

Article

Enhancing Rate-Splitting-Based Distributed Edge Computing via Multi-Group Information Recycling

Wanlin Liang  and Xiaofan He *

The Electronic Information School, Wuhan University, Wuhan 430072, China; liangwl@whu.edu.cn

* Correspondence: xiaofanhe@whu.edu.cn

Abstract: To address the straggling effect in distributed edge computing, existing methods often introduce extra computation or communication costs. Recently, information recycling has emerged as an efficient solution that avoids such extra overhead. Nonetheless, the performance of the existing information recycling-assisted rate-splitting-based distributed edge computing scheme significantly hinges on the single common stream, whose rate is limited by the edge node (EN) with the worst channel quality. To this end, a multi-group information recycling scheme for rate-splitting-based distributed edge computing is proposed, where the ENs are clustered into multiple groups, each with its own common stream. In this way, the proposed scheme alleviates the communication limitation of a single common stream, thereby boosting the information recycling mechanism for promoted computation collaboration. A K-medoids-based grouping algorithm is designed for the general multi-antenna case, and a more efficient continuous partitioning-based grouping algorithm is proposed for the special case of single-antenna. Besides, a convex–concave procedure-based algorithm is developed to solve the corresponding latency optimization problem. Simulations demonstrate that the proposed scheme offers significant and robust advantages across various communication and computation conditions. In the considered scenarios, the proposed scheme can substantially reduce the processing latency by up to 51.0% compared to the conventional information recycling scheme.

Keywords: distributed edge computing; straggling; information recycling; non-orthogonal transmission



Citation: Liang, W.; He, X. Enhancing Rate-Splitting-Based Distributed Edge Computing via Multi-Group Information Recycling. *Electronics* **2024**, *13*, 4403. <https://doi.org/10.3390/electronics13224403>

Academic Editors: Daniele Tarchi, Swapnil Sadashiv Shinde and Saira Bano

Received: 28 September 2024

Revised: 2 November 2024

Accepted: 5 November 2024

Published: 11 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, edge computing has played an increasingly vital role in enhancing the quality of computational services [1,2]. However, the mounting scale of edge intelligence applications, (e.g., big data, augmented reality, and artificial intelligence) is pushing the limits of edge computing. These large-scale and computationally intensive tasks have rendered traditional single-node edge computing insufficient, entailing distributed edge computing, which has emerged as a promising computation paradigm [3–5]. This approach partitions and distributes the original large-scale and intensive tasks among multiple edge nodes (ENs), exploiting their computational resources for faster processing.

Nonetheless, the straggling effect presents a notable challenge in distributed edge computing. Specifically, some ENs, known as stragglers, may experience substantial delays or even fail to complete their tasks due to various factors, including hardware heterogeneity, imbalanced workload allocation, I/O contentions, and hardware failures [6–8]. This significantly degrades the overall performance, as the system needs to wait for feedback from all the ENs in distributed edge computing [9]. To alleviate the straggling issues, several strategies have been considered in the literature. One typical approach is task replication [10–12], where redundant copies of the original task are distributed across multiple ENs, and the execution delay is determined by the fastest EN despite some stragglers. Besides task replication, coded computing [8] has recently been advocated, where the partitioned subtasks are encoded into a set of coded subtasks via coding theory [13]. Thanks to the well-designed coding redundancy, the mobile user can reconstruct the original

task result using the feedback from only a subset of ENs, thus avoiding the need to wait for stragglers. In addition, workload reallocation [14,15] is another commonly adopted method, which allows slower ENs to re-transmit part of their remaining workloads to other faster ENs to minimize the overall processing latency. Nevertheless, despite the effectiveness of these pioneering works in mitigating the straggling effect, they often introduce redundant computation or require additional data transmission, which may still limit the overall performance.

To handle straggling without incurring extra computation and communication costs, a physical-layer mechanism known as information recycling has been recently proposed [16], in view of the widespread employment of the spectrum-efficient non-orthogonal transmission techniques (e.g., rate-splitting multiple access (RSMA) [17,18] and power-domain non-orthogonal multiple access (NOMA) [19]) in distributed edge computing [20–24]. The key insight of information recycling is that, under many non-orthogonal transmission techniques involving successive interference cancellation (SIC), each EN has to decode part of the information intended for other ENs [25,26]. Instead of directly discarding this part of the information after interference cancellation, the information recycling mechanism proactively recycles this information (for free), by exploiting the SIC procedure. With the recycled information, faster ENs that complete their own task earlier can help execute the corresponding part of the task for the straggling ENs, instead of waiting for the straggling ENs, thereby mitigating the straggling effect without extra overhead. A concrete example of the information recycling mechanism has been proposed for rate-splitting-based distributed edge computing systems [16]. The performance of this conventional scheme heavily hinges on the single common stream in rate-splitting. Specifically, a larger common stream can increase the portion of tasks available for information recycling and thus promote computation collaboration. However, the rate of the common stream is fundamentally limited by the EN with the worst channel quality. This constraint undermines the benefits of information recycling, since enforcing a large common stream may aggravate the impact of the rate limitation on transmission, leading to sub-optimal communication performance. To the best of our knowledge, developing a new distributed edge computing scheme that can overcome the communication limitation of the single common stream remains an open challenge.

To this end, a novel scheme, coined as *multi-group information recycling* for rate-splitting-based distributed edge computing, is proposed in this work. Particularly, it will cluster the ENs into multiple groups and offer one common stream for each group, where collaborative computation can still be enabled through information recycling. Since each common stream is decoded by a smaller number of ENs, the communication bottleneck of the single common stream can be alleviated, thereby facilitating the common stream transmission. As a result, the information recycling mechanism can be further boosted, promoting computation collaboration among the ENs for processing latency reduction. The major contributions of this work are summarized as follows:

- A multi-group information recycling scheme for rate-splitting-based distributed edge computing is proposed in this work for processing latency reduction;
- To obtain a good EN grouping strategy, a K-medoids-based grouping algorithm is designed by jointly considering the channel alignment and the channel strength for the general multi-antenna case, and a more efficient continuous partitioning-based grouping algorithm is proposed for the special case of single-antenna;
- By exploiting the inherent structure of the corresponding latency optimization problem and bounding via difference of convex (DC) programming, a convex–concave procedure (CCCP)-based algorithm is developed to find a reasonably good solution.

The remainder of this paper is organized as follows. Related works are briefly reviewed in Section 2, and the system model and the conventional information recycling mechanism are presented in Section 3. In Section 4, the proposed scheme is presented, and the corresponding latency optimization problem is formulated. The latency optimization

algorithms are developed in Section 5. Simulation results are presented in Section 6. Finally, conclusions and future works are discussed in Section 7.

Notations: $(\cdot)^H$ denotes the Hermitian transpose. $\mathbb{C}^{a \times b}$ represents a complex matrix with dimension $a \times b$. $\mathcal{CN}(\mathbf{a}, \mathbf{b})$ represents the distribution of a circularly symmetric complex Gaussian random vector with mean vector $\mathbf{a}$ and covariance matrix $\mathbf{b}$. $\mathbb{E}[\cdot]$ represents expectation. $\Re[\cdot]$ obtains the real part of a complex number. $|\cdot|$ is the absolute value. $\text{tr}(\cdot)$ is the trace.

2. Related Work

Several methods have been considered in the literature to mitigate the straggling effect in distributed edge computing, including task replication [10–12], coded computing [27–29], as well as workload reallocation [14,15]. For example, in [10], the original computation task is duplicated and assigned to multiple ENs so that the execution latency is determined by the fastest EN rather than the stragglers. The tradeoff between expected completion time and task replication ratio is explored in [11] to identify the optimal replication strategy for various computation straggling models. In [12], a balanced grouping replication strategy is further proposed, focusing on optimizing the redundancy level to minimize average job compute time while preserving task completion time predictability. In addition to task replication, coded computing has gained traction for the ability to mitigate straggling via coding theory [13]. Specifically, the maximum distance separable (MDS) codes are used in [27] for distributed task offloading. It is shown that MDS-coded offloading can effectively mitigate the straggling effect as the mobile user can reconstruct the original task result using feedback from only a subset of fast ENs. Besides, entangled polynomial codes are proposed in [28] to achieve the optimal tradeoff between the recovery threshold (i.e., the minimum number of ENs required to recover the original task) and the per-EN computation workload through different matrix partitioning. In [29], the impact of the coding parameter selection under different task execution models is investigated to alleviate the straggling effect and minimize the processing latency. Moreover, workload reallocation is another widely employed method for straggler mitigation. For instance, in [14], when a faster EN completes its task and becomes idle, part of the straggling ENs' remaining workloads can be re-transmitted to this faster EN, thereby rebalancing workload and addressing straggling. In [15], by detecting straggling ENs continuously in the computation process, the mobile user can promptly reassign tasks to faster ENs, thereby minimizing idle time and improving overall system performance. Nevertheless, while these methods effectively mitigate the straggling effect, their performance remains constrained. Specifically, task replication and coded computing incur increased communication and computational costs, because replicated or coded subtasks are often larger than the subtasks in traditional distributed computing due to the introduced task redundancy. Besides, workload reallocation necessitates task data retransmission, which brings extra communication costs and limits the overall performance.

On the other hand, significant research has been conducted on strengthening communication in distributed edge computing using non-orthogonal transmission techniques. For instance, in [22], the computation subtasks are offloaded to the ENs via NOMA, and the processing latency is minimized by jointly optimizing transmission duration and workload allocation. In [23], the benefits of NOMA are investigated for both uplink and downlink transmissions in mobile edge computing. In [24], the total multi-user processing latency of an RSMA-based distributed edge computing system is optimized and the superiority of RSMA in serving distributed edge computing is manifested as compared to NOMA. However, most of these pioneering works mainly consider how non-orthogonal transmission strengthens the communication in distributed edge computing, while the straggling issue is largely ignored.

To handle straggling without incurring extra computation and communication costs, the information recycling mechanism is proposed in [16], which can be integrated into the non-orthogonal transmission techniques. In particular, by recycling the task information

intended for other ENs in the decoded common stream, the faster ENs can help execute the corresponding part of the task for the slower ENs, thereby mitigating the straggling effect. However, the effectiveness of the existing information recycling-assisted distributed edge computing scheme highly hinges on the single common stream in rate-splitting and the rate of the common stream is inherently limited by the EN with the poorest channel quality. This constraint diminishes the advantages of information recycling, as dedicating resources to a single common stream can promote collaborative computation but may also degrade overall communication performance. In contrast to these pioneering works, by clustering ENs into multiple groups and transmitting several common streams, the proposed multi-group information recycling scheme can alleviate the single common stream's communication limitation to facilitate the transmission of the common streams, thus promoting the information recycling mechanism for enhanced computation collaboration among the ENs.

3. System Model

In this section, the system model of rate-splitting-based distributed edge computing with stragglers will be presented, together with the conventional information recycling mechanism.

3.1. System Model

Consider a distributed edge computing system consisting of a mobile user and K ENs indexed by $\mathcal{K} = \{1, 2, \dots, K\}$. The mobile user wirelessly offloads its computation-intensive task $\mathcal{W}$ to these K ENs for faster processing. Specifically, the computation task $\mathcal{W}$ can be divided into K independent subtasks $\{W_1, W_2, \dots, W_K\}$ (Take (large-scale) matrix multiplication, a fundamental operation in many machine learning algorithms [30], as an example of the practical application. The original matrix multiplication task $\mathbf{F}\mathbf{b}$ will be partitioned into K sub-tasks $\{\mathbf{F}_1\mathbf{b}, \mathbf{F}_2\mathbf{b}, \dots, \mathbf{F}_K\mathbf{b}\}$ with $\mathbf{F}^T = [\mathbf{F}_1^T, \mathbf{F}_2^T, \dots, \mathbf{F}_K^T]$). Each sub-task W_k will be offloaded to EN k for processing (As in the existing literature [1,23,31,32], it is assumed that the size of the computing results is much smaller compared to the offloaded subtasks, making the downlink transmission latency negligible).

3.1.1. Communication Phase

For better spectrum efficiency in communications, it is assumed that the mobile user employs RSMA to offload the subtasks to the K ENs during the communication phase. Specifically, each subtask W_k is split into a common part W_k^c and a private part W_k^p (In the matrix multiplication, this corresponds to partition $\mathbf{F}_k$ along the rows into two parts, $\mathbf{F}_k^c$ and $\mathbf{F}_k^p$, where $\mathbf{F}_k^T = [\mathbf{F}_k^{c,T}, \mathbf{F}_k^{p,T}]$). In the wireless physical layer, all the common parts $\{W_1^c, W_2^c, \dots, W_K^c\}$ are encoded into a single common stream s^c , while each private part W_k^p is encoded into a private stream s_k^p . By employing superposition coding, the transmitted signal $\mathbf{x}$ from the mobile user is given by

$$\mathbf{x} = \mathbf{w}^c s^c + \sum_{k \in \mathcal{K}} \mathbf{w}_k^p s_k^p, \quad (1)$$

where $\mathbf{w}^c \in \mathbb{C}^{N_t \times 1}$ and $\mathbf{w}_k^p \in \mathbb{C}^{N_t \times 1}$ are the precoding vectors for the common stream s^c and the private stream s_k^p , respectively, with N_t denoting the number of antennas of the mobile user. The received wireless signal y_k at EN k can be expressed as

$$y_k = \mathbf{h}_k^H \mathbf{x} + n_k, \quad (2)$$

where $\mathbf{h}_k \in \mathbb{C}^{N_t \times 1}$ represents the gain of the wireless channel between the mobile user and EN k , and $n_k \sim \mathcal{CN}(0, \sigma^2)$ is the corresponding additive white Gaussian noise. According to the decoding rules of rate-splitting [18], each EN k first decodes the common stream s^c by treating the interference from all private streams as noise and then subtracts it from the received signal using SIC. Thereafter, EN k decodes its private stream s_k^p by treating

the remaining interference from the other private streams as noise. Hence, by plugging (1) into (2), the signal-to-interference-plus-noise ratios (SINRs) of the common stream s^c and the private stream s_k^p to EN k are, respectively, given by

$$\gamma_k^c = \frac{|\mathbf{h}_k^H \mathbf{w}^c|^2}{\sum_{j \in \mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2}, \forall k \in \mathcal{K}, \quad (3)$$

$$\gamma_k^p = \frac{|\mathbf{h}_k^H \mathbf{w}_k^p|^2}{\sum_{j \in \mathcal{K} \setminus k} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2}, \forall k \in \mathcal{K}. \quad (4)$$

With the above SINRs, the rates R_k^p 's of the private streams should admit

$$R_k^p \leq B \log_2(1 + \gamma_k^p), \quad \forall k \in \mathcal{K}, \quad (5)$$

where B is the system bandwidth. To ensure that the common stream s^c can be successfully decoded by all ENs, the rate R^c of the common stream should admit

$$R^c \leq \min_{k \in \mathcal{K}} \{B \log_2(1 + \gamma_k^c)\}. \quad (6)$$

As the rate R^c of the common stream is shared among all ENs to offload $\{W_1^c, W_2^c, \dots, W_K^c\}$, it has $\sum_{k=1}^K R_k^c = R^c$, where R_k^c is the uplink transmission rate of the common part W_k^c of subtask W_k . Hence, the overall uplink transmission rate R_k to EN k is given by

$$R_k = R_k^c + R_k^p. \quad (7)$$

3.1.2. Computation Phase

In the computation phase, each EN k processes its received offloaded subtask W_k . To account for the straggling effect [8,9], the computation rate C_k of EN k can be modeled as follows [29]

$$C_k = \bar{C} - \Delta_k. \quad (8)$$

Here, $\bar{C} = \frac{f_e}{\beta}$ denotes the maximum computation rate of the ENs, where f_e is the maximum processor frequency, and β represents the number of CPU cycles required to process one bit of data. Δ_k is a non-negative random variable representing the degradation in computation capability due to the straggling effect. The variables $\{\Delta_k\}_{k=1}^K$ are assumed to be independent and identically distributed (i.i.d.).

3.2. Conventional Information Recycling

The idea of conventional information recycling for rate-splitting-based distributed edge computing [16] is introduced in the sequel. According to the rate-splitting transmission principle [18], each EN k will decode the task messages $\{W_1^c, \dots, W_{k-1}^c, W_{k+1}^c, \dots, W_K^c\}$ intended for other ENs when decoding the common stream s^c to obtain its own message W_k . In conventional distributed computing, after decoding, these messages will be discarded directly. As a result, each EN can only compute its own subtask W_k , and the faster EN that completes its own task earlier can do nothing but wait for the straggling ENs. As shown in Figure 1a, both EN 1 and EN 3 can do nothing but wait for the straggling EN 3 and remain idle, which undoubtedly wastes the computation capability of the system. In contrast, the information recycling mechanism enables the ENs to proactively recycle the messages $\{W_1^c, \dots, W_{k-1}^c, W_{k+1}^c, \dots, W_K^c\}$, as shown in Figure 1b. In this way, each EN will have full knowledge of the common parts $\{W_1^c, W_2^c, \dots, W_K^c\}$ of all the subtasks $\{W_k\}_{k=1}^K$. When a certain EN i is straggling, another faster EN j can help compute (the common part of) its subtask based on the recycled information W_i^c .

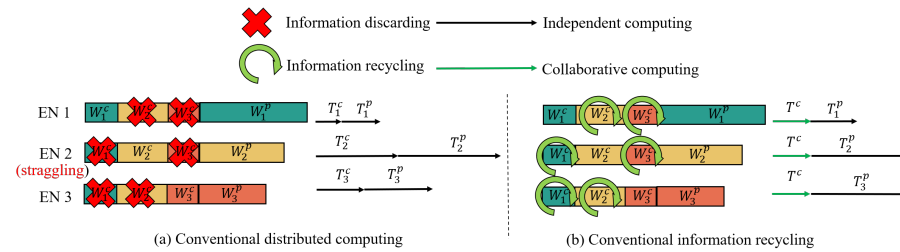


Figure 1. Comparison between the conventional distributed computing scheme and the conventional information recycling scheme.

The specific process of collaborative computation for the common parts is introduced as follows and summarized in Algorithm 1. Each common part task W_k^c is divided into A_k mini-tasks $\{W_{k,u}^c\}_{u=1}^{A_k}$, where $A_k = \left\lceil \frac{D_k^c}{D_m} \right\rceil$. In the matrix multiplication, each common part $\mathbf{F}_k^c$ is further partitioned along the rows into A_k smaller parts $\{\mathbf{F}_{k,1}^c, \mathbf{F}_{k,2}^c, \dots, \mathbf{F}_{k,A_k}^c\}$ to form mini-tasks $\{\mathbf{F}_{k,u}^c \mathbf{b}\}_{u=1}^{A_k}$, where $\mathbf{F}_k^{c,T} = [\mathbf{F}_{k,1}^{c,T}, \mathbf{F}_{k,2}^{c,T}, \dots, \mathbf{F}_{k,A_k}^{c,T}]$. Here, D_k^c and D_m are the sizes of the task W_k^c and each mini-task, respectively, (note that D_m is a parameter determined during the engineering implementation process; therefore, its specifics are beyond the scope of this paper). The system also maintains an ordered task queue for each EN k , which is initialized as $\{W_{k,u}^c\}_{u=1}^{A_k}$. Each EN sequentially processes the mini-tasks according to their queuing order. Upon completing the mini-task $W_{k,u}^c$, EN j sends a finishing signal $S_{k,u}^{\text{fin}}$ to inform the mobile user. Besides, whenever a certain EN j completes all its assigned mini-tasks and the overall computation task has not yet been finished, it can assist other straggling ENs in computing their mini-tasks by information recycling. Specifically, if straggling EN i has the longest unfinished task queue at the current time, and task $W_{k,v}^c$ at the end of its task queue, the mobile user will send a recycling signal $S_{k,v}^{\text{rec}}$ to EN j and a canceling signal $S_{k,v}^{\text{can}}$ to EN i . The signals $S_{k,v}^{\text{rec}}$ and $S_{k,v}^{\text{can}}$ direct EN j to begin computing $W_{k,v}^c$ using information recycling and EN i to remove $W_{k,v}^c$ from its task queue, respectively. This collaboration continues until the overall computation task is completed, allowing fast ENs and slow ENs to collaboratively compute the common parts $\{W_1^c, W_2^c, \dots, W_K^c\}$. An illustrative example is depicted in Figure 2. With the information recycling mechanism enabling computation collaboration, theoretically, the computation capabilities of all ENs can be captured for the computation of $\{W_1^c, W_2^c, \dots, W_K^c\}$ and the overall computation rate C^{rec} can be made to

$$C^{\text{rec}} = \sum_{k=1}^K C_k. \quad (9)$$

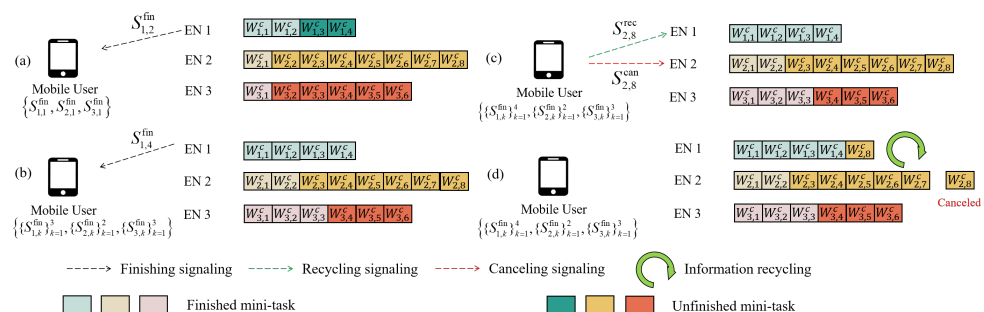


Figure 2. An illustrative example of the collaborative computing among ENs under information recycling. Suppose that the task queues of the three ENs are initialized as $\{W_{1,k}^c\}_{k=1}^4$, $\{W_{2,k}^c\}_{k=1}^8$, and $\{W_{3,k}^c\}_{k=1}^6$, respectively. (a) After it finished the mini-task $W_{1,2}^c$, EN 1 sends the finishing signal $S_{1,2}^{\text{fin}}$ to the mobile user to indicate the completion. Meanwhile, the mobile user has already received three finishing signals $\{S_{1,1}^{\text{fin}}, S_{2,1}^{\text{fin}}, S_{3,1}^{\text{fin}}\}$ from the three ENs. (b) EN 1 finishes its last assigned task $W_{1,4}^c$ and sends a finishing signal $S_{1,4}^{\text{fin}}$ to the mobile user. (c) Upon receiving the finishing signal $S_{1,4}^{\text{fin}}$,

the mobile user is notified that EN 1 has completed all its assigned mini-tasks. As EN 2 has the a recycling signal $S_{2,8}^{\text{rec}}$ and a canceling signal $S_{2,8}^{\text{can}}$ to EN 1 and EN 2, respectively. (d) EN 1 receives the recycling signal $S_{2,8}^{\text{rec}}$ and starts the computation of $W_{2,8}^c$. EN 2 receives the canceling signal $S_{2,8}^{\text{can}}$ and removes the mini-task $W_{2,8}^c$ from its task queue.

Algorithm 1 The collaborative computing process of information recycling

```

1: Initialize: For each EN  $k$ , divide the common part  $W_k^c$  into mini-tasks and initialize the ordered task queue
    $\{W_{k,u}^c\}_{u=1}^{A_k}$ ;
2: Each EN starts to sequentially process the mini-tasks according to their queuing order;
3: while The overall computation task is not completed do
4:   for  $j = 1 \dots K$  do
5:     if EN  $j$  completes a mini-task  $W_{k,u}^c$  then
6:       EN  $j$  sends a finishing signal  $S_{k,u}^{\text{fin}}$  to the mobile user;
7:     end if
8:     if EN  $j$  receives a recycling signal  $S_{k,v}^{\text{rec}}$  then
9:       EN  $j$  adds  $W_{k,v}^c$  to its task queue and begins computing  $W_{k,v}^c$  via information recycling;
10:    end if
11:    if EN  $j$  receives a canceling signal  $S_{k,v}^{\text{can}}$  then
12:      EN  $j$  removes  $W_{k,v}^c$  from its task queue;
13:    end if
14:  end for
15:  if The mobile user receives a finishing signal  $S_{k,u}^{\text{fin}}$  from EN  $j$  and determines that EN  $j$  has completed all
    assigned mini-tasks then
16:    Finding the EN  $i$  with the longest unfinished task queue, and task  $W_{k,v}^c$  at the end of its task queue, the
    mobile user sends a recycling signal  $S_{k,v}^{\text{rec}}$  to EN  $j$  and a canceling signal  $S_{k,v}^{\text{can}}$  to EN  $i$ ;
17:  end if
18: end while

```

4. Proposed Multi-Group Information Recycling

4.1. Multi-Group Information Recycling Scheme

The performance of the conventional information recycling assisted distributed edge computing highly depends on the single common stream [16]. Specifically, when more resources are allocated to the common stream, the common parts $\{W_1^c, W_2^c, \dots, W_K^c\}$ will occupy a larger portion of the task, thus facilitating the information recycling mechanism for straggler mitigation. Nonetheless, enforcing a too-strong common stream may not be necessarily good for communication, as the rate of the common stream s^c is restricted by the EN with the worst channel quality. This may offset the performance gain brought by the information recycling mechanism. For this reason, a new information recycling scheme that can overcome the communication limitation of the single common stream is required. To this end, a multi-group information recycling scheme is proposed for rate-splitting-based distributed edge computing below.

Assume that the K ENs are divided into G mutually-exclusive groups and each group g contains $\mathcal{K}_g$ ENs with $\bigcup_{g \in \mathcal{G}} \mathcal{K}_g = \mathcal{K}$. For the communication phase, as shown in Figure 3a, the common parts $\{W_k^c | k \in \mathcal{K}_g\}$ of the ENs in each group g are combined as $W_{\mathcal{K}_g}^c$ and encoded into the common stream $s_{\mathcal{K}_g}^c$ in the wireless layer. Meanwhile, the private part W_k^p is still encoded into s_k^p . By employing superposition coding, the transmitted signal $\mathbf{x}$ from the mobile user is given by

$$\mathbf{x} = \sum_{g \in \mathcal{G}} \mathbf{w}_{\mathcal{K}_g}^c s_{\mathcal{K}_g}^c + \sum_{k \in \mathcal{K}} \mathbf{w}_k^p s_k^p, \quad (10)$$

where $\mathbf{w}_{\mathcal{K}_g}^c \in \mathbb{C}^{N_t \times 1}$ is the precoding vector for the common stream $s_{\mathcal{K}_g}^c$. By plugging (10) into (2), the received signal y_k at EN k is given by

$$y_k = \sum_{g \in \mathcal{G}} \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_g}^c s_{\mathcal{K}_g}^c + \sum_{j \in \mathcal{K}} \mathbf{h}_k^H \mathbf{w}_j^p s_j^p + n_k. \quad (11)$$

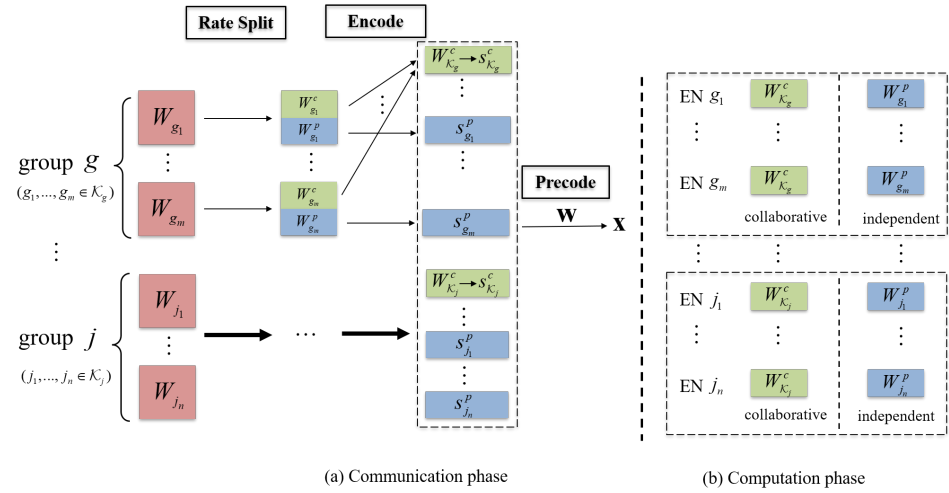


Figure 3. An illustration of the proposed multi-group information recycling scheme.

The decoding procedure of EN k in group g (i.e., $k \in \mathcal{K}_g$) is performed as follows: EN k in group g decodes its common stream $s_{\mathcal{K}_g}^c$ by treating other common streams and all private streams as interference and removes it from the received signal using SIC. Thereafter, EN k decodes its private stream s_k^p by treating the other common streams and private streams as interference. Accordingly, the signal-to-interference-plus-noise ratios (SINRs) of the common stream $s_{\mathcal{K}_g}^c$ and the private stream s_k^p to EN k in group g are given as follows

$$\gamma_{\mathcal{K}_g, k}^c = \frac{|\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_g}^c|^2}{\sum_{g' \in \mathcal{G}, g' \neq g} |\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c|^2 + \sum_{j \in \mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2}, g \in \mathcal{G}, k \in \mathcal{K}_g, \quad (12)$$

$$\gamma_k^p = \frac{|\mathbf{h}_k^H \mathbf{w}_k^p|^2}{\sum_{g' \in \mathcal{G}, g' \neq g} |\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c|^2 + \sum_{j \in \mathcal{K}, j \neq k} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2}, k \in \mathcal{K}. \quad (13)$$

With the above SINRs, the achievable rate R_k^p of the private stream s_k^p admits

$$R_k^p \leq B \log_2(1 + \gamma_k^p), \forall k \in \mathcal{K}. \quad (14)$$

To ensure that each common stream $s_{\mathcal{K}_g}^c$ can be successfully decoded by the ENs in group g , the achievable rate $R_{\mathcal{K}_g}^c$ of the common stream $s_{\mathcal{K}_g}^c$ should admit

$$R_{\mathcal{K}_g}^c \leq \min_{k \in \mathcal{K}_g} \left\{ B \log_2(1 + \gamma_{\mathcal{K}_g, k}^c) \right\}, \forall g \in \mathcal{G}. \quad (15)$$

For the computation phase, as shown in Figure 3b, the ENs in each group g will first collaboratively compute the common parts $\{W_k^c | k \in \mathcal{K}_g\}$ with information recycling. Next, each EN k will compute its private part W_k^p independently.

Remark 1. In the proposed scheme, each common stream $s_{\mathcal{K}_g}^c$ is decoded by a smaller number of ENs instead of all ENs. As a result, the communication bottleneck of the single common stream can be alleviated, which allows more resources to be allocated to form stronger common streams. Consequently, the size of the common parts $\{W_1^c, W_2^c, \dots, W_K^c\}$ can be increased, thereby boosting the advantage of information recycling.

4.2. Problem Formulation

As presented before, the original task W with size D will finally be converted into G combined common parts $\{W_{\mathcal{K}_1}^c, W_{\mathcal{K}_2}^c, \dots, W_{\mathcal{K}_G}^c\}$ and K private parts $\{W_1^p, W_2^p, \dots, W_K^p\}$ for transmission. The uplink transmission time $T_{u,\mathcal{K}_g}^c$ of the combined common part $W_{\mathcal{K}_g}^c$ is given by

$$T_{u,\mathcal{K}_g}^c = \frac{D_{\mathcal{K}_g}^c}{R_{\mathcal{K}_g}^c}, \quad (16)$$

where $D_{\mathcal{K}_g}^c$ is the size of $W_{\mathcal{K}_g}^c$. The uplink transmission time $T_{u,k}^p$ of the private part W_k^p of the EN k is given by

$$T_{u,k}^p = \frac{D_k^p}{R_k^p}, \quad (17)$$

where D_k^p is the size of W_k^p . As the common streams and the private streams are transmitted concurrently, the overall uplink transmission time T_u should admit

$$T_u \geq \max \left\{ \max_{g \in \mathcal{G}} T_{u,\mathcal{K}_g}^c, \max_{k \in \mathcal{K}} T_{u,k}^p \right\} = \max \left\{ \max_{g \in \mathcal{G}} \frac{D_{\mathcal{K}_g}^c}{R_{\mathcal{K}_g}^c}, \max_{k \in \mathcal{K}} \frac{D_k^p}{R_k^p} \right\}. \quad (18)$$

In the computation phase, the ENs in each group g will first collaboratively compute the combined common part $W_{\mathcal{K}_g}^c = \{W_k^c | k \in \mathcal{K}_g\}$. By (9), the corresponding computation latency $T_{e,\mathcal{K}_g}^c$ in group g is given by

$$T_{e,\mathcal{K}_g}^c = \frac{D_{\mathcal{K}_g}^c}{\sum_{k \in \mathcal{K}_g} C_k} + \eta, \quad (19)$$

where η is the cost for the signaling transmission (c.f., Section 3.2). Next, the ENs in group g start to compute their own private part, and the computation latency $T_{e,k}^p$ of the private part W_k^p is given by

$$T_{e,k}^p = \frac{D_k^p}{C_k}. \quad (20)$$

Accordingly, the computation latency $T_{e,\mathcal{K}_g}^p$ for the private parts in group g is given by

$$T_{e,\mathcal{K}_g}^p = \max_{k \in \mathcal{K}_g} \left\{ \frac{D_k^p}{C_k} \right\}. \quad (21)$$

Hence, the overall latency T_e for the total computation phase is given by

$$T_e = \max_{g \in \mathcal{G}} \left\{ T_{e,\mathcal{K}_g}^c + T_{e,\mathcal{K}_g}^p \right\} = \max_{g \in \mathcal{G}} \left\{ \frac{D_{\mathcal{K}_g}^c}{\sum_{k \in \mathcal{K}_g} C_k} + \eta + \max_{k \in \mathcal{K}_g} \left\{ \frac{D_k^p}{C_k} \right\} \right\}. \quad (22)$$

Based on the above, the expected latency minimization problem can be formulated as follows

$$\begin{aligned} \mathcal{P}_1 : \min_{\mathcal{N}, \mathbf{w}, \mathbf{D}, \mathbf{R}} \quad & T_u + \mathbb{E}[T_e] \\ \text{s.t.} \quad & (14), (15), (18), (22), \\ & \sum_{g \in \mathcal{G}} D_{\mathcal{K}_g}^c + \sum_{k \in \mathcal{K}} D_k^p = D, \end{aligned} \quad (23)$$

$$\text{tr}(\mathbf{w}\mathbf{w}^H) \leq P_t, \quad (24)$$

$$\bigcup_{g \in \mathcal{G}} \mathcal{K}_g = \mathcal{K}, \quad (25)$$

$$\mathcal{K}_i \cap \mathcal{K}_j = \emptyset, \forall i, j \in \mathcal{G}, \quad (26)$$

where $\mathcal{N} = [\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_G]$, $\mathbf{w} = [\mathbf{w}_{\mathcal{K}_1}^c, \mathbf{w}_{\mathcal{K}_2}^c, \dots, \mathbf{w}_{\mathcal{K}_G}^c, \mathbf{w}_1^p, \mathbf{w}_2^p, \dots, \mathbf{w}_K^p]$, $\mathbf{D} = [D_{\mathcal{K}_1}^c, D_{\mathcal{K}_2}^c, \dots, D_{\mathcal{K}_G}^c, D_1^p, D_2^p, \dots, D_K^p]$, and $\mathbf{R} = [R_{\mathcal{K}_1}^c, R_{\mathcal{K}_2}^c, \dots, R_{\mathcal{K}_G}^c, R_1^p, R_2^p, \dots, R_K^p]$ are compact representations of the corresponding variables. The expectation in $\mathcal{P}_1$ is over the randomness of the computation rates $\{C_k\}_{k=1}^K$ of the ENs (c.f. (8)). Constraint (24) restricts the total transmit power of the mobile user to P_t .

5. Optimization Algorithm

Since the original problem $\mathcal{P}_1$ is a non-convex combinatorial problem involving random variables, it turns out to be quite challenging to solve it directly. To this end, the optimization of the problem $\mathcal{P}_1$ is suboptimally decomposed into two steps: deriving an effective grouping strategy, and optimizing the rest of the variables under the given grouping strategy.

5.1. EN Grouping

The optimal grouping strategy can be identified by exhaustively searching through all potential grouping strategies $\mathcal{N}'$'s. However, this brute-force approach is computationally infeasible due to its prohibitively high complexity, which scales approximately as $O(G^K)$. To this end, alternative methods can be employed to derive a sub-optimal grouping strategy with significantly lower complexity. Intuitively, if EN i and EN j exhibit sufficiently different channel conditions, clustering them together and transmitting a shared common stream will likely harm the communication rate due to the mismatch in their channel characteristics. Hence, it is essential to design an algorithm that can cluster ENs based on channel similarity to obtain a good grouping strategy. With this consideration, a K-medoids [33] based grouping algorithm is designed by jointly considering the channel alignment and the channel strength for the general multi-antenna case (It is worth noting that other heuristic clustering algorithms, such as K-means [34], can also be used for EN grouping in the proposed scheme). Besides, a more efficient continuous partitioning-based grouping algorithm is proposed for the special case of single-antenna.

Remark 2. As collaborated by the simulation results in Section 6, both the K-medoids-based grouping algorithm and the continuous partitioning-based grouping algorithm can lead to a reasonably good grouping strategy.

For the K-medoids-based grouping algorithm, both channel strength and channel alignment are jointly considered to assess the similarity between the channel characteristics of different ENs. Accordingly, the distance of the EN i and j is defined as

$$d(i, j) = l \left| \frac{\|\mathbf{h}_i\|_2 - \|\mathbf{h}_j\|_2}{\max\{\|\mathbf{h}_i\|_2, \|\mathbf{h}_j\|_2\}} \right| + (1 - l) \left(1 - \frac{\|\mathbf{h}_i^H \mathbf{h}_j\|_2}{\|\mathbf{h}_i\|_2 \|\mathbf{h}_j\|_2} \right), \quad (27)$$

where $l \in [0, 1]$ is a weighting parameter; the first term and the second term account for the channel similarity in magnitude and direction, respectively.

Remark 3. The intuition of (22) is the following: as shown in (12) and (15), decreasing the variation in channel strengths within each group positively impacts the transmission of common streams. Similarly, reducing differences in channel alignment yields comparable benefits.

With this distance metric, the standard K-medoids clustering algorithm can be applied, yielding a grouping strategy with complexity $O(K^2G)$ [33]. The K-medoids-based grouping algorithm has significantly lower complexity compared to the naive brute-force search while

achieving a reasonable performance for problem $\mathcal{P}_1$. The specific process is summarized in Algorithm 2.

Algorithm 2 K-medoids-based grouping algorithm

Input: Channel state information $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K\}$, number of groups G , distance metric $d(i, j)$;

Output: Groups $\{\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_G\}$;

1: Randomly select G ENs $\mathcal{M} = \{M_1, M_2, \dots, M_G\}$ as the initial medoids and set $\mathcal{M}' = \mathcal{M}$;

2: **repeat**

3: Set $\mathcal{M} = \mathcal{M}'$ and $\mathcal{K}_g = \emptyset, \forall g \in \mathcal{G}$;

4: **for** $i = 1, \dots, K$ **do**

5: Compute $j = \arg \min_{j \in \mathcal{G}} d(i, M_j)$ and Set $\mathcal{K}_j = \mathcal{K}_j \cup \{i\}$;

6: **end for**

7: **for** $g = 1, \dots, G$ **do**

8: Select a new medoid M'_g that minimizes the total intra-group distance: $M'_g = \arg \min_{j \in \mathcal{K}_g} \sum_{i \in \mathcal{K}_g} d(i, j)$;

9: **end for**

10: **until** $\mathcal{M} = \mathcal{M}'$

In the single-antenna case, besides the K-medoids-based grouping algorithm, a more efficient continuous partitioning-based grouping algorithm is further proposed. Specifically, in this case, the channel alignment term in the distance metric becomes irrelevant, focusing solely on channel strength. Without loss of generality, assume that the channel strengths of all ENs satisfy $|h_1| \leq |h_2| \leq \dots \leq |h_K|$. The continuous partitioning-based grouping algorithm defines each group g as $\mathcal{K}_g = \{c_{g-1} + 1, c_{g-1} + 2, \dots, c_g\}$, where $c_0 = 0 < c_1 < \dots < c_{g-1} < c_g = K$. By selecting different values for $\{c_1, c_2, \dots, c_{g-1}\}$ from the set $\{1, 2, \dots, K-1\}$, this algorithm can generate $\binom{K-1}{G-1}$ candidate strategies, denoted as $\mathcal{C}$. Besides, through searching through all grouping strategies in $\mathcal{C}$, the continuous partitioning-based grouping algorithm can also achieve a good performance with much lower complexity than the brute-force search approach. In particular, this algorithm can achieve the same performance as the naive brute-force search approach in a special case of problem $\mathcal{P}_1$, where $D_k^p = D_1^p, \forall k \in \mathcal{K}$. The latency optimization problem of this special case can be formulated as follows

$$\begin{aligned} \mathcal{P}_2 : \min_{\mathcal{N}, \mathbf{w}, \mathbf{D}, \mathbf{R}} \quad & T_u + \mathbb{E}[T_e] \\ \text{s.t.} \quad & (14), (15), (18), (22), (23), (24), (25), (26), \\ & D_k^p = D_1^p, \forall k \in \mathcal{K}. \end{aligned} \quad (28)$$

Proposition 1. For problem $\mathcal{P}_2$, the optimal grouping strategy belongs to $\mathcal{C}$.

Proof. Please see Appendix A. $\square$

5.2. Optimization Algorithm with Given Grouping

In this subsection, the optimization algorithm with a given grouping strategy $\mathcal{N}$ will be introduced. Specifically, consider the following problem

$$\begin{aligned} \mathcal{P}_3 : \min_{\mathbf{w}, \mathbf{D}, \mathbf{R}} \quad & T_u + \mathbb{E}[T_e] \\ \text{s.t.} \quad & (14), (15), (18), (22), (23), (24). \end{aligned} \quad (29)$$

In the objective function of $\mathcal{P}_3$, due to the randomness of the computation capabilities C_k 's, it turns out to be difficult to find $\mathbb{E}[T_e]$ in a closed form. To this end, by (22), a lower bound of the objective function can be obtained as follows

$$\begin{aligned}
& T_u + \mathbb{E}[T_e] \\
& \geq T_u + \max_{g \in \mathcal{G}} \left\{ \mathbb{E} \left[\frac{D_{\mathcal{K}_g}^c}{\sum_{k \in \mathcal{K}_g} C_k} + \eta + \max_{k \in \mathcal{K}_g} \left\{ \frac{D_k^p}{C_k} \right\} \right] \right\} \\
& \geq T_u + \max_{g \in \mathcal{G}} \left\{ \mathbb{E} \left[\frac{D_{\mathcal{K}_g}^c}{\sum_{k \in \mathcal{K}_g} C_k} + \eta \right] + \max_{k \in \mathcal{K}_g} \left\{ \mathbb{E} \left[\frac{D_k^p}{C_k} \right] \right\} \right\}.
\end{aligned} \tag{30}$$

Accordingly, the following optimization problem will be considered in the sequel

$$\begin{aligned}
\mathcal{P}_4 : \min_{\mathbf{w}, \mathbf{D}, \mathbf{R}} \quad & T_u + \max_{g \in \mathcal{G}} \left\{ \mathbb{E} \left[\frac{D_{\mathcal{K}_g}^c}{\sum_{k \in \mathcal{K}_g} C_k} + \eta \right] + \max_{k \in \mathcal{K}_g} \left\{ \mathbb{E} \left[\frac{D_k^p}{C_k} \right] \right\} \right\} \\
\text{s.t.} \quad & (14), (15), (18), (23), (24).
\end{aligned} \tag{31}$$

Remark 4. As corroborated by the simulation results, using the solution obtained by optimizing this lower bound, a quite good expected processing latency can be achieved for $\mathcal{P}_1$.

By introducing the auxiliary variables $\mathbf{t} = [t_e, t_{e, \mathcal{K}_1}, \dots, t_{e, \mathcal{K}_G}]$, problem $\mathcal{P}_4$ can be readily transformed into the following equivalent problem

$$\begin{aligned}
\mathcal{P}_5 : \min_{\mathbf{w}, \mathbf{D}, \mathbf{R}, \mathbf{t}} \quad & T_u + t_e \\
\text{s.t.} \quad & (14), (15), (18), (23), (24), \\
& t_e \geq t_{e, \mathcal{K}_g}, \forall g \in \mathcal{G},
\end{aligned} \tag{32}$$

$$t_{e, \mathcal{K}_g} \geq \mathbb{E} \left[\frac{D_{\mathcal{K}_g}^c}{\sum_{k \in \mathcal{K}_g} C_k} + \eta \right] + \mathbb{E} \left[\frac{D_k^p}{C_k} \right], \forall k \in \mathcal{K}_g. \tag{33}$$

In the sequel, the non-convex problem $\mathcal{P}_5$ will be first transformed into a DC programming problem. In particular, by introducing the auxiliary variables $\mathbf{v} = \{v_{\mathcal{K}_g, k}^c, v_k^p | \forall g \in \mathcal{G}, \forall k \in \mathcal{K}_g\}$ to $\mathcal{P}_5$, constraints (14) and (15) can be equivalently written as

$$R_{\mathcal{K}_g}^c \leq B \log_2 v_{\mathcal{K}_g, k}^c, \forall g \in \mathcal{G}, k \in \mathcal{K}_g, \tag{34}$$

$$R_k^p \leq B \log_2 v_k^p, \forall k \in \mathcal{K}, \tag{35}$$

$$v_{\mathcal{K}_g, k}^c \leq 1 + \gamma_{\mathcal{K}_g, k}^c, \forall g \in \mathcal{G}, \forall k \in \mathcal{K}_g, \tag{36}$$

$$v_k^p \leq 1 + \gamma_k^p, \forall k \in \mathcal{K}. \tag{37}$$

By invoking (12) and (13), (36) and (37) can be further transformed into the following DC form

$$\begin{aligned}
& \sum_{g' \in \mathcal{G}, g' \neq g} \left| \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{h}_k^H \mathbf{w}_j^p \right|^2 + \sigma^2 - \frac{\sum_{g' \in \mathcal{G}} \left| \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{h}_k^H \mathbf{w}_j^p \right|^2 + \sigma^2}{v_{\mathcal{K}_g, k}^c} \leq 0, \\
& \forall g \in \mathcal{G}, k \in \mathcal{K}_g,
\end{aligned} \tag{38}$$

$$\begin{aligned}
& \sum_{g' \in \mathcal{G}, g' \neq g} \left| \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c \right|^2 + \sum_{j \in \mathcal{K}, j \neq k} \left| \mathbf{h}_k^H \mathbf{w}_j^p \right|^2 + \sigma^2 - \frac{\sum_{g' \in \mathcal{G}, g' \neq g} \left| \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{h}_k^H \mathbf{w}_j^p \right|^2 + \sigma^2}{v_k^p} \leq 0, \\
& \forall g \in \mathcal{G}, k \in \mathcal{K}_g.
\end{aligned} \tag{39}$$

Besides, it can be obtained that (18) can be rewritten into the following DC form

$$D_{\mathcal{K}_g}^c + \Omega(T_u, R_{\mathcal{K}_g}^c) \leq 0, \forall g \in \mathcal{G}, \quad (40)$$

$$D_k^p + \Omega(T_u, R_k^p) \leq 0, \forall k \in \mathcal{K}, \quad (41)$$

where $\Omega(a, b) \triangleq 1/4 \cdot (a - b)^2 - 1/4 \cdot (a + b)^2$. Besides, it is not difficult to verified that constraint (32) is linear about t_e and $t_{e, \mathcal{K}_g}$, and constraint (33) is linear about $t_{e, \mathcal{K}_g}$, $D_{\mathcal{K}_g}^c$ and D_k^p . Consequently, $\mathcal{P}_5$ can be equivalently transformed into the following DC programming problem

$$\begin{aligned} \mathcal{P}_6 : \min_{\mathbf{w}, \mathbf{D}, \mathbf{R}, \mathbf{t}, \nu} \quad & T_u + t_e \\ \text{s.t.} \quad & (23), (24), (32), (33), (34), (35), (38), (39), (40), (41). \end{aligned} \quad (42)$$

The above DC programming problem $\mathcal{P}_6$ can be efficiently solved by the CCCP technique [35], which is briefly introduced in the following. Consider DC programming problems which have the form

$$\begin{aligned} \min_x \quad & f_0(x) - g_0(x) \\ \text{s.t.} \quad & f_a(x) - g_a(x) \leq 0, \quad a = 1, \dots, m, \end{aligned} \quad (43)$$

where x is the optimization variable and f_a and g_a are convex for $a = 0, 1, \dots, m$. The fundamental concept of CCCP is to solve a sequence of approximate convex problems iteratively. At the $i + 1$ -th iteration, the approximated problem is given by the following:

$$\begin{aligned} \min_x \quad & f_0(x) - \hat{g}_0(x; x^i) \\ \text{s.t.} \quad & f_a(x) - \hat{g}_a(x; x^i) \leq 0, \quad a = 1, \dots, m, \end{aligned} \quad (44)$$

where $\hat{g}_a(x; x^i) = g_a(x^i) + \nabla g_a(x^i)(x - x^i)$ and x^i is the optimal solution of problem (44) at the i -th iteration. As the number of iterations increases, the optimal solution x^i will converge to a Karush–Kuhn–Tucker (KKT) point of the original problem [35].

For the considered problem $\mathcal{P}_6$, at the $(i + 1)$ -th iteration of the CCCP procedure, the concave parts in both the objective function and the constraints will be linearized at the point $(\mathbf{w}^i, \mathbf{D}^i, \mathbf{R}^i, \mathbf{t}^i, \nu^i)$, where $\mathbf{w}^i = [\mathbf{w}_{\mathcal{K}_1}^{c,i}, \mathbf{w}_{\mathcal{K}_2}^{c,i}, \dots, \mathbf{w}_{\mathcal{K}_G}^{c,i}, \mathbf{w}_1^{p,i}, \mathbf{w}_2^{p,i}, \dots, \mathbf{w}_K^{p,i}]$, $\mathbf{D}^i = [D_{\mathcal{K}_1}^{c,i}, D_{\mathcal{K}_2}^{c,i}, \dots, D_{\mathcal{K}_G}^{c,i}, D_1^{p,i}, D_2^{p,i}, \dots, D_K^{p,i}]$, $\mathbf{R}^i = [R_{\mathcal{K}_1}^{c,i}, R_{\mathcal{K}_2}^{c,i}, \dots, R_{\mathcal{K}_G}^{c,i}, R_1^{p,i}, R_2^{p,i}, \dots, R_K^{p,i}]$, $\mathbf{t}^i = [t_e^i, t_{e, \mathcal{K}_1}^i, \dots, t_{e, \mathcal{K}_G}^i]$ and $\nu^i = \{\nu_{\mathcal{K}_g, k}^{c,i}, \nu_k^{p,i} | \forall k \in \mathcal{K}, \forall g \in \mathcal{G}\}$ are compact representations of the corresponding optimization variables obtained in the i -th iteration. Specifically, as the objective function (42) and the constraints (24), (32), (33), (34) and (35) are convex, they do not require the treatment of linearization. For the constraints (38) and (39), by linearizing their concave parts, their approximations for the $(i + 1)$ -th iteration are, respectively, given by

$$\begin{aligned} & \sum_{g' \in \mathcal{G}, g' \neq g} |\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c|^2 + \sum_{j \in \mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2 + \left(\sum_{g' \in \mathcal{G}} |\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c|^2 + \sum_{j \in \mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2 \right) \times \\ & \nu_{\mathcal{K}_g, k}^c / \left(\nu_{\mathcal{K}_g, k}^{c,i} \right)^2 - 2\Re \left[\sum_{g' \in \mathcal{G}} \left(\mathbf{w}_{\mathcal{K}_{g'}}^{c,i} \right)^H \mathbf{h}_k \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c + \sum_{j \in \mathcal{K}} \left(\mathbf{w}_j^{p,i} \right)^H \mathbf{h}_k \mathbf{h}_k^H \mathbf{w}_j^p + \sigma^2 \right] / \nu_{\mathcal{K}_g, k}^{c,i} \leq 0, \\ & \forall g \in \mathcal{G}, k \in \mathcal{K}_g, \end{aligned} \quad (45)$$

$$\sum_{g' \in \mathcal{G}, g' \neq g} |\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c|^2 + \sum_{j \in \mathcal{K}, j \neq k} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2 + \left(\sum_{g' \in \mathcal{G}, g' \neq g} |\mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c|^2 + \sum_{j \in \mathcal{K}} |\mathbf{h}_k^H \mathbf{w}_j^p|^2 + \sigma^2 \right) \times$$

$$v_k^p / (v_k^{p,i})^2 - 2\Re \left[\sum_{g' \in \mathcal{G}, g' \neq g} \left(\mathbf{w}_{\mathcal{K}_{g'}}^{c,i} \right)^H \mathbf{h}_k \mathbf{h}_k^H \mathbf{w}_{\mathcal{K}_{g'}}^c + \sum_{j \in \mathcal{K}} \left(\mathbf{w}_j^{p,i} \right)^H \mathbf{h}_k \mathbf{h}_k^H \mathbf{w}_j^p + \sigma^2 \right] / v_k^{p,i} \leq 0, \\ \forall g \in \mathcal{G}, k \in \mathcal{K}_g. \quad (46)$$

Similarly, for the constraints (40) and (41), their approximations for the $(i+1)$ -th iteration are, respectively, given by

$$D_{\mathcal{K}_g}^c + \Phi(T_u, R_{\mathcal{K}_g}^c; T_u^i, R_{\mathcal{K}_g}^{c,i}) \leq 0, \forall g \in \mathcal{G}, \quad (47)$$

$$D_k^p + \Phi(T_u, R_k^p; T_u^i, R_k^{p,i}) \leq 0, \forall k \in \mathcal{K}, \quad (48)$$

where $\Phi^i(a, b; a^i, b^i) \triangleq \frac{1}{4}(a-b)^2 + \frac{1}{4}(a^i + b^i)^2 - \frac{1}{2}(a^i + b^i)(a+b)$ and T_u^i can be computed by using (18) and $(\mathbf{D}^i, \mathbf{R}^i)$.

Based on the above, given the optimal solution $(\mathbf{w}^i, \mathbf{D}^i, \mathbf{R}^i, \mathbf{t}^i, v^i)$ at the i -th iteration, the approximate convex problem of $\mathcal{P}_6$ for the $(i+1)$ -th iteration is given by

$$\mathcal{P}_7 : \min_{\mathbf{w}, \mathbf{D}, \mathbf{R}, \mathbf{t}, v} T_u + t_e \\ \text{s.t. (23), (24), (32), (33), (34), (35), (45), (46), (47), (48)}.$$

The convex problem $\mathcal{P}_7$ can be solved by standard methods [36]. Specifically, with the interior point method, the computational complexity for solving problem $\mathcal{P}_7$ is approximately $\mathcal{O}(((K+G)N_t)^{3.5})$ [36]. The above procedure will be repeated until convergence as summarized in Algorithm 3. Additionally, since the problem size of $\mathcal{P}_7$ does not scale with the number of iterations, the computation complexity of Algorithm 3 is $\mathcal{O}(((K+G)N_t)^{3.5}I)$, where I is the total number of iterations required for the convergence.

Algorithm 3 CCCP-based optimization for problem $\mathcal{P}_3$

Input: Tolerance ϵ , power constraint P_t , distributions of the computation rates of all ENs $\{C_k\}_{k=1}^K$, task size D ;
Output: Precoder $\mathbf{w}^*$, task partitioning vector $\mathbf{D}^*$, rate vector $\mathbf{R}^*$;
 1: Set iteration index $i = 0$;
 2: Find an initial feasible point $(\mathbf{w}^0, \mathbf{D}^0, \mathbf{R}^0, \mathbf{t}^0, v^0)$ of $\mathcal{P}_5$;
 3: **repeat**
 4: Using $(\mathbf{w}^i, \mathbf{D}^i, \mathbf{R}^i, \mathbf{t}^i, v^i)$ obtained from the last iteration, solve problem $\mathcal{P}_7$ and obtain an optimal solution $(\mathbf{w}^{i+1}, \mathbf{D}^{i+1}, \mathbf{R}^{i+1}, \mathbf{t}^{i+1}, v^{i+1})$;
 5: Compute the corresponding objective function O_E^{i+1} ;
 6: Update iteration index $i = i + 1$;
 7: **until** $|O_E^i - O_E^{i-1}| \leq \epsilon$;
 8: **return** $\mathbf{w}^* = \mathbf{w}^i, \mathbf{D}^* = \mathbf{D}^i, \mathbf{R}^* = \mathbf{R}^i$.

6. Simulation Results

In this section, simulation results are presented to collaborate on the effectiveness of the proposed scheme. In the simulations, the number K of the ENs is set to 8. The size of the original computation task is set to 1×10^6 bits. For communication, the one-ring model [37] is adopted to characterize the channel between the mobile user and each EN, assuming that the mobile user's antennas have a uniform linear array structure [25,38,39]. Besides, the location angle of EN k with respect to the mobile user's antennas is set to $\frac{2\pi k}{K}$ and the angular spread of the scattering associated with EN k follows uniform distribution on $[\frac{\pi}{36}, \frac{\pi}{18}]$. The average channel power gain ρ is set to -85 dB. The bandwidth B is set to 1 (MHz); the noise power σ^2 is set to 1×10^{-10} (W); the maximum transmit power P_t of the mobile user is set to 1 (W). The number of antennas of the mobile user N_t is set to 6. For computation, processing each bit of the computation task is assumed to take $\beta = 400$ CPU cycles. The maximum processor frequency f_e is set to 2.5 (GHz). The disturbed part Δ_k of the computation rate is assumed to follow Bernoulli distribution [29,40,41] with parameter

$p_k = 0.3$; that is, $\mathbb{P}\{\Delta_k = S\} = p_k$ and $\mathbb{P}\{\Delta_k = 0\} = 1 - p_k$. Here, S is the maximum value of the disturbed part and is set to $\frac{19}{20}\bar{C}$. The cost η for the signaling transmission of the collaboration is set to 1×10^{-2} second. The weighting parameter l in (27) is set to 0.5. To demonstrate the advantage of the proposed scheme, three baseline schemes are considered. The first baseline is the naive (uncoded) distributed edge computing with neither information recycling nor coded computing. The second baseline is conventional coded computing with the optimal coding parameter. The third baseline is the conventional information recycling for distributed edge computing, (i.e., the proposed scheme with $G = 1$ and $\mathcal{K}_{G_1} = \mathcal{K}$). Besides, the K-medoids algorithm and continuous partitioning algorithm are both demonstrated in the single-antenna case for the proposed scheme.

The task processing latency of the proposed scheme and those of the three baselines under different B with $N_t = 1$ and $N_t = 6$ are compared in Figure 4 and Figure 5, respectively. It can be seen that the performance gain brought by the proposed scheme is significant. For instance, when $N_t = 1$ and the bandwidth is 1 (MHz), the proposed scheme with continuous partitioning algorithm can substantially reduce the processing latency to about 0.29 (s) and achieves about 59.8 % and 51.0 % performance gains as compared to Baseline 2 and Baseline 3. Intuitively, the reason is that the proposed scheme does not introduce extra task coding redundancy as compared to Baseline 2 and alleviates the communication limitation by transmitting multiple common streams as compared to Baseline 3. Besides, it can be observed from Figure 4 that the proposed scheme with the continuous partitioning algorithm achieves a better performance than that with the heuristic K-medoids algorithm. To better demonstrate the advantage of the proposed scheme over Baseline 3, the common

streams' power ratio $\frac{\sum_{g=1}^G |\mathbf{w}_{\mathcal{K}_g}^c|^2}{P_t}$ and task ratio $\frac{\sum_{g=1}^G \mathcal{D}_{\mathcal{K}_g}}{D}$ of the proposed scheme and Baseline 3 are compared in Figure 6. It can be observed that the proposed scheme has a higher

power ratio $\frac{\sum_{g=1}^G |\mathbf{w}_{\mathcal{K}_g}^c|^2}{P_t}$ and task ratio $\frac{\sum_{g=1}^G \mathcal{D}_{\mathcal{K}_g}}{D}$ as compared to Baseline 3. For example, when the bandwidth is 0.5 (MHz), the power ratio and task ratio of the proposed scheme are 0.91 and 0.75, respectively, while the power ratio and task ratio of Baseline 3 are 0.62 and 0.23, respectively. The reason is that, by alleviating the rate limitation of the single common stream through the transmission of multiple common streams, the proposed scheme can allocate more power to enlarge the common streams, thereby increasing the task ratio of the common part(s) and promoting computation collaboration for processing latency reduction.

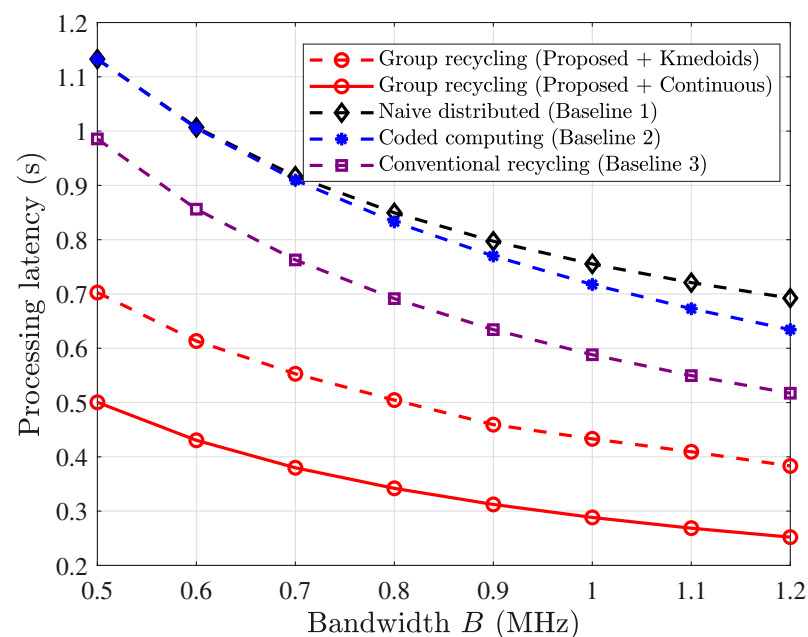


Figure 4. Comparison of processing latency under different B ($N_t = 1$).

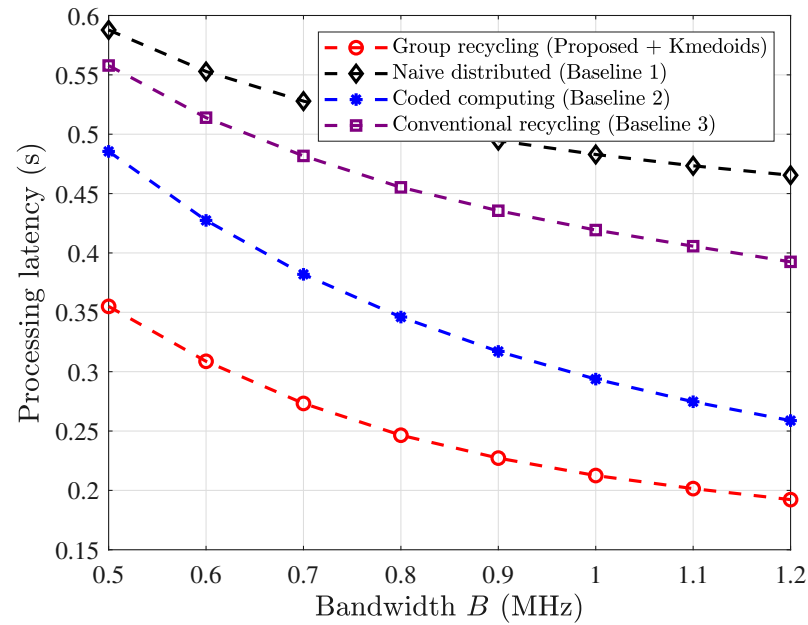


Figure 5. Comparison of the processing latency under different B ($N_t = 6$).

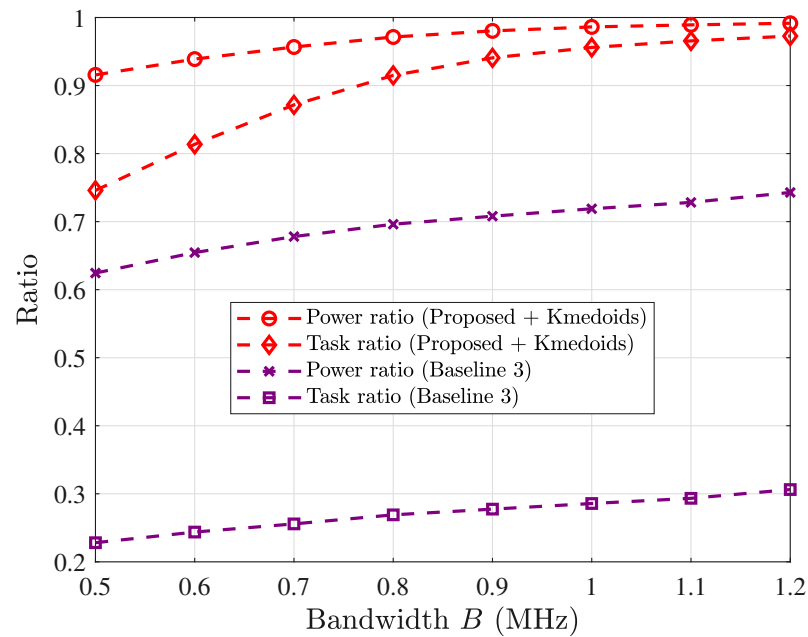


Figure 6. Comparison of the power ratio $\frac{\sum_{g=1}^G |\mathbf{w}_{K_g}^t|^2}{P_t}$ and task ratio $\frac{\sum_{g=1}^G \mathcal{D}_{K_g}}{D}$ under different B ($N_t = 6$).

Besides, the task processing latency of the proposed scheme and those of the three baselines under different ρ with $N_t = 1$ and $N_t = 6$ are compared in Figure 7 and Figure 8, respectively. It can be seen that the advantages of the proposed scheme are still robust. For instance, when the average power gain ρ is -90 (dB), the proposed scheme can substantially reduce the processing latency to about 0.39 (s) and 0.30 (s), respectively, which further demonstrates the effectiveness of the proposed scheme.

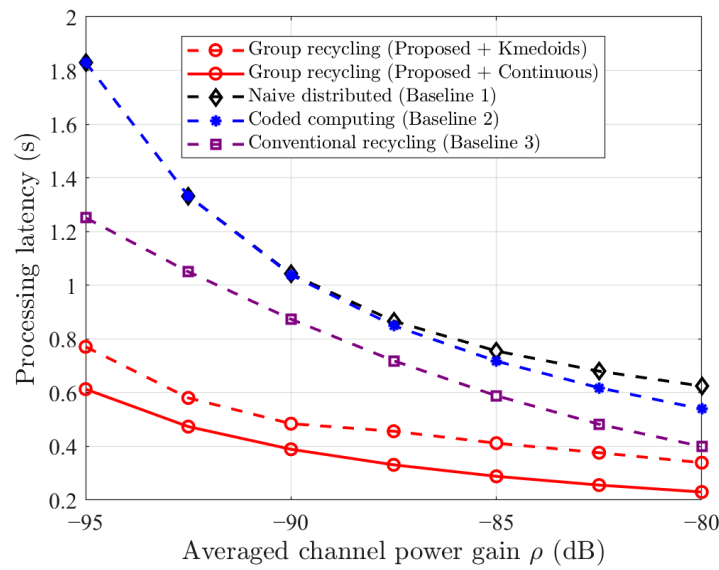


Figure 7. Comparison of processing latency under different ρ ($N_t = 1$).

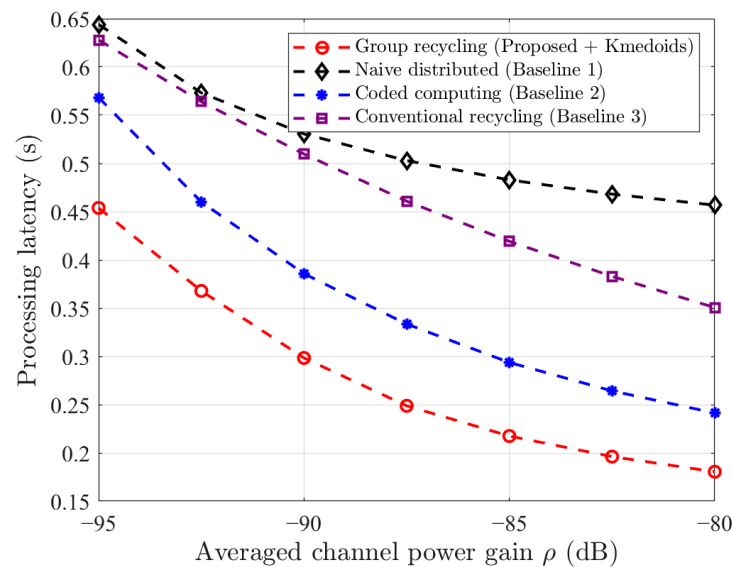


Figure 8. Comparison of the processing latency under different ρ ($N_t = 6$).

Moreover, the communication phase latency and the computation phase latency under different f_e with $N_t = 6$ are compared in Figure 9 and Figure 10, respectively. It can be seen that the proposed scheme achieves both lower communication latency and computation latency as compared to Baseline 3, further demonstrating its superiority in overcoming the communication limitation of the single common stream to facilitate communication and promote information recycling for enhanced computation collaboration. Meanwhile, it can be seen from Figure 9 that, the proposed scheme has much lower communication latency than Baseline 2. The reason is that, due to the introduction of task redundancy for straggler mitigation, Baseline 2 (i.e., coded computing) incurs additional communication overhead, resulting in relatively high communication latency.

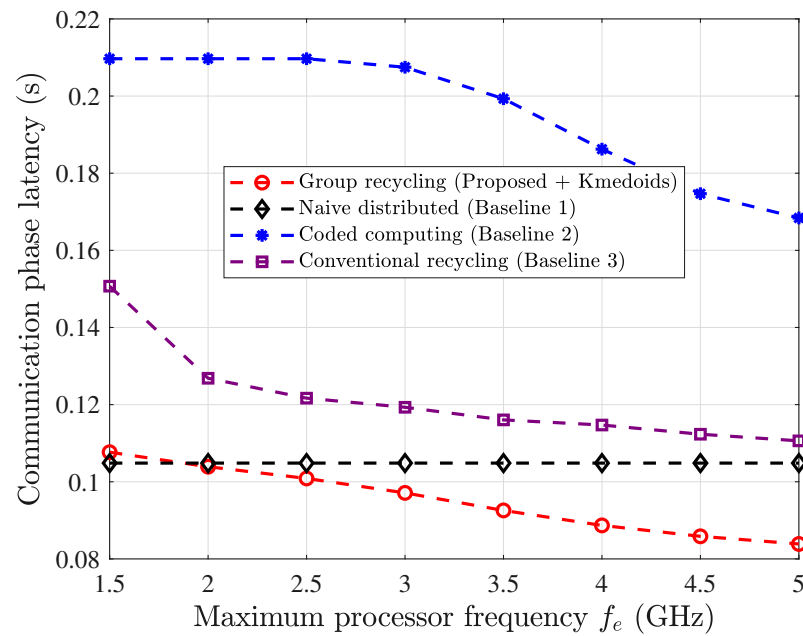


Figure 9. Comparison of communication phase latency under different f_e ($N_t = 6$).

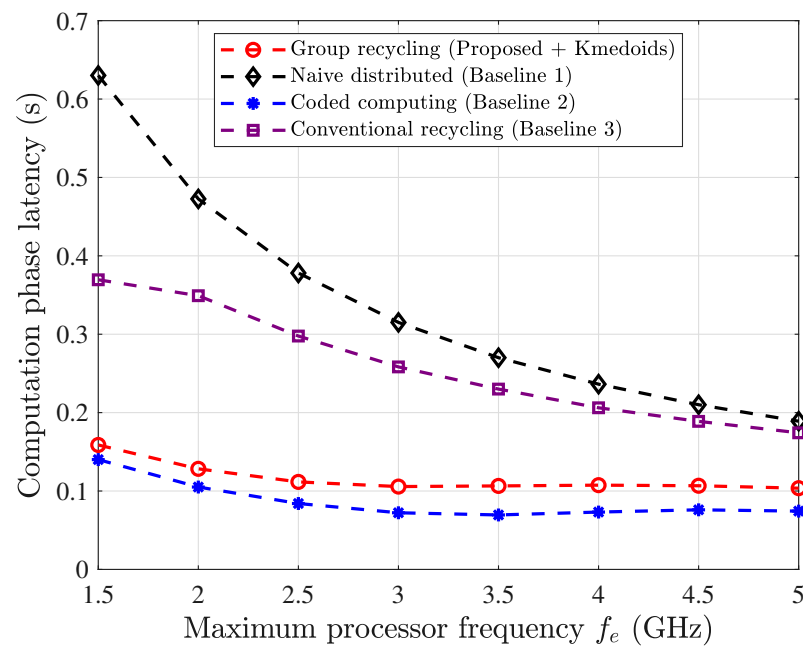


Figure 10. Comparison of computation phase latency under different f_e ($N_t = 6$).

Furthermore, the task processing latency of the proposed scheme and those of the three baselines with $K = 12$ under different B and ρ are compared in Figure 11 and Figure 12, respectively.

It can be observed from Figures 11 and 12 that, the performance gain brought by the proposed scheme is still significant. For example, when the bandwidth is 0.5 (MHz), the proposed scheme can substantially reduce the processing latency to about 0.33 (s) and achieve about 28.5 % and 31.3% performance gains as compared to Baseline 2 and Baseline 3.

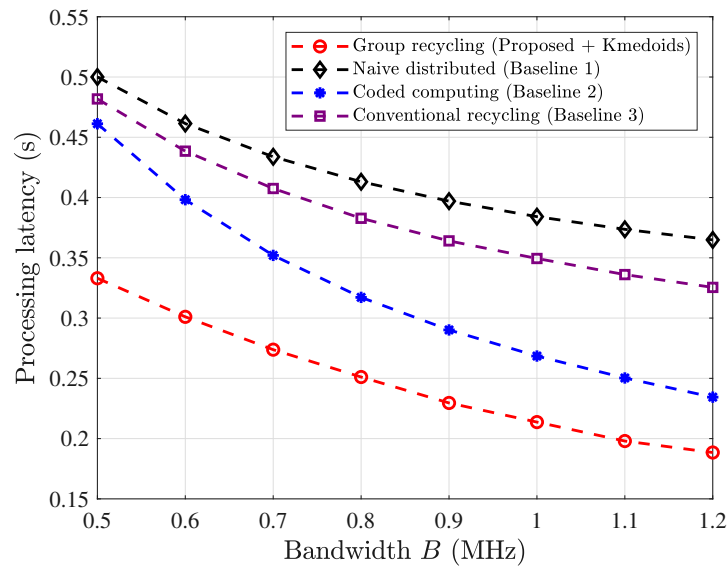


Figure 11. Comparison of processing latency under different B ($K = 12$).

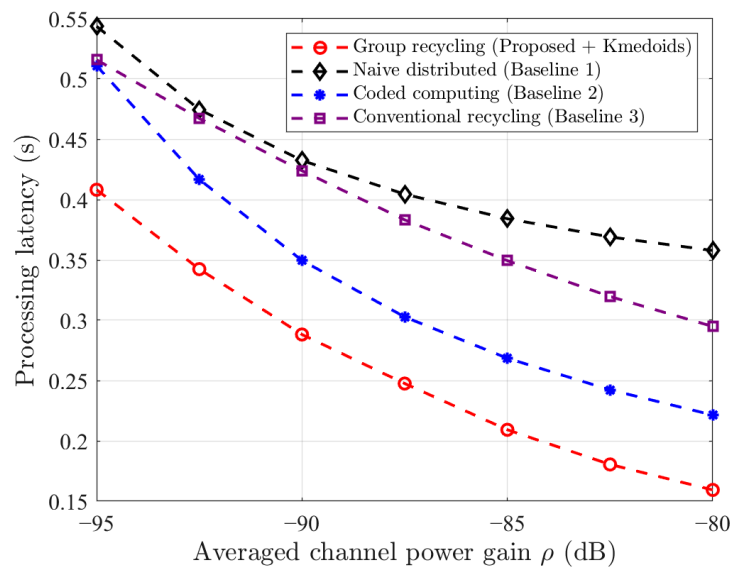


Figure 12. Comparison of processing latency under different ρ ($K = 12$).

7. Conclusions and Further Works

In this work, a novel multi-group information recycling scheme is proposed for rate-splitting-based distributed edge computing, which can better utilize the information recycling mechanism for processing latency reduction. In particular, by clustering ENs into different groups and transmitting multiple common streams, the proposed scheme can overcome the communication limitation of the single common stream and allocate more resources to the common streams to foster information recycling for straggler mitigation. To achieve an effective EN grouping strategy, a K-medoids-based algorithm is developed by jointly considering the channel alignment and channel strength for the general multi-antenna case, and a more efficient continuous partitioning-based algorithm is proposed for the specific case of single-antenna. Besides, through bounding the corresponding latency optimization problem via DC programming, a CCCP-based algorithm is developed to find a good solution. Simulation results show that the proposed scheme can substantially outperform the baselines and greatly enhance the system's performance.

Extension to group number optimization, advanced grouping strategies exploration, practical implementation and compatibility with existing infrastructures, imperfect channel state information problem, security and vulnerability issues related to the common streams transmission, energy consumption optimization and distributed network congestion, as well as integrating information recycling into other non-orthogonal transmission techniques for distributed edge computing are all worthwhile future works.

Author Contributions: Conceptualization, W.L. and X.H.; methodology, W.L. and X.H.; software, W.L.; validation, W.L.; formal analysis, W.L.; writing—original draft preparation, W.L. and X.H.; writing—review and editing, W.L. and X.H.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Proof of Proposition 1

Proof. Consider a grouping strategy $\mathcal{N}' = \{\mathcal{K}'_g\}_{g=1}^G \notin \mathcal{C}$, which admits

$$\exists a, b \in \mathcal{G}, \quad \left[\inf \mathcal{K}'_a, \sup \mathcal{K}'_a \right] \cap \left[\inf \mathcal{K}'_b, \sup \mathcal{K}'_b \right] \neq \emptyset. \quad (\text{A1})$$

Without loss of generality, assume that $\inf \mathcal{K}'_b < \inf \mathcal{K}'_a$ and the optimal point of $\mathcal{P}_2$ is $(\mathcal{N}', \mathbf{w}', \mathbf{D}', \mathbf{R}')$. By contradiction, it can be verified that $\mathcal{K}'_a \cap [\inf \mathcal{K}'_b, \sup \mathcal{K}'_b] \neq \emptyset$. In the following, it will be proved that the performance of $(\mathcal{N}', \mathbf{w}', \mathbf{D}', \mathbf{R}')$ can be achieved by another point $(\mathcal{N}, \mathbf{w}, \mathbf{D}, \mathbf{R})$ constructed as follows

$$\mathcal{K}_g = \mathcal{K}'_g, \forall g \in \mathcal{G}, g \neq a, b, \quad (\text{A2})$$

$$\mathcal{K}_a = \{\mathcal{K}'_a \setminus o\} \cup \{\sup \mathcal{K}'_b\}, \quad (\text{A3})$$

$$\mathcal{K}_b = \{\mathcal{K}'_b \setminus \sup \mathcal{K}'_b\} \cup \{o\}, \quad (\text{A4})$$

$$\mathbf{w}_{\mathcal{K}_g}^c = \mathbf{w}_{\mathcal{K}'_g}^{c'}, \forall g \in \mathcal{G}, \quad (\text{A5})$$

$$\mathbf{w}_k^p = \mathbf{w}_k^{p'}, \forall k \in \mathcal{K}, k \neq o, \sup \mathcal{K}'_b, \quad (\text{A6})$$

$$\mathbf{w}_o^p = \mathbf{w}_o^{p'} \left(\frac{\sum_{g' \in \mathcal{G}, g' \neq b} \left| \mathbf{w}_{\mathcal{K}'_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_o|^2}{\sum_{g' \in \mathcal{G}, g' \neq a} \left| \mathbf{w}_{\mathcal{K}'_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_o|^2} \right)^{1/2}, \quad (\text{A7})$$

$$\mathbf{w}_{\sup \mathcal{K}'_b}^p = \mathbf{w}_{\sup \mathcal{K}'_b}^{p'} \left(\frac{\sum_{g' \in \mathcal{G}, g' \neq a} \left| \mathbf{w}_{\mathcal{K}'_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_{\sup \mathcal{K}'_b}|^2}{\sum_{g' \in \mathcal{G}, g' \neq b} \left| \mathbf{w}_{\mathcal{K}'_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_{\sup \mathcal{K}'_b}|^2} \right)^{1/2}, \quad (\text{A8})$$

$$\mathbf{D} = \mathbf{D}', \quad (\text{A9})$$

$$\mathbf{R} = \mathbf{R}', \quad (\text{A10})$$

where $o \in \mathcal{K}'_a \cap [\inf \mathcal{K}'_b, \sup \mathcal{K}'_b]$. The feasibility of the point $(\mathcal{N}, \mathbf{w}, \mathbf{D}, \mathbf{R})$ is presented in the following. First, it is shown that the optimal point $(\mathcal{N}', \mathbf{w}', \mathbf{D}', \mathbf{R}')$ satisfies $\min_{k \in \mathcal{K}} R_k^{p'} \leq$

$\min_{k \in \mathcal{K}} \log(1 + \gamma_k^{p'})$. Assume that $\min_{k \in \mathcal{K}} R_k^{p'} > \min_{k \in \mathcal{K}} \log(1 + \gamma_k^{p'})$ and let $j = \arg \min_{k \in \mathcal{K}} \{\log(1 + \gamma_k^{p'})\}$. As $R_j^{p'} \leq \log(1 + \gamma_j^{p'})$ and $\min_{k=1}^K \{R_k^{p'}\} \leq R_j^{p'}$, a contradiction arises, following that $\min_{k=1}^K \{R_k^{p'}\} \leq \min_{k=1}^K \{\log(1 + \gamma_k^{p'})\}$. Next, it is shown that the optimal point $(\mathcal{N}', \mathbf{w}', \mathbf{D}', \mathbf{R}')$ admits

$$\gamma_i^{p'} = \gamma_j^{p'}, \forall i, j \in \mathcal{K}. \quad (\text{A11})$$

Assume that $\exists k' \in \mathcal{K}, \gamma_{k'}^{p'} > \min_{k \in \mathcal{K}} \gamma_k^{p'}$, where $\gamma_{k'}^{p'}$ admits (13). Multiplying $\mathbf{w}_{k'}^{p'}$ by a factor α , where $0 < \alpha < 1$, a new precoder $\mathbf{w}'' = [\mathbf{w}_{\mathcal{K}_1}^{c'}, \dots, \mathbf{w}_{\mathcal{K}_G}^{c'}, \mathbf{w}_1^{p'}, \dots, \alpha \mathbf{w}_{k'}^{p'}, \dots, \mathbf{w}_K^{p'}]$ can be constructed, where $\mathbf{w}''$ admits that

$$\begin{aligned} \gamma_{k'}^{p''} &= \min_{k \in \mathcal{K}} \gamma_k^{p'}, \\ \gamma_k^{p''} &> \gamma_k^{p'}, \quad \forall k \in \mathcal{K} \setminus k', \\ \gamma_{\mathcal{K}_g, k}^{c''} &> \gamma_{\mathcal{K}_g, k}^{c'}. \end{aligned} \quad (\text{A12})$$

By (14) and (15), a feasible point $(\mathcal{N}', \mathbf{w}'', \mathbf{D}', \mathbf{R}'')$ can be further constructed, satisfying

$$\begin{aligned} R_k^{p''} &= \min_{k \in \mathcal{K}} \log(1 + \gamma_k^{p''}) = \min_{k \in \mathcal{K}} \log(1 + \gamma_k^{p'}) \geq \min_{k \in \mathcal{K}} R_k^{p'}, \forall k \in \mathcal{K}, \\ R_{\mathcal{K}_g}^{c''} &= R_{\mathcal{K}_g}^{c'}, \forall g \in \mathcal{G}. \end{aligned} \quad (\text{A13})$$

By (18) and (22), it can be readily obtained that the point $(\mathcal{N}', \mathbf{w}'', \mathbf{D}', \mathbf{R}'')$ can achieve the same performance as the point $(\mathcal{N}', \mathbf{w}', \mathbf{D}', \mathbf{R}')$. Further using the spare power $|(1 - \alpha) \mathbf{w}_{k'}^{p'}|^2$, a point $(\mathcal{N}', \frac{P_t}{\text{tr}(\mathbf{w}' \mathbf{w}'^H) - |(1 - \alpha) \mathbf{w}_{k'}^{p'}|^2} \mathbf{w}'', \mathbf{D}', \hat{\mathbf{R}})$ with better performance, where $\hat{\mathbf{R}} \succ \mathbf{R}''$, can be constructed, thus contradicting the assumption. Therefore, it follows that the point $(\mathcal{N}', \mathbf{w}', \mathbf{D}', \mathbf{R}')$ admits $\forall k' \in \mathcal{K}, \gamma_{k'}^{p'} = \min_{k \in \mathcal{K}} \gamma_k^{p'}$ (i.e., $\gamma_i^{p'} = \gamma_j^{p'}, \forall i, j \in \mathcal{K}$), which yields that

$$\begin{aligned} \frac{|\mathbf{h}_0^H \mathbf{w}_0^{p'}|^2}{\sum_{g' \in \mathcal{G}, g' \neq a} |\mathbf{h}_0^H \mathbf{w}_{\mathcal{K}_{g'}}^{c'}|^2 + \sum_{j \in \mathcal{K}, j \neq 0} |\mathbf{h}_0^H \mathbf{w}_j^{p'}|^2 + \sigma^2} &= \frac{|\mathbf{h}_{\sup \mathcal{K}_b}^H \mathbf{w}_{\sup \mathcal{K}_b}^{p'}|^2}{\sum_{g' \in \mathcal{G}, g' \neq b} |\mathbf{h}_{\sup \mathcal{K}_b}^H \mathbf{w}_{\mathcal{K}_{g'}}^{c'}|^2 + \sum_{j \in \mathcal{K}, j \neq \sup \mathcal{K}_b} |\mathbf{h}_{\sup \mathcal{K}_b}^H \mathbf{w}_j^{p'}|^2 + \sigma^2} \\ \frac{|\mathbf{h}_0^H \mathbf{w}_0^{p'}|^2}{\sum_{g' \in \mathcal{G}, g' \neq a} |\mathbf{h}_0^H \mathbf{w}_{\mathcal{K}_{g'}}^{c'}|^2 + \sum_{j \in \mathcal{K}} |\mathbf{h}_0^H \mathbf{w}_j^{p'}|^2 + \sigma^2} &= \frac{|\mathbf{h}_{\sup \mathcal{K}_b}^H \mathbf{w}_{\sup \mathcal{K}_b}^{p'}|^2}{\sum_{g' \in \mathcal{G}, g' \neq b} |\mathbf{h}_{\sup \mathcal{K}_b}^H \mathbf{w}_{\mathcal{K}_{g'}}^{c'}|^2 + \sum_{j \in \mathcal{K}} |\mathbf{h}_{\sup \mathcal{K}_b}^H \mathbf{w}_j^{p'}|^2 + \sigma^2}. \end{aligned} \quad (\text{A14})$$

Accordingly, in the special case of single-antenna, it can be computed that

$$\begin{aligned}
& \left| \mathbf{w}_o^p \right|^2 + \left| \mathbf{w}_{\sup \mathcal{K}_b}^p \right|^2 - \left| \mathbf{w}_o^{p'} \right|^2 - \left| \mathbf{w}_{\sup \mathcal{K}_b}^{p'} \right|^2 \\
&= \left(\left| \mathbf{w}_{\mathcal{K}_a}^{c'} \right|^2 - \left| \mathbf{w}_{\mathcal{K}_b}^{c'} \right|^2 \right) \left(\frac{\left| \mathbf{w}_o^{p'} \right|^2}{\sum_{g' \in \mathcal{G}, g' \neq a} \left| \mathbf{w}_{\mathcal{K}_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_o|^2} - \frac{\left| \mathbf{w}_{\sup \mathcal{K}_b}^{p'} \right|^2}{\sum_{g' \in \mathcal{G}, g' \neq b} \left| \mathbf{w}_{\mathcal{K}_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_{\sup \mathcal{K}_b}|^2} \right) \\
&= 0.
\end{aligned} \tag{A15}$$

By (A5), (A6), and (A15), it can be obtained that $\mathbf{w}$ is feasible for $\mathcal{P}_2$. Moreover, by (13), (A7) and (A15), it can be computed that

$$\begin{aligned}
\gamma_o^p &= \frac{\left| \mathbf{w}_o^p \right|^2}{\sum_{g' \in \mathcal{G}, g' \neq b} \left| \mathbf{w}_{\mathcal{K}_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^p \right|^2 - \left| \mathbf{w}_o^p \right|^2 + \sigma^2 / |\mathbf{h}_o|^2} \\
&= \frac{1}{\frac{\sum_{g' \in \mathcal{G}, g' \neq b} \left| \mathbf{w}_{\mathcal{K}_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}} \left| \mathbf{w}_j^{p'} \right|^2 + \sigma^2 / |\mathbf{h}_o|^2}{\left| \mathbf{w}_o^p \right|^2} - 1} \\
&= \frac{\left| \mathbf{h}_o^H \mathbf{w}_o^{p'} \right|^2}{\sum_{g' \in \mathcal{G}, g' \neq a} \left| \mathbf{h}_o^H \mathbf{w}_{\mathcal{K}_{g'}}^{c'} \right|^2 + \sum_{j \in \mathcal{K}, j \neq o} \left| \mathbf{h}_o^H \mathbf{w}_j^{p'} \right|^2 + \sigma^2} = \gamma_o^{p'}.
\end{aligned} \tag{A16}$$

Similarly, it can be derived that $\gamma_{\sup \mathcal{K}_b}^p = \gamma_{\sup \mathcal{K}_b}^{p'}$. By (12), (13), (A5), (A6), and (A15), it can be readily obtained that

$$\begin{aligned}
\min_{k \in \mathcal{K}_g} \gamma_{\mathcal{K}_g}^c &= \min_{k \in \mathcal{K}_g'} \gamma_{\mathcal{K}_g'}^{c'}, \forall g \in \mathcal{G}, \\
\gamma_k^p &= \gamma_k^{p'}, \forall k \in \mathcal{K}, k \neq o, \sup \mathcal{K}_b.
\end{aligned} \tag{A17}$$

Further, by (14) and (15), it can be obtained that the point $(\mathcal{N}, \mathbf{w}, \mathbf{D}, \mathbf{R})$ is feasible for $\mathcal{P}_2$ and achieves the same performance as the optimal point of the previous grouping strategy $\mathcal{N}'$. Note that $\mathcal{N}$ and $\mathcal{N}'$ admits the following relationship

$$\mathcal{K}_a' \cap [\inf \mathcal{K}_b', \sup \mathcal{K}_b'] = \mathcal{K}_a \cap [\inf \mathcal{K}_b, \sup \mathcal{K}_b] \cup \{o\}. \tag{A18}$$

Hence, by recursively adjusting the grouping strategy as (A2)–(A4), a grouping strategy $\mathcal{N}^* = \{\mathcal{K}_g^*\}_{g=1}^G$ can achieve the same performance as $\mathcal{N}'$, where

$$\begin{aligned}
\mathcal{K}_g^* &= \mathcal{K}_g', \forall g \in \mathcal{G}, g \neq a, b, \\
\mathcal{K}_a^* \cap [\inf \mathcal{K}_b^*, \sup \mathcal{K}_b^*] &= \emptyset.
\end{aligned} \tag{A19}$$

Further adjusting the grouping strategy $\mathcal{N}^*$, it can be obtained that a grouping strategy $\mathcal{N}^{**} = \{\mathcal{K}_g^{**}\}_{g=1}^G$ can achieve the same performance as $\mathcal{N}'$, where

$$\mathcal{K}_i^{**} \cap [\inf \mathcal{K}_j^{**}, \sup \mathcal{K}_j^{**}] = \emptyset, \forall i, j \in \mathcal{G}. \tag{A20}$$

It can be readily verified that $\mathcal{N}^{**}$ belongs to $\mathcal{C}$, which yields the fact that $\forall \mathcal{N}' \notin \mathcal{C}$ is not the optimal grouping strategy. Thus, the proposition is proved. $\square$

References

1. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tuts.* **2017**, *19*, 2322–2358. [\[CrossRef\]](#)
2. Wang, X.; Han, Y.; Leung, V.C.M.; Niyato, D.; Yan, X.; Chen, X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Commun. Surv. Tuts.* **2020**, *22*, 869–904. [\[CrossRef\]](#)
3. Gong, X. Delay-Optimal Distributed Edge Computing in Wireless Edge Networks. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 6–9 July 2020.
4. Chen, M.; Gunduz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V. Distributed Learning in Wireless Networks: Recent Progress and Future Challenges. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3579–3605. [\[CrossRef\]](#)
5. Nguyen, C.T.; Nguyen, D.N.; Hoang, D.T.; Phan, K.T.; Niyato, D.; Pham, H.A.; Dutkiewicz, E. Elastic Resource Allocation for Coded Distributed Computing Over Heterogeneous Wireless Edge Networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 2636–2649. [\[CrossRef\]](#)
6. Dutta, S.; Jeong, H.; Yang, Y.; Cadambe, V.; Low, T.M.; Grover, P. Addressing unreliability in emerging devices and non-von Neumann architectures using coded computing. *Proc. IEEE* **2020**, *108*, 1219–1234. [\[CrossRef\]](#)
7. Zhu, J.; Li, S. Generalized Lagrange Coded Computing: A Flexible Computation-Communication Tradeoff. In Proceedings of the IEEE ISIT, Espoo, Finland, 26 June–1 July 2022.
8. Ng, J.S.; Lim, W.Y.B.; Luong, N.C.; Xiong, Z.; Asheralieva, A.; Niyato, D.; Leung, C.; Miao, C. A Comprehensive Survey on Coded Distributed Computing: Fundamentals, Challenges, and Networking Applications. *IEEE Commun. Surv. Tuts.* **2021**, *23*, 1800–1837. [\[CrossRef\]](#)
9. Lee, K.; Lam, M.; Pedarsani, R.; Papailiopoulos, D.; Ramchandran, K. Speeding Up Distributed Machine Learning Using Codes. *IEEE Trans. Inf. Theory* **2018**, *64*, 1514–1529. [\[CrossRef\]](#)
10. Shah, N.B.; Lee, K.; Ramchandran, K. When Do Redundant Requests Reduce Latency? *IEEE Trans. Commun.* **2016**, *64*, 715–722. [\[CrossRef\]](#)
11. Wang, D.; Joshi, G.; Wornell, G.W. Efficient Straggler Replication in Large-Scale Parallel Computing. *ACM Trans. Model. Perform. Eval. Comput. Syst.* **2019**, *4*, 1–23. [\[CrossRef\]](#)
12. Behrouzi-Far, A.; Soljanin, E. Efficient Replication for Fast and Predictable Performance in Distributed Computing. *IEEE/ACM Trans. Netw.* **2021**, *29*, 1467–1476. [\[CrossRef\]](#)
13. Ramamoorthy, A.; Das, A.B.; Tang, L. Straggler-Resistant Distributed Matrix Computation via Coding Theory: Removing a Bottleneck in Large-Scale Data Processing. *IEEE Signal Process. Mag.* **2020**, *37*, 136–145. [\[CrossRef\]](#)
14. Blumofe, R.D.; Leiserson, C.E. Scheduling multithreaded computations by work stealing. *J. ACM* **1999**, *46*, 720–748. [\[CrossRef\]](#)
15. Harlap, A.; Cui, H.; Dai, W.; Wei, J.; Ganger, G.R.; Gibbons, P.B.; Gibson, G.A.; Xing, E.P. Addressing the straggler problem for iterative convergent parallel ML. In Proceedings of the ACM Symposium on Cloud Computing, Santa Clara, CA, USA, 5–7 October 2016.
16. Liang, W.; Li, T.; He, X. Information Recycling Assisted Collaborative Edge Computing for Distributed Learning. In Proceedings of the IEEE INFOCOM WKSHPS, Hoboken, NJ, USA, 17–20 May 2023.
17. Clerckx, B.; Mao, Y.; Jorswieck, E.A.; Yuan, J.; Love, D.J.; Erkip, E.; Niyato, D. A Primer on Rate-Splitting Multiple Access: Tutorial, Myths, and Frequently Asked Questions. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1265–1308. [\[CrossRef\]](#)
18. Mao, Y.; Dizdar, O.; Clerckx, B.; Schober, R.; Popovski, P.; Poor, H.V. Rate-Splitting Multiple Access: Fundamentals, Survey, and Future Research Trends. *IEEE Commun. Surv. Tuts.* **2022**, *24*, 2073–2126. [\[CrossRef\]](#)
19. Dai, L.; Wang, B.; Ding, Z.; Wang, Z.; Chen, S.; Hanzo, L. A Survey of Non-Orthogonal Multiple Access for 5G. *IEEE Commun. Surv. Tuts.* **2018**, *20*, 2294–2323. [\[CrossRef\]](#)
20. Park, S.H.; Lee, H. Completion Time Minimization of Fog-RAN-Assisted Federated Learning With Rate-Splitting Transmission. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10209–10214. [\[CrossRef\]](#)
21. Nguyen, T.H.; Park, H.; Kim, M.; Park, L. DRL-Enabled RSMA-Assisted Task Offloading in Multi-Server Edge Computing. In Proceedings of the ICOIN, Ho Chi Minh City, Vietnam, 17–19 January 2024.
22. Zhu, B.; Chi, K.; Liu, J.; Yu, K.; Mumtaz, S. Efficient Offloading for Minimizing Task Computation Delay of NOMA-Based Multiaccess Edge Computing. *IEEE Trans. Commun.* **2022**, *70*, 3186–3203. [\[CrossRef\]](#)
23. Ding, Z.; Fan, P.; Poor, H.V. Impact of Non-Orthogonal Multiple Access on the Offloading of Mobile Edge Computing. *IEEE Trans. Wirel. Commun.* **2019**, *67*, 375–390. [\[CrossRef\]](#)
24. Diamanti, M.; Pelekis, C.; Tsiropoulou, E.E.; Papavassiliou, S. Delay Minimization for Rate-Splitting Multiple Access-Based Multi-Server MEC Offloading. *IEEE/ACM Trans. Netw.* **2024**, *32*, 1035–1047. [\[CrossRef\]](#)
25. Dai, M.; Clerckx, B.; Gesbert, D.; Caire, G. A Rate Splitting Strategy for Massive MIMO With Imperfect CSIT. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 4611–4624. [\[CrossRef\]](#)
26. Wang, Y.; Wong, V.W.S.; Wang, J. Flexible Rate-Splitting Multiple Access With Finite Blocklength. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1398–1412. [\[CrossRef\]](#)
27. Ko, D.; Chae, S.H.; Choi, W. MDS Coded Task Offloading in Stochastic Wireless Edge Computing Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 2107–2121. [\[CrossRef\]](#)
28. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, A.S. Straggler Mitigation in Distributed Matrix Multiplication: Fundamental Limits and Optimal Coding. *IEEE Trans. Inf. Theory* **2020**, *66*, 1920–1933. [\[CrossRef\]](#)

29. Peng, P.; Soljanin, E.; Whiting, P. Diversity/Parallelism Trade-Off in Distributed Systems with Redundancy. *IEEE Trans. Inf. Theory* **2022**, *68*, 1279–1295. [[CrossRef](#)]
30. Bekkerman, R.; Bilenko, M.; Langford, J. *Scaling up Machine Learning: Parallel and Distributed Approaches*; Cambridge University Press: Cambridge, UK, 2011.
31. Ding, Z.; Xu, J.; Dobre, O.A.; Poor, H.V. Joint Power and Time Allocation for NOMA–MEC Offloading. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6207–6211. [[CrossRef](#)]
32. Kiani, A.; Ansari, N. Edge Computing Aware NOMA for 5G Networks. *IEEE Internet Things J.* **2018**, *5*, 1299–1306. [[CrossRef](#)]
33. Park, H.S.; Jun, C.H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341. [[CrossRef](#)]
34. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* **2020**, *9*, 1295. [[CrossRef](#)]
35. Sun, Y.; Babu, P.; Palomar, D.P. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. *IEEE Trans. Signal Process.* **2017**, *65*, 794–816. [[CrossRef](#)]
36. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
37. Goldsmith, A. *Wireless Communications*; Cambridge University Press: Cambridge, UK, 2005.
38. Li, Z.; Ye, C.; Cui, Y.; Yang, S.; Shamai, S. Rate Splitting for Multi-Antenna Downlink: Precoder Design and Practical Implementation. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1910–1924. [[CrossRef](#)]
39. Sadeghabadi, E.; Blostein, S. Low Complexity Rate Splitting Using Hierarchical User Grouping. In Proceedings of the IEEE ICC WKSHPs, Montreal, QC, Canada, 14–23 June 2021.
40. Behrouzi-Far, A.; Soljanin, E. Redundancy Scheduling in Systems with Bi-Modal Job Service Time Distributions. In Proceedings of the Allerton, Monticello, IL, USA, 24–27 September 2019.
41. Tandon, R.; Lei, Q.; Dimakis, A.G.; Karampatziakis, N. Gradient Coding: Avoiding Stragglers in Distributed Learning. In Proceedings of the ICML, Sydney, NSW, Australia, 6–11 August 2017.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.