{T}{NQ}$$

which denotes the normalized communication blocks used in the Shuffle phase. In our setting, the computation load and communication load pair (r, L) is said to be achievable if there exists a wireless MapReduce scheme consisting of Map, Shuffle and Reduce phases under the above one-shot linear assumptions, in which all the intermediate values can be decoded with vanishing error probability as B increases. We also define the *computation-communication function* of this wireless distributed computing system, as

$$L^*(r) \triangleq \inf\{L : (r, L) \text{ is feasible}\}.$$

III. MAIN RESULTS

This section provides the main results of this work for the wireless distributed computing system defined in Section II. The converse and achievability proofs are presented in Sections V and VI, respectively.

Theorem 1. *For the wireless distributed computing system defined in Section II, with the assumption of one-shot linear schemes and a sufficiently large N , the computation-communication function, $L^*(r)$, is characterized as*

$$L^*(r) = \frac{1 - \frac{r}{K}}{\min\{K, 2r\}}, \quad r \in \{1, 2, \dots, K\}. \quad (10)$$

Theorem 1 provides a fundamental tradeoff between the communication load L and the computation load r for the wireless distributed computing system defined in Section II. The achievability of Theorem 1 is based on a one-shot linear scheme that utilizes the methods of zero-forcing and interference cancellation with side information. The proposed scheme turns out to be optimal for integer r . For non-integer r , our converse proof shows that $L^*(r) \geq \frac{1 - \frac{r}{K}}{\min\{K, 2r\}}$; our achievability results can be extended using time-sharing such that the line connecting the adjacent integer points $(r, L^*(r))$ and $(r+1, L^*(r+1))$ is achievable, for any $1 \leq r \leq K-1$, as plotted in Fig. 2. When $\frac{K}{2} \leq r \leq K$, the expression in (10) is linear in r . Therefore, the expression (10) gives the optimal computation-communication function for all integer r for $1 \leq r \leq K$, and all real r , for $\frac{K}{2} \leq r \leq K$.

From the achievability proof in Section VI, Theorem 1 holds when N is a multiple of some N_0 that depends on (K, r) , or when N is sufficiently large for fixed K, Q, r . Note that, in practice, the dataset to be processed is typically big (big data) for the distributed computing systems. The whole dataset can be partitioned into N files and N can be much larger than the number of servers K . Moreover, Q is often a small multiple of K [9]. We also assume that r is fixed to ensure bounded computation load.

Since the Reduce functions indexed by \mathcal{W}_k need QN/K intermediate values as inputs and $Q \cdot |\mathcal{M}_k|/K$ of them have been cached at Node k , it implies that the total number of intermediate values required by Node k is $\frac{Q}{K}(N - |\mathcal{M}_k|)$. Therefore, the total number of intermediate values required to be delivered in the Shuffle phase, denoted as C_{total} , can be expressed as

$$C_{\text{total}} = \sum_{k=1}^K \frac{Q}{K} (N - |\mathcal{M}_k|) = QN \left(1 - \frac{r}{K}\right). \quad (11)$$

Remark 1 (Uncoded TDMA scheme). *In the uncoded TDMA scheme, only one node delivers one (uncoded) intermediate value at each transmission block. From (11), the communication load L is expressed as*

$$L^{\text{Uncoded-TDMA}}(r) = 1 - \frac{r}{K}, \quad r \in \{1, 2, \dots, K\}. \quad (12)$$

Remark 2 (Coded TDMA scheme). *In the coded TDMA scheme, one node delivers one coded intermediate value at each transmission block. From the result in [10], the communication load L of this coded TDMA scheme is*

$$L^{\text{Coded-TDMA}}(r) = \frac{1}{r} \cdot \left(1 - \frac{r}{K}\right), \quad r \in \{1, 2, \dots, K\}. \quad (13)$$

Remark 3. *The significant improvement of our scheme compared to uncoded and coded TDMA schemes is depicted in Fig. 2. Note that, the communication load of the proposed one-shot linear scheme is $\left(1 - \frac{1}{\min\{K, 2r\}}\right) \times 100\%$ lower than that of uncoded TDMA. Furthermore, the communication load of the proposed one-shot linear scheme is $\left(1 - \frac{r}{\min\{K, 2r\}}\right) \times 100\%$ lower than that of coded TDMA.*

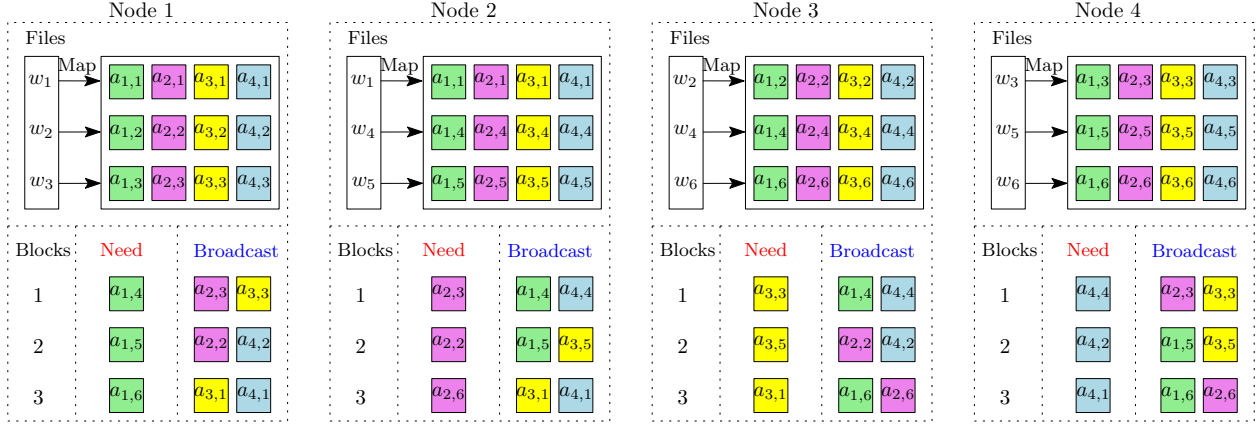


Fig. 3. An example of wireless distributed computing with $K = Q = 4$, $N = 6$ and $r = 2$.

IV. EXAMPLES

In the introduction, we saw an example of one-shot linear scheme in the Shuffle phase with $K = Q = N = 3$ and $r = 2$. The scheme exploits the side information for interference cancellation. In this section, we use two examples to illustrate the proposed one-shot linear schemes in the Shuffle phase. In the first example with $r \geq K/2$, the scheme exploits side information cancellation and zero-forcing, while in the second example with $r < K/2$, the scheme uses side information cancellation and *partial* zero-forcing. We introduce important notations including virtual transmitters, beamforming vectors and channel coefficient vectors for the virtual transmitters. These notations will be used in our converse and achievability proofs in Sections V and VI.

A. The example of $K = Q = 4$, $N = 6$ and $r = 2$ ($r \geq K/2$)

Let us consider the case of ($K = Q = 4$, $N = 6$, $r = 2$). As shown in Fig. 3, we assign three files for each node such that $\mathcal{M}_1 = \{1, 2, 3\}$, $\mathcal{M}_2 = \{1, 4, 5\}$, $\mathcal{M}_3 = \{2, 4, 6\}$ and $\mathcal{M}_4 = \{3, 5, 6\}$. Without loss of generality we consider the case where the k -th Reduce function is assigned to Node k , for $k = 1, 2, 3, 4$.

In the Map phase, each node generates a set of intermediate values. Then, each intermediate value (e.g., $a_{1,4}$) is mapped into a coded packet (e.g., $\tilde{\mathbf{a}}_{1,4}$). Let $\mathcal{S}_n = \{i : n \in \mathcal{M}_i\}$ represent the indices of all the nodes having file w_n , $n \in [1 : N]$. The transmitters indexed by \mathcal{S}_n are defined to be a *virtual transmitter* (i.e., virtual Transmitter \mathcal{S}_n). We use

$$\mathbf{h}_{k, \mathcal{S}_n} \triangleq [h_{k, \mathcal{S}_n^1}, h_{k, \mathcal{S}_n^2}, \dots, h_{k, \mathcal{S}_n^{|\mathcal{S}_n|}}]^T \quad (14)$$

to denote the *channel vector* from virtual Transmitter \mathcal{S}_n to Receiver k , where \mathcal{S}_n^j denotes the j th element of set \mathcal{S}_n . Let

$$\mathbf{v}_{\mathcal{S}_n, q, n} \triangleq [\beta_{\mathcal{S}_n^1, q, n}, \beta_{\mathcal{S}_n^2, q, n}, \dots, \beta_{\mathcal{S}_n^{|\mathcal{S}_n|}, q, n}]^T \quad (15)$$

denote the *beamforming vector* for coded packet $\tilde{\mathbf{a}}_{q, n}$ that is transmitted from virtual Transmitter \mathcal{S}_n , where $\beta_{\mathcal{S}_n^j, q, n}$ is the beamforming coefficient of node \mathcal{S}_n^j for the coded packet $\tilde{\mathbf{a}}_{q, n}$. For example, for virtual Transmitter $\mathcal{S}_n = \{2, 3\}$ and Receiver 1, we have the channel vector $\mathbf{h}_{1, \{2, 3\}}^T = [h_{1, 2}, h_{1, 3}]$. And

$\mathbf{v}_{\{2, 3\}, 1, 4}^T = [\beta_{2, 1, 4}, \beta_{3, 1, 4}]$ is the beamforming vector for the coded packet $\tilde{\mathbf{a}}_{1, 4}$.

In order to compute the first Reduce function, Node 1 needs the intermediate values $(a_{1, 1}, a_{1, 2}, a_{1, 3}, a_{1, 4}, a_{1, 5}, a_{1, 6})$. Since three intermediate values $(a_{1, 1}, a_{1, 2}, a_{1, 3})$ are already available at Node 1 after the Map phase, Node 1 only needs to obtain $(a_{1, 4}, a_{1, 5}, a_{1, 6})$ in the Shuffle phase. Similarly, $(a_{2, 2}, a_{2, 3}, a_{2, 6})$, $(a_{3, 1}, a_{3, 3}, a_{3, 5})$ and $(a_{4, 1}, a_{4, 2}, a_{4, 4})$ need to be delivered to Nodes 2, 3 and 4, respectively (see Fig. 3). We will show that in each transmission block, $K = 4$ intermediate values are transmitted to K receivers without interference, and three blocks ($T = 3$) are sufficient for delivering all the required intermediate values.

In the first block, four required intermediate values $a_{1, 4}, a_{2, 3}, a_{3, 3}$ and $a_{4, 4}$ are transmitted to Nodes 1, 2, 3 and 4, respectively. Specifically, the transmitted signals of four nodes are given as

$$\mathbf{x}_1[1] = \beta_{1, 2, 3} \tilde{\mathbf{a}}_{2, 3} + \beta_{1, 3, 3} \tilde{\mathbf{a}}_{3, 3}, \quad (16)$$

$$\mathbf{x}_2[1] = \beta_{2, 1, 4} \tilde{\mathbf{a}}_{1, 4} + \beta_{2, 4, 4} \tilde{\mathbf{a}}_{4, 4}, \quad (17)$$

$$\mathbf{x}_3[1] = \beta_{3, 1, 4} \tilde{\mathbf{a}}_{1, 4} + \beta_{3, 4, 4} \tilde{\mathbf{a}}_{4, 4}, \quad (18)$$

$$\mathbf{x}_4[1] = \beta_{4, 2, 3} \tilde{\mathbf{a}}_{2, 3} + \beta_{4, 3, 3} \tilde{\mathbf{a}}_{3, 3}, \quad (19)$$

where the beamforming coefficients $\{\beta_{i, q, n}\}$ are designed such that

$$\mathbf{v}_{\{2, 3\}, 4, 4} \in \text{Null}(\mathbf{h}_{1, \{2, 3\}}), \quad \mathbf{v}_{\{1, 4\}, 3, 3} \in \text{Null}(\mathbf{h}_{2, \{1, 4\}}), \quad (20)$$

$$\mathbf{v}_{\{1, 4\}, 2, 3} \in \text{Null}(\mathbf{h}_{3, \{1, 4\}}), \quad \mathbf{v}_{\{2, 3\}, 1, 4} \in \text{Null}(\mathbf{h}_{4, \{2, 3\}}), \quad (21)$$

where $\text{Null}(\mathbf{e})$ denotes the null space of the vector \mathbf{e} .

At the receiver side, Node 1 receives the following signal

$$\begin{aligned} \mathbf{y}_1[1] &= \sum_{i=1}^K h_{1, i} \mathbf{x}_i[1] + \mathbf{z}_1[1] \\ &= \underbrace{\mathbf{h}_{1, \{2, 3\}}^T \mathbf{v}_{\{2, 3\}, 1, 4} \tilde{\mathbf{a}}_{1, 4}}_{\text{desired intermediate value}} + \underbrace{\mathbf{h}_{1, \{1, 4\}}^T \mathbf{v}_{\{1, 4\}, 2, 3} \tilde{\mathbf{a}}_{2, 3}}_{\text{side information}} + \\ &\quad \underbrace{\mathbf{h}_{1, \{1, 4\}}^T \mathbf{v}_{\{1, 4\}, 3, 3} \tilde{\mathbf{a}}_{3, 3}}_{\text{side information}} + \underbrace{\mathbf{h}_{1, \{2, 3\}}^T \mathbf{v}_{\{2, 3\}, 4, 4} \tilde{\mathbf{a}}_{4, 4}}_{\text{interference}} + \mathbf{z}_1[1]. \end{aligned}$$

In the above expansion of $\mathbf{y}_1[1]$, the second and the third terms can be removed by using side information $\tilde{\mathbf{a}}_{2, 3}$ and $\tilde{\mathbf{a}}_{3, 3}$ at

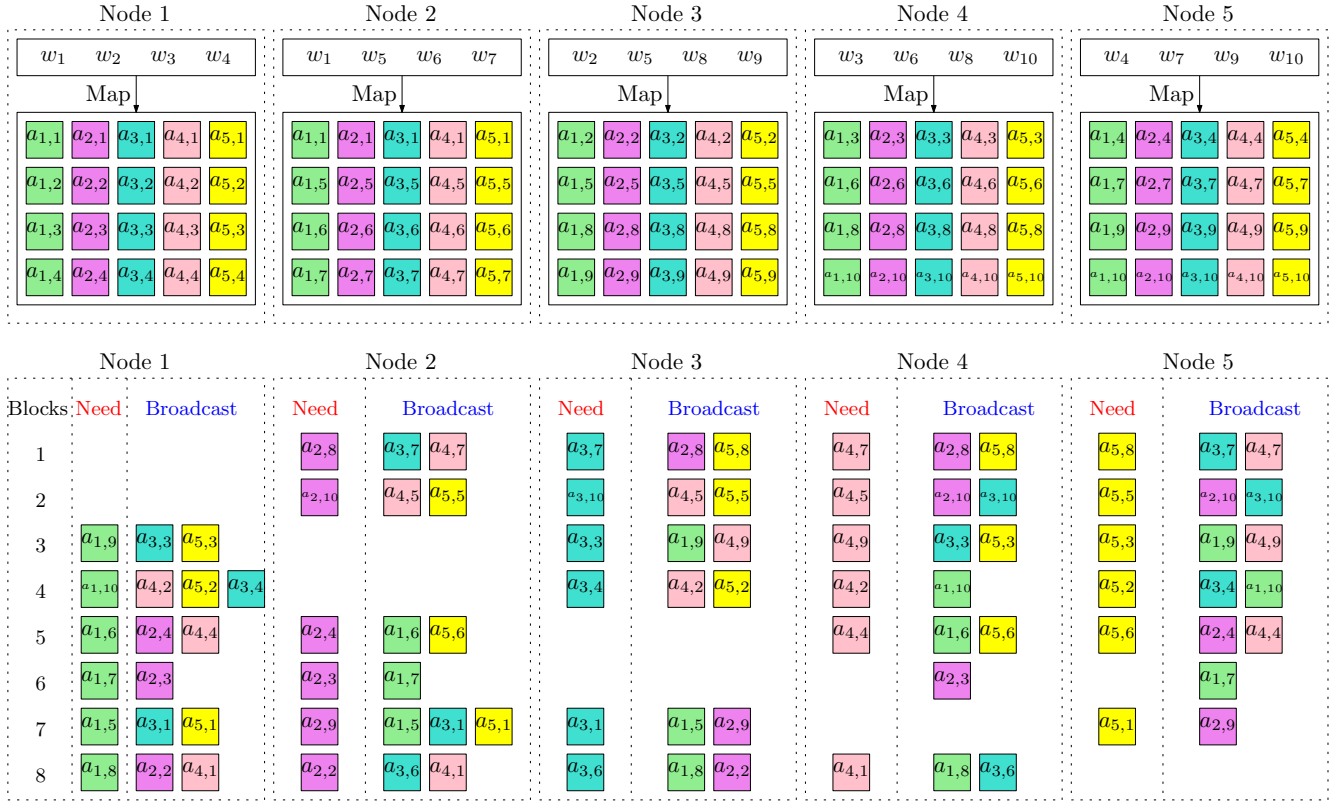


Fig. 4. An example of wireless distributed computing with $K = Q = 5$, $N = 10$ and $r = 2$.

Node 1, while the fourth term can be canceled out due to our design in (20). In our setting, since we consider the full rank assumption for the channels, once a beamforming vector is orthogonal to the channel vector associated with the interference, e.g., $\mathbf{v}_{\{2,3\},1,4} \in \text{Null}(\mathbf{h}_{4,\{2,3\}})$, then this beamforming vector is not orthogonal to the channel vector associated with the desired intermediate value, e.g., $\mathbf{v}_{\{2,3\},1,4} \notin \text{Null}(\mathbf{h}_{1,\{2,3\}})$.

Therefore, Node 1 can decode the desired intermediate value $a_{1,4}$. Similarly, Nodes 2, 3 and 4 can decode the desired $a_{2,3}$, $a_{3,3}$ and $a_{4,4}$, respectively.

By applying the same methods, in the second block the desired intermediate values $a_{1,5}$, $a_{2,2}$, $a_{3,5}$ and $a_{4,2}$ can be delivered to Nodes 1, 2, 3 and 4, respectively, while in the third block, the desired intermediate values $a_{1,6}$, $a_{2,6}$, $a_{3,1}$ and $a_{4,1}$ can be delivered to Nodes 1, 2, 3 and 4, respectively.

Therefore, with the methods of side information cancellation and zero-forcing, each node can obtain the desired intermediate values after using three blocks ($T = 3$) in the Shuffle phase.

B. The example of $K = Q = 5$, $N = 10$ and $r = 2$ ($r < K/2$)

Let us consider the example of $K = Q = 5$, $r = 2$ and $N = \binom{K}{r} = 10$ (see Fig. 4). This case is different from the case mentioned in Section IV-A. In the previous case with $r \geq K/2$, K intermediate values are delivered without interference in each transmission block. However, in this case with $r < K/2$, it is impossible to deliver K intermediate values without interference in each transmission block. Instead, $2r$ intermediate values are delivered in each

transmission block, by using partial zero-forcing and side information cancellation.

In this example, given 10 independent files, we assign 4 independent files for each node such that $\mathcal{M}_1 = \{1, 2, 3, 4\}$, $\mathcal{M}_2 = \{1, 5, 6, 7\}$, $\mathcal{M}_3 = \{2, 5, 8, 9\}$, $\mathcal{M}_4 = \{3, 6, 8, 10\}$, and $\mathcal{M}_5 = \{4, 7, 9, 10\}$, as shown in Fig. 4. Again, without loss of generality we consider the case where the k -th Reduce function is assigned to Node k , for $k \in [1 : K]$.

After the Map phase, each node generates a set of intermediate values. In order to complete the computation of each Reduce function, all the nodes need to exchange a subset of intermediate values in the Shuffle phase. For instance, in order to compute the first Reduce function at Node 1, the following intermediate values

$$(a_{1,5}, a_{1,6}, a_{1,7}, a_{1,8}, a_{1,9}, a_{1,10})$$

need to be delivered to Node 1 in the Shuffle phase.

We select $2r = 4$ nodes to exchange the intermediate values at each transmission block. Let us focus on the first block. As shown in Fig. 4, in this block, we select only four nodes, i.e., Nodes 2, 3, 4 and 5, to exchange four intermediate values $(a_{2,8}, a_{3,7}, a_{4,7}, a_{5,8})$. Note that, $a_{2,8}$, $a_{3,7}$, $a_{4,7}$ and $a_{5,8}$ are intended for Nodes 2, 3, 4 and 5, respectively. The beamforming coefficients $\{\beta_{i,q,n}\}$ are designed such that

$$\mathbf{v}_{\{3,4\},5,8} \in \text{Null}(\mathbf{h}_{2,\{3,4\}}), \mathbf{v}_{\{2,5\},4,7} \in \text{Null}(\mathbf{h}_{3,\{2,5\}}), \quad (22)$$

$$\mathbf{v}_{\{2,5\},3,7} \in \text{Null}(\mathbf{h}_{4,\{2,5\}}), \mathbf{v}_{\{3,4\},2,8} \in \text{Null}(\mathbf{h}_{5,\{3,4\}}). \quad (23)$$

At the receiver side, Node 2 receives the following signal

$$\mathbf{y}_2[1] = \underbrace{\mathbf{h}_{2,\{3,4\}}^T \mathbf{v}_{\{3,4\},2,8} \tilde{\mathbf{a}}_{2,8}}_{\text{desired intermediate value}} + \underbrace{\mathbf{h}_{2,\{2,5\}}^T \mathbf{v}_{\{2,5\},3,7} \tilde{\mathbf{a}}_{3,7}}_{\text{side information}} + \underbrace{\mathbf{h}_{2,\{2,5\}}^T \mathbf{v}_{\{2,5\},4,7} \tilde{\mathbf{a}}_{4,7}}_{\text{side information}} + \underbrace{\mathbf{h}_{2,\{3,4\}}^T \mathbf{v}_{\{3,4\},5,8} \tilde{\mathbf{a}}_{5,8}}_{\text{interference}} + \mathbf{z}_2[1].$$

Due to the side information cancellation and zero-forcing, $a_{2,8}$ can be decoded at Node 2 without interference. In a similar way, $a_{3,7}$, $a_{4,7}$ and $a_{5,8}$ can be decoded at Nodes 3, 4 and 5, respectively.

With the same argument, in each of the other blocks (see Fig. 4), only four nodes are selected to receive four intermediate values. In this way, at each block all the interference can be either canceled with side information or zero-forced at the selected nodes (*partial zero-forcing*). The only exception is that block 6 transmits only two intermediate values, because the total number of transmitted intermediate values 30 is not a multiple of $2r = 4$. Therefore, all the required intermediate values can be delivered with $T = 8$ transmission blocks in the Shuffle phase. The communication load is $L = \frac{T}{NQ} = 0.16$. In fact, it will be seen from the converse proof in Section V that, for any feasible scheme, one needs to transmit a total of $C_{\text{total}} = 30$ intermediate values in the shuffle phase, and at most $2r = 4$ intermediate values can be delivered in each block. Thus the number of transmission blocks is $T \geq \lceil \frac{30}{4} \rceil = 8$. Therefore, the scheme of this example is optimal for $N = 10$. Note that if there are $N = 20$ files, then it is possible to extend the scheme in this example and obtain a lower communication load $L = 0.15$, matching the result of Theorem 1. The details are shown in Section VI-B.

C. Discussion on time varying channels

Note that our achievability and converse also work for the setting with varying channel gains. One simply needs to replace the channel vector and the beamforming vector with the channel matrix and the beamforming matrix, respectively. In the following we explain this point by focusing on the example in Section IV-A.

For this example with varying channel gains, the received signal of Node 1 at block 1 takes the following form

$$\begin{aligned} \mathbf{y}_1[1] &= \sum_{i=1}^K \mathbf{H}_{1,i}[1] \mathbf{x}_i[1] + \mathbf{z}_1[1] \\ &= \underbrace{\mathbf{H}_{1,\{2,3\}}^T[1] \mathbf{V}_{\{2,3\},1,4} \tilde{\mathbf{a}}_{1,4}}_{\text{desired intermediate value}} + \underbrace{\mathbf{H}_{1,\{1,4\}}^T[1] \mathbf{V}_{\{1,4\},2,3} \tilde{\mathbf{a}}_{2,3}}_{\text{side information}} \\ &\quad + \underbrace{\mathbf{H}_{1,\{1,4\}}^T[1] \mathbf{V}_{\{1,4\},3,3} \tilde{\mathbf{a}}_{3,3}}_{\text{side information}} + \underbrace{\mathbf{H}_{1,\{2,3\}}^T[1] \mathbf{V}_{\{2,3\},4,4} \tilde{\mathbf{a}}_{4,4}}_{\text{interference}} \\ &\quad + \mathbf{z}_1[1] \end{aligned} \quad (24)$$

where

$$\mathbf{H}_{k,i}[\ell] = \begin{pmatrix} h_{k,i}^{(1)}[\ell] & 0 & \cdots & 0 \\ 0 & h_{k,i}^{(2)}[\ell] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{k,i}^{(\tau)}[\ell] \end{pmatrix}$$

and $h_{k,i}^{(n)}[\ell]$, $n \in [1, \tau]$ denotes the channel gain of the n -th channel use in block ℓ , for Transmitter i and Receiver k . In the above expression of $\mathbf{y}_1[1]$, we have the following notations

$$\mathbf{H}_{1,\{2,3\}}^T[1] = \begin{pmatrix} h_{1,2}^{(1)}[1] & h_{1,3}^{(1)}[1] & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & h_{1,2}^{(2)}[1] & h_{1,3}^{(2)}[1] & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & h_{1,2}^{(\tau)}[1] & h_{1,3}^{(\tau)}[1] \end{pmatrix}$$

and

$$\mathbf{V}_{\{2,3\},4,4} = \begin{pmatrix} \beta_{2,4,4}^{(1)} & 0 & \cdots & 0 \\ \beta_{3,4,4}^{(1)} & 0 & \cdots & 0 \\ 0 & \beta_{2,4,4}^{(2)} & \cdots & 0 \\ 0 & \beta_{3,4,4}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_{2,4,4}^{(\tau)} \\ 0 & 0 & \cdots & \beta_{3,4,4}^{(\tau)} \end{pmatrix}$$

where $\beta_{i,q,n}^{(n)}$ denotes the beamforming coefficient of the n -th channel use. By designing the beamforming coefficients $\{\beta_{i,q,n}^{(n)}\}$ such that $\mathbf{H}_{1,\{2,3\}}^T[1] \mathbf{V}_{\{2,3\},4,4} = \mathbf{0}$, the interference can be removed.

With this approach, one can conclude that the proposed general scheme and the converse argument also hold for the setting with time varying channel gains. For simplicity of presentation, we omit the details and just assume fixed channel gains in the remaining sections.

V. CONVERSE PROOF FOR THEOREM 1

In this section we show the converse of Theorem 1. In fact, we show the following lower bound of the communication load:

$$L \geq \frac{1 - \frac{r}{K}}{\min\{K, 2r\}}, \quad r \in \mathbb{R}, 1 \leq r \leq K. \quad (25)$$

We first bound the maximum number of coded packets (of the corresponding intermediate values) that can be transmitted simultaneously in block ℓ , denoted by $|\mathcal{D}_\ell|$, for $\ell \in [1 : T]$. We take a similar approach as in [32], [50]. Recall that in block ℓ we have coded packets \mathcal{D}_ℓ to be transmitted to the receivers indexed by \mathcal{R}_ℓ , with $|\mathcal{R}_\ell| = |\mathcal{D}_\ell|$.

In block ℓ , the transmitted signal from Node i takes the form as in (5). Then, the received signal of Node k , $k \in \mathcal{R}_\ell$,

takes the following form

$$\begin{aligned}
\mathbf{y}_k[\ell] &= \sum_{i=1}^K h_{k,i} \mathbf{x}_i[\ell] + \mathbf{z}_k[\ell] \\
&= \sum_{i=1}^K h_{k,i} \sum_{(q,n): \tilde{\mathbf{a}}_{q,n} \in \mathcal{D}_\ell, n \in \mathcal{M}_i} \beta_{i,q,n} \tilde{\mathbf{a}}_{q,n} + \mathbf{z}_k[\ell] \\
&= \sum_{(q,n): \tilde{\mathbf{a}}_{q,n} \in \mathcal{D}_\ell} \sum_{i \in \mathcal{S}_n} h_{k,i} \beta_{i,q,n} \tilde{\mathbf{a}}_{q,n} + \mathbf{z}_k[\ell] \\
&= \sum_{(q,n): \tilde{\mathbf{a}}_{q,n} \in \mathcal{D}_\ell} \mathbf{h}_{k,\mathcal{S}_n}^T \mathbf{v}_{\mathcal{S}_n,q,n} \tilde{\mathbf{a}}_{q,n} + \mathbf{z}_k[\ell] \quad (26)
\end{aligned}$$

where the channel vector $\mathbf{h}_{k,\mathcal{S}_n}$, the beamforming vector $\mathbf{v}_{\mathcal{S}_n,q,n}$ are defined in (14) and (15), respectively. From (26), we can conclude that the channel of packet transmission can be transformed into a MISO interference channel. The MISO interference channel has $|\mathcal{R}_\ell|$ single-antenna receivers and $|\mathcal{D}_\ell|$ virtual transmitters, where virtual Transmitter \mathcal{S}_n has $|\mathcal{S}_n|$ antennas, for $n \in [1 : N]$.

In what follows let us first consider the case where each file w_n is stored at $|\mathcal{S}_n| = r$ nodes (symmetric file replications), for $n = 1, 2, \dots, N$ and integer $r \in \{1, 2, \dots, K\}$. For the other case where different files may be replicated different times (asymmetric file replications), the proof is provided in Section V-A.

Let us focus on the transmission of one coded packet $\tilde{\mathbf{a}}_{q,n}$ associated with the intermediate value $a_{q,n}$, for a given pair (q, n) . Assume it is transmitted in block ℓ , and is intended for Receiver k , for $\ell \in [1 : T]$ and $k \in [1 : K]$. Based on a MISO interference channel, a beamforming vector $\mathbf{v}_{\mathcal{S}_n,q,n} \in \mathbb{C}^{|\mathcal{S}_n|}$ is used by virtual Transmitter \mathcal{S}_n to transmit the corresponding coded packet $\tilde{\mathbf{a}}_{q,n}$. At the receiver side, let $\mathcal{J}_n = \mathcal{R}_\ell \setminus \{k\} \cup \mathcal{S}_n$ denote the indices of receivers excluding the intended Receiver k and the transmitters indexed by \mathcal{S}_n , where the packet $\tilde{\mathbf{a}}_{q,n}$ should be zero forced. Then

$$|\mathcal{J}_n| \geq |\mathcal{R}_\ell| - |\mathcal{S}_n| - 1, \quad (27)$$

and the inequality holds with equality when \mathcal{S}_n is a subset of \mathcal{R}_ℓ . Therefore, for $\mathbf{H} \in \mathbb{C}^{|\mathcal{J}_n| \times |\mathcal{S}_n|}$ denoting the channel from virtual Transmitter \mathcal{S}_n to the receivers indexed by \mathcal{J}_n , we should have

$$\mathbf{H} \mathbf{v}_{\mathcal{S}_n,q,n} = 0 \quad (28)$$

in order to remove the interference associated with $\tilde{\mathbf{a}}_{q,n}$ at the receivers indexed by \mathcal{J}_n . Given that \mathbf{H} is full rank and $\mathbf{v}_{\mathcal{S}_n,q,n}$ should be nonzero, a necessary condition for the existence of the solution to (28) becomes

$$|\mathcal{J}_n| \leq |\mathcal{S}_n| - 1, \quad (29)$$

which combined with (27) gives

$$|\mathcal{D}_\ell| = |\mathcal{R}_\ell| \leq 2|\mathcal{S}_n| = 2r. \quad (30)$$

Furthermore, it is obvious that $|\mathcal{D}_\ell| \leq K$. Then, we can conclude that, at block ℓ the maximum number of transmitted coded packets satisfies

$$|\mathcal{D}_\ell| \leq \min\{K, 2r\}, \quad \forall \ell \in [1 : T]. \quad (31)$$

Since in one block we can transmit $|\mathcal{D}_\ell|$ coded packets, combining (11) and (31), the number of blocks used to transmit all the intermediate values should be bounded by

$$T \geq \left\lceil \frac{C_{\text{total}}}{|\mathcal{D}_\ell|} \right\rceil \geq \left\lceil \frac{NQ(1 - \frac{r}{K})}{\min\{K, 2r\}} \right\rceil. \quad (32)$$

Therefore, communication load L should be bounded by

$$L = \frac{T}{NQ} \geq \frac{\left\lceil \frac{NQ(1 - \frac{r}{K})}{\min\{K, 2r\}} \right\rceil}{NQ} \geq \frac{1 - \frac{r}{K}}{\min\{K, 2r\}}. \quad (33)$$

A. The case with asymmetric file replications

Now, let us consider the case where different files may be replicated different times (asymmetric file replications), given an average computation load $r = \frac{\sum_{k=1}^K |\mathcal{M}_k|}{N}$. Note that for this case the value r does not need to be an integer. Let

$$\theta_n \triangleq |\mathcal{S}_n|$$

denote the number of times that File n is replicated across the distributed nodes, $n \in [1 : N]$. By our definitions of θ_n and r , we have,

$$\frac{\sum_{n=1}^N \theta_n}{N} = r.$$

Without loss of generality, we consider the case with

$$\theta_1 \leq \theta_2 \leq \dots \leq \theta_N.$$

Let C_n denote the total number of intermediate values generated by File n and required to be delivered in the Shuffle phase, $n \in [1 : N]$. Then, we have

$$C_n = \frac{(K - \theta_n)Q}{K}. \quad (34)$$

This is because, for each node that does not have File n , it needs Q/K intermediate values generated by File n to complete the computation of its output functions; and the total number of nodes that do not have File n is $(K - \theta_n)$. It is easy to see that

$$\sum_{n=1}^N C_n = C_{\text{total}}, \quad (35)$$

where C_{total} is defined in (11). Let us use the following notations for the ease of our argument:

$$\sigma_n \triangleq \frac{C_n}{\min\{2\theta_n, K\}} = \frac{(K - \theta_n)}{\min\{2\theta_n, K\}} \cdot \frac{Q}{K}, \quad (36)$$

and

$$\sigma_{\text{sum}} \triangleq \sum_{n=1}^N \sigma_n. \quad (37)$$

In the rest of the proof, we show that σ_{sum} is a lower bound on the number of required blocks T . Thus the converse of Theorem 1 follows from bounding σ_{sum} .

In each block ℓ , packets corresponding to $|\mathcal{R}_\ell| = |\mathcal{D}_\ell|$ intermediate values are transmitted, for $\ell \in [1 : T]$. Let $r_{\ell,j}$ denote the total number of nodes that generate (after the Map phase) the j th intermediate value out of these $|\mathcal{D}_\ell|$ intermediate

values. It implies that $r_{\ell,j} \in \{\theta_1, \dots, \theta_N\}$, for $j \in [1 : |\mathcal{D}_\ell|]$. For example, in block ℓ , if we transmit intermediate values corresponding to Files 1, 1, 2 and 3, then we have

$$(r_{\ell,1}, r_{\ell,2}, r_{\ell,3}, r_{\ell,4}) = (\theta_1, \theta_1, \theta_2, \theta_3).$$

Without loss of generality let

$$r_{\ell,1} \leq r_{\ell,2} \leq \dots \leq r_{\ell,|\mathcal{D}_\ell|}.$$

Let $C_{\ell,n}$ denote the total number of intermediate values generated by File n and delivered in block ℓ . By the definitions of $C_{\ell,n}$ and C_n , we have

$$\sum_{\ell=1}^T C_{\ell,n} = C_n. \quad (38)$$

Moreover,

$$|\mathcal{D}_\ell| = \sum_{n=1}^N C_{\ell,n}. \quad (39)$$

Thus

$$\sum_{\ell=1}^T |\mathcal{D}_\ell| = \sum_{\ell=1}^T \sum_{n=1}^N C_{\ell,n} \quad (40)$$

$$= \sum_{n=1}^N C_n \quad (41)$$

$$= \sum_{n=1}^N \frac{C_n}{\min\{2\theta_n, K\}} \min\{2\theta_n, K\}$$

$$= \sum_{n=1}^N \sigma_n \min\{2\theta_n, K\}, \quad (42)$$

where (40) is from (39); (41) is from (38); (42) is from (36).

Normalizing $\sum_{\ell=1}^T |\mathcal{D}_\ell|$ by σ_{sum} (see (37)), we then have

$$\frac{1}{\sigma_{\text{sum}}} \sum_{\ell=1}^T |\mathcal{D}_\ell| = \sum_{n=1}^N \frac{\sigma_n}{\sigma_{\text{sum}}} \min\{2\theta_n, K\} \quad (43)$$

$$\leq \frac{1}{N} \sum_{n=1}^N \min\{2\theta_n, K\} \quad (44)$$

$$\leq \min\left\{\frac{1}{N} \sum_{n=1}^N 2\theta_n, K\right\} \quad (45)$$

$$= \min\{2r, K\}. \quad (46)$$

Here (43) is the weighted average of the non-decreasing sequence $\min\{2\theta_n, K\}$, $1 \leq n \leq N$, with non-increasing weights $\frac{\sigma_n}{\sigma_{\text{sum}}}$, $1 \leq n \leq N$. But $\frac{1}{N} \sum_{n=1}^N \min\{2\theta_n, K\}$ is the simple average of $\min\{2\theta_n, K\}$, $1 \leq n \leq N$, and thus (44) holds. In addition, (45) is due to the property of the minimum function.

Based on (46), we have

$$\sigma_{\text{sum}} \geq \frac{\sum_{\ell=1}^T |\mathcal{D}_\ell|}{\min\{2r, K\}} \quad (47)$$

$$= \frac{\sum_{n=1}^N C_n}{\min\{2r, K\}} \quad (48)$$

$$= \frac{C_{\text{total}}}{\min\{2r, K\}}, \quad (49)$$

where (47) is from (46); (48) is from (38) and (39); (49) is from (35).

Furthermore, by the same argument as (31) we get that

$$|\mathcal{D}_\ell| \leq \min\{2r_{\ell,1}, K\} \quad (50)$$

where $r_{\ell,1}$ is the smallest number in $\{r_{\ell,j}\}_{j=1}^{|\mathcal{D}_\ell|}$ for block ℓ . On the other hand,

$$1 = \frac{|\mathcal{D}_\ell|}{|\mathcal{D}_\ell|} = \frac{\sum_{n=1}^N C_{\ell,n}}{|\mathcal{D}_\ell|} \quad (51)$$

$$\geq \frac{\sum_{n=1}^N C_{\ell,n}}{\min\{2r_{\ell,1}, K\}} \quad (52)$$

$$\geq \sum_{n:C_{\ell,n}=0} \frac{C_{\ell,n}}{\min\{2\theta_n, K\}} + \sum_{n:C_{\ell,n} \neq 0} \frac{C_{\ell,n}}{\min\{2\theta_n, K\}} \quad (53)$$

$$= \sum_{n=1}^N \frac{C_{\ell,n}}{\min\{2\theta_n, K\}}, \quad (54)$$

where (51) is from (39); (52) results from (50); (53) is due to the fact that for all n such that $C_{\ell,n} \neq 0$, we have $\theta_n \in \{r_{\ell,1}, \dots, r_{\ell,|\mathcal{D}_\ell|}\}$, and hence $r_{\ell,1} \leq \theta_n$. Thus,

$$T = \sum_{\ell=1}^T 1 \geq \left\lceil \sum_{\ell=1}^T \sum_{n=1}^N \frac{C_{\ell,n}}{\min\{2\theta_n, K\}} \right\rceil \quad (55)$$

$$\geq \left\lceil \sum_{n=1}^N \frac{C_n}{\min\{2\theta_n, K\}} \right\rceil \quad (56)$$

$$= \lceil \sigma_{\text{sum}} \rceil, \quad (57)$$

where (55) is from (54) and the interger property of T ; (56) is from (38); σ_{sum} is defined in (37). Combining (49) and (57), the total number of transmission blocks T can be bounded by

$$T \geq \lceil \sigma_{\text{sum}} \rceil \quad (58)$$

$$\geq \left\lceil \frac{C_{\text{total}}}{\min\{2r, K\}} \right\rceil \quad (59)$$

$$= \left\lceil \frac{NQ(1 - \frac{r}{K})}{\min\{2r, K\}} \right\rceil, \quad (60)$$

where (58) is from (57); (59) is from (49); C_{total} is defined in (11). Finally, the communication load L is

$$L = \frac{T}{NQ} \geq \frac{\left\lceil \frac{NQ(1 - \frac{r}{K})}{\min\{K, 2r\}} \right\rceil}{NQ} \geq \frac{1 - \frac{r}{K}}{\min\{K, 2r\}}, \quad (61)$$

which completes the proof.

VI. ACHIEVABILITY PROOF FOR THEOREM 1

In this section, we provide the achievability proof for Theorem 1. We present our file placement scheme as well as the one-shot linear transmission scheme. We consider the case when the number of files, N , is sufficiently large². Note that for a sufficiently large number of files N , we have

$$\alpha N_0 < N \leq (\alpha + 1)N_0$$

for some nonnegative integer α , where N_0 is defined by

$$N_0 = \begin{cases} \binom{K}{r}, & \text{if } r \geq K/2, \\ \binom{K-r-1}{r-1} \binom{K}{r}, & \text{if } r < K/2. \end{cases} \quad (62)$$

In our scheme, we add the following number of empty files

$$\Delta = (\alpha + 1)N_0 - N, \quad 0 \leq \Delta < N_0,$$

and then the number of input files becomes

$$\tilde{N} = N + \Delta = (\alpha + 1)N_0. \quad (63)$$

Afterwards, for every $\binom{K}{r}$ files, we design a *symmetric* file placement such that each file is placed at r out of the K nodes (see Fig. 3 for example). Then, the same placement can be copied $\tilde{N}/\binom{K}{r}$ times to complete the placement of \tilde{N} input files. Since communication is not needed when $r \geq K$, we will just focus on the cases when

$$r < K.$$

Similar to (11), the total number of intermediate values to be transmitted is

$$\tilde{N}Q \left(1 - \frac{r}{K}\right). \quad (64)$$

We describe below the intuition of designing an optimal achievable transmission scheme. Let us focus on the transmission of one intermediate value $a_{q,n}$, for a given pair (q, n) . Assume it is transmitted in block ℓ , and is intended for Receiver k for $\ell \in [1 : T]$ and $k \in [1 : K]$. Recall that \mathcal{S}_n denotes the indices of r nodes having the intermediate value $a_{q,n}$. This set of transmitters is viewed as a virtual transmitter. Recall that \mathcal{R}_ℓ denotes the indices of receivers in block ℓ . $\mathcal{J}_n = \mathcal{R}_\ell \setminus \{\{k\} \cup \mathcal{S}_n\}$ denotes the indices of receivers where the packet $\tilde{a}_{q,n}$ is zero forced. Thus $|\mathcal{J}_n| \leq |[1 : K] \setminus \{\{k\} \cup \mathcal{S}_n\}| = K - r - 1$. From the analysis in the converse proof in Section V, the number of receivers without interference from $a_{q,n}$, excluding the intended Receiver k , is:

$$[\text{side information cancellation:}] |\mathcal{S}_n \cap \mathcal{R}_\ell| \leq |\mathcal{S}_n| = r, \quad (65)$$

$$[\text{zero-forcing:}] |\mathcal{J}_n| \leq \min\{r - 1, K - r - 1\}, \quad (66)$$

and the total number of receivers in a block (i.e., $|\mathcal{R}_\ell|$, $\ell \in [1 : T]$) is upper bounded by $1 + |\mathcal{S}_n \cap \mathcal{R}_\ell| + |\mathcal{J}_n| \leq \min\{2r, K\}$.

We will show an optimal scheme such that $|\mathcal{R}_\ell| = \min\{2r, K\}$ for all ℓ . In particular, we show that there exists an assignment of the intermediate values to the blocks, such that for every $a_{q,n}$, the transmitters indexed by \mathcal{S}_n are a subset

²Note that our result also holds for the case with finite N as long as N can be expressed as $N = (\alpha + 1)N_0$, for some nonnegative integer α , where N_0 is defined in (62).

of the receivers indexed by \mathcal{R}_ℓ (i.e., $\mathcal{S}_n \subseteq \mathcal{R}_\ell$) and hence (65) holds with equality. As a result, (66) automatically holds with equality since $|\mathcal{J}_n| = |\mathcal{R}_\ell| - 1 - |\mathcal{S}_n| = \min\{r - 1, K - r - 1\}$.

For a sufficiently large number of files N , the algorithm of the general achievable scheme is described in Algorithm 1. The algorithm of the Shuffle phase is described in Algorithm 2. In what follows, we describe the scheme in details for different cases of $r < K$.

Algorithm 1 Achievable MapReduce Scheme

Map Phase:

- 1: **procedure** FILE PLACEMENT
- 2: Partition \tilde{N} files into $\tilde{N}/\binom{K}{r}$ disjoint groups
- 3: **for** $i = 1 : \tilde{N}/\binom{K}{r}$
- 4: Place $\binom{K}{r}$ files indexed by $[(i - 1)\binom{K}{r} + 1 : i\binom{K}{r}]$ symmetrically across K nodes, with each file placed at r out of the K nodes
- 5: **end for**
- 6: **end procedure**
- 7: **procedure** MAP FUNCTION
- 8: **for** $k = 1 : K$
- 9: Node k computes *Map* functions and outputs $a_{q,n}$, $q \in [1 : Q]$ and $n \in \mathcal{M}_k$
- 10: **end for**
- 11: **end procedure**

Shuffle Phase:

- 12: **procedure** SHUFFLE
- 13: **for** $\ell = 1 : T$
- 14: Deliver $\min\{2r, K\}$ intermediate values in block ℓ
- 15: **end for**
- 16: **end procedure**

Reduce Phase:

- 17: **procedure** REDUCE FUNCTION
 - 18: **for** $k = 1 : K$
 - 19: Node k computes *Reduce* functions indexed by \mathcal{W}_k
 - 20: **end for**
 - 21: **end procedure**
-

A. The case of $r \geq K/2$

In this case we will show that $K = \min\{2r, K\}$ intermediate values can be transmitted in each block. From (62), in this case we have the following number of data files

$$\tilde{N} = (\alpha + 1)N_0 = (\alpha + 1) \binom{K}{r}.$$

Recall that after the Map phase, the following set of intermediate values are cached at Node k , $k \in [1 : K]$,

$$\mathcal{P}_k = \{a_{q,n} : q \in [1 : Q], n \in \mathcal{M}_k\}, \quad (67)$$

with $|\mathcal{P}_k| = Q \cdot |\mathcal{M}_k|$, where $|\mathcal{M}_k| = \frac{\tilde{N}r}{K}$ according to our placement. Furthermore, the following set of intermediate values are required by Node k

$$\mathcal{G}_k = \{a_{q,n} : q \in \mathcal{W}_k, n \in [1 : \tilde{N}], n \notin \mathcal{M}_k\}, \quad (68)$$

with $|\mathcal{G}_k| = \frac{Q}{K}(\tilde{N} - |\mathcal{M}_k|) = \frac{\tilde{N}Q(1-\frac{r}{K})}{K}$.

In our scheme, we design

$$T = \frac{\tilde{N}Q(1-\frac{r}{K})}{K} \quad (69)$$

Algorithm 2 Shuffle Phase

Shuffle Phase:

```

1: procedure SHUFFLE
2:   procedure ENCODING
3:     1. Choose intermediate values:
4:     if  $r \geq K/2$ 
5:       for block index  $\ell = 1 : T$ 
6:         For every  $k \in [1 : K]$ , choose one
7:         undelivered  $a_{q,n}$  from  $\mathcal{G}_k$  as in (68).
8:       end for
9:     else ( $r < K/2$ )
10:      Initialize block index  $\ell = 1$ 
11:      for every  $\mathcal{R} \subseteq [1 : K]$ 
12:        for copy  $y = 1 : (\alpha + 1) \frac{Q}{K}$ 
13:          for  $i = 1 : \binom{2r-1}{r}$ 
14:            Choose one undelivered  $a_{q,n}$  from
15:             $A_{k,\mathcal{S}_{k,i}}$  defined in (72) and (73), for
16:            every  $k \in \mathcal{R}$ .
17:          Increase block index  $\ell = \ell + 1$ .
18:        end for
19:      end for
20:    end if
21:  2. Gaussian coding:  $a_{q,n} \in \mathbb{F}_2^B \rightarrow \tilde{a}_{q,n} \in \mathbb{C}^\tau$ ,
22:  where  $B = \tau \log P + o(\tau \log P)$ ,  $\forall q, n$ .
23:  3. Choose beamforming coefficients  $\beta_{i,q,n}$ ,
24:   $\forall q, n, i \in \mathcal{S}_n$ , to satisfy zero-forcing in (28).
25:  4. Node  $i$ :  $\mathbf{x}_i[\ell] = \sum_{(q,n): \tilde{a}_{q,n} \in \mathcal{D}_\ell, n \in \mathcal{M}_i} \beta_{i,q,n} \tilde{a}_{q,n}$ ,
26:   $i \in [1 : K], \ell \in [1 : T]$ .
27: end procedure
28: procedure DECODING
29:  1. Node  $k$  receives signal:  $\mathbf{y}_k[\ell] = \sum_{i=1}^K h_{k,i} \mathbf{x}_i[\ell] + \mathbf{z}_k[\ell]$ ,  $k \in [1 : K], \ell \in [1 : T]$ .
30:  2. Subtract the interference from  $\mathbf{y}_k[\ell]$  by using
31:  a linear function,  $\mathcal{L}_{k,\ell}(\mathbf{y}_k[\ell], \tilde{\mathcal{P}}_k)$ , where
32:   $\tilde{\mathcal{P}}_k = \{\tilde{a}_{q,n} : a_{q,n} \in \mathcal{P}_k\}$  is side information at
33:  Node  $k$ ,  $k \in [1 : K], \ell \in [1 : T]$ .
34:  3. Decode  $\tilde{a}_{q,n}$  as  $\mathcal{L}_{k,\ell}(\mathbf{y}_k[\ell], \tilde{\mathcal{P}}_k) = \tilde{a}_{q,n} + \mathbf{z}_k[\ell]$ ,
35:   $\forall q, n$ .
36:  4. Decoding:  $\tilde{a}_{q,n} \in \mathbb{C}^\tau \rightarrow a_{q,n} \in \mathbb{F}_2^B$ ,  $\forall q, n$ .
37: end procedure
38: end procedure

```

blocks such that in every block each of the K nodes receives one intermediate value without interference. Specifically, in each block we choose one of the undelivered intermediate values arbitrarily from \mathcal{G}_k , for all $k \in [1 : K]$. As a result, in each block, K intermediate values are selected, each intended for a different receiver. For each selected intermediate value, it interferes with $K - 1$ unintended receivers. However, we note that for any intermediate value $a_{q,n}$, (65) and (66) hold with equality, since $\mathcal{R}_\ell = [1 : K]$, $|\mathcal{S}_n \cap \mathcal{R}_\ell| = |\mathcal{S}_n| = r$, and $|\mathcal{J}_n| = K - r - 1 = \min\{r - 1, K - r - 1\}$. Thus a total of $K = \min\{2r, K\}$ intermediate values can be transmitted in every block.

In our scheme, one intermediate value in \mathcal{G}_k , $\forall k \in [1 : K]$, is delivered at each block. It implies that the number of blocks to deliver all the required intermediate values is

$$T = |\mathcal{G}_1| = \dots = |\mathcal{G}_K| = \frac{\tilde{N}Q(1-\frac{r}{K})}{K},$$

which can be rewritten as

$$T = \frac{NQ(1-\frac{r}{K})}{K} + \frac{\Delta Q(1-\frac{r}{K})}{K}, \quad (70)$$

where $0 \leq \Delta < N_0$, $N_0 = \binom{K}{r}$ (see (62) and (63)). The second term on the right hand side of (70) can be bounded by

$$\frac{\Delta Q(1-\frac{r}{K})}{K} < \frac{N_0 Q(1-\frac{r}{K})}{K} = o(N), \quad (71)$$

where $o(N)/N$ vanishes when N grows and Q, K, r are kept fixed. As mentioned, such scaling of N is seen in many big data applications. Therefore, for a large N , the communication load L is

$$L = \frac{T}{NQ} = \frac{1-\frac{r}{K}}{K}.$$

B. The case of $r < K/2$

In this case, at each transmission block we choose $2r = \min\{2r, K\}$ nodes out of K nodes as receivers, and a subset of them as transmitters. Next, we show that $2r$ intermediate values can be transmitted for each block without interference.

From (62) and (63), in this case we have the following number of data files

$$\tilde{N} = (\alpha + 1) \binom{K-r-1}{r-1} \binom{K}{r}.$$

For any $k \in [1 : K]$ and $\mathcal{S} \subseteq [1 : K] \setminus \{k\}$, $|\mathcal{S}| = r$, let us define a set of intermediate values as

$$A_{k,\mathcal{S}} = \{a_{q,n} : q \in \mathcal{W}_k, n \in \cap_{j \in \mathcal{S}} \mathcal{M}_j\}.$$

By definition, for each intermediate value in $A_{k,\mathcal{S}}$, it is required by Node k for its Reduce functions and it is cached in each of the nodes indexed by \mathcal{S} . Note that due to the

TABLE I
AN EXAMPLE FOR ONE COPY WITH 3 BLOCKS, FOR $r = 2$ AND
 $\mathcal{R} = \{1, 2, 3, 4\}$.

Receiver	1	2	3	4
block 1	$A_{1,\{2,3\}}^1$	$A_{2,\{1,3\}}^1$	$A_{3,\{1,2\}}^1$	$A_{4,\{1,2\}}^1$
block 2	$A_{1,\{2,4\}}^2$	$A_{2,\{1,4\}}^2$	$A_{3,\{1,4\}}^2$	$A_{4,\{1,3\}}^2$
block 3	$A_{1,\{3,4\}}^2$	$A_{2,\{3,4\}}^2$	$A_{3,\{2,4\}}^2$	$A_{4,\{2,3\}}^2$

symmetric file placement, for every pair (k, \mathcal{S}) , the number of intermediate values in $A_{k,\mathcal{S}}$ is

$$|A_{k,\mathcal{S}}| = \frac{Q}{K} \frac{\tilde{N}}{\binom{K}{r}} = (\alpha + 1) \frac{Q}{K} \binom{K-r-1}{r-1}. \quad (72)$$

Let $\mathcal{R} \subseteq [1 : K]$ be the indices of an arbitrary set of $2r$ receivers, $|\mathcal{R}| = 2r$. We next design $(\alpha + 1) \frac{Q}{K} \binom{2r-1}{r}$ blocks such that in every block, every node whose index is in \mathcal{R} receives one intermediate value without interference. Such blocks can be viewed as $(\alpha + 1) \frac{Q}{K}$ copies, each copy corresponding to $\binom{2r-1}{r}$ blocks. We describe the transmission for one copy, and without loss of generality we index the corresponding blocks of that copy by $1, 2, \dots, \binom{2r-1}{r}$. The transmissions for the other copies are the same.

For every $k \in \mathcal{R}$, let

$$\mathcal{S}_{k,1}, \mathcal{S}_{k,2}, \dots, \mathcal{S}_{k,\binom{2r-1}{r}} \quad (73)$$

be the subsets of $\mathcal{R} \setminus \{k\}$ in any given order, each subset with size r , i.e., $|\mathcal{S}_{k,i}| = r$ for $i = 1, 2, \dots, \binom{2r-1}{r}$. These subsets are used as different virtual transmitters for Receiver k . In the i -th block, $1 \leq i \leq \binom{2r-1}{r}$, one intermediate value in $A_{k,\mathcal{S}_{k,i}}$ is transmitted, for all $k \in \mathcal{R}$. From (65) and (66), when an intermediate value in $A_{k,\mathcal{S}_{k,i}}$ is transmitted, it can be canceled using side information at r undesired receivers indexed by $\mathcal{S}_{k,i}$ (because it is cached in the nodes indexed by $\mathcal{S}_{k,i}$); it can be zero-forced at the remaining $r-1 = \min\{r-1, K-r-1\}$ undesired receivers. Hence, in block i , each of $2r$ receivers in \mathcal{R} gets a desired intermediate value without interference. In addition, over the $\binom{2r-1}{r}$ blocks, a total of $2r \cdot \binom{2r-1}{r}$ intermediate values are transmitted, where each of them comes from one (and only one) of the sets $\{A_{k,\mathcal{S}} : k \in \mathcal{R}, \mathcal{S} \subseteq \mathcal{R} \setminus \{k\}, |\mathcal{S}| = r\}$.

For example, let $r = 2$, $\mathcal{R} = \{1, 2, 3, 4\}$. One copy of the scheme has $\binom{2r-1}{r} = 3$ blocks. Some details of one copy are given in Table I. In Table I, $A_{k,\mathcal{S}}^j$ denotes the j -th element of set $A_{k,\mathcal{S}}$ for $j \in [1 : |A_{k,\mathcal{S}}|]$. We can arbitrarily choose the superscript j as long as the intermediate value has not been sent. In this example, every transmitted intermediate value can be decoded at the intended receiver without interference. Note that $\{2, 3\}$, $\{2, 4\}$ and $\{3, 4\}$ are three subsets of $\mathcal{R} \setminus \{1\}$ and we choose $\mathcal{S}_{1,1} = \{2, 3\}$, $\mathcal{S}_{1,2} = \{2, 4\}$ and $\mathcal{S}_{1,3} = \{3, 4\}$, corresponding to the column for Receiver 1. One can also permute these three subsets in any other order and have, e.g., $\mathcal{S}_{1,1} = \{2, 4\}$, $\mathcal{S}_{1,2} = \{2, 3\}$ and $\mathcal{S}_{1,3} = \{3, 4\}$.

Now for every $\mathcal{R} \subseteq [1 : K]$ of size $2r$, we proceed as before and create $(\alpha + 1) \frac{Q}{K} \binom{2r-1}{r}$ blocks. In every block, exactly $2r$

intermediate values can be transmitted without interference. Moreover, the scheme is symmetric, in the sense that a total of $(\alpha + 1) \frac{Q}{K} \binom{K-r-1}{r-1} = |A_{k,\mathcal{S}}|$ intermediate values in $A_{k,\mathcal{S}}$ are transmitted at the end of the scheme, for any $k \in [1 : K]$, $\mathcal{S} \subseteq [1 : K] \setminus \{k\}$, $|\mathcal{S}| = r$. This can be seen from the following facts: there are $\binom{K-r-1}{r-1}$ choices of \mathcal{R} that include k and \mathcal{S} ; for every such \mathcal{R} we create $(\alpha + 1) \frac{Q}{K}$ copies; and for every copy we transmit one intermediate value in $A_{k,\mathcal{S}}$.

One can see an example with $(\tilde{N} = 20, K = Q = 5, r = 2)$ in Table II. Let $\{A_{k,\mathcal{S}}^1\}_{k,\mathcal{S}}$ be the set of intermediate values associated with Files 1 to 10, and $\{A_{k,\mathcal{S}}^2\}_{k,\mathcal{S}}$ be the set of intermediate values associated with Files 11 to 20. Note that focusing on the intermediate values in the set $\{A_{k,\mathcal{S}}^1\}_{k,\mathcal{S}}$ in Table II, we can extract a scheme with $(N = 10, T = 8)$ that is identical to the example (see Fig. 4) in Section IV-B. For example, the four intermediate values $(a_{2,8}, a_{3,7}, a_{4,7}, a_{5,8})$ in block 1 of Fig. 4 correspond to the four intermediate values $(A_{2,\{3,4\}}^1, A_{3,\{2,5\}}^1, A_{4,\{2,5\}}^1, A_{5,\{3,4\}}^1)$ in block 1 of Table II. Similarly, the transmissions of blocks 1, 2, \dots , 8 in Fig. 4 match the transmissions of blocks 1, 2, 4, 5, 7, 8, 10, 13 in Table II, respectively. We note that when the number of files increases from $N = 10$ to $N = 20$, the communication load is reduced from $L = 0.16$ to $L = 0.15$, due to the integral effect in $\lceil \frac{C_{\text{total}}}{2r} \rceil$.

Based on the above scheme, and similar to (70) and (71), the number of transmission blocks T is

$$T = \frac{\tilde{N}Q(1 - \frac{r}{K})}{2r} \quad (74)$$

$$= \frac{NQ(1 - \frac{r}{K})}{2r} + o(N). \quad (75)$$

Finally, for a large N , the communication load L is given as

$$L = \frac{T}{NQ} = \frac{1 - \frac{r}{K}}{2r}. \quad (76)$$

Remark 4. As a sanity check, the total number of blocks is also equal to

$$T = (\alpha + 1) \frac{Q}{K} \binom{2r-1}{r} \binom{K}{2r}, \quad (77)$$

where $(\alpha + 1) \frac{Q}{K} \binom{2r-1}{r}$ is the number of blocks for each $2r$ receiver set \mathcal{R} , and $\binom{K}{2r}$ is the number of choices of receiver sets \mathcal{R} . One can easily verify that (77) is equal to (74).

Remark 5. In the proof, at least one copy of $\binom{2r-1}{r}$ blocks for receivers \mathcal{R} is needed. However, it may be possible to reduce the number of blocks in a copy, and hence reduce the minimum required \tilde{N} . The smallest \tilde{N} for given parameters is an open problem.

VII. CONCLUSION

In this work, we studied the MapReduce-based wireless distributed computing framework, where the distributed nodes exchange information over a wireless interference network. We demonstrated an optimal tradeoff between the computation load and communication load, under the assumption of one-shot linear schemes. One possible future direction is to allow arbitrary given file placement in the Map phase, with a given

TABLE II

AN EXAMPLE WITH $(\tilde{N} = 20, K = Q = 5, r = 2)$. FOR EACH RECEIVER SET \mathcal{R} , WE DESIGN $(\alpha + 1) \frac{Q}{K} = 1$ COPY OF $\binom{2r-1}{r} = 3$ BLOCKS. WE LIST ALL THE TRANSMITTED INTERMEDIATE VALUES AND THE CORRESPONDING RECEIVERS IN EACH BLOCK. INTERMEDIATE VALUE $A_{k,S}^1$ CORRESPONDS TO FILES 1 TO 10; INTERMEDIATE VALUES $A_{k,S}^2$ CORRESPONDS TO FILES 11 TO 20.

Receiver (Node)	1				2				3				4				5			
File placement	w_1	w_2	w_3	w_4	w_1	w_5	w_6	w_7	w_2	w_5	w_8	w_9	w_3	w_6	w_8	w_{10}	w_4	w_7	w_9	w_{10}
	w_{11}	w_{12}	w_{13}	w_{14}	w_{11}	w_{15}	w_{16}	w_{17}	w_{12}	w_{15}	w_{18}	w_{19}	w_{13}	w_{16}	w_{18}	w_{20}	w_{14}	w_{17}	w_{19}	w_{20}
block 1							$A_{2,\{3,4\}}^1$				$A_{3,\{2,5\}}^1$				$A_{4,\{2,5\}}^1$				$A_{5,\{3,4\}}^1$	
block 2							$A_{2,\{4,5\}}^1$				$A_{3,\{4,5\}}^1$				$A_{4,\{2,3\}}^1$				$A_{5,\{2,3\}}^1$	
block 3							$A_{2,\{3,5\}}^2$				$A_{3,\{2,4\}}^2$				$A_{4,\{3,5\}}^2$				$A_{5,\{2,4\}}^2$	
block 4		$A_{1,\{3,5\}}^1$									$A_{3,\{1,4\}}^1$				$A_{4,\{3,5\}}^1$				$A_{5,\{1,4\}}^1$	
block 5		$A_{1,\{4,5\}}^1$									$A_{3,\{1,5\}}^1$				$A_{4,\{1,3\}}^1$				$A_{5,\{1,3\}}^1$	
block 6		$A_{1,\{3,4\}}^2$									$A_{3,\{4,5\}}^2$				$A_{4,\{1,5\}}^2$				$A_{5,\{3,4\}}^2$	
block 7		$A_{1,\{2,4\}}^1$					$A_{2,\{1,5\}}^1$								$A_{4,\{1,5\}}^1$				$A_{5,\{2,4\}}^1$	
block 8		$A_{1,\{2,5\}}^1$					$A_{2,\{1,4\}}^1$								$A_{4,\{1,2\}}^2$				$A_{5,\{1,2\}}^2$	
block 9		$A_{1,\{4,5\}}^2$					$A_{2,\{4,5\}}^2$								$A_{4,\{2,5\}}^2$				$A_{5,\{1,4\}}^2$	
block 10		$A_{1,\{2,3\}}^1$					$A_{2,\{3,5\}}^1$				$A_{3,\{1,2\}}^1$								$A_{5,\{1,2\}}^1$	
block 11		$A_{1,\{2,5\}}^2$					$A_{2,\{1,3\}}^2$				$A_{3,\{1,5\}}^2$								$A_{5,\{1,3\}}^2$	
block 12		$A_{1,\{3,5\}}^2$					$A_{2,\{1,5\}}^2$				$A_{3,\{2,5\}}^2$								$A_{5,\{2,3\}}^2$	
block 13		$A_{1,\{3,4\}}^1$					$A_{2,\{1,3\}}^1$				$A_{3,\{2,4\}}^1$				$A_{4,\{1,2\}}^1$					
block 14		$A_{1,\{2,3\}}^2$					$A_{2,\{1,4\}}^2$				$A_{3,\{1,2\}}^2$				$A_{4,\{1,3\}}^2$					
block 15		$A_{1,\{2,4\}}^2$					$A_{2,\{3,4\}}^2$				$A_{3,\{1,4\}}^2$				$A_{4,\{2,3\}}^2$					

average computation load, and find the corresponding optimal achievable scheme. Moreover, the communication cost is an open problem when channel state information and synchronization are not fully available. Another direction is to characterize the fundamental tradeoff between the computation load and communication load without the assumption of one-shot linear schemes, where it may be possible apply the interference alignment approach to improve the system performance.

REFERENCES

- [1] H. Vardhan, S.-R. Ryu, B. Banerjee, and R. Prakash, "60GHz wireless links in data center networks," *Computer Networks*, vol. 58, pp. 192–205, Jan. 2014.
- [2] J. Bao, D. Dong, B. Zhao, Z. Luo, C. Wu, and Z. Gong, "Flycast: Free-space optics accelerating multicast communications in physical layer," in *ACM SIGCOMM Computer Communication Review*, Oct. 2015, pp. 97–98.
- [3] E. Marinelli, "Hyrax: Cloud computing on mobile devices using mapreduce," 2009, available on: <http://www.dtic.mil/dtic/tr/fulltext/u2/a512601.pdf>.
- [4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing – A key technology towards 5G," *ETSI white paper*, no. 11, pp. 1–16, 2015.
- [5] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, Jun. 2015.
- [6] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
- [7] T. Jeong, J. Chung, J. Hong, and S. Ha, "Towards a distributed computing framework for fog," in *IEEE Fog World Congress*, Nov. 2017.
- [8] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coding for distributed fog computing," *IEEE Commun. Magazine*, vol. 55, no. 4, pp. 34–40, Apr. 2017.
- [9] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
- [10] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.
- [11] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded MapReduce," in *Proc. Allerton Conf. Communication, Control and Computing*, Sep. 2015, pp. 964–971.
- [12] Q. Yu, S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "How to optimally allocate resources for coded distributed computing?" in *Proc. IEEE Int. Conf. Communications (ICC)*, May 2017.
- [13] Y. Ezzeldin, M. Karmoose, and C. Fragouli, "Communication vs distributed computation: an alternative trade-off curve," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2017.
- [14] Q. Yan, S. Yang, and M. Wigger, "A storage-computation-communication tradeoff for distributed computing," 2018, available on: <https://arxiv.org/abs/1805.10462>.
- [15] N. Shakya, F. Li, and J. Chen, "On distributed computing with heterogeneous communication constraints," available on ArXiv:<http://arxiv.org/abs/1802.00413>.
- [16] N. Woolsey, R. Chen, and M. Ji, "A new combinatorial design of coded distributed computing," 2018, available on: <https://arxiv.org/abs/1802.03870>.
- [17] E. Parrinello, E. Lampiris, and P. Elia, "Coded distributed computing with node cooperation substantially increases speedup factors," 2018, available on: <https://arxiv.org/abs/1802.04172>.
- [18] K. Konstantinidis and A. Ramamoorthy, "Leveraging coding techniques for speeding up distributed computing," 2018, arXiv preprint [arXiv:1802.03049](https://arxiv.org/abs/1802.03049).
- [19] S. Prakash, A. Reiszadeh, R. Pedarsani, and A. S. Avestimehr, "Coded computing for distributed graph analytics," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018.
- [20] S. Srinivasavaradhan, L. Song, and C. Fragouli, "Distributed computing trade-offs with random connectivity," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018.
- [21] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Compressed coded distributed computing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018.
- [22] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "A scalable framework for wireless distributed computing," *IEEE/ACM Trans. Netw.*, vol. pp. no. 99, pp. 1–12, May 2017.

- [23] —, “Coded distributed computing: Fundamental limits and practical challenges,” in *Proc. Asilomar Conf. Signals, Systems and Computers*, Nov. 2016, pp. 509–513.
- [24] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, “Communication-aware computing for edge processing,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2885–2889.
- [25] M. Attia and R. Tandon, “On the worst-case communication overhead for distributed data shuffling,” in *Proc. Allerton Conf. Communication, Control and Computing*, Sep. 2016, pp. 961–968.
- [26] L. Song, C. Fragouli, and T. Zhao, “A pliable index coding approach to data shuffling,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2558–2562.
- [27] M. Attia and R. Tandon, “Information theoretic limits of data shuffling for distributed learning,” in *Proc. IEEE Global Conf. Communications (GLOBECOM)*, Dec. 2016.
- [28] M. A. Maddah-Ali and U. Niesen, “Cache-aided interference channels,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 809–813.
- [29] J. Hachem, U. Niesen, and S. Diggavi, “Degrees of freedom of cache-aided wireless interference networks,” Nov. 2016, available on ArXiv: <https://arxiv.org/pdf/1606.03175v3.pdf>.
- [30] A. Sengupta, R. Tandon, and O. Simeone, “Fog-aided wireless networks for content delivery: Fundamental latency tradeoffs,” *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6650–6678, Oct. 2017.
- [31] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of caching in wireless D2D networks,” *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
- [32] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, “Fundamental limits of cache-aided interference management,” *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3092–3107, Feb. 2017.
- [33] F. Xu, M. Tao, and K. Liu, “Fundamental tradeoff between storage and latency in cache-aided wireless interference networks,” *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7464–7491, Nov. 2017.
- [34] S. Dutta, V. Cadambe, and P. Grover, “‘Short-dot’: computing large linear transforms distributedly using coded short dot products,” in *Advances In Neural Information Processing Systems (NIPS)*, Dec. 2016.
- [35] —, “Coded convolution for parallel and distributed computing within a deadline,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2403–2407.
- [36] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, “Coded distributed computing: Straggling servers and multistage dataflows,” in *Proc. Allerton Conf. Communication, Control and Computing*, Sep. 2016, pp. 164–171.
- [37] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, “Speeding up distributed machine learning using codes,” *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.
- [38] M. Attia and R. Tandon, “Combating computational heterogeneity in large-scale distributed computing via work exchange,” 2017, available on ArXiv: <https://arxiv.org/abs/1711.08452>.
- [39] K. Lee, C. Suh, and K. Ramchandran, “High-dimensional coded matrix multiplication,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2418–2422.
- [40] C. Karakus, Y. Sun, and S. Diggavi, “Encoded distributed optimization,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2890–2894.
- [41] A. Reiszadeh, S. Prakash, R. Pedarsani, and S. Avestimehr, “Coded computation over heterogeneous clusters,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2408–2412.
- [42] A. Reiszadeh and R. Pedarsani, “Latency analysis of coded computation schemes over wireless networks,” in *Proc. Allerton Conf. Communication, Control and Computing*, Oct. 2017, pp. 1256–1263.
- [43] R. Tandon, Q. Lei, A. Dimakis, and N. Karampatziakis, “Gradient coding: Avoiding stragglers in distributed learning,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- [44] H. Park, K. Lee, J. Sohn, C. Suh, and J. Moon, “Hierarchical coding for distributed computing,” 2018, available on ArXiv: <https://arxiv.org/abs/1801.04686>.
- [45] H. McMahan, F. Moore, D. Ramage, S. Hampson, and B. Arcas, “Communication-efficient learning of deep networks from decentralized data,” available on ArXiv: <https://arxiv.org/abs/1602.05629>.
- [46] J. Konecny, H. McMahan, X. Yu, P. Richtarik, A. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” 2017, available on: <https://arxiv.org/abs/1610.05492>.
- [47] N. Agarwal, A. Suresh, F. Yu, S. Kumar, and H. McMahan, “cpSGD: Communication-efficient and differentially-private distributed SGD,” available on ArXiv: <https://arxiv.org/abs/1805.10559>.
- [48] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley-Interscience, 2006.
- [49] U. Erez and R. Zamir, “Achieving $\frac{1}{2} \log(1 + \text{SNR})$ on the AWGN channel with lattice encoding and decoding,” *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.
- [50] M. Razaviyayn, G. Lyubeznik, and Z. Q. Luo, “On the degrees of freedom achievable through interference alignment in a MIMO interference channel,” *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 812–821, Feb. 2012.

Fan Li is currently pursuing the Ph.D. degree with the Electrical Engineering Department of Louisiana Tech University. She received the B.Sc. degree from Shandong Technology and Business University in 2012, and the M.Sc. degree from University of Shanghai for Science and Technology in 2015. Her research interests include information theory, distributed computing and machine learning.

Jinyuan Chen is an assistant professor in the Electrical Engineering Department of Louisiana Tech University. Before joining Louisiana Tech, he was a postdoctoral scholar at Stanford University from 2014 to 2016. He received the B.Sc. degree from Tianjin University in 2007, the M.Sc. degree from Beijing University of Posts and Telecommunications in 2010, and the Ph.D. degree from Télécom ParisTech in 2014. His research interests include information theory, communication theory, security and privacy, distributed computing, and machine learning.

Zhiying Wang received the B.Sc. degree in Information Electronics and Engineering from Tsinghua University in 2007, M. Sc. and Ph.D degrees in Electrical Engineering from California Institute of Technology in 2009 and 2013, respectively. She was a postdoctoral fellow in Department of Electrical Engineering, Stanford University. She is currently Assistant Professor at Center for Pervasive Communications and Computing, University of California, Irvine. Dr. Wang is the recipient of NSF Center for Science of Information (CSoI) Postdoctoral Research Fellow, 2013. She received IEEE Communication Society Data Storage Best Paper Award. Her research focuses on information theory, coding theory, with an emphasis on coding for data storage.