Exploiting Computation Replication for Mobile Edge Computing: A Fundamental Computation-Communication Tradeoff Study

Kuikui Li, Meixia Tao, Zhiyong Chen

Abstract

Existing works on task offloading in mobile edge computing (MEC) networks often assume a task be executed once at a single edge node (EN). Downloading the computed result from the EN back to the mobile user may suffer long delay if the downlink channel experiences strong interference or deep fading. This paper exploits the idea of computation replication in MEC networks to speed up the downloading phase. Computation replication allows each user to offload its task to multiple ENs for repetitive execution so as to create multiple copies of the computed result at different ENs which can then enable transmission cooperation and hence reduce the communication latency for result downloading. Yet, computation replication may also increase the communication latency for task uploading, despite the obvious increase in computation load. The main contribution of this work is to characterize asymptotically an order-optimal upload-download communication latency pair for a given computation load in a multi-user multi-server MEC network. Analysis shows when the computation load increases within a certain range, the downloading time can decrease in an inversely proportional way if it is binary offloading or decrease linearly if it is partial offloading, both at the expense of increasing the uploading time linearly.

Index Terms

Mobile Edge Computing, Computation Replication, Computation-Communication Tradeoff, Transmission Cooperation

I. INTRODUCTION

The explosive growth of Internet of Things is driving the emergence of new mobile applications that demand intensive computation and stringent latency, such as intelligent navigation, online gaming, virtual reality (VR), and augmented reality (AR) [3]. The limited computation and energy resources at mobile devices pose a great challenge for supporting these new applications [4]. Mobile edge computing (MEC) is envisioned as a promising network architecture to address this challenge by providing cloud-computing services at the edge nodes (ENs) of mobile networks, such as wireless access points and base stations [5], [6]. By offloading computation-intensive tasks from mobile users to their nearby server-enabled ENs for processing, MEC systems have great potential to prolong the battery lifetime of mobile devices and reduce the overall task execution latency.

Task offloading and resource allocation are crucial problems in MEC systems. This is because offloading a task from a user to its associated EN involves extra overhead in transmission energy and communication latency due to the input data uploading and computation result downloading

[7]–[9]. In the existing literature, the task offloading and resource (both radio resource and computation resource) allocation problems have been studied for single-user single-server MEC systems in [10], [11], for multi-user single-server MEC systems in [12]–[19], for single-user multi-server MEC systems in [20], and for multi-user multi-server MEC systems in [21]. These problems are often formulated as minimizing the energy consumption under latency constraints or minimizing the latency subject to energy constraints, so as to strike a good balance between communication efficiency and computation efficiency. The task offloading strategy also depends on whether the task is dividable, known as partial offloading [11]–[15], or has to be executed as a whole, known as binary offloading [16]–[21].

Note that in the aforementioned literature [10]–[21], each task or subtask is executed once at one EN only. As a result, the computed result of the task/subtask being offloaded is only available at one EN without any diversity. Consider the scenario where the downlink channel from the EN to the user experiences deep fading or suffers from strong interference, downloading the computed results back to the user may incur very low transmission rate and consequently cause significantly long delay in completing the overall task. This work aims to address this issue by exploiting computation replication when there are multiple edge servers.

The main idea of computation replication is to let mobile users offload their tasks or subtasks to multiple ENs for repeated execution so as to create multiple copies of the computed result at different ENs which then can enable the multiple ENs to cooperatively transmit the computed result back to users in the downlink. This transmission cooperation can mitigate interferences across users, overcome deep fading, and hence increase the transmission rate of downlink channels. Thus, the downloading time can be reduced. But note that a side effect of replicating tasks on multiple ENs is that it introduces more data traffic in the uplink and hence may increase the uploading time compared with offloading to a single EN. That means computation replication can induce a tradeoff between uploading time and downloading time. Take a 3-user 3-server MEC network for example. First, consider that each user offloads its task to all 3 servers for repeated computing. In the uploading phase, each user takes turn to multicast its task input to all 3 ENs, consuming 3 time slots in total. In the downloading phase, all the ENs have the same message to transmit, and the downlink channel thus becomes a virtual MISO broadcast channel, and all computed results can be delivered to users within 1 time slot by using zero-forcing precoding [22]. For the baseline scheme where each task is offloaded to an individual EN, both the uplink and downlink channels are 3-user interference channels [23]. Both task uploading and downloading can be completed within 2 time slots by using interference alignment. It is seen that by computation replication, the downloading time is reduced from 2 time slots to 1 time slot while the uploading time is increased from 2 time slots to 3 time slots. For those computation tasks whose output data size is much larger than the input data size or the input data size is negligible, computation replication can bring significant benefits in reducing the overall communication latency. In general, computation replication should be carefully chosen by considering the balance between upload and download times, despite the obvious increase in the computation load.

Computation replication, i.e., replicating a task on multiple servers, has already demonstrated significant advantages in modern computer systems. It can mitigate the random server straggling and thus reduce the task service delay in queuing systems [24]–[26]. It can also create coded multicasting for data shuffling and thus reduce the communication load in distributed computing frameworks, like MapReduce and Spark [27], [28]. Our work is an attempt to exploit this idea of computation replication in MEC systems to enable the transmission cooperation for speeding up the computed result downloading. To our best knowledge, the only work that uses the similar idea

of computation replication to enable transmission cooperation for computed result downloading in MEC systems is [29]. However, [29] relies on a strong assumption that the computation functions are linear so that the tasks can be executed on some linear combinations of the inputs, i.e., coded input, and then the computed coded outputs on all ENs are utilized to zero-force the downlink interference at each user. In addition, the work [29] ignores the task uploading phase.

In this work, we exploit computation replication in a general multi-user multi-server MEC system with general computation function. Unlike [29], the transmission cooperation in the downlink enabled by computation replication does not assume linearity of the computation function. It relies purely on the replication of the computed result of each individual task at multiple ENs. Moreover, we consider both task uploading and results downloading and adopt the upload-download latency pair as the performance metric. In specific, we consider an MEC network, where a set of N mobile users offload their tasks to a set of M computing-enabled ENs. Each task has an input data and an output data. We define the *computation load r* as the average number of ENs to compute a task, i.e., the degrees of replication for executing a task. The *communication latency* is defined as the upload-download time pair, denoted as (τ^u, τ^d). A fundamental question we would like to address is: *Given a computation load r, what is the minimum achievable communication latency boundary* $\big(\tau^u(r), \tau^d(r)\big)$?

Our work attempts to address the above question for a general MEC network with any M (≥ 2) ENs and any N (≥ 2) users, in both binary and partial offloading cases from an information-theoretic perspective. We reveal a fundamental tradeoff between computation load and communication latency, and also present the uploading time and downloading time tradeoff.

We first consider binary offloading where the computation tasks are not dividable. We propose a task assignment scheme where each task is offloaded to r different ENs for repeated computing, and each EN has an even assignment of $\frac{Nr}{M}$ tasks. By utilizing the duplicated computation results on multiple ENs, transmission cooperation in the form of interference neutralization can be exploited in the data downloading phase. We characterize the communication latency by the pair of the normalized uploading time (NULT) $\tau^u(r)$ and normalized downloading time (NDLT) $\tau^d(r)$. The main distinction in the communication latency $\big(\tau^u(r), \tau^d(r)\big)$ analysis lies at the degree of freedom (DoF) analysis of the so-called circular cooperative interference-multicast channels. We obtain the optimal per-receiver DoF for the uplink channel, and an order-optimal per-receiver DoF for the downlink channel. Based on these DoF regions, we then develop an order-optimal achievable communication latency pair at any integer computation load. In particular, the NULT is exactly optimal and the NDLT is within a multiplicative gap of 2 to the optimum. We show that the NDLT is an ***inversely proportional function*** of the computation load in the interval $\frac{MN}{M+N} \leq r \leq M$, which presents the ***computation-communication tradeoff***. We also reveal that the decrease of NDLT is at the expense of increasing the NULT linearly, which forms another NULT-NDLT tradeoff. Part of this result is submitted to IEEE ISIT 2019 [2].

Next, we consider partial offloading where each task can be divided arbitrarily. We propose a task partition scheme where the task generated by each user is partitioned into $\binom{M}{r}$ subtasks, and each is offloaded to a distinct subset of r ENs chosen from the total M ENs for repeated computing. The uplink is formed as the X-multicast channel whose optimal per-receiver DoF is obtained. The downlink is the cooperative X channel, and an achievable per-receiver DoF is derived with order optimality. We thus develop an order-optimal achievable communication latency pair at any given computation load, and both the achievable NULT and NDLT are within multiplicative gaps of 2 to their lower bounds. Moreover, the NDLT ***decreases linearly*** with the computation load in the interval $1 \leq r \leq \min\{M,N\}$, which is also at the expense of increasing the NULT linearly. Part of this result is presented in [1].
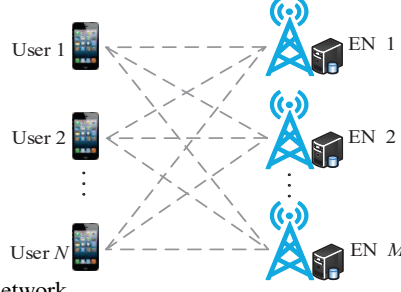
Fig. 1. A multi-user multi-server MEC network.

The rest of this paper is organized as follows. Section II presents the problem formulation and definitions. The computation-communication tradeoffs are presented in Section III for binary offloading and Section IV for partial offloading. The conclusions are drawn in Section V.

Notations: $\mathbb{C}$ denotes the set of complex numbers. $\mathbb{Z}^+$ denotes the set of positive integers. $(\cdot)^T$ denotes the transpose. $\mathcal{K}$ denotes the set of indexes $\{1, 2, \cdots, K\}$. $\lfloor r \rfloor$ denotes the largest integer no greater than r while $\lceil r \rceil$ denotes the minimum integer no smaller than r. $|\Phi|$ denotes the cardinality of set Φ. $[a]$ denotes the set of integers $\{1, 2, \cdots, a\}$. $[a:b]$ denotes the set of integers $\{a, a+1, \cdots, b\}$. $(x^j)_{j=1}^L$ denotes the vector $(x_1, x_2, \cdots, x_L)^T$. $\{U_j : j \in [a]\}$ denotes the set $\{U_1, U_2, \cdots, U_a\}$.

II. PROBLEM FORMULATION

A. MEC Network Model

We consider an MEC network consisting of M single-antenna ENs and N single-antenna users, as shown in Fig. 1. Each EN is equipped with a computing server and they all communicate with all users via a shared wireless channel. Denote by $\mathcal{M} = \{1, 2, \ldots, M\}$ the set of ENs and $\mathcal{N} = \{1, 2, \ldots, N\}$ the set of users. The communication link between each EN and each user experiences both channel fading and an additive white Gaussian noise. Let $h_{ij}(g_{ji})$ denote the uplink (downlink) channel fading from user $j \in \mathcal{N}$ (EN $i \in \mathcal{M}$) to EN $i \in \mathcal{M}$ (user $j \in \mathcal{N}$). It is assumed to be independent and identically distributed (i.i.d.) as some continuous distribution.

The network is time-slotted. At each time slot, each user generates an independent computation task to be offloaded to the ENs for execution. The computation task on each user j, for $j \in \mathcal{N}$, is characterized by the input data to be computed, denoted as W_j, with size $|W_j| = L$ bits, the computed output data, denoted as $\widetilde{W}_j$, with size $|\widetilde{W}_j| = \widetilde{L}$ bits[1]. We consider both binary and partial computation task offloading models. Binary offloading requires a task to be executed as a whole. Partial offloading, on the other hand, allows a task to be partitioned into multiple modules and executed in different nodes in a distributed manner. While binary offloading is suitable for simple tasks that are tightly integrated in structure and are not separable, partial offloading is more suitable for data-oriented applications that can be separated into multiple parts and executed in distributed nodes in parallel, such as image processing, voice-to-text conversion, and components rendering in $360°$ VR video display. For partial offloading, it is further assumed that the task partition is exclusive without intra-task or inter-task coding and the computed output size of each subtask is proportional to its corresponding input data size.

[1]Here the equal size for both input data and output data from all tasks is assumed for analytical tractability. In the general case when each task has a distinct input or output data size, the problem may not be tractable.

B. Task Offloading Procedure

Before the task offloading procedure begins, the system needs to decide which EN or which set of ENs should each task (for binary offloading) or subtask (for partial offloading) be assigned to for execution. We denote by $W_{j,\Phi}$ the part of task j that is assigned exclusively to the set of ENs $\Phi \subseteq \mathcal{M}$ for computation with repetition order $|\Phi|$. Every task must be computed. Thus, for $\forall j \in \mathcal{N}$, we have $\bigcup_{\Phi \subseteq \mathcal{M}} W_{j,\Phi} = W_j$, and $W_{j,\Phi} \bigcap W_{j,\Psi} = \varnothing$ for $\forall \Psi \neq \Phi$. By such task assignment, the set of tasks or subtasks to be computed at each EN $i \in \mathcal{M}$ can be denoted as $\{W_{j,\Phi} : \forall \Phi \supseteq \{i\}, \forall j \in \mathcal{N}\}$.

Definition 1. *For a given task assignment scheme $\{W_{j,\Phi}\}$, the computation load r, $1 \leq r \leq M$, is defined as the total number of task input bits computed at all the M ENs, normalized by the total number of task input bits from all the N users, i.e.,*

$$r \triangleq \frac{\sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} \sum_{\Phi : i \in \Phi} |W_{j,\Phi}|}{NL}. \tag{1}$$

Similar to [27], the computation load r can be interpreted as the average number of ENs to compute each task (for binary offloading) or each input bit (for partial offloading) and hence is a measure of computation repetition.

Given a feasible task assignment strategy $\{W_{j,\Phi}\}$, the overall offloading procedure contains two communication phases, an input data uploading phase and an output data downloading phase.

1) Uploading phase: Each user j employs an encoding function to map its task inputs W_j and channel coefficients $\mathbf{H} \triangleq [h_{ij}]_{i \in \mathcal{M}, j \in \mathcal{N}}$ to a length-T^u codeword $\mathbf{X}_j \triangleq (X_j(t))_{t=1}^{T^u}$, where $X_j(t) \in \mathbb{C}$ is the transmitted symbol at time $t \in [T^u]$. Each codeword has an average power constraint of P_u, i.e., $\frac{1}{T^u}||\mathbf{X}_j|| \leq P_u$. Then, the received signal $Y_i(t) \in \mathbb{C}$ of each EN i at time $t \in [T^u]$ is given by

$$Y_i(t) = \sum_{j \in \mathcal{N}} h_{ij}(t) X_j(t) + Z_i(t), \quad \forall i \in \mathcal{M}, \tag{2}$$

where $Z_i(t) \sim \mathcal{CN}(0, 1)$ is the noise at EN i. Each EN i uses a decoding function to map received signals $(Y_i(t))_{t=1}^{T^u}$ and channel coefficients $\mathbf{H}$ to the estimate $\{\hat{W}_{j,\Phi} : \forall j \in \mathcal{N}, \forall \Phi \supseteq \{i\}\}$ of its assigned task inputs $\{W_{j,\Phi} : \forall j \in \mathcal{N}, \forall \Phi \supseteq \{i\}\}$. The error probability is given by

$$P_e^u = \max_{i \in \mathcal{M}} \ \mathbb{P}\left(\bigcup_{j \in \mathcal{N}, \Phi \supseteq \{i\}} \left\{\hat{W}_{j,\Phi} \neq W_{j,\Phi}\right\}\right). \tag{3}$$

2) Downloading phase: After receiving the assigned task input data and executing them at the server, each EN i obtains the output data of its assigned tasks, $\left\{\widetilde{W}_{j,\Phi} : \forall j \in \mathcal{N}, \forall \Phi \supseteq \{i\}\right\}$, and begins to transmit these computed results back to users. The computed results downloading is similar to the task uploading operation. Briefly, each EN maps the task outputs $\left\{\widetilde{W}_{j,\Phi} : \forall j \in \mathcal{N}, \forall \Phi \supseteq \{i\}\right\}$ and channel coefficients $\mathbf{G} \triangleq [g_{ji}]_{j \in \mathcal{N}, i \in \mathcal{M}}$ into a codeword of block length T^d over the downlink interference channel, with an average power constraint of P_d. Each user j decodes its desired task output data $\widetilde{W}_j$ from its received signals and obtains the estimate $\hat{\widetilde{W}}_j$. The error probability is given by

$$P_e^d = \max_{j \in \mathcal{N}} \ \mathbb{P}\left(\hat{\widetilde{W}}_j \neq \widetilde{W}_j\right). \tag{4}$$

A task offloading policy with computation load r, denoted as $\left(\{W_{j,\Phi}\}, (L, \widetilde{L}), r\right)$, consists of a sequence of task assignment schemes $\{W_{j,\Phi}\}$, task input uploading schemes with time T^u, and task output downloading schemes with time T^d, indexed by the task input and output data size pair $(L, \widetilde{L})$. It is said to be feasible when the error probabilities P_e^u and P_e^d approach to zero when $L \to \infty$ and $\widetilde{L} \to \infty$.

C. Performance Metric

We characterize the performance of the considered MEC network by the computation load r as well as the asymptotic communication time for task input uploading and output downloading.

Definition 2. *The normalized uploading time (NULT) and normalized downloading time (NDLT) for a given feasible task offloading policy with computation load r are defined, respectively, as*

$$\tau^u(r) \triangleq \lim_{P_u \to \infty} \lim_{L \to \infty} \frac{\mathbb{E}_{\mathbf{H}}[T^u]}{L/\log P_u}, \tag{5}$$

$$\tau^d(r) \triangleq \lim_{P_d \to \infty} \lim_{\widetilde{L} \to \infty} \frac{\mathbb{E}_{\mathbf{H}}[T^d]}{\widetilde{L}/\log P_d}. \tag{6}$$

Further, the minimum NULT and NDLT are defined, respectively, as

$$\tau^{u^*}(r) \triangleq \inf\{\tau^u(r) : \forall \tau^u(r) \text{ is achievable at the computation load } r\}, \tag{7}$$

$$\tau^{d^*}(r) \triangleq \inf\{\tau^d(r) : \forall \tau^d(r) \text{ is achievable at the computation load } r\}. \tag{8}$$

Note that $L/\log P_d$ (or $\widetilde{L}/\log P_d$) is the reference time to transmit the input (or output) data of L (or $\widetilde{L}$) bits for one task in a Gaussian point-to-point baseline system in the high SNR regime. Thus, an NULT (or NDLT) of $\tau^{u^*}(r)$ $\left(\text{or } \tau^{d^*}(r)\right)$ indicates that the time required to upload (or download) the tasks of all users is $\tau^{u^*}(r)$ $\left(\text{or } \tau^{d^*}(r)\right)$ times of this reference time period.

Definition 3. *A communication latency pair $(\tau^u(r), \tau^d(r))$ at a computation load r is said to be achievable if there exists a feasible task offloading policy $\left(\{W_{j,\Phi}\}, (L, \widetilde{L}), r\right)$. The optimal communication latency region is the closure of the set of all achievable communication latency pairs $\left\{(\tau^u(r), \tau^d(r)\right\}$ at all possible computation load r's, i.e.,*

$$\mathscr{T} \triangleq \text{closure}\left\{\left(\tau^u(r), \tau^d(r)\right) : \forall \left(\tau^u(r), \tau^d(r)\right) \text{ is achievable}, \ \forall r \in [1, M]\right\}. \tag{9}$$

Our goal is to characterize the optimal communication latency pair at any given computation load r for both binary offloading and partial offloading.

III. COMMUNICATION LATENCY ANALYSIS FOR BINARY OFFLOADING

In this section, we present the analysis of the optimal communication latency pair at any given computation load, including both achievable scheme and converse, for binary offloading.

A. Main Results

Theorem 1. *(Achievable result). An achievable communication latency pair $\left(\tau_a^u(r), \tau_a^d(r)\right)$ at an integer computation load $r \in [M]$, for binary task offloading in the MEC network with M ENs*
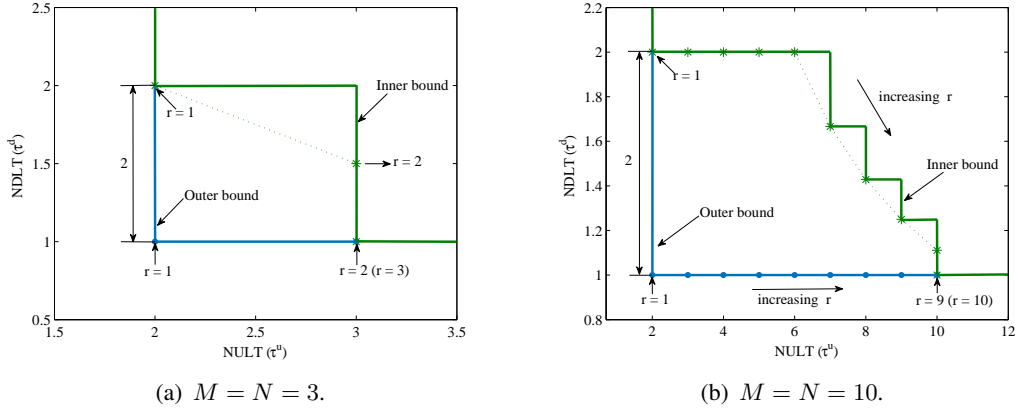
Fig. 2. The inner bound and outer bound of the optimal communication latency region for binary offloading. At a given integer computation load $r \in [M]$, the achievable NULT is optimal while the gap of NDLT is within 2.

and N users, is given by

$$\tau_a^u(r) = \min\left\{1 + \frac{Nr}{M}, N\right\}, \tag{10}$$

$$\tau_a^d(r) = \min\left\{1 + \frac{N}{M}, \frac{N}{r}\right\}, \tag{11}$$

when $\frac{N}{M} \in \mathbb{Z}^+$. If $\frac{N}{M}$ is not an integer, one can always find two integers δ_1 and δ_2 so that $\frac{N+\delta_1}{M-\delta_2}$ is the closest integer to $\frac{N}{M}$ and the above results still hold by adding δ_1 more users and deactivating δ_2 ENs.

We prove the achievability of Theorem 1 in Section III-B.

Theorem 2. *(Converse). The optimal communication latency pair $\left(\tau^{u^*}(r), \tau^{d^*}(r)\right)$ at any given computation load $r \in \left\{r: \sum_{i \in \mathcal{M}} a_i = Nr, a_i \in [0:N], \forall i \in \mathcal{M}\right\}$, for binary task offloading in the MEC network with M ENs and N users, is lower bounded by*

$$\tau^{u^*}(r) \geq \min\left\{1 + \frac{Nr}{M}, N\right\}, \tag{12}$$

$$\tau^{d^*}(r) \geq \frac{N}{\min\{M, N\}}. \tag{13}$$

Based on Theorem 1 and Theorem 2, we can obtain an inner bound denoted as $\mathscr{T}_{in}$ and an outer bound denoted as $\mathscr{T}_{out}$, respectively, of the optimal communication latency region by collecting the latency pairs $\left(\tau^u(r), \tau^d(r)\right)$ at all the considered computation loads r's. Fig. 2 shows the bounds in the MEC networks with $M = N \in \{3, 10\}$.

Corollary 1. *(Optimality). At an integer computation load $r \in [M]$, the achievable NULT in (10) is **optimal**, and the achievable NDLT in (11) is within a multiplicative gap of 2 to its minimum.*

The proof for Theorem 2 and Corollary 1 is given in Section III-C.

Now, we demonstrate how the computation load r affects the achievable communication latency $(\tau_a^u(r), \tau_a^d(r))$. By discussing the $\min$ function terms in (10) and (11), we have the monotonicity of the achievable computation-communication function $(\tau_a^u(r), \tau_a^d(r))$:

- The NULT $\tau_a^u(r)$ increases strictly with the computation load r for $1 \leq r \leq M - \frac{M}{N}$, and then keeps a constant N for $M - \frac{M}{N} \leq r \leq M$.

- The NDLT $\tau_a^d(r)$ keeps a constant $1+\frac{N}{M}$ for $1\leq r\leq\frac{MN}{M+N}$, and then is **inversely proportional** to the computation load r for $\frac{MN}{M+N}\leq r\leq M$.

Remark 1. *The achievable computation-communication function $(\tau_a^u(r),\tau_a^d(r))$ has two corner points $\left(N,\frac{N^2}{MN-M}\right)$ and $\left(1+\frac{N^2}{M+N},1+\frac{N}{M}\right)$, corresponding to $r=M-\frac{M}{N}$ and $r=\frac{MN}{M+N}$, respectively. They are explained as follows:*

- *For input data uploading, before r increases to $M-\frac{M}{N}$, the NULT is increasing since more traffic is introduced in the uplink. When r grows to more than $M-\frac{M}{N}$, there is no need to increase the NULT since all tasks can be uploaded within N time slots by using TDMA.*
- *For output data downloading, before r increases to $\frac{MN}{M+N}$, the potential transmission co-operation gain brought by computation replication cannot exceed the existing interference alignment gain without computation replication and thus the NDLT keeps fixed. When r grows to more than $\frac{MN}{M+N}$, interference neutralization can be exploited which outperforms interference alignment, and thus the NDLT begins to decrease with r.*

It can be easily proved that $M-\frac{M}{N}\geq\frac{MN}{M+N}$ for all $M,N\geq2$. Hence, we have the following remark to characterize the envelope of the inner bound of the optimal communication latency region, present the tradeoff between computation load and communication latency, and illustrate the interaction between the NULT and NDLT.

Remark 2. *The envelope of the inner bound $\mathcal{T}_{in}$ of the optimal communication latency region for binary offloading can be divided into three sections, each corresponding to a distinct interval of the computation load r:*

1) *Constant-NDLT section: $\tau^u(r)=1+\frac{Nr}{M}$, $\tau^d(r)=1+\frac{N}{M}$, when $1\leq r\leq\frac{MN}{M+N}$;*
2) *NULT-NDLT tradeoff section: $\tau^u(r)=1+\frac{Nr}{M}$, $\tau^d(r)=\frac{N}{r}$, when $\frac{MN}{M+N}\leq r\leq M-\frac{M}{N}$;*
3) *Constant-NULT section: $\tau^u(r)=N$, $\tau^d(r)=\frac{N}{r}$, when $M-\frac{M}{N}\leq r\leq M$.*

In particular, in the NULT-NDLT tradeoff section, as the computation load increases, the NDLT decreases in an inversely proportional way, at the expense of increasing the NULT linearly.

It is seen from Fig. 2(b) that the envelope of the inner bound is composed of three sections corresponding to three different intervals of the computation load, and the middle section at $5\leq r\leq9$ (dotted line) presents the NULT-NDLT tradeoff, in an inversely proportional form.

B. Achievable task offloading scheme

1) Task assignment and uploading: Consider that the system parameters M and N satisfy $\frac{N}{M}\in\mathbb{Z}^+$ such that $Nr=Mn$ holds for $\forall r\in[M]$, where r is the given integer computation load and n is an integer in $[N]$. In the proposed task assignment method, we let each task be executed at exactly r different ENs and let each EN execute n distinct tasks with even load. Note that if $\frac{N}{M}$ is not an integer, we can inject δ_1 (≥0) tasks and let δ_2 (≥0) ENs being idle and use the remaining ENs for task offloading, such that $\frac{N+\delta_1}{M-\delta_2}$ is the integer closest to $\frac{N}{M}$, denoted as n_1. In this way, we still have $(N+\delta_1)r=(M-\delta_2)rn_1$ for $\forall r\in[M-\delta_2]$, and can use the new $N+\delta_1$ and $M-\delta_2$ to replace N and M to obtain the corresponding analytical results.

To ensure even task assignment on each EN, we perform circular assignment. Specifically, the set of tasks assigned to EN $i\in\mathcal{M}$ is given by

$$\mathcal{T}_i = \big\{W_{j+1}: j\in[(i-1)n:(in-1)]\,(\mathrm{mod}\,N)\big\}. \tag{14}$$

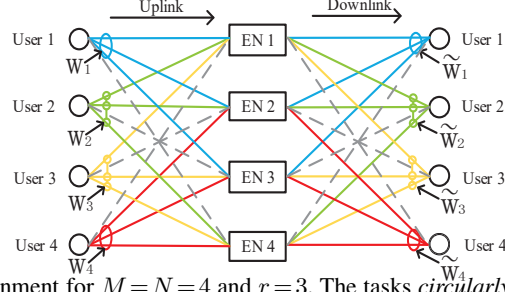An example of the task uploading for $M=N=4$ and $r=3$ is shown in Fig. 3.

Fig. 3. Illustration of even task assignment for $M = N = 4$ and $r = 3$. The tasks *circularly* assigned to 4 ENs are $\{W_1, W_2, W_3\}$, $\{W_4, W_1, W_2\}$, $\{W_3, W_4, W_1\}$, and $\{W_2, W_3, W_4\}$. The uplink channel is the circular interference-multicast channel whose per-receiver DoF is $\frac{3}{4}$ achieved via TDMA, while the downlink is the circular cooperative interference channel whose per-receiver DoF is $\frac{3}{4}$ achieved by interference neutralization. Solid circle denotes that the channels inside it carry the same information.

Given the above task assignment in (14), the uplink channel formed by uploading the N tasks to their corresponding ENs is referred to as *the circular interference-multicast channel with multicast group size* r. This channel is different from the X-multicast channel with multicast group size r defined in [30], [31], where any subset of r receivers can form a multicast group, resulting in $\binom{M}{r}$ multicast groups, and each transmitter needs to communicate with all the $\binom{M}{r}$ multicast groups. In our considered circular interference-multicast channel, there are only N multicast groups which are performed circularly by the M receivers and each transmitter only needs to communicate with one multicast group. The optimal per-receiver DoF of this uplink channel is given as follows.

Lemma 1. *The optimal per-receiver DoF of the circular interference-multicast channel with N transmitters and M receivers satisfying $\frac{N}{M} \in \mathbb{Z}^+$ and multicast group size r is given by*

$$DoF_r^u = \max\left\{ \frac{Nr}{Nr + M}, \frac{r}{M} \right\}, \ r \in [M]. \tag{15}$$

Proof. First, we use partial interference alignment scheme to achieve a DoF of $\frac{n}{n+1} = \frac{Nr}{Nr+M}$ for each receiver, which is similar to the achievability proof of the DoF of K-user interference channels in [23]. Then, we compare it to the DoF of $\frac{r}{M}$ achieved by TDMA. The detailed achievable scheme and proof of optimality are given in Appendix A. □

The per-receiver rate of this channel in the high SNR regime can be approximated as $DoF_r^u \times \log P_u + o(\log P_u)$. The traffic load for each EN to receive its assigned tasks is $\frac{Nr}{M}L$ bits, then the uploading time can be approximately given by $T^u = \frac{\frac{Nr}{M}L}{DoF_r^u \log P_u + o(\log P_u)}$. Let $P_u \to \infty$ and $L \to \infty$, by Definition 2, the NULT for each EN at computation load r can be given by

$$\tau_a^u(r) = \frac{\frac{Nr}{M}}{DoF_r^u} = \min\left\{ \frac{Nr}{M} + 1, N \right\}, \quad r \in [M]. \tag{16}$$

2) Results downloading: After computing all offloaded tasks, ENs begin to transmit the computed results back to users via downlink channels. Recall that each task is computed at r different ENs and each EN i has the computed results of n different tasks $\mathcal{T}_i$, as given in (14). Each user j wants the computed results $\widetilde{W}_j$ for $\forall j \in \mathcal{N}$, which is owned in r different ENs. Multiple ENs with the same computed results can exploit transmission cooperation to neutralize interferences across users [22], [32]. The computation results downloading for $M = N = 4$ and $r = 3$ is shown in Fig. 3. We refer to the downlink channel formed by downloading the N tasks as *the circular cooperative interference channel with transmitter cooperation group size* r. This channel is different from the cooperative X channel with transmitter cooperation group

size r defined in [30], [33], where any subset of r transmitters can form a cooperation group, resulting in $\binom{M}{r}$ groups in total, and each transmitter cooperation group has messages to send to all receivers. In our considered downlink channel, there are only N cooperation groups which are performed circularly by the M transmitters and each group only needs to communicate with one receiver. An achievable per-receiver DoF of this downlink channel is given as below.

Lemma 2. *An achievable per-receiver DoF of the circular cooperative interference channel with M transmitters and N receivers satisfying $\frac{N}{M} \in \mathbb{Z}^+$ and transmitter cooperation group size r is given by*

$$DoF_r^d = \max\left\{\frac{M}{N+M}, \frac{r}{N}\right\}, \quad r \in [M], \tag{17}$$

and it is within a multiplicative gap of 2 to the optimal DoF.

Proof. When $r = 1$, we use partial interference alignment scheme to achieve a DoF of $\frac{M}{N+M}$ for each receiver. The achievable scheme is similar to that for the $M \times N$ user X channel [34]. We then compare it to a DoF of $\frac{1}{N}$ achieved via TDMA. When $r \geq 2$, we prove that the achievable per-receiver DoF is $\max\{\frac{M}{N+M}, \frac{r}{N}\}$, where we first use interference neutralization to achieve a DoF of $\frac{r}{N}$ for each receiver, and then compare it with the per-receiver DoF of $\frac{NM}{N+M}$ achieved by only using interference alignment. Summarizing these two cases, we have (17). Please refer to Appendix B for the detailed achievable scheme and optimality proof. □

The per-receiver channel rate in the high SNR regime can be approximated as $DoF_r^d \times \log P_d + o(\log P_d)$. The traffic load for each user to download its task output data is $\widetilde{L}$ bits, then the downloading time can be approximately given by $T^d = \frac{\widetilde{L}}{DoF_r^d \log P_d + o(\log P_d)}$. Let $P_d \to \infty$ and $\widetilde{L} \to \infty$, by Definition 2, the NDLT for each user at computation load r is given by

$$\tau_a^d(r) = \frac{1}{DoF_r^d} = \min\left\{\frac{N}{M} + 1, \frac{N}{r}\right\}, \quad r \in [M]. \tag{18}$$

By (16) and (18), we thus have the achievable communication latency pair $(\tau_a^u(r), \tau_a^d(r))$ at an integer computation load $r \in [M]$ for binary offloading.

C. Proof of Converse

1) Lower bound and optimality of NULT: We prove the lower bound of the NULT at any given computation load $r \in \left\{r : \sum_{i=1}^M a_i = Nr, a_i \in [0:N], \forall i \in \mathcal{M}\right\}$, i.e., $\tau^{u^*}(r) \geq \min\{\frac{Nr}{M} + 1, N\}$. First, we use genie-aided arguments to derive a lower bound on the NULT of any given feasible task assignment policy with computation load r. Then, we optimize the lower bound over all feasible task assignment policies to obtain the minimum NULT for a given computation load r.

Given a computation load r. Consider an arbitrary task assignment policy where the number of tasks assigned to each EN i is denoted as a_i, $\forall i \in \mathcal{M}$, and satisfies

$$\sum_{i \in \mathcal{M}} a_i = Nr, \tag{19}$$

$$a_i \in [0:N], \quad i \in \mathcal{M}. \tag{20}$$

Note that we only need consider $a_i > 0$ case since $a_i = 0$ means no task is assigned to EN i and we can remove EN i from the EN set $\mathcal{M}$, which will not change the results. Consider the following three disjoint subsets of task input data (or message):

$$\mathcal{W}_r = \{W_{j,\mathcal{S}_j} : j \in \mathcal{N}, i \in \mathcal{S}_j\}, \tag{21}$$

$$\mathcal{W}_t = \{W_{j,\mathcal{S}_j} : j = t_o, i \notin \mathcal{S}_j\}, \tag{22}$$

$$\overline{\mathcal{W}} = \{W_{j,\mathcal{S}_j} : j \neq t_o \text{ and } i \notin \mathcal{S}_j\}, \tag{23}$$

where $W_{j,\mathcal{S}_j}$ denotes the input message of task j that is assigned to all ENs in subset $\mathcal{S}_j$, and t_o denotes one of the users that do not offload their tasks to EN i, i.e., $\mathcal{W}_r \cap \mathcal{W}_t = \varnothing$. It is seen that the set $\mathcal{W}_r$ indicates the messages that EN i need decode, i.e., $|\mathcal{W}_r| = a_i$; The set $\mathcal{W}_t$ is a nonempty set with cardinality $|\mathcal{W}_t| = 1$ when EN i is not assigned all N tasks (or $a_i < N$), since user t_o exists in this case; Otherwise, we have $\mathcal{W}_t = \varnothing$ for $a_i = N$. We will show that set $\mathcal{W}_r \cup \mathcal{W}_t$ has the maximum number of messages that can be decoded by EN i.

Let a genie provide the messages $\overline{\mathcal{W}}$ to all ENs, and additionally provide messages $\mathcal{W}_r$ to ENs in $\mathcal{M}/\{i\}$. The received signal of EN i can be represented as

$$\hat{\mathbf{y}}_i = \sum_{j=1,\neq t_o}^{M} \mathbf{H}_{ij}\mathbf{x}_j + \mathbf{H}_{it_o}\mathbf{x}_{t_o} + \hat{\mathbf{z}}_i, \tag{24}$$

where $\mathbf{H}_{ij}$, $\mathbf{x}_j$, $\mathbf{z}_i$ are diagonal matrices representing the channel coefficients from user j to EN i, signal transmitted by user j, noise received at EN i, over the block length T^u, respectively. Note that we reduce the noise at EN i from $\mathbf{z}_i$ to $\hat{\mathbf{z}}_i$ by a fixed amount such that its received signal $\mathbf{y}_i$ can be replaced by $\hat{\mathbf{y}}_i$. The ENs in $\mathcal{M}/\{i\}$ have messages $\overline{\mathcal{W}} \cup \mathcal{W}_r$, which do not include the message of user t_o. Using these genie-aided information, each EN $k \in \mathcal{M}/\{i\}$ can compute the transmitted signals $\{\mathbf{x}_j : j \neq t_o\}$ and subtract them from the received signal. Thus, the received signal of EN $k \neq i$ can be rewritten as

$$\bar{\mathbf{y}}_k = \mathbf{y}_k - \sum_{j=1,\neq t_o}^{N} \mathbf{H}_{kj}\mathbf{x}_j = \mathbf{H}_{kt_o}\mathbf{x}_{t_o} + \mathbf{z}_k. \tag{25}$$

Since the message $\mathcal{W}_t$ is intended for some ENs in $\mathcal{M}/\{i\}$, denoted as $\mathcal{R}_t$, the ENs in $\mathcal{R}_t$ can decode it. By Fano's inequality and (25), we have

$$H(\mathcal{W}_t|\mathbf{y}_k, \overline{\mathcal{W}}, \mathcal{W}_r) \leq T^u\epsilon, \quad k \in \mathcal{R}_t. \tag{26}$$

Consider EN i, it can decode messages $\mathcal{W}_r$ intended for it. By Fano's inequality, we have

$$H(\mathcal{W}_r|\hat{\mathbf{y}}_i, \overline{\mathcal{W}}) \leq |\mathcal{W}_r|T^u\epsilon. \tag{27}$$

Using genie-aided messages $\overline{\mathcal{W}}$ and decoded messages $\mathcal{W}_r$, EN i can compute the transmitted signals $\{\mathbf{x}_j : j \neq t_o\}$, and subtract them from the received signal. We thus have

$$\bar{\mathbf{y}}_i = \hat{\mathbf{y}}_i - \sum_{j=1,\neq t_o}^{N} \mathbf{H}_{ij}\mathbf{x}_j = \mathbf{H}_{it_o}\mathbf{x}_{t_o} + \hat{\mathbf{z}}_i. \tag{28}$$

By reducing noise and multiplying the constructed signal $\bar{\mathbf{y}}_i$ at EN i by $\mathbf{H}_{kt_o}\mathbf{H}_{it_o}^{-1}$, we have

$$\bar{\mathbf{y}}_i^k = \mathbf{H}_{kt_o}\mathbf{H}_{it_o}^{-1}\bar{\mathbf{y}}_i = \mathbf{H}_{kt_o}\mathbf{x}_{t_o} + \hat{\mathbf{z}}_k', \tag{29}$$

where $\hat{\mathbf{z}}'_k$ represents the reduced noise. It is seen that $\bar{\mathbf{y}}^k_i$ is a degraded version of $\bar{\mathbf{y}}_k$ at EN k in $\mathcal{R}_t$, so EN i must be able to decode the messages that ENs in $\mathcal{R}_t$ can decode. Thus, we have

$$H(\mathcal{W}_t|\hat{\mathbf{y}}_i, \overline{\mathcal{W}}, \mathcal{W}_r) \leq H(\mathcal{W}_t|\mathbf{y}_k, \overline{\mathcal{W}}, \mathcal{W}_r) \leq T^u\epsilon, \quad i \in \mathcal{R}_t. \tag{30}$$

All the above changes including genie-aided information, receiver cooperation, and noise reducing can only improve capacity. Therefore, we have the following chain of inequalities,

$$(|\mathcal{W}_r| + |\mathcal{W}_t|)L = H(\mathcal{W}_r, \mathcal{W}_t) \tag{31}$$

$$\overset{(a)}{=} H(\mathcal{W}_r, \mathcal{W}_t|\overline{\mathcal{W}}) \tag{32}$$

$$\overset{(b)}{=} I(\mathcal{W}_r, \mathcal{W}_t : \hat{\mathbf{y}}_i|\overline{\mathcal{W}}) + H(\mathcal{W}_r, \mathcal{W}_t|\hat{\mathbf{y}}_i, \overline{\mathcal{W}}) \tag{33}$$

$$\overset{(c)}{=} I(\mathcal{W}_r, \mathcal{W}_t : \hat{\mathbf{y}}_i|\overline{\mathcal{W}}) + H(\mathcal{W}_r|\hat{\mathbf{y}}_i, \overline{\mathcal{W}}) + H(\mathcal{W}_t|\hat{\mathbf{y}}_i, \mathcal{W}_r, \overline{\mathcal{W}}) \tag{34}$$

$$\overset{(d)}{\leq} I(\mathcal{W}_r, \mathcal{W}_t : \hat{\mathbf{y}}_i|\overline{\mathcal{W}}) + |\mathcal{W}_r|T^u\epsilon + T^u\epsilon \tag{35}$$

$$\overset{(e)}{\leq} I(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{a_i}, \mathbf{x}_{t_o} : \hat{\mathbf{y}}_i|\overline{\mathcal{W}}) + (|\mathcal{W}_r| + 1)T^u\epsilon \tag{36}$$

$$\overset{(f)}{\leq} T^u \log P_u + (|\mathcal{W}_r| + 1)T^u\epsilon, \tag{37}$$

where (a) is due to the independence of messages, (b) and (c) follow from the chain rule, (d) uses Fano's inequalities (27) and (30), (e) is the data processing inequality, and (f) uses the DoF bound of the MAC channel. By dividing on $\frac{L}{\log P_u}$, and taking $P_u \to \infty$ and $\epsilon \to 0$, we have $\tau^u \geq |\mathcal{W}_r| + |\mathcal{W}_t| = \min\{a_i + 1, N\}$.

Thus, for any given feasible task assignment $\mathbf{a} \triangleq [a_i]_{i \in \mathcal{M}}$, the NULT satisfies $\tau^u \geq \min\{a_i+1, N\}$ for $\forall i \in \mathcal{M}$, i.e., the minimum NULT of the task assignment policy $\mathbf{a}$ is lower bounded by

$$\tau^{u^*}(r, \mathbf{a}) \geq \max_{i \in \mathcal{M}} \min\{a_i + 1, N\} = \min\left\{\max_{i \in \mathcal{M}} a_i + 1, N\right\}. \tag{38}$$

Hence, the minimum NULT of all feasible task assignment is given by $\tau^{u^*}(r) = \min_{\mathbf{a}} \tau^{u^*}(r, \mathbf{a})$. It can be lower bounded by the optimal solution of the following linear programming problem,

$$\text{P1}: \quad \min_{\mathbf{a}} \quad \min\left\{\max_{i \in \mathcal{M}} a_i + 1, N\right\}$$

$$s.t. \quad (19), (20)$$

By relaxing the integer constraint $a_i \in [0 : N]$ into a real-value constraint $0 \leq a_i \leq N$, the optimal solution is still a lower bound of the minimum NULT $\tau^{u^*}(r)$. Since the objective is equivalent to minimizing the term $\max_{\forall i \in \mathcal{M}} a_i$, the optimal solution can be obtained easily as $a_i^* = \frac{Nr}{M}, \forall i \in \mathcal{M}$. Hence, the minimum NULT is lower bounded by

$$\tau^{u^*}(r) \geq \min\left\{\frac{Nr}{M} + 1, N\right\}. \tag{39}$$

The proof of the lower bound of NULT is thus completed. Comparing (39) with (10) in Theorem 1, we see that they are the same. Thus, the achievable NULT in (10) is optimal.

2) Lower bound and gap of NDLT: Let $\mathbf{x_i}$ denote the signal transmitted by each EN i, and $\mathbf{y_j}$ the signal received at each user j, over the block length T^d. Consider the N computed results

decoded by N users, we have the following chain of inequalities,

$$N\widetilde{L} = H(\widetilde{W}_1, \cdots, \widetilde{W}_N) = I(\widetilde{W}_1, \cdots, \widetilde{W}_N : \mathbf{y}_1, \cdots, \mathbf{y}_N) + H(\widetilde{W}_1, \cdots, \widetilde{W}_N | \mathbf{y}_1, \cdots, \mathbf{y}_N) \quad (40)$$

$$\overset{(g)}{\leq} I(\widetilde{W}_1, \cdots, \widetilde{W}_N : \mathbf{y}_1, \cdots, \mathbf{y}_N) + \sum_{j \in \mathcal{N}} H(\widetilde{W}_j | \mathbf{y}_j) \quad (41)$$

$$\overset{(h)}{\leq} I(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M : \mathbf{y}_1, \cdots, \mathbf{y}_N) + NT^d \epsilon \quad (42)$$

$$\overset{(i)}{\leq} \min\{M, N\} T^d \log P_d + NT^d \epsilon, \quad (43)$$

where (g) follows from $H(\widetilde{W}_1, \cdots, \widetilde{W}_N | \mathbf{y}_1, \cdots, \mathbf{y}_N) \leq \sum_{j \in \mathcal{N}} H(\widetilde{W}_j | \mathbf{y}_1, \cdots, \mathbf{y}_N) \leq \sum_{j \in \mathcal{N}} H(\widetilde{W}_j | \mathbf{y}_j)$, (h) follows from the data processing inequality and Fano's inequality, and (i) uses the capacity bound of the MISO broadcast channel with a M-antenna transmitter and N single-antenna receivers. By dividing on $\frac{\widetilde{L}}{\log P_d}$, and taking $P_d \to \infty$ and $\epsilon \to 0$, we have

$$\tau^d \geq \frac{N}{\min\{M, N\}}. \quad (44)$$

Hence, the minimum NDLT is lower bounded by $\tau^{d*} \geq \frac{N}{\min\{M,N\}}$. It can be easily proved that the multiplicative gap between the achievable NDLT $\tau_a^d(r)$ in Theorem 1 and this lower bound is within 2 for $\frac{N}{M} \in \mathbb{Z}^+$, i.e., $\frac{\tau_a^d(r)}{\tau^{d*}(r)} \leq \frac{\min\{\frac{N}{M}+1, \frac{N}{r}\}}{N/M} \leq 2$. We complete the proof of the lower bound and gap of the NDLT for binary offloading.

IV. COMMUNICATION LATENCY ANALYSIS FOR PARTIAL OFFLOADING

In this section, we present the analysis of the optimal communication latency pair at any given computation load, including achievable scheme and converse, for partial offloading.

A. Main Results

Theorem 3. *(Achievable result). An achievable communication latency pair $(\tau_a^u(r), \tau_a^d(r))$ at an integer computation load $r \in [M]$, for partial task offloading in the MEC network with M ENs and N users, is given by*

$$\tau_a^u(r) = \frac{N-1}{M} r + 1, \quad (45)$$

$$\tau_a^d(r) = \max\left\{ \frac{N-r}{M} + 1, 1 \right\}. \quad (46)$$

For general $1 \leq r \leq M$, the achievable communication latency pair is given by the lower convex envelope of the above points $\left\{ \left(\tau_a^u(r), \tau_a^d(r) \right) : r \in [M] \right\}$.

The achievable task offloading scheme for Theorem 3 is given in Section IV-B.
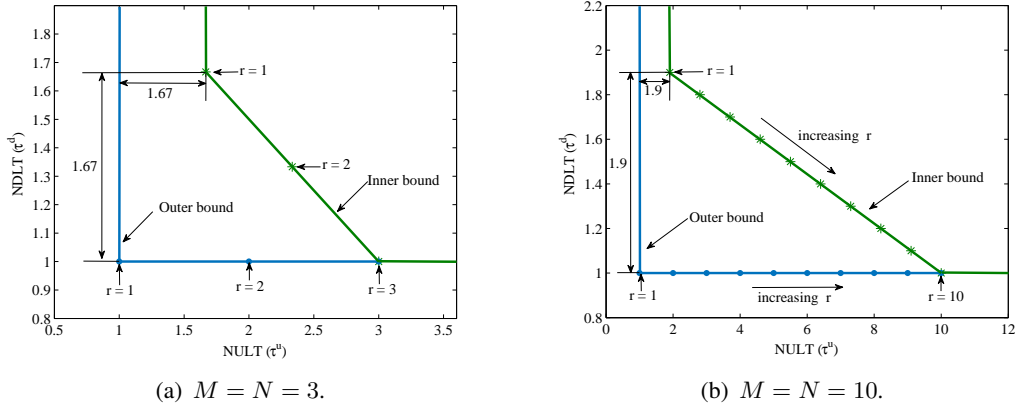
(a) $M = N = 3$.

(b) $M = N = 10$.

Fig. 4. The inner bound and outer bound of the optimal communication latency region for partial offloading. At a given computation load $r \in [1, M]$, both the gaps of NULT and NDLT are within 2.

Theorem 4. *(Converse). The optimal communication latency pair $\left(\tau^{u^*}(r), \tau^{d^*}(r)\right)$ at any given computation load $r \in [1, M]$, for partial task offloading in the MEC network with M ENs and N users, is lower bounded by*

$$\tau^{u^*}(r) \geq \max\left\{\frac{Nr}{M}, 1\right\}, \tag{47}$$

$$\tau^{d^*}(r) \geq \frac{N}{\min\{M, N\}}. \tag{48}$$

According to Theorem 3 and Theorem 4, we can obtain an inner bound $\mathcal{T}_{in}$ and an outer bound $\mathcal{T}_{out}$, respectively, of the optimal communication latency region by collecting the latency pairs $(\tau^u(r), \tau^d(r))$ at all the computation loads r's. Fig. 4 presents the bounds in the MEC networks with $M = N \in \{3, 10\}$.

Corollary 2. *(Optimality). At a computation load $r \in [1, M]$, both the achievable NULT in (45) and NDLT in (46) are within multiplicative gaps of 2 to their lower bounds.*

The proof for Theorem 4 and Corollary 2 is given in Section IV-C.

Observing (45) and (46), we have the monotonicity of the computation-communication function $\left(\tau_a^u(r), \tau_a^d(r)\right)$:

- The NULT $\tau_a^u(r)$ increases with the computation load r strictly for $1 \leq r \leq M$.
- The NDLT $\tau_a^d(r)$ **decreases linearly** with the computation load r for $1 \leq r \leq \min\{M, N\}$, and keeps a constant 1 for $\min\{M, N\} \leq r \leq M$.

Then, we have the following remark to characterize the inner bound of the optimal communication latency region, present the computation-communication tradeoff, and show the interaction between NULT and NDLT.

Remark 3. *The inner bound $\mathcal{T}_{in}$ of the optimal communication latency region of partial task offloading is composed of two sections corresponding to two different intervals of the computation load r:*

1) *NULT-NDLT tradeoff section: $\tau^u(r) = \frac{N-1}{M}r + 1$, $\tau^d(r) = \frac{N-r}{M} + 1$, when $1 \leq r \leq \min\{M, N\}$;*
2) *Constant-NDLT section: $\tau^u(r) = \frac{N-1}{M}r + 1$, $\tau^d(r) = 1$, when $\min\{M, N\} \leq r \leq M$.*

In particular, in the NULT-NDLT tradeoff section, the NDLT decreases linearly with the computation load, at the expense of increasing the NULT linearly.

It is clearly seen from Fig. 4 that in the $M = N$ case, a linear tradeoff between NDLT and NULT is established when the computation load increases within $[1, M]$.

B. Achievable task offloading scheme

1) Task partition and uploading: All tasks are treated equally in this work without taking user priority into account. We thus focus on the task partition of W_j without loss of generality, for $\forall j \in \mathcal{N}$. We partition W_j into $\binom{M}{r}$ equal-sized subtasks, each denoted as $W_{j,\Phi}$ and to be computed at the ENs in subset $\Phi \subseteq \mathcal{M}$, with *repetition order* $|\Phi| = r$. Based on Definition 1, the computation load is calculated as $\dfrac{NM\binom{M-1}{r-1}\frac{L}{\binom{M}{r}}}{NL} = r$, which equals the *repetition order* of subtasks. The output data size of each subtask is assumed to be proportional to its input data size.

According to the above task partition scheme, each user uploads the subtasks to the assigned ENs via uplink channels. Specifically, each EN i wants subtasks $\{W_{j,\Phi} : \forall j \in \mathcal{N}, \forall \Phi \supseteq \{i\}, |\Phi| = r\}$. There are a total of $N\binom{M}{r}$ subtasks in this transmission at the computation load r, and each EN desires $N\binom{M-1}{r-1}$ subtasks of them. Clearly, the channel formed by uploading the subtasks at the computation load r is the *X-multicast channel with multicast group size* r, defined in [30], [31]. The optimal per-receiver DoF of this channel is given below via interference alignment.

Corollary 3. *([30, Lemma 1], [31, Theorem 2]) The optimal per-receiver DoF of the X-multicast channel with N transmitters, M receivers, and multicast group size r is given by*

$$DoF_r^u = \frac{Nr}{Nr + M - r}, \quad r \in [M]. \tag{49}$$

The traffic load for each EN to receive its assigned subtasks is $N\binom{M-1}{r-1}\frac{L}{\binom{M}{r}} = \frac{Nr}{M}L$ bits. Similar to (16) in binary offloading, the NULT for each EN at computation load r is given by

$$\tau_a^u(r) = \frac{\frac{Nr}{M}}{DoF_r^u} = \frac{N-1}{M}r + 1. \tag{50}$$

2) Results downloading: After computing all the offloaded subtasks, ENs begin to transmit the computed results back to users via downlink channels. In specific, each user j wants the results of subtasks $\{W_{j,\Phi} : \forall \Phi \subseteq \mathcal{M}, |\Phi| = r\}$. There are a total of $N\binom{M}{r}$ subtasks in this transmission at the computation load r, of which $\binom{M}{r}$ subtasks are desired by each user j. The duplication of computed results on multiple ENs can enable the transmitter cooperation to be exploited in results downloading. The downlink channel formed by downloading the computed results at the computation load r is the *cooperative X channel with transmitter cooperation group size r* defined in [30], [33]. An achievable per-receiver DoF of this channel is given below.

Corollary 4. *([33, Theorem 1]) An achievable per-receiver DoF of the cooperative X channel with M transmitters, N receivers, and transmitter cooperation group size r is given by*

$$DoF_r^d = \min\left\{\frac{M}{M + N - r}, 1\right\}, \quad r \in [M], \tag{51}$$

and it is within a multiplicative gap of 2 to the optimal DoF.

The traffic load for each user to download its task output data is $\widetilde{L}$ bits. Similar to (18) in

binary offloading, the NDLT for each user at computation load r is given by

$$\tau_a^d(r) = \frac{1}{DoF_r^d} = \max\left\{\frac{N-r}{M}+1, 1\right\}, \quad r \in [M]. \tag{52}$$

By (50) and (52), we thus obtain an achievable communication latency pair $\left(\tau_a^u(r), \tau_a^d(r)\right)$ at an integer computation load $r \in [M]$ for partial offloading.

3) Non-integer computation load: Based on the above scheme, given an integer computation load $r \in [M]$, the achievable NULT $\tau_a^u(r)$ and NDLT $\tau_a^d(r)$ are given by (50) and (52), respectively. If r is a non-integer value, it can be rewritten as as a convex combination of $\lceil r \rceil$ and $\lfloor r \rfloor$, i.e., $r = \lambda\lceil r\rceil + (1-\lambda)\lfloor r\rfloor$ for some $\lambda \in [0,1]$. Then, we can partition the input data of each task into two parts with ratio λ and $1-\lambda$, respectively. Then, we apply the achievable schemes with computation load $\lceil r \rceil$ and $\lfloor r \rfloor$ to offload these two parts, respectively, via a time-sharing way. The corresponding communication latency pair is given by $\tau_a^u(r) = \lambda\tau_a^u(\lceil r\rceil) + (1-\lambda)\tau_a^u(\lfloor r\rfloor)$, and $\tau_a^d(r) = \lambda\tau_a^d(\lceil r\rceil) + (1-\lambda)\tau_a^d(\lfloor r\rfloor)$, i.e., $\left(\tau_a^u(r), \tau_a^d(r)\right) = \lambda\left(\tau_a^u(\lceil r\rceil), \tau_a^d(\lceil r\rceil)\right) + (1-\lambda)\left(\tau_a^u(\lfloor r\rfloor), \tau_a^d(\lfloor r\rfloor)\right)$. Actually, for any two integer-valued computation loads r_1 and r_2, the points on the line segment between $(\tau_a^u(r_1), \tau_a^d(r_1))$ and $(\tau_a^u(r_2), \tau_a^d(r_2))$ can be achieved via data partition and time sharing. Thus, the lower convex envelope of the achievable points $\left\{\left(\tau_a^u(r), \tau_a^d(r)\right): r \in [M]\right\}$ is also achievable.

C. Proof of Converse

1) Lower bound and gap of NULT: This proof is similar to the lower bound proof of the NULT for binary offloading. First, we derive an lower bound on the NULT of any given feasible subtask assignment policy with computation load $r \in [1, M]$. Then, we construct an optimization problem to obtain the lower bound of the NULT for all feasible subtask assignment policies.

Given a computation load $r \in [1, M]$. Consider an arbitrary feasible subtask assignment policy $\{W_{j,\Phi}\}$, where the total subtasks partitioned from task W_j and assigned to EN i are denoted as $\mathcal{W}_{j,i} \triangleq \{W_{j,\Phi}: \forall\Phi \supseteq \{i\}\}$ for $\forall i \in \mathcal{M}$ and $\forall j \in \mathcal{N}$, and the size of $\mathcal{W}_{j,i}$ is assumed to be $\gamma_{j,i}L$, and $\gamma_{j,i}$ satisfies

$$\sum_{i\in\mathcal{M}} \gamma_{j,i} = r, \quad \forall j \in \mathcal{N} \tag{53}$$

$$0 \le \gamma_{j,i} \le 1, \quad \forall j \in \mathcal{N}, \ \forall i \in \mathcal{M}. \tag{54}$$

Let $\mathbf{y}_i$ denote the signal received at each EN i, and $\mathbf{x}_j$ the signal transmitted by each user j, over the block length T^u. Hence, for any EN $i \in \mathcal{M}$, we have the following chain of inequalities,

$$\sum_{j\in\mathcal{N}} \gamma_{j,i}L = H(\mathcal{W}_{1,i}, \cdots, \mathcal{W}_{N,i}) \tag{55}$$

$$= I(\mathcal{W}_{1,i}, \cdots, \mathcal{W}_{N,i} : \mathbf{y}_i) + H(\mathcal{W}_{1,i}, \cdots, \mathcal{W}_{N,i}|\mathbf{y}_i) \tag{56}$$

$$\le I(\mathcal{W}_{1,i}, \cdots, \mathcal{W}_{N,i} : \mathbf{y}_i) + \sum_{j\in\mathcal{N}} H(\mathcal{W}_{j,i}|\mathbf{y}_i) \tag{57}$$

$$\le I(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N : \mathbf{y}_i) + NT^u\epsilon \tag{58}$$

$$\le T^u \log P_u + NT^u\epsilon. \tag{59}$$

By dividing on $\frac{L}{\log P_u}$, and taking $P_u \to \infty$ and $\epsilon \to 0$, we have $\tau^u \ge \sum_{j\in\mathcal{N}} \gamma_{j,i}$.

On the other hand, for any user $j \in \mathcal{N}$ and its task input data $W_j = \cup_{i \in \mathcal{M}} \mathcal{W}_{j,i}$, we also have

$$L = H(W_j) = H(\mathcal{W}_{j,1}, \cdots, \mathcal{W}_{j,M}) \tag{60}$$

$$= I(\mathcal{W}_{j,1}, \cdots, \mathcal{W}_{j,M} : \mathbf{y}_1, \cdots, \mathbf{y}_M) + H(\mathcal{W}_{j,1}, \cdots, \mathcal{W}_{j,M} | \mathbf{y}_1, \cdots, \mathbf{y}_M) \tag{61}$$

$$\leq I(\mathbf{x}_j : \mathbf{y}_1, \cdots, \mathbf{y}_M) + \sum_{i \in \mathcal{M}} H(\mathcal{W}_{j,i} | \mathbf{y}_i) \tag{62}$$

$$\leq T^u \log P_u + M T^u \epsilon. \tag{63}$$

By dividing on $\frac{L}{\log P_u}$, and taking $P_u \to \infty$ and $\epsilon \to 0$, we have $\tau^u \geq 1$. Combining these two final inequalities of τ^u, for any given feasible subtask assignment $\boldsymbol{\gamma} \triangleq [\gamma_{j,i}]_{i \in \mathcal{M}, \forall j \in \mathcal{N}}$, the NULT satisfies $\tau^u \geq \max \left\{ \sum_{j \in \mathcal{N}} \gamma_{j,i}, 1 \right\}$ for $\forall i \in \mathcal{M}$, i.e., the minimum NULT of the subtask assignment policy $\boldsymbol{\gamma}$ is lower bounded by

$$\tau^{u^*}(r, \boldsymbol{\gamma}) \geq \max_{i \in \mathcal{M}} \max \left\{ \sum_{j \in \mathcal{N}} \gamma_{j,i}, 1 \right\} = \max \left\{ \max_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} \gamma_{j,i}, 1 \right\}. \tag{64}$$

Hence, the minimum NULT of all subtask assignment policies is given by $\tau^{u^*}(r) = \min_{\boldsymbol{\gamma}} \tau^{u^*}(r, \boldsymbol{\gamma})$, and can be lower bounded by the optimal solution of the following linear programming problem,

$$\text{P2}: \quad \min_{\boldsymbol{\gamma}} \ \max \left\{ \max_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} \gamma_{j,i}, 1 \right\}$$

$$s.t. \quad (53), (54)$$

By defining a new variable $\mu_i = \sum_{j \in \mathcal{N}} \gamma_{j,i}$, Problem P2 can be transformed into P3 below,

$$\text{P3}: \quad \min_{\boldsymbol{\mu}} \ \max \left\{ \max_{i \in \mathcal{M}} \mu_i, 1 \right\}$$

$$s.t. \quad \sum_{i \in \mathcal{M}} \mu_i = Nr, \tag{65}$$

$$0 \leq \mu_i \leq N, \quad \forall i \in \mathcal{M}. \tag{66}$$

For any given feasible solution $\{\mu_i\}$ to P3, we can also construct a feasible solution to P2 by let $\gamma_{j,i} = \frac{\mu_i}{N}$ for $\forall j \in \mathcal{N}$, $\forall i \in \mathcal{M}$. Thus, Problem P2 and P3 are equivalent. It can be easily proved that the optimal solution to P3 is given by $\mu_i^* = \frac{Nr}{M}$, and consequently, the optimal solution to P2 is $\gamma_{j,i}^* = \frac{r}{M}$. Therefore, the minimum NULT $\tau^{u^*}(r)$ is lower bounded by

$$\tau^{u^*}(r) \geq \max \left\{ \frac{Nr}{M}, 1 \right\}. \tag{67}$$

Comparing it with the achievable NULT in (45) of Theorem 3, it can be easily proved that the multiplicative gap between them is within 2, i.e., $\frac{\tau_a^u(r)}{\tau^{u^*}(r)} \leq \frac{\frac{N-1}{M}r+1}{\max\{Nr/M,1\}} \leq 2$. Thus, we complete the proof of the lower bound and gap of the NULT for partial offloading.

2) Lower bound and gap of NDLT: Since the downlink channel capacity in this problem cannot exceed the capacity of the N user MISO broadcast channel with a single M-antenna transmitter, so the NDLT can be lower bounded by

$$\tau^{d^*} \geq \frac{N}{\min\{M, N\}}. \tag{68}$$

The rigorous proof is the same as the lower bound proof of NDLT for binary offloading in Section III-C2. The multiplicative gap between the achievable NDLT in (46) of Theorem 3 and the lower bound in (68) is within 2, i.e., $\frac{\tau_a^d(r)}{\tau^{d*}(r)} \leq \frac{\max\left\{\frac{N-r}{M}+1,1\right\}}{N/\min\{M,N\}} \leq 2$.

V. Conclusion

This paper studies a fundamental tradeoff between computation load r and communication latency $\left(\tau^u(r), \tau^d(r)\right)$ defined as the pair of normalized uploading time (NULT) and normalized downloading time (NDLT), in the MEC network with M ENs and N users. We exploit the idea of computation replication in task offloading schemes to speed up the computed result downloading via transmission cooperation, in binary and partial offloading cases. We develop an order-optimal achievable communication latency pair at a given computation load, and both the NULT and NDLT are within multiplicative gaps of 2 to their lower bounds. Particularly, the NULT in binary offloading is optimal. We show that the NDLT can be traded by the computation load r in the specific interval. It is an ***inversely proportional function*** for $\frac{MN}{M+N} \leq r \leq M$ in binary offloading, and a ***linear decreasing function*** for $1 \leq r \leq \min\{M, N\}$ in partial offloading, both of which decrease at the expense of increasing the NULT linearly. Hence, computation replication is very beneficial in reducing the communication latency for offloading tasks whose output data size is larger than the input data size. Our results reveal a fundamental relationship between computation load and communication latency in MEC systems.

Appendix

A. Proof of Lemma 1

1) Achievable scheme: We use the partial interference alignment scheme with a $u_s = ns^\Gamma + (s+1)^\Gamma$ symbol extension over the original channel, where $s \in \mathbb{N}$ and $\Gamma = M(N-n)$. Specifically, each transmitter j encodes the task input message W_j into s^Γ independent streams x_j^l, $l \in [s^\Gamma]$, each beamformed along a $u_s \times 1$ column vector $\mathbf{v}_j^l$. So the symbol $\bar{\mathbf{X}}_j$ transmitted at transmitter j can be expressed as

$$\bar{\mathbf{X}}_j = \sum_{l=1}^{s^\Gamma} x_j^l \mathbf{v}_j^l = \bar{\mathbf{V}}_j \mathbf{X}_j, \tag{69}$$

where $\mathbf{X}_j \triangleq (x_j^l)_{l=1}^{s^\Gamma}$ is a $s^\Gamma \times 1$ column vector, and $\bar{\mathbf{V}}_j = [\mathbf{v}_j^l]_{l=1}^{s^\Gamma}$ is a $u_s \times s^\Gamma$ matrix. Then, the received signal at EN i can be written as

$$\bar{\mathbf{Y}}_i = \sum_{j=1}^{N} \bar{\mathbf{H}}_{ij} \bar{\mathbf{V}}_j \mathbf{X}_j + \bar{\mathbf{Z}}_i, \tag{70}$$

where $\bar{\mathbf{Y}}_i$ and $\bar{\mathbf{Z}}_i$ represent the u_s symbol extension of the received signal Y_i and noise Z_i, respectively. $\bar{\mathbf{H}}_{ij}$ is a $u_s \times u_s$ diagonal matrix representing the u_s symbol extension of the channel, whose l-th diagonal element is $h_{ij}(l)$.

Next, we design the beamforming vectors such that each receiver i can decode the n desired signals $\{\mathbf{X}_{k+1} : k \in [(i-1)n : (in-1)] (\bmod N)\}$ by zero-forcing the interferences. To align the $N-n$ interference signals at each receiver together in the space with dimension $(s+1)^\Gamma$, beamforming vectors need to satisfy the following conditions:

$$\text{span}(\bar{\mathbf{H}}_{ij}\bar{\mathbf{V}}_j) \prec \text{span}(\mathbf{V}), \forall i \in \mathcal{M}, \forall j \in \mathcal{N}, (i,j) \notin \{(i,k+1) : k \in [(i-1)n : (in-1)] (\bmod N)\} \tag{71}$$

where span($\mathbf{P}$) denotes the space spanned by the column vectors of matrix $\mathbf{P}$, and $\mathbf{V}$ is a $u_s \times (s+1)^\Gamma$ matrix. Now we need to design the column vectors of $\bar{\mathbf{V}}^{[j]}$ and $\mathbf{V}$ to satisfy (71). Let $\mathbf{w}$ be a $u_s \times 1$ column vector $\mathbf{w} = (1, 1, \cdots, 1)^\mathrm{T}$. The sets of column vectors of $\bar{\mathbf{V}}_j$ and $\mathbf{V}$, denoted as $\bar{\mathcal{V}}_j$ and $\mathcal{V}$, respectively, are given as below

$$\bar{\mathcal{V}}_j = \left\{ \underbrace{\left(\prod_{t \in \mathcal{M}, q \in \mathcal{N}, (t,q) \notin \{(t,k+1) : k \in [(t-1)n:(tn-1)] \,(\mathrm{mod}\,N)\}} \left(\bar{\mathbf{H}}_{tq} \right)^{\alpha_{tq}} \right) \mathbf{w} : \alpha_{tq} \in [0:s-1]}_{\text{a total of } s^\Gamma \text{ columns}} \right\}, \forall j \in \mathcal{N}, \quad (72)$$

$$\mathcal{V} = \left\{ \underbrace{\left(\prod_{t \in \mathcal{M}, q \in \mathcal{N}, (t,q) \notin \{(t,k+1) : k \in [(t-1)n:(tn-1)] \,(\mathrm{mod}\,N)\}} \left(\bar{\mathbf{H}}_{tq} \right)^{\alpha_{tq}} \right) \mathbf{w} : \alpha_{tq} \in [0:s]}_{\text{a total of } (s+1)^\Gamma \text{ columns}} \right\}. \quad (73)$$

We then show that the desired signal streams received at each receiver are linearly independent of each other and interference signal streams such that the desired streams can be decoded by zero-forcing interferences. At any receiver i, the desired signal streams are beamformed along the ns^Γ vectors of $\left[\bar{\mathbf{H}}_{i,i_1} \bar{\mathbf{V}}_{i_1} \quad \bar{\mathbf{H}}_{i,i_2} \bar{\mathbf{V}}_{i_2} \quad \cdots \quad \bar{\mathbf{H}}_{i,i_n} \bar{\mathbf{V}}_{i_n} \right]$, where $i_m = (i-1)n + m \,(\mathrm{mod}\,N)$, $m \in [n]$. By condition (71), the interference streams at any receiver i from transmitter j are aligned at the column vector space of $\mathbf{V}$ for $j \in [N] / \{k+1 : k \in [(i-1)n : in-1] \,(\mathrm{mod}\,N)\}$. To decode the desired ns^Γ streams successfully, it suffices to show that the $u_s \times u_s$ matrix

$$\mathbf{A}_i = \left[\bar{\mathbf{H}}_{i,i_1} \bar{\mathbf{V}}_{i_1} \quad \bar{\mathbf{H}}_{i,i_2} \bar{\mathbf{V}}_{i_2} \quad \cdots \quad \bar{\mathbf{H}}_{i,i_n} \bar{\mathbf{V}}_{i_n} \quad \mathbf{V} \right] \quad (74)$$

has a full rank of u_s almost surely for $\forall i \in \mathcal{M}$. By the beamforming vectors in (72) and (73), we can observe that the u_s elements in the l-th row of $\mathbf{A}_i$ have the following forms

$$\underbrace{\left\{ h_{i,i_m}(l) \prod_{t \in \mathcal{M}, q \in \mathcal{N}, (t,q) \notin \{(t,k+1) : k \in [(t-1)n:(tn-1)] (\mathrm{mod}\,N)\}} (h_{tq}(l))^{\alpha_{tq}} : \alpha_{tq} \in [0:s-1], m \in [n] \right\}}_{\text{a total of } ns^\Gamma \text{ elements}} \bigcup$$

$$\underbrace{\left\{ \prod_{t \in \mathcal{M}, q \in \mathcal{N}, (t,q) \notin \{(t,k+1) : k \in [(t-1)n:(tn-1)] \,(\mathrm{mod}\,N)\}} (h_{tq}(l))^{\beta_{tq}} : \beta_{tq} \in [0:s] \right\}}_{\text{a total of } (s+1)^\Gamma \text{ elements}}, \quad (75)$$

where $\{h_{ij}(l)\}$ are drawn independently from a continuous probability distribution. We observe from (75) that all the elements of $\mathbf{A_i}$ meet the two conditions of [34, Lemma 1]. Hence, the matrix $\mathbf{A_i}$ is a full-rank matrix for $\forall i \in \mathcal{M}$. Taking s to infinity, the DoF for each receiver achieved by above scheme is given by $\lim_{s \to +\infty} \frac{ns^\Gamma}{ns^\Gamma + (s+1)^\Gamma} = \frac{n}{n+1} = \frac{Nr}{Nr+M}$.

Further, consider the basic scheme that N transmitters deliver their messages to the assigned receivers in the time division manner, which achieves a DoF of $\frac{n}{N} = \frac{r}{M}$ for each receiver. Therefore, the per-receiver DoF of the considered interference-multicast channel with multicast group size r is given by $DoF_r^u = \max\left\{ \frac{Nr}{Nr+M}, \frac{r}{M} \right\}$.

2) Converse Proof of Lemma 1: In Section III-C1, we have proved that the NULT of the task assignment and uploading scheme in Section III-B1 is information-theoretically optimal, so the per-receiver DoF of the considered uplink channel in (15) of Lemma 1 must also be optimal.

B. Proof of Lemma 2

1) Achievability: We show the achievable schemes in two cases, $r = 1$ and $r \geq 2$, respectively. $i)$ $r = 1$: By (14), the task output messages at each transmitter i can be represented as $\{\widetilde{W}_j : j \in [(i-1)n+1 : in]\}$. Let $\Gamma = M(N-1)$ and consider a $u_s = s^\Gamma + n(s+1)^\Gamma$ symbol extension. Each message $\widetilde{W}_j$ are encoded into s^Γ independent streams x_j^l, $l \in [s^\Gamma]$, each beamformed along a $u_s \times 1$ column vector $\mathbf{v}_j^l$, for $\forall j \in \mathcal{N}$. Then, the signal transmitted at transmitter i can be expressed as

$$\bar{\mathbf{X}}_i = \sum_{j=(i-1)n+1}^{in} \sum_{l=1}^{s^\Gamma} x_j^l \mathbf{v}_j^l = \sum_{j=(i-1)n+1}^{in} \bar{\mathbf{V}}_j \mathbf{X}_j, \tag{76}$$

where $\mathbf{X}_j = (x_j^l)_{l=1}^{s^\Gamma}$ is a $s^\Gamma \times 1$ column vector and $\bar{\mathbf{V}}_j = [\mathbf{v}_j^l]_{l=1}^{s^\Gamma}$ is a $u_s \times s^\Gamma$ matrix. The signal received at receiver j can be expressed as

$$\bar{\mathbf{Y}}_j = \sum_{i=1}^{M} \bar{\mathbf{G}}_{ji} \sum_{k=(i-1)n+1}^{in} \bar{\mathbf{V}}_k \mathbf{X}_k + \bar{\mathbf{Z}}_j, \tag{77}$$

where $\bar{\mathbf{G}}_{ji}$ is a $u_s \times u_s$ diagonal matrix representing the u_s symbol extension of the channel, $\bar{\mathbf{Y}}_j$ and $\bar{\mathbf{Z}}_j$ represent the u_s symbol extension of the received signal Y_j and noise Z_j, respectively.

Next, we align the interferences at each receiver j such that the total dimension of the spaces spanned by the interference vectors is $n(s+1)^\Gamma$. Then, the desired s^Γ streams corresponding to the desired signal $\mathbf{X}_j$ can be decoded by zero-forcing the interferences from an $u_s = s^\Gamma + n(s+1)^\Gamma$-dimensional received signal vector. We ensure this by designing the beamforming vectors $\{\bar{\mathbf{V}}_j\}$ as follows, where the message $\widetilde{W}_j$ desired by receiver j is at transmitter $\lceil \frac{j}{n} \rceil \in \mathcal{M}$ by (14),

$$\left.\begin{aligned} \text{span}(\bar{\mathbf{G}}_{ji}\bar{\mathbf{V}}_{(i-1)n+1}) &\subset \text{span}(\mathbf{U}_1) \\ \text{span}(\bar{\mathbf{G}}_{ji}\bar{\mathbf{V}}_{(i-1)n+2}) &\subset \text{span}(\mathbf{U}_2) \\ &\vdots \\ \text{span}(\bar{\mathbf{G}}_{ji}\bar{\mathbf{V}}_{(i-1)n+k}) &\subset \text{span}(\mathbf{U}_k) \\ &\vdots \\ \text{span}(\bar{\mathbf{G}}_{ji}\bar{\mathbf{V}}_{in}) &\subset \text{span}(\mathbf{U}_n) \end{aligned}\right\} \forall j \in \mathcal{N}, \forall i \in \mathcal{M}, j \neq (i-1)n+k \text{ for } \forall k \in [n], \tag{78}$$

where $\mathbf{U}_k$ is a $u_s \times (s+1)^\Gamma$ matrix, $\forall k \in [n]$. Next, we design $\{\mathbf{V}_j\}$ and $\{\mathbf{U}_k\}$ to satisfy above conditions. First, we generate n $u_s \times 1$ column vectors $\mathbf{w}_k = (w_k^l)_{l=1}^{u_s}$, $k \in [n]$. All elements of these n vectors are chosen i.i.d from some continuous distribution whose support lies between a finite minimum value and a finite maximum value. Then, the sets of column vectors of $\bar{\mathbf{V}}_j$ and $\mathbf{U}_k$ are denoted as $\bar{\mathcal{V}}_j$ and $\mathcal{U}_k$, respectively, and are given as follows,

$$\bar{\mathcal{V}}_{(i-1)n+k} = \left\{ \left(\prod_{q \in \mathcal{N}, t \in \mathcal{M}, (q,t) \neq ((t-1)n+k,t)} (\bar{\mathbf{G}}_{qt})^{\alpha_{qt}} \right) \mathbf{w}_k : \alpha_{qt} \in [0 : s-1] \right\}, \forall i \in \mathcal{M}, \forall k \in [n], \tag{79}$$

$$\mathcal{U}_k = \left\{ \left(\prod_{t \in \mathcal{M}, q \in \mathcal{N}, (q,t) \neq ((t-1)n+k,t)} (\bar{\mathbf{G}}_{qt})^{\alpha_{qt}} \right) \mathbf{w}_k : \alpha_{qt} \in [0 : s] \right\}, \forall k \in [n]. \tag{80}$$

In the following, we show that the desired signal streams are linearly independent with the

interference signal streams, and hence can be decoded by zero-forcing the interference. Consider the signal vectors received at any receiver $j = (i-1)n+k$, $i \in \mathcal{M}$, $k \in [n]$. By (78), the desired signal streams are beamformed along the s^Γ vectors of $\bar{\mathbf{G}}_{(i-1)n+k,i}\bar{\mathbf{V}}_{(i-1)n+k}$, while the interference vectors are aligned at the column vector spaces of $\mathbf{U}_{k'}, \forall k' \in [n]$. To decode the desired streams successfully, it suffices to show that the $u_s \times u_s$ matrix

$$\mathbf{\Lambda}_j = \mathbf{\Lambda}_{(i-1)n+k} = \begin{bmatrix} \bar{\mathbf{G}}_{(i-1)n+k,i}\bar{\mathbf{V}}_{(i-1)n+k} & \mathbf{U}_1 & \mathbf{U}_2 & \cdots & \mathbf{U}_n \end{bmatrix} \tag{81}$$

is a full-rank matrix almost surely for $\forall j \in \mathcal{N}$ or $\forall i \in \mathcal{M}$ and $\forall k \in [n]$. It is seen that the l-th row elements of $\mathbf{\Lambda_j}$ have the following forms,

$$\left\{ h_{(i-1)n+k,i}(l) \prod_{q \in \mathcal{N}, t \in \mathcal{M}, (q,t) \neq ((t-1)n+k,t)} (g_{qt}(l))^{\alpha_{qt}} w_k(l): \ \alpha_{qt} \in [0:s-1] \right\} \bigcup$$

$$\left\{ \prod_{q \in \mathcal{N}, t \in \mathcal{M}, (q,t) \neq ((t-1)n+k',t)} (g_{qt}(l))^{\beta_{qt}} w_{k'}(l): \ \beta_{qt} \in [0:s], \ k' \in [n] \right\}. \tag{82}$$

By (82), we have:

1) The product term in the l-th row of $\mathbf{U}_k$ contains $w_k(l)$ with exponent 1, but do not contain $w_{k'}(l), \forall k' \neq k$. Thus, all the monomial elements in the l-th row of $[\mathbf{U}_1 \ \mathbf{U}_2 \ \cdots \ \mathbf{U}_n]$ are unique.

2) The equations corresponding to $\mathbf{G}_{(i-1)n+k,i}$ are not contained in the interference alignment relations of (78) for $\mathbf{U}_k$, so the monomial elements in the l-th row of $\mathbf{U}_k$ do not contain $h_{(i-1)n+k,i}$, $\forall i \in \mathcal{M}$. It means that all the monomial terms in $\bar{\mathbf{G}}_{(i-1)n+k,i}\bar{\mathbf{V}}_{(i-1)n+k}$ are different from those in $\mathbf{U}_k$. They are also different from the monomial terms in $\mathbf{U}_{k'}$, $\forall k' \neq k$, due to $w_k(l)$.

Therefore, we can conclude that these u_s vectors in $\mathbf{\Lambda_j}$ are independent, and hence $\mathbf{\Lambda_j}$ is a full-rank matrix. Taking s to infinity, the scheme achieves a per-receiver DoF of $\lim_{s \to +\infty} \frac{s^\Gamma}{s^\Gamma + n(s+1)^\Gamma} = \frac{1}{1+n} = \frac{M}{N+M}$. Comparing it with the DoF of $\frac{1}{N}$ achieved by TDMA, the per-receiver DoF of the considered downlink channel for $r = 1$ is given by $DoF_1^d = \max\left\{\frac{M}{N+M}, \frac{1}{N}\right\}$.

$ii)$ $r \geq 2$: We first consider interference neutralization enabled by transmitter cooperation. Encode the task output message $\widetilde{W}_j$ into r independent streams x_j^p, $p \in [r]$. For better illustration, each stream x_j^p is given an index $(p-1)N+j$. There are a total of Nr (or Mn) different streams corresponding to all N messages. Based on the index order, these Nr different streams can be divided into N groups, each group with r different streams, where the k-th group is given by

$$\mathcal{Q}_k = \left\{ x_j^p: (p-1)N+j \in [(k-1)r+1: kr] \right\}, \ k \in [N]. \tag{83}$$

Since each message exists at r different transmitters, each stream is also owned by r different transmitters. Each group of streams is downloaded in the time division manner. The downlink channel formed by transmitting each group of r streams can be treated as a MISO broadcast channel with perfect transmitter cooperation, whose sum DoF is r [22], [35] achieved by using interference neutralization. Thus, a DoF of $\frac{r}{N}$ is obtained for each receiver in the $r \geq 2$ case.

Then, we apply asymptotic interference alignment scheme in the $r = 1$ case. Since $\frac{N}{M} = n_1$, the messages at each transmitter i are $\left\{\widetilde{W}_k : k \in [(i-1)n_1+1 : in_1]\right\}$ for $r = 1$ and $\left\{\widetilde{W}_{k+1} : k \in [(i-1)rn_1 : (irn_1-1)](\bmod N)\right\}$ for $r \geq 2$. In $r \geq 2$ case, we can let each transmitter only transmit the n_1 messages among the total rn_1 messages, and different transmitters transmit non-overlapped messages. By doing so, we construct a downlink channel with the same information flow as $r = 1$ case. Utilizing the alignment scheme in $r = 1$ case, we thus obtain a per-receiver DoF of $\frac{M}{N+M}$ for $r \geq 2$ case. Comparing above two schemes, a Dof of $\max\{\frac{M}{N+M}, \frac{r}{N}\}$ is obtained.

Summarizing the per-receiver DoF for $r = 1$ and $r \geq 2$ cases, we thus prove Lemma 2.

2) Converse proof of Lemma 2: By using the maximum sum DoF $\min\{M, N\}$ of MISO broadcast channels [36], the optimal per-receiver DoF DoF_r^{d*} is upper bounded by $\frac{\min\{M, N\}}{N}$, and the gap between this upper bound and the lower bound in (17) satisfies $\frac{DoF_r^{d*}}{DoF_r^d} \leq 2$.

REFERENCES

[1] K. Li, M. Tao, and Z. Chen, "Exploiting computation replication in multi-user multi-server mobile edge computing networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[2] K. Li, M. Tao, and Z. Chen, "A computation-communication tradeoff study for mobile edge computing networks," *Submitted to IEEE ISIT 2019*. [Online]. Available: http://iwct.sjtu.edu.cn/Personal/mxtao/paper/isit19-Li-long.pdf

[3] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[4] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conf. on Hot Topics in Cloud Comput.*, Jun. 2010, pp. 4–4.

[5] ETSI, "Mobile edge computing: A key technology towards 5G," *White Paper*, Sep. 2015. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf

[6] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[7] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[8] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[9] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[10] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[11] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[12] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[13] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[14] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[15] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Letters*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[16] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[17] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. on Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[18] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[19] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[20] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[21] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan 2019.

[22] V. S. Annapureddy, A. E. Gamal, and V. V. Veeravalli, "Degrees of freedom of interference channels with CoMP transmission and reception," *IEEE Trans. Inf. Theory*, vol. 58, no. 9, pp. 5740–5760, Sep. 2012.

[23] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[24] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. PP, no. 99, pp. 1–1, 2017.

[25] D. Wang, G. Joshi, and G. Wornell, "Efficient task replication for fast response times in parallel computation," in *Proc. ACM SIGMETRICS*, 2014.

[26] Y. Sun, C. E. Koksal, and N. B. Shroff, "On delay-optimal scheduling in queueing systems with replications," *CoRR*, vol. abs/1603.07322, 2016. [Online]. Available: http://arxiv.org/abs/1603.07322

[27] 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. PP, no. 99, pp. 1–1, 2017.

[28] 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, 2017.

[29] 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.

[30] 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.

[31] J. Hachem, U. Niesen, and S. Diggavi, "Degrees of freedom of cache-aided wireless interference networks," *IEEE Trans. Inf. Theory*, pp. 1–1, 2018.

[32] I. Wang and D. N. C. Tse, "Interference mitigation through limited transmitter cooperation," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 2941–2965, May 2011.

[33] A. M. Girgis, Ö. Erçetin, M. Nafie, and T. A. ElBatt, "Degrees of freedom of interference networks with transmitter-side caches," *CoRR*, vol. abs/1712.05957, 2017. [Online]. Available: http://arxiv.org/abs/1712.05957

[34] V. R. Cadambe, S. A. Jafar, "Interference alignment and the degrees of freedom of wireless X networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 3893–3908, Sep. 2009.

[35] T. Gou and S. A. Jafar, "Degrees of freedom of the k user $M \times N$ MIMO interference channel," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6040–6057, Dec 2010.

[36] H. Weingarten, Y. Steinberg, and S. S. Shamai, "The capacity region of the gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936–3964, Sep. 2006.