Joint MDS Codes and Weighted Graph Based Coded Caching in Fog Radio Access Networks

Yanxiang Jiang, *Senior Member, IEEE*, Bao Wang, Fu-Chun Zheng, *Senior Member, IEEE*,
Mehdi Bennis, *Fellow, IEEE*, and Xiaohu You, *Fellow, IEEE*

***Abstract*—In this paper, we investigate maximum-distance separable (MDS) codes and weighted graph based coded caching in fog radio access networks (F-RANs). In the placement phase, the redundant MDS based coded placement scheme is used to provide redundant coded packets and homogeneous cached contents. The redundant coded packets can be used to construct multicast opportunities for similar requests. In the delivery phase, the weighted graph based coded delivery scheme is conducted based on homogeneous cached contents, which can induce considerable multicast opportunities. By integrating the above two schemes, a joint MDS codes and weighted graph based coded caching policy is proposed to minimize the fronthaul load. Finally, we theoretically analyze the performance of the proposed policy by deriving the lower and upper bounds of the fronthaul load. Simulation results show that our proposed policy can provide** 44% **savings in the fronthaul load compared to the MDS-based uncoded delivery policy.**

***Index Terms*—Coded caching, fog radio access networks, maximum-distance separable code, weighted graph.**

I. INTRODUCTION

As video consumption has become the mainstream of demands, heavy traffic can happen during the peak periods of wireless networks, which often leads to network congestion and poor quality of services (QoS). Although traditional cloud radio access networks (C-RANs) can provide reliable and stable services, it is a massive challenge to realize real-time requests processing and high quality services [1]. Most of all, the ever increasing mobile data traffic brings tremendous pressure on fronthaul links, which may cause network congestion, especially at peak periods. In order to alleviate this, fog radio access networks (F-RANs) have raised widespread attention [2]–[14]. Fog access points (F-APs) in F-RANs are close to users and able to improve network congestion as well as QoS by using edge caching resources [15]. Furthermore, coded caching was firstly proposed in [16] and [17] to reduce the network congestion by elaborately designing the cache content and content delivery.

Recently, large body of literature has focused on coded caching from different aspects. In [18], a decentralized asynchronous coded caching scheme was proposed, which established the relationship between the coded-multicast contents in asynchronous and synchronous coded caching. The authors in [19] derived the upper and lower bounds of the fronthaul load for the general case where the number of requests during each time slot is random. An index coding achievable scheme was proposed in [20] under the constraint that the content was placed uncoded within the caches. In [21], a new algorithm based on the perfect match of a bipartite graph was proposed, which offered full multiplexing as well as full coded-caching gains to both cache-aided and cache-less users. The use of coded caching in device-to-device (D2D) system was studied in [22], which jointly designed the uncoded cache placement and linear coded D2D delivery to minimize the D2D delivery load. In [23], the authors pointed out that the coded caching scheme in [16] may cause privacy issues regarding the cache contents stored at other users' caches. Then, a two phase private caching scheme was proposed to both minimize the load and preserve the information-theoretic privacy of each user's demand. In [24], the authors considered the general scenario with arbitrary file sizes and cache sizes, and proposed an optimization framework for decentralized coded caching to minimize the worst-case and average load. In [25], a joint storage and proactive caching scheme was proposed, which jointly exploited MDS-coded storage at the multi-servers, and uncoded prefetch and coded delivery at the users.

The use of MDS codes in heterogeneous wireless networks was studied in [26]–[34]. The authors in [26] formulated the optimal MDS encoding caching scheme as a convex optimization problem to minimize the fronthaul rate. A caching strategy combining file splitting with MDS encoding was proposed in [27], which can improve the overall energy efficiency. By designing the number of MDS coded contents cached at each small base station, the authors in [28] were able to minimize the power consumption. A deep reinforcement learning method was proposed in [29], where each F-AP in F-RANs obtained the number of MDS coded packets for caching by autonomous learning. An analytical expression for the delay in downloading contents from wireless networks was derived in [30], where the authors studied the distributed caching networks by using MDS codes to reduce the download delay of wireless content delivery. In [31], the authors utilized an Markov decision process to capture the popularity dynamics

Manuscript received April 16, 2021, revised August 30, 2021, December 20, 2021, and accepted January 14, 2022. This work was supported in part by the National Key Research and Development Program under Grant 2021YFB2900300, the National Natural Science Foundation of China under grant 61971129, and the Shenzhen Science and Technology Program under Grant KQTD20190929172545139.

Y. Jiang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, and the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: yxjiang@seu.edu.cn).

B. Wang and X. You, are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: 220180884@seu.edu.cn, xhyu@seu.edu.cn).

F. Zheng is with the School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China, and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: fzheng@ieee.org).

M. Bennis is with Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland (e-mail: mehdi.bennis@oulu.fi).

and maximize the long term expected cumulative traffic load, which were served by the small base stations with an MDS coding based coopreative caching scheme. A novel mobility-aware content storage and delivery scheme was proposed in [32], which jointly exploited the MDS coded packets stored at the small base stations and coded delivery to reduce the fronthaul rate. In [33], the authors utilized the MDS codes in the placement phase and utilized the real interference alignment in the delivery phase to minimize the normalized delivery time. In [34], a strategy of probabilistic content caching based on MDS codes was designed, which minimized the fronthaul rate in the mobile edge caching.

However, the papers mentioned above mainly focus on designing the cache placement phase by the MDS codes, but no multicast messages are constructed in the delivery phase. Hence, an MDS-based coded caching scheme which can construct multicast messages during the delivery phase is needed. An interesting idea was proposed in [35], which used the scheme in [16] to construct multicast messages and further encoded the multicast messages by MDS codes. In [36], the authors utilized fountain codes to satisfy the requests of multiple users for the same file, where fountain codes are sub-optimal MDS codes. An MDS codes-aided transmission scheme (MCAT) was proposed in [37], which constructed multicast opportunities for similar requests by using the redundancy of MDS codes. In the placement phase, MCAT lets each F-AP randomly and uniformly store the same fraction of the most popular N^t files. In the delivery phase, it optimizes the clustering of F-APs and then each cluster serves one multicast group which is composed of the users requesting the same files. Nevertheless, the benefits of MDS codes in fronthaul multicast and cooperative transmissions have not been well unlocked yet [37]. In [38], we proposed a joint redundant MDS codes[1] and cluster cooperation based coded caching policy, which utilized the MDS codes in the placement phase and constructed considerable multicast messages based on the MDS coded packets during the delivery phase.

Motivated by the aforementioned facts, we in this paper propose a new MDS based coded caching policy in F-RANs by taking both MDS codes and coded delivery into account. The proposed policy combines the construction of multicast messages, the full utilization of MDS codes properties, and the cooperative transmission between F-APs in a new way. In our proposed policy, multicast messages are constructed based on the redundancy of MDS codes and the homogeneous cache contents of F-APs.

Our main contributions are summarized below.

1) We propose a redundant MDS based coded placement scheme, which achieves a homogeneity of the cache contents among F-APs and stores independent l coded packets at the cloud server. We show that the homogeneity and the l coded packets offers the opportunities to construct multicast messages for both consistent requests and inconsistent requests[2] during the delivery phase.
2) We propose a weighted graph based coded delivery scheme, which can fully utilize the cache contents among F-APs. We formulate the coded delivery optimization problem to minimize the fronthaul load based on weighted graph, where for a given graph, the problem can be transformed into an integer programming problem.
3) We propose a joint MDS codes and weighted graph based coded caching policy (JMWG), which can construct considerable multicast messages during the delivery phase. The proposed policy can construct two kinds of multicast messages by using the l redundant coded packets and the cache contents between any two clusters.
4) We analyze the performance of our proposed policy by deriving the upper and lower bounds of the fronthaul load. Furthermore, we show through theoretical analysis that the performance gap between our proposed policy and its lower bound is extremely small.

The rest of this paper is organized as follows. The system model is briefly described in Section II. The proposed joint MDS codes and weighted graph based coded caching policy and performance analysis are presented in Section III. Numerical results are shown in Section IV. Final conclusions are drawn in Section V.

II. SYSTEM MODEL

We consider the F-RAN as illustrated in Fig. 1, where there are K F-APs and one wireless cloud server. The cloud server has access to a library of N files $W_1, W_2, \ldots, W_n, \ldots, W_N$. The size of each file is F bits. Let $\mathcal{N} = \{1, 2, \ldots, n, \ldots, N\}$ denote the index set of N files. Let $\mathcal{K} = \{1, 2, \ldots, k, \ldots, K\}$ denote the index set of the considered K F-APs. Each F-AP k has an isolated cache memory Z_k and the size of Z_k is $M \cdot F$ bits for $0 < M < N$. C clusters are formed based on the geographical locations of the K F-APs. Each cluster contains $S = \lfloor \frac{K}{C} \rfloor$ F-

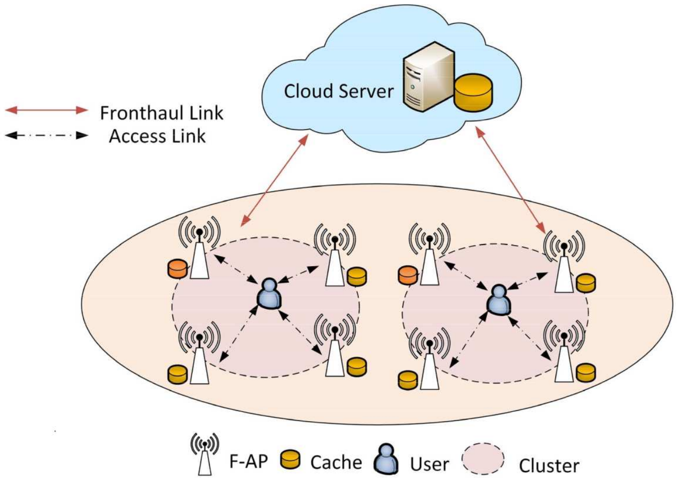

Fig. 1. Illustration of the F-RANs with $K = 8$, $U = C = 2$, $S = 4$.

---

[1] The “redundant MDS” here indicates leveraging the redundancy of MDS codes to create multicast opportunities.

[2] We refer to multiple users' requests for the same file and different files as *consistent requests* and *inconsistent requests*, respectively.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

APs and only serves a single user at a time [3][4]. It is easy to see that $K \geq C \cdot S$. Let $\mathcal{C} = \{1, 2, \ldots, c, \ldots, C\}$ denote the index set of C clusters and $\mathcal{C}_c$ the c-th cluster. We assume that the F-APs in cluster $\mathcal{C}_c$ are indexed by $\{S \cdot (c-1) + 1, \ldots, S \cdot c\} \subset \mathcal{K}$. U users are associated with the F-APs through access links, where $U \leq C$. Let $\mathcal{U} = \{1, 2, \ldots, u, \ldots, U\}$ denote the index set of U users. Without loss of generality, we assume that user u is served by cluster $\mathcal{C}_u$, where $0 < u \leq U \leq C$.

The popularities of N files follow a Zipf distribution, and the popularity of file W_n is defined as:

$$p_n = \frac{1/n^{\alpha}}{\sum_{\tilde{n}=1}^{N} 1/\tilde{n}^{\alpha}}, \tag{1}$$

where α is the parameter of the Zipf distribution. Let $\boldsymbol{p} = [p_1, p_2, \ldots, p_n, \ldots, p_N]^T$ denote the popularity vector of the N files. Without loss of generality, we assume that the elements in $\boldsymbol{p}$ are in a descending order.

Firstly, we use MDS codes to encode the caching files. Specifically, file W_n is split into L fragments and each fragment has F/L bits. Then, we use $(K+l, L)$ MDS codes to encode the L fragments into $K+l$ coded packets $W_{n,1}, W_{n,2}, ..., W_{n,K+l}$, where $K + l \geq L$. The l redundant coded packets are used to construct multicast messages for similar requests in the delivery phase. The size of each coded packet is the same as the file fragment. The cloud server has all copies of these coded packets.

In the placement phase, due to the limitation of each F-AP's cache capacity, we take a truncated caching strategy in this paper. That is, each F-AP selects the coded packets of the N^{t} most popular files for caching, where $1 \leq N^{\mathrm{t}} \leq N$. The cache contents of K F-APs are different from each other. Note that the cache size of each F-AP is $M \cdot F$ bits, and $N^{\mathrm{t}} \leq M \cdot L$ is needed to satisfy the cache constraint.

In the delivery phase, U users initiate requests for files in the library. Let $\boldsymbol{d} = [d_1, d_2, \ldots, d_u, \ldots, d_U]^T$ denote the request vector of U users, where $d_u \in \mathcal{N}$ is the index of file requested by user u. Based on $\boldsymbol{d}$, U users can be classified into $\mathbb{U}_1, \mathbb{U}_2, \ldots, \mathbb{U}_n, \ldots, \mathbb{U}_N$, where $\mathbb{U}_n = \{u \mid d_u = n, \forall u \in \mathcal{U}\}$ denotes the index set of users who request file W_n. Furthermore, $\mathcal{U}^{\mathrm{c}} = \{\mathbb{U}_n \mid |\mathbb{U}_n| \geq 2, \forall n \in \mathcal{N}\}$ and $\mathcal{U}^{\mathrm{i}} = \{\mathbb{U}_n \mid |\mathbb{U}_n| = 1, \forall n \in \mathcal{N}\}$ are used to represent the index set of users for consistent requests and inconsistent requests, respectively. Similarly, $\mathcal{C}$ can be classified into $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_n, \ldots, \mathbb{C}_N$, where $\mathbb{C}_n = \{\mathcal{C}_u \mid \forall u \in \mathbb{U}_n\}$ denotes the set of served clusters whose users request file W_n. Furthermore, $\mathcal{C}^{\mathrm{c}} = \{\mathbb{C}_n \mid |\mathbb{C}_n| \geq 2, \forall n \in \mathcal{N}\}$ and $\mathcal{C}^{\mathrm{i}} = \{\mathbb{C}_n \mid |\mathbb{C}_n| = 1, \forall n \in \mathcal{N}\}$ are used to represent the set of served clusters for consistent requests and inconsistent requests, respectively.

If $d_u \leq N^{\mathrm{t}}$, user u can receive S distinct coded packets of file W_{d_u} from the local cluster $\mathcal{C}_u$. According to the property of the MDS codes, at least L distinct coded packets are needed to recover the desirable file. If $S \geq L$, user u is able to recover W_{d_u} without using the fronthaul link. However, if $S < L$, user u would need another $L-S$ distinct coded packets to recover the desirable file. In this case, the cloud server randomly selects an F-AP in cluster $\mathcal{C}_u$ as a relay to transmit the remaining coded packets to user u. For ease of description, we use the expression that the cloud server transmits the coded packets to cluster $\mathcal{C}_u$ instead of the cloud server transmits the coded packets to the corresponding F-AP in cluster $\mathcal{C}_u$. Let R denote the overall fronthaul load, which is defined as the size of messages that the cloud server transmits to all clusters for satisfying all requests. For conventional uncoded delivery schemes based on the MDS codes, $R = U \cdot \max(L-S, 0) \cdot \frac{F}{L}$.

The objective of this paper is to find a feasible coded caching policy to minimize the fronthaul load by utilizing the optimality of MDS codes in terms of redundancy-reliability tradeoff and the cache contents between any two clusters. For convenience, a summary of major notations is presented in Table I.

III. JOINT MDS CODES AND WEIGHTED GRAPH BASED CODED CACHING POLICY

In this section, we first propose the redundant MDS based coded placement scheme and show the construction of multicast messages based on the l redundant coded packets. Then, we propose the weighted graph based coded delivery scheme and show the construction of multicast messages based on the cache contents between any two clusters. Furthermore, the joint MDS codes and weighted graph based coded caching policy is proposed to reduce the fronthaul load. Finally, the performance and complexity of the proposed policy are analyzed.

A. Redundant MDS Based Coded Placement Scheme

In order to construct multicast opportunities, motivated by [16], the cache content of each F-AP should be dedicated designed. The redundant MDS based coded placement scheme is presented in Algorithm 1: each caching file is encoded into $K + l$ coded packets, the K coded packets are placed in all K F-APs, and the l redundant coded packets are stored in the cloud server without being cached in any F-AP. It is easy to see that the k-th coded packet of file W_n is placed in F-AP k and we refer to this correspondence as the *homogeneity of cache contents* in F-APs. The construction of multicast messages based on the l redundant coded packets is presented as follows.

Consider that $\mathbb{U}_n \in \mathcal{U}^{\mathrm{c}}$, $\mathbb{C}_n \in \mathcal{C}^{\mathrm{c}}$, users in $\mathbb{U}_n$ request the same file W_n. If file W_n is cached in F-APs, i.e., $n \leq N^{\mathrm{t}}$, the number of distinct coded packets of file W_n cached in the clusters in $\mathbb{C}_n$ is $|\mathbb{C}_n| \cdot S$. Note that the cloud server has all $K + l$ distinct coded packets of file W_n, so there are other $K + l - |\mathbb{C}_n| \cdot S$ distinct coded packets of file W_n can be used to offer multicast opportunities for users in $\mathbb{U}_n$. It is easy to see that at most $K + l - |\mathbb{C}_n| \cdot S$ multicast messages useful to $|\mathbb{C}_n|$ clusters can be constructed.

---

[3] To facilitate analysis, we assume that the K F-APs form C non-overlapping clusters and the cluster size are all the same. In future work, the discussion of system models with different cluster sizes will be expanded.

[4] The single-user assumption is only for description convenience. The multi-user case can also be handled by our proposed scheme, but some simple modifications should be conducted. Multiple users will result in unbalanced requests between different clusters. Thus, clusters that previously request files may not request them again and the number of requests in each cluster may be distinct, which influences coded multicasting. We think the generalization from single-user to the multi-user case is not difficult, so we only focus on the single-user case in our work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

TABLE I. Summary of Major Notations

Notation	Defination
N	Number of files
K	Total Number of F-APs
C	Number of clusters
S	Number of F-APs in each cluster
W_n	File with index n
Z_k	Cache memory of F-AP k
p_n	Popularity of file W_n
L	Number of fragments each file split into
l	Number of redundant coded packets
N^t	Number of files to encode and cache
d_u	Index of file requested by user u
$\mathbb{U}_n$	Set of users who request file W_n
$\mathcal{U}^c, \mathcal{U}^i$	Set of users for consistent requests and inconsistent requests
$\mathbb{C}_n$	Set of clusters whose users request file W_n
C^c, C^i	Set of clusters for consistent requests and inconsistent requests
$\mathcal{G}(\mathcal{V}, \mathcal{E})$	Undirected graph with vertices set $\mathcal{V}$ and edge set $\mathcal{E}$
$D(v)$	Number of edges incident to vertex v
$w_{i,j}$	Weight of the edge between vertice i and vertice j
$\mathcal{W}(\mathcal{G})$	Sum of all weights of graph $\mathcal{G}$
$E(\mathcal{G})$	Number of edges of graph $\mathcal{G}$

By storing the l redundant coded packets at the cloud server instead of caching at any F-AP, our proposed redundant MDS based coded placement scheme can provide multicast opportunities for consistent requests, especially when the consistency of requests is relatively high. Furthermore, it also provides homogeneity of cache contents in F-APs, which can be used to construct multicast messages for both consistent and inconsistent requests in the delivery phase.

B. Weighted Graph Based Coded Delivery Scheme

In this subsection, we first propose the cluster cooperation based coded delivery scheme and show the construction of multicast messages based on the cache contents between any two clusters. Then, we extend the cluster cooperation based coded delivery scheme to the general situation and propose the weighted graph based coded delivery scheme.

1) Cluster Cooperation Based Coded Delivery Scheme: In Algorithm 1, each F-AP has coded packets of the N^{t} most popular files, and the cache contents of the K F-APs are different from each other. It is easy to see that if the requested files are cached in F-APs, the cache contents of other clusters are useful to the user served by the local cluster. Similarly, the cache contents of the local cluster are also useful to the users served by other clusters. The construction of multicast messages based on the the homogeneity of cache contents between two clusters is presented as follows.

Algorithm 1 Redundant MDS Based Coded Placement Scheme

```
Initialize $K$, $l$, $L$, $N^{\mathrm{t}}$, $(W_1, W_2, \ldots, W_n, \ldots, W_{N^{\mathrm{t}}})$.
for $n \in \{1, 2, \ldots, N^{\mathrm{t}}\}$ do
    Split $W_n$ into $L$ fragments of equal size,
    Use $(K + l, L)$ MDS codes to encode the $L$ fragments into $K + l$ coded packets
        $W_{n,1}, W_{n,2}, \ldots, W_{n,K+l}$.
end for
for $k \in \{1, 2, \ldots, K\}$ do
    $Z_k = \{W_{1,k}, W_{2,k}, \ldots, W_{N^{\mathrm{t}},k}\}$
end for
```

Consider user i and user j, where $0 < i \neq j \leq U$. The served clusters of user i and user j are C_i and C_j, respectively. The desirable files are file W_{d_i} and file W_{d_j}, respectively. If $d_i, d_j \leq N^t$, according to Algorithm 1, the cache contents of file W_{d_j} in cluster C_i are $\{W_{d_j,S\cdot(i-1)+1}, W_{d_j,S\cdot(i-1)+2}, \ldots, W_{d_j,S\cdot i}\}$. Similarly, the cache contents of file W_{d_i} in cluster C_j are $\{W_{d_i,S\cdot(j-1)+1}, W_{d_i,S\cdot(j-1)+2}, \ldots, W_{d_i,S\cdot j}\}$. For $s = 1, 2, \ldots, S$, multicast message $W_{d_i,S\cdot(j-1)+s} \oplus W_{d_j,S\cdot(i-1)+s}$ is useful to both C_i and C_j. Specifically, after receiving message $W_{d_i,S\cdot(j-1)+s} \oplus W_{d_j,S\cdot(i-1)+s}$, F-AP $S \cdot (i-1) + s$ in C_i is able to decode $W_{d_i,S\cdot(j-1)+s}$ by using $W_{d_j,S\cdot(i-1)+s}$ stored in its own cache. Then, user i can receive $W_{d_i,S\cdot(j-1)+s}$ through access link. Similarly, user j can receive $W_{d_j,S\cdot(i-1)+s}$. It is easy to see that at most S multicast messages can be constructed.

The cluster cooperation based coded delivery scheme [5] is presented in Algorithm 2. Assume that $C^{\mathrm{a}} \subset C$ is any subset of C and n_{a} is the number of coded packets of the requested file the user already has. When the S multicast messages are enough to satisfy the requests, i.e. $L - n_{\mathrm{a}} \leq S$, the cloud server transmits $L - n_{\mathrm{a}}$ of the S coded packets to satisfy the requests of users served by the clusters in C^{a}. When the S multicast messages are not enough to satisfy the requests, i.e., $L - n_{\mathrm{a}} > S$,

---

[5] If there exists a cluster that cannot be paired in C^{a}, the required coded packets of the cluster are directly transmitted by the cloud server.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888

Algorithm 2 Cluster Cooperation Based Coded Delivery Scheme

```
Initialize $C^{\mathrm{a}}$, $\boldsymbol{d}$, $S$, $L$, $n_{\mathrm{a}}$.
Randomly pair the clusters in $C^{\mathrm{a}}$.
if $L - n_{\mathrm{a}} \leq S$ then
    $n_{\mathrm{r}} = L - n_{\mathrm{a}}$.
else
    $n_{\mathrm{r}} = S$.
end if
for $\{C_i, C_j\} \subset C^{\mathrm{a}}$ do
    Generate the index sets of F-APs in $C_i$ and $C_j$,
    $C_i : \{S \cdot (i-1)+1, S \cdot (i-1)+2, \ldots, S \cdot i\}$
    $C_j : \{S \cdot (j-1)+1, S \cdot (j-1)+2, \ldots, S \cdot j\}$.
    for $s = 1, 2, \ldots, n_{\mathrm{r}}$ do
        Cloud server transmits $W_{d_j, S\cdot(i-1)+s} \oplus W_{d_i, S\cdot(j-1)+s}$ to $C_i$ and $C_j$.
    end for
    while $n_{\mathrm{a}} + n_{\mathrm{r}} < L$ do
        Cloud server separately transmits $L - n_{\mathrm{a}} - n_{\mathrm{r}}$ of the $K + l - n_{\mathrm{a}} - n_{\mathrm{r}}$ coded packets of $W_n$ to $C_i$ and $C_j$.
    end while
end for
```

the cloud server transmits the S multicast messages first, and each user served by the clusters in C^{a} needs another $L - n_{\mathrm{a}} - S$ coded packets to recover the desirable file. Since each caching file is encoded into $K + l$ coded packets with $K + l \geq L$, there are $K + l - n_{\mathrm{a}} - S$ distinct coded packets stored in the cloud server. Then, the cloud server separately transmits $L - n_{\mathrm{a}} - S$ of the remaining $K + l - n_{\mathrm{a}} - S$ coded packets to C_i and C_j to satisfy the requests of user i and user j, respectively.

By using the homogeneity of cache contents between two clusters, our proposed cluster cooperation based coded delivery scheme can construct multicast messages for both consistent ($d_i = d_j$) and inconsistent requests ($d_i \neq d_j$), which is crucial for reducing the fronthaul load.

2) Weighted Graph Based Coded Delivery Scheme: In the proposed cluster cooperation based coded delivery scheme, it is easy to see that for a particular cluster, it only uses the cache content of another cluster. In order to fully utilize the cache contents of all F-APs, we take a graph theoretic perspective to generalize the coded delivery scheme. To make the connection, some graph theoretic notions are given here first. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an undirected graph on the set of vertices $\mathcal{V}$ and edge set $\mathcal{E}$. The degree of a vertex $v \in \mathcal{V}$ is the number of edges incident to vertex v, which is denoted by $D(v)$. $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is said to be a λ-regular graph if $D(v) = \lambda$ for each $v \in \mathcal{V}$, where $1 \leq \lambda \leq |\mathcal{V}| - 1$. Particularly, $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a complete graph when $D(v) = |\mathcal{V}| - 1$, for each $v \in \mathcal{V}$. For each edge $e \in \mathcal{E}$, there is a real number $w(e)$ associated to the edge e, then the real number $w(e)$ is called the weight of edge e. We use $\mathcal{W}(\mathcal{G})$ to represent the sum of all weights of graph $\mathcal{G}$.

We now describe the connections between our system model and graph theory. Let the clusters represent the vertices, and the multicast opportunities between two clusters represent the edges. According to the cluster cooperation based coded delivery scheme, it is easy to see that if there exits an edge between cluster C_i and cluster C_j, at most S multicast messages can be constructed. Let $w_{i,j}$ denote the weight of the edge between cluster C_i and cluster C_j. We define $w_{i,j}$ as the number of multicast messages constructed for cluster C_i and cluster C_j, where $w_{i,j}$ is an integer and $0 \leq w_{i,j} = w_{j,i} \leq S$. For given $C^{\mathrm{a}} \subset C$, there exits a set of vertices $\mathcal{V} = \{C_i \mid \forall i \in C^{\mathrm{a}}\}$ constructed by the clusters in C^{a}, where $|\mathcal{V}| = |C^{\mathrm{a}}| = V$.

In order to reduce the energy consumption of the cloud server, the graph constructed by the clusters in C^{a} should have the least number of edges. Besides, the sum of all weights of edges of the graph should be minimized to reduce the fronthaul load.

The objective of the proposed delivery scheme is to generate a graph $\mathcal{G}^*$ with the least number of edges and the smallest $\mathcal{W}(\mathcal{G}^*)$. The cloud sever transmits multicast messages according to the edges of graph $\mathcal{G}^*$, then all users served by clusters in C^{a} can recover their desirable files.

Assume that V users already have n_{a} distinct coded packets. For the case of $n_{\mathrm{a}} = L$, $D(v) = 0$ for each $v \in \mathcal{V}$ is optimal. Because n_{a} distinct coded packets are already enough to recover the requested file, there is no need to construct any more multicast messages. For the case of $L - n_{\mathrm{a}} \geq (V-1) \cdot S$, $D(v) = V - 1$ for each $v \in \mathcal{V}$ is the optimal. Under this condition, $\mathcal{G}^*$ is a complete graph with $\frac{V \cdot (V-1)}{2}$ edges and the weight of each edge is S. Then all the contents cached at $V \cdot S$ F-APs are fully utilized. The remaining $L - n_{\mathrm{a}} - (V-1) \cdot S$ distinct coded packets for each cluster in C^{a} will be transmitted by the cloud server separately.

Focusing on the general case when $0 < L - n_{\mathrm{a}} < (V-1) \cdot S$, the coded delivery problem is equivalent to finding a graph $\mathcal{G}^*$, which has the least number of edges and satisfies $D(v) \geq \gamma$ for each $v \in \mathcal{V}$, where $\gamma = \lceil \frac{L - n_{\mathrm{a}}}{S} \rceil$. Meanwhile, in order to minimize the fronthaul load, we also need to minimize the sum of weights of graph $\mathcal{G}^*$, i.e., $\mathcal{W}(\mathcal{G}^*)$ should be minimized.

Theorem 1: The number of edges of graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is $E(\mathcal{G}) = \sum_{v=1}^{|\mathcal{V}|} D(v)/2$.

Proof: The degree of a vertex is the number of edges incident to the vertex. According to the definition of graph, an edge connects two vertices. So the sum of all vertices' degrees is two times of the number of edges, i.e., the number of edges of graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is $\sum_{v=1}^{|\mathcal{V}|} D(v)/2$. This completes the proof. ■

Theorem 2: For a vertex set $\mathcal{V}$ with V vertices, if $\mu \cdot V$ is even, where μ is an integer and $1 \leq \mu \leq V - 1$, there exists a μ-regular graph $\mathcal{G}^{\mathrm{r}}$ with V vertices.

Proof: Please see Appendix A. ■

Corollary 1: For a vertex set $\mathcal{V}$ with V vertices, if $\mu \cdot V$ is even, where μ is an integer and $1 \leq \mu \leq V - 1$, then a μ-regular graph $\mathcal{G}^{\mathrm{r}}$ with V vertices is the optimal graph, which has the least number of edges and satisfies $D(v) \geq \mu$ for any $v \in \mathcal{V}$, i.e., $\mathcal{G}^* = \mathcal{G}^{\mathrm{r}}$.

Proof: From the assumption above, the lower bound of $D(v)$ is μ. In order to generate a graph $\mathcal{G}^*$ with the least number of edges, we let $D(v) = \mu$ for any $v \in \mathcal{V}$, i.e., $\mathcal{G}^*$ is a $\mu-$regular graph. From Theorem 2, there exits a μ-regular graph under such condition. This completes the proof. ■

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

***Corollary** 2:* For a vertex set $\mathcal{V}$ with V vertices, if $\mu \cdot V$ is odd, where μ is an integer and $1 \le \mu \le V-1$, there exists a sequence $\boldsymbol{\theta} = (\mu+1, \mu, \ldots, \mu)$ which can be graphic. The graph generated by $\boldsymbol{\theta}$ is the optimal graph, which has the least number of edges and satisfies $D(v) \ge \mu$ for any $v \in \mathcal{V}$.

Proof: Please see Appendix B. ■

Now we describe the proposed weighted graph based coded delivery scheme. Consider the case of $0 < L - n_{\mathrm{a}} < (U-1) \cdot S$. The user needs another $L - n_{\mathrm{a}}$ distinct coded packets to reconstruct their desirable files. Consider cluster C_i in $\mathcal{C}^{\mathrm{a}}$, at least another γ clusters' cache contents are needed to reconstruct the desirable file. From the perspective of graph theory, the lower bound of the degree of vertex C_i would be γ. In order to guarantee that all users served by the clusters in $\mathcal{C}^{\mathrm{a}}$ can recover their desirable files, $D(C_i) \ge \gamma$ needs to be satisfied for any $i \in \mathcal{C}^{\mathrm{a}}$. On one hand, when $V \cdot \gamma$ is even, the graph generated by the sequence $(\gamma, \ldots, \gamma)$ is the optimal graph according to Corollary 1. In this case, the optimal graph $\mathcal{G}^*$ is a $\gamma-$regular graph, and the upper bound of the fronthual load consumed by the multicast messages is $S \cdot E(\mathcal{G}^*) \cdot \frac{F}{L} = S \cdot \frac{V \cdot \gamma}{2} \cdot \frac{F}{L}$. On the other hand, when $V \cdot \gamma$ is odd, the optimal graph $\mathcal{G}^*$ can be generated from the sequence $(\gamma+1, \gamma, \ldots, \gamma)$ according to Corollary 2, and the upper bound of the fronthual load consumed by the multicast messages is $S \cdot E(\mathcal{G}^*) \cdot \frac{F}{L} = S \cdot \frac{V \cdot \gamma + 1}{2} \cdot \frac{F}{L}$.

Furthermore, in order to minimize the fronthaul load, we should elaborately design the number of multicast messages for each cluster pair, i.e., the weights of graph $\mathcal{G}^*$ should be elaborately designed. Then, the optimization problem can be formulated as follows:

$$\min_{w_{i,j}} \quad \frac{1}{2} \cdot \sum_{i=1}^{V} \sum_{j=1, j \ne i}^{V} w_{i,j} \tag{2}$$

$$\text{s.t.} \quad \sum_{j=1, j \ne i}^{V} w_{i,j} \ge L - n_{\mathrm{a}}, \quad \forall i \in [1:V] \tag{2a}$$

$$\sum_{j=2}^{V} \mathbb{I}_{\{w_{1,j} > 0\}} = \zeta, \tag{2b}$$

$$\sum_{j=1, j \ne i}^{V} \mathbb{I}_{\{w_{i,j} > 0\}} = \gamma, \quad \forall i \in [2:V] \tag{2c}$$

$$w_{i,j} = w_{j,i}, \quad \forall i \in [1:V], j \in [1:V], j \ne i \tag{2d}$$

$$w_{i,j} \in [0:S], \quad \forall i \in [1:V], j \in [1:V], j \ne i, \tag{2e}$$

where $[\mathrm{a}:\mathrm{b}]$ denotes the set of integers $\{\mathrm{a}, \mathrm{a}+1, \ldots, \mathrm{b}\}$ for $\mathrm{a} \le \mathrm{b}$. In constraint (2b), $\zeta = \gamma + 1$ when $V \cdot \gamma$ is odd and $\zeta = \gamma$ when $V \cdot \gamma$ is even. Constraint (2a) guarantees that each user can receive at least $L - n_{\mathrm{a}}$ coded packets to successfully reconstruct their desirable files. Without loss of generality, we assume that the degree of cluster C_1 is $\gamma+1$ when $V \cdot \gamma$ is odd. Constraints (2a) and (2b) ensure that enough cluster pairs are generated to provide sufficient multicast messages.

Algorithm 3 Weighted Graph Based Coded Delivery Scheme

```
Initialize $\mathcal{C}^{\mathrm{a}}$, $\boldsymbol{d}$, $S$, $L$, $n_{\mathrm{a}}$.
if $(|\mathcal{C}^{\mathrm{a}}| - 1) \cdot S < (L - n_{\mathrm{a}})$ then
    Generate a complete graph $\mathcal{G}^{\mathrm{c}}$ from $|\mathcal{C}^{\mathrm{a}}|$ vertices.
    Let the weights of all edges of graph $\mathcal{G}^{\mathrm{c}}$ be $S$.
    Cloud sever sends multicast messages to the corresponding cluster pairs according to edges of graph $\mathcal{G}^{\mathrm{c}}$.
    Cloud server separately sends the remaining $L - n_{\mathrm{a}} - (|\mathcal{C}^{\mathrm{a}}| - 1) \cdot S$ coded packets to each cluster in $\mathcal{C}^{\mathrm{a}}$.
else
    Let $\gamma = \lceil \frac{L - n_{\mathrm{a}}}{S} \rceil$.
    if $|\mathcal{C}^{\mathrm{a}}| \cdot \gamma \mod 2 = 0$ then
        Generate a $\gamma$-regular graph $\mathcal{G}^*$ from $|\mathcal{C}^{\mathrm{a}}|$ vertices from the degree sequence $(\gamma, \ldots, \gamma)$.
    else
        Generate a graph $\mathcal{G}^*$ from $|\mathcal{C}^{\mathrm{a}}|$ vertices from the degree sequence $(\gamma+1, \gamma, \ldots, \gamma)$.
    end if
    Solve Problem (3) with the given $\mathcal{G}^*$.
    Cloud sever sends multicast messages to the corresponding cluster pairs according to the edges of graph $\mathcal{G}^*$.
end if
```

For given V and γ, the optimal graph $\mathcal{G}^*$ can be generated according to Corollary 1 and Corollary 2. For a given graph $\mathcal{G}^*$, constraints (5c) and (5d) are guaranteed. Then, Problem (2) can be transformed to the following equivalent form:

$$\min_{w_{i,j}} \quad \frac{1}{2} \cdot \sum_{i=1}^{V} \sum_{j=1, j \ne i}^{V} w_{i,j} \tag{3}$$

$$\text{s.t.} \quad \sum_{j=1, j \ne i}^{V} w_{i,j} \ge L - n_{\mathrm{a}}, \quad \forall i \in [1:V] \tag{3a}$$

$$w_{i,j} = w_{j,i}, \quad \forall i \in [1:V], j \in [1:V], j \ne i \tag{3b}$$

$$w_{i,j} \in [0:S], \quad \forall i \in [1:V], j \in [1:V], j \ne i. \tag{3c}$$

Problem (3) is an integer programming problem and can be solved by the mixed-integer linear programming package in python or the existing optimization solvers such as *intlinprog* in matlab.

Based on the cluster cooperation based coded delivery scheme and the above analysis, we now propose the weighted graph based coded delivery scheme, which futher explores the opportunities to create multicast messages between clusters. The detailed description of the proposed scheme is presented in Algorithm 3.

In Algorithm 3, each local cluster (or user) in $\mathcal{C}^{\mathrm{a}}$ needs another $L - n_{\mathrm{a}}$ distinct coded packets to recover their desirable files. For any cluster C_i in $\mathcal{C}^{\mathrm{a}}$, there are $(|\mathcal{C}^{\mathrm{a}}| - 1) \cdot S$ distinct coded packets cached in the other clusters in $\mathcal{C}^{\mathrm{a}}$. If $\mathcal{C}^{\mathrm{a}}$ has enough coded packets of users' requested files, i.e., $(|\mathcal{C}^{\mathrm{a}}| - 1) \cdot S \ge L - n_{\mathrm{a}}$, an optimal weighted graph can be generated based on all the clusters in $\mathcal{C}^{\mathrm{a}}$. The cloud server transmits the corresponding multicast messages according to the edges of the optimal weighted graph, then the requests served by the clusters in $\mathcal{C}^{\mathrm{a}}$ can be satisfied. The fronthaul load under such an occasion is $\mathcal{W}(\mathcal{G}^*) \cdot \frac{F}{L}$, where $\mathcal{G}^*$ is the optimal weighted

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

graph. However, if $(|C^{\mathrm{a}}|-1)\cdot S<(L-n_{\mathrm{a}})$, the cache contents of C^{a} are not enough to satisfy the requests. In order to fully utilize all the cache contents of C^{a}, a complete graph will be generated based on all the clusters in C^{a}, and the weight of each edge in the complete graph is S. The cloud server will then transmit the remaining $L-n_{\mathrm{a}}-(|C^{\mathrm{a}}|-1)\cdot S$ distinct coded packets to each cluster in C^{a} separately. In this case, the fronthaul load is $\frac{|C^{\mathrm{a}}|\cdot(|C^{\mathrm{a}}|-1)}{2}\cdot\frac{F\cdot S}{L}+|C^{\mathrm{a}}|\cdot[L-n_{\mathrm{a}}-(|C^{\mathrm{a}}|-1)\cdot S]\cdot\frac{F}{L}$. Let $\mathcal{R}(C^{\mathrm{a}},n_{\mathrm{a}})$ denote the fronthaul load consumed by the weighted graph based coded delivery scheme, which can be formulated as:

$$\mathcal{R}(C^{\mathrm{a}},n_{\mathrm{a}})=\begin{cases}\left[\frac{(|C^{\mathrm{a}}|-1)\cdot S}{2}+(L-n_{\mathrm{a}}-\delta)\right]\cdot|C^{\mathrm{a}}|\cdot\frac{F}{L} & L-n_{\mathrm{a}}>\delta,\\ \mathcal{W}(\mathcal{G}^{*})\cdot\frac{F}{L} & L-n_{\mathrm{a}}\leq\delta,\end{cases} \tag{4}$$

where $\delta=(|C^{\mathrm{a}}|-1)\cdot S$.

We will now explain the proposed weighted graph delivery scheme with two examples, which illustrate the cases where $V\cdot\gamma$ is odd and $V\cdot\gamma$ is even, respectively.

Example 1: Consider the scenario in which five clusters are formed by twenty F-APs as shown in Fig. 2. Each cluster has 4 F-APs. The F-APs in C_i are indexed by $\{4\cdot(i-1)+1,\ldots,4\cdot i\}$, where $1\leq i\leq 5$. Assume that each file is split into $L=14$ fragments, which are encoded into 21 coded packets. From Algorithm 1, the cache contents of C_i are $\{W_{n,4\cdot(i-1)+1},W_{n,4\cdot(i-1)+2},\ldots,W_{n,4\cdot i}\}$, where $1\leq n\leq N^{\mathrm{t}}$. Assume that five users' desirable files are all cached in the F-APs, i.e., $1\leq d_i\leq N^{\mathrm{t}}$. During the delivery phase, the local cluster can provide the corresponding user with 4 coded packets of W_{d_i}. Then each user needs another 10 distinct coded packets to reconstruct the desirable file. According to Corollary 2, a graph $\mathcal{G}^{*}$ can be constructed as shown in Fig. 2 by viewing the clusters as vertices and the multicast opportunities between two clusters as edges. It is easy to see that the degree sequence of $\mathcal{G}^{*}$ is $(4,3,3,3,3)$. Given graph $\mathcal{G}^{*}$, we can obtain the optimal weight of each edge of graph $\mathcal{G}^{*}$ by solving Problem (3). The weight of each edge is marked in Fig. 2. The coded-multicast contents represented by the edges of $\mathcal{G}^{*}$ are presented in Table II. It is easy to see that the fronthual load of Fig. 2 is $25\cdot\frac{F}{L}$. If traditional unicast scheme is utilized to satisfy all the requests, a total of 50 unicast messages are required, and the fronthaul load consumed by these messages is $50\cdot\frac{F}{L}$.

Example 2: Consider the scenario in which four clusters are formed by sixteen F-APs as shown in Fig. 3. Each cluster has 4 F-APs and the F-APs in C_i are indexed by $\{4\cdot(i-1)+1,\ldots,4\cdot i\}$, where $1\leq i\leq 4$. Assume that each file is split into 16 fragments, which are encoded into 17 coded packets. From Algorithm 1, the cache contents of C_i are $\{W_{n,4\cdot(i-1)+1},W_{n,4\cdot(i-1)+2},\ldots,W_{n,4\cdot i}\}$, where $1\leq n\leq N^{\mathrm{t}}$. Assume that the requested files of the four users are all cached in the F-APs, i.e., $1\leq d_i\leq N^{\mathrm{t}}$. During the delivery phase, the local cluster can provide the corresponding user with 4 coded packets of W_{d_i}. Each user needs another 12 distinct coded packets to reconstruct the desirable file. According to Corollary 1, a 3-regular graph $\mathcal{G}^{*}$ can be constructed as shown in Fig. 3, and the degree sequence of $\mathcal{G}^{*}$ is $(3,3,3,3)$. Given graph $\mathcal{G}^{*}$, the optimal weight of each edge of graph $\mathcal{G}^{*}$ can be obtained by solving Problem (3). The weight of each edge is marked in Fig. 3. The coded-multicast contents represented by the edges of $\mathcal{G}^{*}$ are presented in Table. III. It is easy to see that the fronthual load of Fig. 3 is $24\cdot\frac{F}{L}$, whereas the traditional unicast scheme is $48\cdot\frac{F}{L}$.

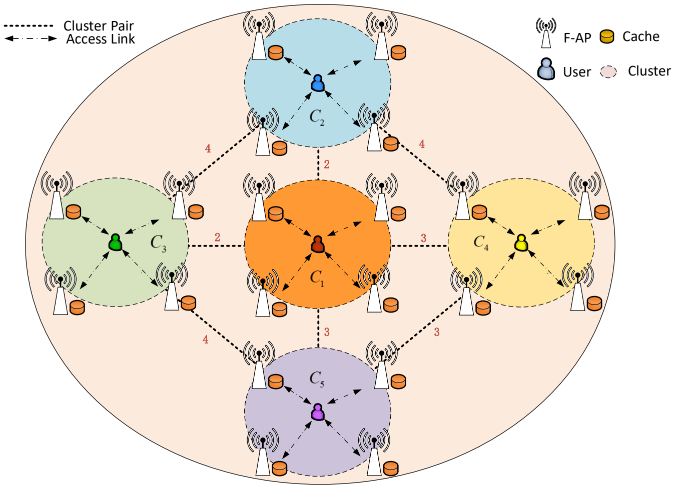

Fig. 2. Illustration of graph $\mathcal{G}^{*}$ constructed by $V=5$, $\gamma=3$.

TABLE II. The coded-multicast contents represented by the edges of $\mathcal{G}^{*}$ in Fig. 2

Edges	Coded-multicast contents	
(C_1,C_2)	$W_{d_2,1}\oplus W_{d_1,5}$	$W_{d_2,2}\oplus W_{d_1,6}$
(C_1,C_3)	$W_{d_3,1}\oplus W_{d_1,9}$	$W_{d_3,2}\oplus W_{d_1,10}$
(C_1,C_4)	$W_{d_4,1}\oplus W_{d_1,13}$	$W_{d_4,2}\oplus W_{d_1,14}$
	$W_{d_4,3}\oplus W_{d_1,15}$	
(C_1,C_5)	$W_{d_5,1}\oplus W_{d_1,17}$	$W_{d_5,2}\oplus W_{d_1,18}$
	$W_{d_5,3}\oplus W_{d_1,19}$	
(C_2,C_3)	$W_{d_3,5}\oplus W_{d_2,9}$	$W_{d_3,6}\oplus W_{d_2,10}$
	$W_{d_3,7}\oplus W_{d_2,11}$	$W_{d_3,8}\oplus W_{d_2,12}$
(C_2,C_4)	$W_{d_5,5}\oplus W_{d_2,13}$	$W_{d_5,6}\oplus W_{d_2,14}$
	$W_{d_5,7}\oplus W_{d_2,15}$	$W_{d_5,8}\oplus W_{d_2,16}$
(C_3,C_5)	$W_{d_4,9}\oplus W_{d_3,17}$	$W_{d_4,10}\oplus W_{d_3,18}$
	$W_{d_4,11}\oplus W_{d_3,19}$	$W_{d_4,12}\oplus W_{d_3,20}$
(C_4,C_5)	$W_{d_5,13}\oplus W_{d_4,17}$	$W_{d_5,14}\oplus W_{d_4,18}$
	$W_{d_5,15}\oplus W_{d_4,19}$	

C. Joint MDS Codes and Weighted Graph Based Coded Caching Policy

In this subsection, we focus on the case of $S<L$, where a single cluster is not able to satisfy the request of its served user. In order to fully utilize the multicast opportunities offered by the l redundant coded packets and the cache contents between any two clusters, the proposed policy is performed on consistent requests and inconsistent requests independently.

1) Consistent Requests: Consider that $\mathbb{U}_n\in\mathcal{U}^{\mathrm{c}}$, $\mathbb{C}_n\in C^{\mathrm{c}}$. Users in $\mathbb{U}_n$ request the same file W_n. If W_n is not cached in any F-AP, i.e., $n>N^{\mathrm{t}}$, the cloud server transmits a multicast message W_n with F bits to the clusters in $\mathbb{C}_n$. However, $n\leq N^{\mathrm{t}}$ is the typical case, where the requested file is cached in all F-APs. In this case, each user in $\mathbb{U}_n$ can receive S coded packets of file W_n from the local cluster. According to the property of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

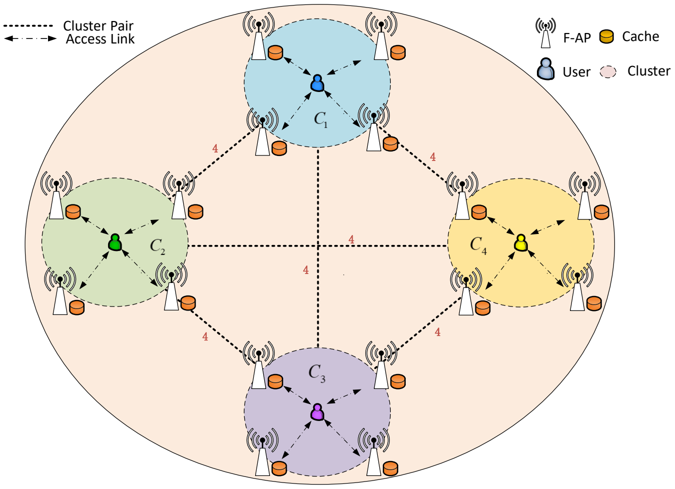

Fig. 3. Illustration of graph $\mathcal{G}^*$ constructed by $V = 4$, $\gamma = 3$.

TABLE III. The coded-multicast contents represented by the edges of $\mathcal{G}^*$ in Fig. 3

Edges	Coded-multicast contents	
(C_1, C_2)	$W_{d_2,1} \oplus W_{d_1,5}$	$W_{d_2,2} \oplus W_{d_1,6}$
	$W_{d_2,3} \oplus W_{d_1,7}$	$W_{d_2,4} \oplus W_{d_1,8}$
(C_1, C_3)	$W_{d_3,1} \oplus W_{d_1,9}$	$W_{d_3,2} \oplus W_{d_1,10}$
	$W_{d_3,3} \oplus W_{d_1,11}$	$W_{d_2,4} \oplus W_{d_1,8}$
(C_1, C_4)	$W_{d_4,1} \oplus W_{d_1,13}$	$W_{d_4,2} \oplus W_{d_1,14}$
	$W_{d_4,3} \oplus W_{d_1,15}$	$W_{d_4,4} \oplus W_{d_1,16}$
(C_2, C_3)	$W_{d_3,5} \oplus W_{d_2,9}$	$W_{d_3,6} \oplus W_{d_2,10}$
	$W_{d_3,7} \oplus W_{d_2,11}$	$W_{d_3,8} \oplus W_{d_2,12}$
(C_2, C_4)	$W_{d_4,5} \oplus W_{d_2,13}$	$W_{d_4,6} \oplus W_{d_2,14}$
	$W_{d_4,7} \oplus W_{d_2,15}$	$W_{d_4,8} \oplus W_{d_2,16}$
(C_3, C_4)	$W_{d_4,9} \oplus W_{d_3,13}$	$W_{d_4,10} \oplus W_{d_3,14}$
	$W_{d_4,11} \oplus W_{d_3,15}$	$W_{d_4,12} \oplus W_{d_3,16}$

MDS codes, another $L-S$ distinct coded packets are needed to recover file W_n. Note that there are $K+l$ coded packets of file W_n stored at the cloud server. Since there are $|\mathbb{C}_n| \cdot S$ distinct coded packets of file W_n cached in the F-APs of the clusters in $\mathbb{C}_n$, $K + l - |\mathbb{C}_n| \cdot S$ coded packets of file W_n are stored in the cloud server, which are different from all the coded packets cached in the clusters in $\mathbb{C}_n$. In other words, there are $K+l-|\mathbb{C}_n| \cdot S$ coded packets of file W_n which are useful for the users in $\mathbb{U}_n$, and can be used to construct multicast messages. On one hand, if these coded packets are enough to satisfy the requests for file W_n, i.e., $K+l-|\mathbb{C}_n| \cdot S \geq L-S$, the cloud server transmits $L-S$ out of the $K+l-|\mathbb{C}_n| \cdot S$ coded packets to the clusters in $\mathbb{C}_n$. Then, the users in $\mathbb{U}_n$ will have enough coded packets to recover file W_n. The fronthaul load consumed by the $L-S$ coded packets is $(L-S) \cdot \frac{F}{L}$. On the other hand, if $K+l-|\mathbb{C}_n| \cdot S < L-S$, these coded packets are not enough for the users to recover file W_n. The cloud server transmits all the $K + l - |\mathbb{C}_n| \cdot S$ coded packets to the clusters in $\mathbb{C}_n$ first, then the proposed weighted graph based coded caching scheme is used to transmit the remaining coded packets. In this case, the fronthaul load is $(K+l-|\mathbb{C}_n| \cdot S) \cdot \frac{F}{L} + \mathcal{R}(\mathbb{C}_n, K+l-|\mathbb{C}_n| \cdot S + S)$. Let $r(\mathbb{C}_n)$ denote the fronthaul load consumed by satisfying the consistent requests for file W_n, where $n \leq N^{\mathrm{t}}$, which can be formulated as:

$$r(\mathbb{C}_n) = \begin{cases} (L-S) \cdot \frac{F}{L} & L-S \leq \eta \\ \eta \cdot \frac{F}{L} + \mathcal{R}(\mathbb{C}_n, \eta + S) & L-S > \eta, \end{cases} \tag{5}$$

where $\eta = K + l - |\mathbb{C}_n| \cdot S$. The fronthaul load consumed by satisfying all the consistent requests on cached files can be further derived as:

$$R^{\mathrm{c}} = \sum_{n=1}^{N^{\mathrm{t}}} \mathbb{I}_{\{\mathbb{C}_n \in C^{\mathrm{c}}\}} \cdot r(\mathbb{C}_n), \tag{6}$$

where $\mathbb{I}_{\{x\}}$ is the indicator function, $\mathbb{I}_{\{x\}} = 1$ when x is true; otherwise $\mathbb{I}_{\{x\}} = 0$.

2) Inconsistent Requests: Consider $\mathcal{U}^{\mathrm{i}}$ and C^{i}, the requested files of users in $\mathcal{U}^{\mathrm{i}}$ are all different. The cloud server directly transmits the uncoded files to the corresponding clusters in C^{i} to satisfy all the inconsistent requests for uncached files. The requests for cached files are satisfied similarly to the case of consistent requests. Let $C^{\mathrm{I}} = \{\mathbb{C}_n \mid \mathbb{C}_n \in C^{\mathrm{i}}, n \leq N^{\mathrm{t}}\}$. Specifically, each cluster in C^{I} can provide its served user with S distinct coded packets of the desirable file. Then, users served by the clusters in C^{I} would need another $L-S$ distinct coded packets to recover their desirable files, which will be transmitted as described in Algorithm 3. Let R^{i} denote the fronthaul load consumed by satisfying all the inconsistent requests for cached files, which can be expressed as:

$$R^{\mathrm{i}} = \mathcal{R}(C^{\mathrm{I}}, S). \tag{7}$$

The detailed description of our proposed joint MDS codes and weighted graph based coded caching policy is presented in Algorithm 4.

D. Performance Analysis

According to the above discussions, we have derived the expressions of the fronthaul load for both consistent requests and inconsistent requests for cached files, which are given in (6) and (7), respectively. If a user requests the uncached file W_n, i.e., $n > N^{\mathrm{t}}$, then regardless of the category of $\mathbb{C}_n$, the cloud server only needs to transmit the whole file W_n (size of F bits) to the clusters in $\mathbb{C}_n$. The fronthaul load consumed by satisfying the requests for uncached files is therefore $\sum_{n=\mathrm{N}^{\mathrm{t}}+1}^{N} \mathbb{I}_{\{|\mathbb{C}_n|>0\}} \cdot F$.

The total fronthaul load R of the proposed JMWG policy is composed of the fronthaul load consumed by satisfying the consistent requests for cached files R^{c}, the fronthaul load consumed by satisfying the inconsistent requests for cached files R^{i} and the fronthaul load consumed by satisfying the requests for uncached files:

$$\begin{aligned} R &= R^{\mathrm{c}} + R^{\mathrm{i}} + \sum_{n=N^{\mathrm{t}}+1}^{N} \mathbb{I}_{\{|\mathbb{C}_n|>0\}} \cdot F \\ &= \sum_{n=1}^{N^{\mathrm{t}}} \mathbb{I}_{\{\mathbb{C}_n \in C^{\mathrm{c}}\}} \cdot r(\mathbb{C}_n) + \mathcal{R}(C^{\mathrm{I}}, S) + \sum_{n=N^{\mathrm{t}}+1}^{N} \mathbb{I}_{\{|\mathbb{C}_n|>0\}} \cdot F, \end{aligned} \tag{8}$$

where $r(\mathbb{C}_n)$ and $\mathcal{R}(C^{\mathrm{I}}, S)$ are given in (5) and (4), respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

According to (5) and (7), the values of R^{c} and R^{i} depend on the values of $\mathcal{R}(\mathbb{C}_n, \eta + S)$ and $\mathcal{R}(C^{\mathrm{I}}, S)$, respectively. Furthermore, the values of $\mathcal{R}(\mathbb{C}_n, \eta + S)$ and $\mathcal{R}(C^{\mathrm{I}}, S)$ are dependent on (4). Specifically, in (4), for given V and γ, $\mathcal{W}(\mathcal{G}^*)$ is the optimization target of Problem (3). Since Problem (3) is an integer programming problem, it is hard to give an exact expression for $\mathcal{W}(\mathcal{G}^*)$. However, for given V and γ, the optimal graph $\mathcal{G}^*$ can be generated from the degree sequence $(\gamma, \gamma, \ldots, \gamma)$ or $(\gamma+1, \gamma, \ldots, \gamma)$, each with V elements, according to Corollary 1 and Corollary 2. And the number of edges of the corresponding graph is

$$E(\mathcal{G}^*) = \begin{cases} \frac{V\cdot\gamma}{2} & (V\cdot\gamma) \bmod 2 = 0 \\ \frac{V\cdot\gamma+1}{2} & (V\cdot\gamma) \bmod 2 = 1. \end{cases}$$

As mentioned previously, the range of the edge weight is $0 \leq w_{i,j} \leq S$, so the lower and upper bounds of $\mathcal{W}(\mathcal{G}^*)$ are 0 and $E(\mathcal{G}^*)\cdot S$, respectively.

Note that the lower bound of $\mathcal{W}(\mathcal{G}^*)$ can be further improved. Remind that $\mathcal{W}(\mathcal{G}^*)$ represents the number of multicast messages for satisfying all the users served by the clusters in C^{a}. Since each user already has n_{a} coded packets of the desirable file, for any cluster C_i in C^{a}, the user served by cluster C_i needs another $L - n_{\mathrm{a}}$ distinct coded packets to recover the desirable file. Consider that cluster C_i is also a vertex in graph $\mathcal{G}^*$, according to constraint (3a), the sum of the weights of all edges containing the vertex should be equal or greater than $L - n_{\mathrm{a}}$. So the lower bound of $\mathcal{W}(\mathcal{G}^*)$ can be further improved as $\frac{V\cdot(L-n_{\mathrm{a}})}{2}$.

Algorithm 4 Joint MDS Codes and Weighted Graph Based Coded Caching Policy

```
    Placement phase: Execute Algorithm 1.
    Delivery phase:
    Initialize $\boldsymbol{d}$, $C$, $\mathcal{U}$, $S$ $L$, $l$, $N^{\mathrm{t}}$.
    Generate $\mathcal{U}^{\mathrm{c}}$, $\mathcal{U}^{\mathrm{i}}$, $C^{\mathrm{c}}$, $C^{\mathrm{i}}$.
 5: for $\mathbb{C}_n \in C^{\mathrm{c}}$ do
        if $n \leq N^t$ then
            if $L - S \leq K + l - |\mathbb{C}_n| \cdot S$ then
                Cloud server sends $L-S$ out of $K+l-|\mathbb{C}_n|\cdot S$ coded packets of file $W_n$ to the
                clusters in $\mathbb{C}_n$.
10:         else
                Cloud server sends $K+l-|\mathbb{C}_n|\cdot S$ coded packets of file $W_n$ to the clusters in $\mathbb{C}_n$,
                $n_{\mathrm{a}} = S + K + l - |\mathbb{C}_n| \cdot S$, $C^{\mathrm{a}} = \mathbb{C}_n$,
                Execute Algorithm 3.
            end if
15:     else
            Cloud server transmits a multicast message $W_n$ to the clusters in $\mathbb{C}_n$.
        end if
    end for
    $C^{\mathrm{a}} = \emptyset$.
20: for $\mathbb{C}_n \in C^{\mathrm{i}}$ do
        if $n \leq N^t$ then
            $C^{\mathrm{a}} = C^{\mathrm{a}} \cup \mathbb{C}_n$
        else
            Cloud server transmits file $W_n$ to the cluster in $\mathbb{C}_n$.
25:     end if
    end for
    $n_{\mathrm{a}} = S$
    Execute Algorithm 3.
```

Let $\mathcal{R}^{\mathrm{lb}}(C^{\mathrm{a}}, n_{\mathrm{a}})$ and $\mathcal{R}^{\mathrm{ub}}(C^{\mathrm{a}}, n_{\mathrm{a}})$ denote the lower and upper bounds of $\mathcal{R}(C^{\mathrm{a}}, n_{\mathrm{a}})$, respectively. Then $\mathcal{R}^{\mathrm{lb}}(C^{\mathrm{a}}, n_{\mathrm{a}})$ and $\mathcal{R}^{\mathrm{ub}}(C^{\mathrm{a}}, n_{\mathrm{a}})$ can be formulated as:

$$\mathcal{R}^{\mathrm{lb}}(C^{\mathrm{a}}, n_{\mathrm{a}}) = \begin{cases} \left[\frac{(|C^{\mathrm{a}}|-1)\cdot S}{2} + (L - n_{\mathrm{a}} - \delta)\right]\cdot|C^{\mathrm{a}}|\cdot\frac{F}{L} & L - n_{\mathrm{a}} > \delta \\ \frac{(L-n_{\mathrm{a}})\cdot|C^{\mathrm{a}}|}{2}\cdot\frac{F}{L} & L - n_{\mathrm{a}} \leq \delta, \end{cases} \tag{9}$$

$$\mathcal{R}^{\mathrm{ub}}(C^{\mathrm{a}}, n_{\mathrm{a}}) = \begin{cases} \left[\frac{(|C^{\mathrm{a}}|-1)\cdot S}{2} + (L - n_{\mathrm{a}} - \delta)\right]\cdot|C^{\mathrm{a}}|\cdot\frac{F}{L} & L - n_{\mathrm{a}} > \delta \\ E(\mathcal{G}^*)\cdot S\cdot\frac{F}{L} & L - n_{\mathrm{a}} \leq \delta, \end{cases} \tag{10}$$

where $\delta = (|C^{\mathrm{a}}| - 1)\cdot S$.

Let $r^{\mathrm{lb}}(\mathbb{C}_n)$ and $r^{\mathrm{ub}}(\mathbb{C}_n)$ denote the lower and upper bounds of $r(\mathbb{C}_n)$, respectively. Based on (9) and (10), $r(\mathbb{C}_n)$ can be formulated as:

$$r^{\mathrm{lb}}(\mathbb{C}_n) = \begin{cases} (L-S)\cdot\frac{F}{L} & L - S \leq \eta \\ \eta\cdot\frac{F}{L} + \mathcal{R}^{\mathrm{lb}}(\mathbb{C}_n, \eta + S) & L - S > \eta, \end{cases} \tag{11}$$

$$r^{\mathrm{ub}}(\mathbb{C}_n) = \begin{cases} (L-S)\cdot\frac{F}{L} & L - S \leq \eta \\ \eta\cdot\frac{F}{L} + \mathcal{R}^{\mathrm{ub}}(\mathbb{C}_n, \eta + S) & L - S > \eta, \end{cases} \tag{12}$$

where $\eta = K + l - |\mathbb{C}_n|\cdot S$.

From the above analysis, we can now obtain the lower bound R^{lb} and upper bound R^{ub} of the overall fronthaul load of the proposed JMWG policy:

$$R^{\mathrm{lb}} = \sum_{n=1}^{N^{\mathrm{t}}} \mathbb{I}_{\{\mathbb{C}_n \in C^{\mathrm{c}}\}}\cdot r^{\mathrm{lb}}(\mathbb{C}_n) + \mathcal{R}^{\mathrm{lb}}(C^{\mathrm{I}}, S) + \sum_{n=N^{\mathrm{t}}+1}^{N} \mathbb{I}_{\{|\mathbb{C}_n|>0\}}\cdot F, \tag{13}$$

$$R^{\mathrm{ub}} = \sum_{n=1}^{N^{\mathrm{t}}} \mathbb{I}_{\{\mathbb{C}_n \in C^{\mathrm{c}}\}}\cdot r^{\mathrm{ub}}(\mathbb{C}_n) + \mathcal{R}^{\mathrm{ub}}(C^{\mathrm{I}}, S) + \sum_{n=N^{\mathrm{t}}+1}^{N} \mathbb{I}_{\{|\mathbb{C}_n|>0\}}\cdot F. \tag{14}$$

By fully utilizing the property of MDS codes and the cache contents between any two clusters, our proposed policy can construct considerable multicast messages, leading to a significant reduction in the overall fronthaul load.

E. Complexity Analysis

Problem (3) is an integer programming problem, which has been proven to be NP-hard. If ignoring constraint (3a), the number of combinations of weights is $S^{E(\mathcal{G}^*)}$. Therefore, the complexity of problem (3) in the worst case would be $O(S^{E(\mathcal{G}^*)})$. Algorithm 4 needs $|C^{\mathrm{c}}| + 1$ loops to satisfy all requests and $|C^{\mathrm{i}}|$ loops to generate C^{a} for inconsistent requets. So the worst complexity of Algorithm 4 is $O((|C^{\mathrm{c}}|+1)\cdot S^{E(\mathcal{G}^*)} + |C^{\mathrm{i}}|)$. Although the complexity of the proposed algorithm is exponential, Corollary 1 and Corollary 2 guarantee that the optimal graph $\mathcal{G}^*$ has the least number of edges, i.e., $E(\mathcal{G}^*)$ is minimized, which can greatly reduce the complexity. According to Theorem 1 and Corollary 1, $E(\mathcal{G}^*) = \frac{V\cdot\gamma}{2}$ ($V\cdot\gamma$ is even) or $E(\mathcal{G}^*) = \frac{V\cdot\gamma+1}{2}$ ($V\cdot\gamma$ is odd). Consider that $V\cdot\gamma$ is odd. Then, we have $E(\mathcal{G}^*) = \frac{V\cdot\gamma+1}{2} = \frac{|C^{\mathrm{a}}|\cdot\gamma+1}{2}$, where C^{a} is generated according to line 11 or line 22 in Algorithm 4 and is related to the number of clusters C. Due to the nonuniform popularity,

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

the set of clusters $\mathbb{C}_1$ served for the most popular requests should contain the maximum number of clusters, which results in the maximum $|C^{\mathrm{a}}|$. Generally, $|\mathbb{C}_n|$ is much less than C, and so $|C^{\mathrm{a}}|$ is much less than C. Therefore, the time consumption grows exponentially with C but the exponential term $S^{E(\mathcal{G}^*)}$ will be in an acceptable range if C is not very large.

IV. Simulation Results

In this section, the performance of the proposed joint MDS codes and weighted graph based coded caching policy (JMWG) is evaluated via simulations. In order to investigate the performance of our proposed policy, we adopt traditional MDS-based uncoded delivery policy (TMBU) and MCAT in [37] as baselines. Furthermore, the lower bound (LB) and the upper bound (UB) of the proposed JMWG policy are also evaluated. The cache placement of all policies is performed by Algorithm 1, and all users request files according to $\boldsymbol{p}$ as given in (1). The parameters are set as follows: $F = 1$ Gb, $M = 20$, $N = 100$, $N^{\mathrm{t}} = 80$, and $U = C = 50$.

In Fig. 4, we depict how the number of file fragments, i.e., L, affects the fronthaul load of each policy with $K = 300$, $l = 5$, and $\alpha = 0.5$. As shown, the performance of JMWG is superior to the other policies. Specifically, compared with TMBU policy, the proposed JMWG policy can provide 44% savings in the fronthaul load. It can be observed that the fronthaul loads of all the three policies increase with the number of file fragments. The reason is that with L increasing, the number of coded packets needed to reconstruct the requested file also increases, which means more messages will be transmitted via the fronthaul link. It can also be observed that the fronthaul load of all policies are the same when $L < S$. The reason is that the local S clustered F-APs are able to provide their served users with required cached L coded packets. Therefore, the fronthaul load is only constituted by the transmission cost of the requested but uncached files. It can also be observed that when L is larger, the performance gaps of JMWG with the other policies become larger, which reveals that our proposed policy can construct more multicast messages than the other policies. Furthermore, the performance gap between JMWG and its LB is extremely small, which verifies that JMWG can fully utilize the multicast opportunities among the clusters.

In Fig. 5, we show the effect of the number of F-APs, i.e., K, on the fronthaul load of each policy with $L = 30$, $l = 5$ and $\alpha = 0.5$. As shown, JMWG offers a better performance than TMBU and MCAT. It can be observed that with K increasing, the fronthaul loads of all policies decrease. The reason is that the local cluster can provide its served user with more coded packets. When K is larger, the number of F-AP in a cluster, i.e., S, is larger, and the local cluster can provide more coded packets. Then the number of coded packets transmitted from the cloud server decreases, which results in the sightly fluctuation in the performance of UB. It can also be observed that the performance gap between JMWG and the other two policies is very large across all K values. The result verified that the proposed JMWG policy can construct considerable multicast messages, which significantly reduces the fronthaul load. Moreover, the better performance of its UB and the smaller gap between JMWG and LB ensure the superiority of the proposed coded caching policy in reducing the fronthaul load.

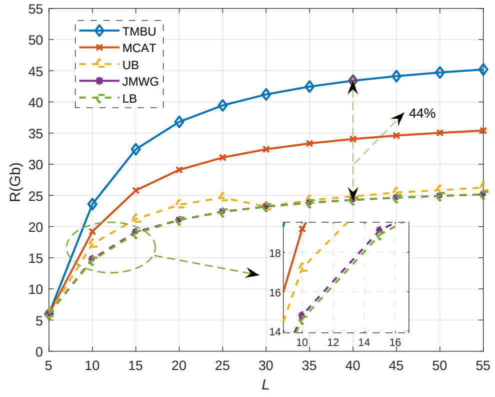

Fig. 4. Fronthaul load versus the number of file fragments L with $K = 300$, $l = 5$.

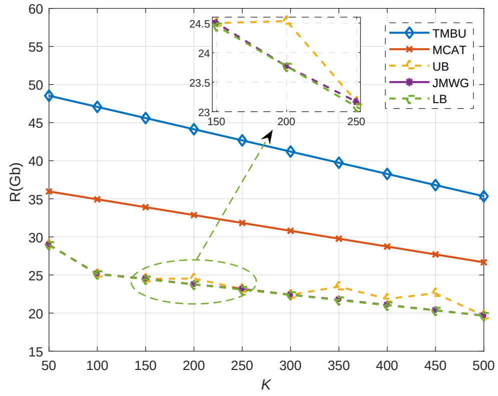

Fig. 5. Fronthaul load versus the number of F-APs K with $L = 30$, $l = 5$.

In Fig. 6, we show the effect of the number of the redundant coded packets, i.e., l, on the fronthaul load of each policy with $K = 300$ and $L = 310$. Since the l redundant MDS coded packets are meant to construct multicast messages for consistent requests, we focus on cases with higher consistency of requests by letting $\alpha = 1.5$. As shown, JMWG outperforms TMBU and MCAT in reducing the fronthaul load. It can be observed that with l increasing, the fronthaul load of TMBU remains as a constant. The reason is that TMBU cannot construct multicast messages during the delivery phase. It can also be observed that the fronthaul loads of MCAT and JMWG decrease with increasing l. The reason is that the cloud server can construct multicast messages for consistent requests based on the l redundant coded packets, which greatly reduces the fronthaul load. The performance gaps between JMWG, UB and LB are decreasing with increasing l. This is due to the fact that with l increasing, the number of multicast messages constructed based on the l redundant coded packets also increases, which leads to a smaller number of coded packets transmitted by Algorithm 3. When l is large enough, consistent requests are satisfied by the redundant coded packets without

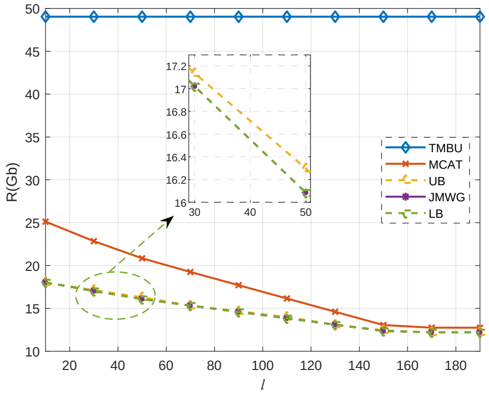

Fig. 6. Fronthaul load versus the number of the redundant coded packets l with $K = 300$, $L = 310$, $\alpha = 1.5$.

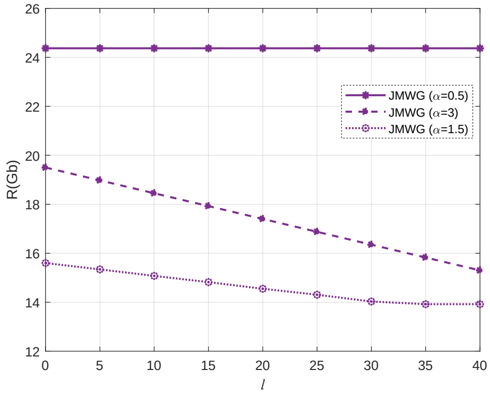

Fig. 7. Fronthaul load versus the number of the redundant coded packets l with $K = 300$, $L = 200$ and different α values.

using Algorithm 3. Therefore, according to the performance analysis in Section III-D, the performances of UB, LB and JMWG become the same.

In Fig. 7, we show the effect of the number of the redundant coded packets, i.e., l, on the fronthaul load of our proposed JMWG policy with $L = 200$, $K = 300$ and different α values. Note that α is the Zipf parameter which characterizes the popularities of files in the library. It can be observed that the fronthaul loads decrease with increasing l when $\alpha = 1.5$ and $\alpha = 3$, whereas the fronthaul load with $\alpha = 0.5$ remains as a constant. The reason is that when $\alpha = 1.5$ and $\alpha = 3$, the consistency of all users' requests is high, i.e., there will be multiple users requesting the same file under such occasions. For file W_n, the number of users who request file W_n is $|\mathbb{C}_n|$. It is easy to see that the number of coded packets of file W_n stored in the cloud server, which are different from the coded packets cached in the $|\mathbb{C}_n|$ clusters is $K+l-S\cdot|\mathbb{C}_n|$. For given relative larger $|\mathbb{C}_n|$ (which is characterized by different α values), $K+l-S\cdot|\mathbb{C}_n|$ increases with l. Therefore, the proposed JMWG policy can construct more multicast messages for consistent requests when $\alpha = 1.5$ and $\alpha = 3$. On the contrary, when $\alpha = 0.5$, which means that the value of $|\mathbb{C}_n|$ is relatively smaller, $K-S\cdot|\mathbb{C}_n|$ distinct coded packets are already enough for $|\mathbb{C}_n|$ users to recover file W_n, and the influence of l is therefore weakened. It can also be observed that our proposed JMWG policy performs better for $\alpha = 1.5$ compared with $\alpha = 3$. The reason is that when $\alpha = 3$, the consistency of requests is higher, meaning that most of users request the same file, i.e., $|\mathbb{C}_n|_{\alpha=3} > |\mathbb{C}_n|_{\alpha=1.5}$, where $|\mathbb{C}_n|_{\alpha=\alpha_0}$ is the number of users requesting the same file W_n when $\alpha = \alpha_0$. As a consequence, $K+l-S\cdot|\mathbb{C}_n|_{\alpha=1.5} > K+l-S\cdot|\mathbb{C}_n|_{\alpha=3}$, which leads to the performance gap of the two cases.

V. CONCLUSIONS

In this paper, we have proposed a joint MDS codes and weighted graph based coded caching policy for F-RANs. Our proposed coded caching policy constructs multicast messages for both consistent and inconsistent requests by using the redundant coded packets and homogeneous cache contents between clusters. The analytical results have shown that more multicast messages can be constructed by the redundant coded packets when the consistency of requests is relatively higher. The simulation results have shown that the upper bound of our proposed policy offers a better performance than the other policies and the performance of our proposed policy is extremely close to its lower bound.

Besides, there are some ideal assumptions in the system model, which are expected to be generalized in our future work. As for the situation of different request probability of users, the file selection strategy in Algorithm 1 should be elaborately designed to trade off between the local caching gain and the global multicasting gain, which is currently an open problem. As for the situation of distinct file size, different files can be encoded by MDS codes with different code rates, which keeps the multicasting packets with the same size. As for the situation of distinct number of F-APs in clusters, different numbers of packets are needed for different clusters to recover files. For both of the two aforementioned cases, a new method is needed to construct the optimal graph G^* in Algorithm 3, which will be an NP-hard problem with much higher complexity.

APPENDIX A
PROOF OF THEOREM 2

In order to prove Theorem 2, it is equivalent to proving the following: given a sequence $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_v, \ldots, \theta_V)$, where $\theta_v = \mu$, $1 \le v \le V$ and $1 \le \mu \le V-1$, prove that $\boldsymbol{\theta}$ is graphic.

From [39], we know that for a given sequence $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_v, \ldots, \theta_V)$, $\boldsymbol{\theta}$ is graphic if it satisfies the following conditions:

C1. $\sum\limits_{i=1}^{V} \theta_i$ is even;

C2. $\sum\limits_{i=1}^{v} \theta_i \le v\cdot(v-1) + \sum\limits_{i=v+1}^{V} \min\{v, \theta_i\}$ for $1 \le v \le V$.

Since $\mu \cdot V$ is even, C1 is already satisfied. To facilitate analysis, we reformulate C2 as:

C2. $v\cdot(v-1) + \sum\limits_{i=v+1}^{V} \min\{v, \theta_i\} - \sum\limits_{i=1}^{v} \theta_i \ge 0$ for $1 \le v \le V$.

Consider $\theta_v = \mu$ for $1 \le v \le V$, C2 can be transformed to $v\cdot(v-1) + \sum\limits_{i=v+1}^{V} \min\{v, \mu\} - v\cdot\mu \ge 0$, for $1 \le v \le V$. When $v \le \mu$, C2 can be further transformed to $v\cdot(v-1)+(V-v)\cdot v - v\cdot\mu \ge 0$.

Then we only need to prove that $v \cdot (v-1) + (V-v) \cdot v - v \cdot \mu \geq 0$ holds for $1 \leq v \leq V$ when $v \leq \mu$. As

$$\begin{aligned} & v \cdot (v-1) + (V-v) \cdot v - v \cdot \mu \\ = \; & v^2 - v + V \cdot v - v^2 - v \cdot \mu \\ = \; & v \cdot (V - \mu - 1), \end{aligned} \tag{15}$$

the sign of (15) is the same as the sign of $V - \mu - 1$. Since $\mu \leq V - 1$ holds, then $V - \mu - 1 \geq 0$ holds, i.e., C2 holds when $v \leq \mu$.

Now consider the case of $v > \mu$. C2 can be further transformed to $v \cdot (v-1) + (V-v) \cdot \mu - v \cdot \mu \geq 0$. Now we only need to prove that $v \cdot (v-1) + (V-v) \cdot \mu - v \cdot \mu \geq 0$ holds for $1 \leq v \leq V$ and $v > \mu$. Since

$$\begin{aligned} & v \cdot (v-1) + (V-v) \cdot \mu - v \cdot \mu \\ = \; & v \cdot (v-1) + (V - 2 \cdot v) \cdot \mu \\ > \; & \mu \cdot (v-1) + (V - 2 \cdot v) \cdot \mu \\ = \; & (V - v - 1) \cdot \mu, \end{aligned} \tag{16}$$

it is easy to see that $(V - v - 1) \cdot \mu \geq 0$, i.e., C2 holds when $1 \leq v \leq V - 1$. Then we only need to prove that C2 also holds for $v = V$. C2 can be transformed to $v \cdot (v-1) + (V-v) \cdot \mu - v \cdot \mu$ when $v = V$. Then, we have

$$\begin{aligned} & v \cdot (v-1) + (V-v) \cdot \mu - v \cdot \mu \\ = \; & V \cdot (V-1) + (V-V) \cdot \mu - V \cdot \mu \\ = \; & V \cdot (V-1) - V \cdot \mu \\ = \; & V \cdot (V - 1 - \mu). \end{aligned} \tag{17}$$

From (15), we know that $V - 1 - \mu \geq 0$, i.e., C2 holds when $v = V$.

C1 and C2 are hence both satisfied, i.e., $\boldsymbol{\theta}$ is graphic. This completes the proof.

APPENDIX B
PROOF OF COROLLARY 2

We first prove that the sequence $\boldsymbol{\theta} = (\mu+1, \mu, \ldots, \mu)$ can be graphic, then we discuss the optimality of this construction.

We denote the sequence $(\mu-1, \mu-1, \ldots, \mu-1)$ by $\boldsymbol{\theta}'$, where $\boldsymbol{\theta}'$ is constructed by subtracting 1 from all elements of $\boldsymbol{\theta}$ and removing the first element of $\boldsymbol{\theta}$. From [39], we know that $\boldsymbol{\theta}$ can be graphic if and only if $\boldsymbol{\theta}'$ is graphic. It is easy to see that $(V-1) \cdot (\mu - 1)$ is even. From Theorem 2, there exits a $(\mu - 1)$−regular graph with $V - 1$ vertices, i.e., $\boldsymbol{\theta}'$ can be graphic. So the sequence $\boldsymbol{\theta} = (\mu+1, \mu, \ldots, \mu)$ can be graphic.

Now let us prove the optimality of this construction. In order to minimize the number of edges, the optimal case is when we let the degree of each vertex equals to the lower bound, i.e., $\boldsymbol{\theta}^* = (\mu, \mu, \ldots, \mu)$. However, since $\mu \cdot V$ is odd, the sequence $(\mu, \mu, \ldots, \mu)$ can not be graphic [39]. It is easy to see that $\mu \cdot V + 1$ is the closest even number to $\mu \cdot V$. So the graph $\mathcal{G}^*$ constructed by the sequence $\boldsymbol{\theta} = (\mu+1, \mu, \ldots, \mu)$ is optimal for all graphs satisfying $D(v) \geq \mu$ for $1 \leq v \leq V$. This completes the proof.

REFERENCES

[1] X. You, C. Wang, J. Huang, and et al., "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China-Inf. Sci.*, vol. 64, no. 1, pp. 1–74, Jan. 2021.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, Jul. 2016.

[3] Y. Jiang, A. Peng, C. Wan, Y. Cui, X. You, F. Zheng, and S. Jin, "Analysis and optimization of cache-enabled fog radio access networks: Successful transmission probability, fractional offloaded traffic and delay," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5219–5231, May 2020.

[4] A. Peng, Y. Jiang, M. Bennis, and et al., "Performance analysis and caching design in fog radio access networks," in *2018 IEEE Globecom Workshops*, Dec. 2018, pp. 1–6.

[5] M. Zhang, Y. Jiang, F. Zheng, and et al., "Cooperative edge caching via federated deep reinforcement learning in fog-rans," in *IEEE ICC 2021 Workshop DAWNZ*, Jun. 2021, pp. 1–6.

[6] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[7] Y. Jiang, M. Ma, M. Bennis, and et al., "A novel caching policy with content popularity prediction and user preference learning in fog-ran," in *2017 IEEE Globecom Workshops*, Dec. 2017, pp. 1–6.

[8] H. Feng, Y. Jiang, D. Niyato, and et al., "Content popularity prediction via deep learning in cache-enabled fog radio access networks," in *2019 IEEE Glob. Commun. Conf.*, Dec. 2019, pp. 1–6.

[9] L. Lu, Y. Jiang, M. Bennis, and et al., "Distributed edge caching via reinforcement learning in fog radio access networks," in *2019 IEEE 89th Veh. Technol. Conf.*, Apr. 2019, pp. 1–6.

[10] Y. Jiang, Y. Hu, M. Bennis, F. Zheng, and X. You, "A mean field game-based distributed edge caching in fog radio access networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1567–1580, Mar. 2020.

[11] Y. Jiang, H. Feng, F. C. Zheng, D. Niyato, and X. You, "Deep learning-based edge caching in fog radio access networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8442–8454, Dec. 2020.

[12] Y. Jiang, C. Wan, M. Tao, F. C. Zheng, P. Zhu, X. Gao, and X. You, "Analysis and optimization of fog radio access networks with hybrid caching: Delay and energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 69–82, Jan. 2021.

[13] X. Cui, Y. Jiang, X. Chen, F. Zheng, and X. You, "Graph-based cooperative caching in fog-ran," in *2018 Int. Conf. Comput. Netw. Commun.*, Mar. 2018, pp. 166–171.

[14] Y. Jiang, X. Cui, M. Bennis, and F. C. Zheng, "Cooperative caching in fog radio access networks: A graph-based approach," *IET Commun.*, vol. 13, no. 20, pp. 3519–3528, Nov. 2019.

[15] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[16] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[17] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.

[18] W. Huang, Y. Jiang, M. Bennis, F. Zheng, H. Gacanin, and X. You, "Decentralized asynchronous coded caching in Fog-RAN," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug. 2018, pp. 1–6.

[19] Y. Jiang, W. Huang, M. Bennis, and F. Zheng, "Decentralized asynchronous coded caching design and performance analysis in fog radio access networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 540–551, Mar. 2020.

[20] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1318–1332, Mar. 2020.

[21] E. Lampiris and P. Elia, "Full coded caching gains for cache-less users," *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7635–7651, Dec. 2020.

[22] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Device-to-device coded-caching with distinct cache sizes," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2748–2762, May 2020.

[23] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.

[24] Q. Wang, Y. Cui, S. Jin, J. Zou, C. Li, and H. Xiong, "Optimization-based decentralized coded caching for files and caches with arbitrary sizes," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2090–2105, Apr. 2020.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

[25] N. Mital, D. Gndz, and C. Ling, "Coded caching in a multi-server system with random topology," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4620–4631, Aug. 2020.

[26] V. Bioglio, F. Gabry, and I. Land, "Optimizing MDS codes for caching at the edge," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2015, pp. 1–6.

[27] F. Gabry, V. Bioglio, and I. Land, "On energy-efficient edge caching in heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3288–3298, Dec. 2016.

[28] F. Yin, A. Wang, D. Liu, and Z. Zhang, "Energy-aware joint user association and resource allocation for coded cache-enabled HetNets," *IEEE Access*, vol. 7, pp. 94 128–94 142, Jun. 2019.

[29] Y. Zhou, M. Peng, S. Yan, and Y. Sun, "Deep reinforcement learning based coded caching scheme in fog radio access networks," in *2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Aug. 2018, pp. 309–313.

[30] A. Piemontese and A. Graell i Amat, "MDS-coded distributed caching for low delay wireless content delivery," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1600–1612, Feb. 2019.

[31] S. Gao, P. Dong, Z. Pan, and G. Y. Li, "Reinforcement learning based cooperative coded caching under dynamic popularities in ultra-dense networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5442–5456, May 2020.

[32] E. Ozfatura and D. Gndz, "Mobility-aware coded storage and delivery," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3275–3285, Jun. 2020.

[33] A. R. Elkordy, A. S. Motahari, M. Nafie, and D. Gndz, "Cache-aided combination networks with interference," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 148–161, Jan. 2020.

[34] D. Ko and W. Choi, "Probabilistic caching based on MDS code in cooperative mobile edge caching networks," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, Aug. 2020, pp. 1–6.

[35] M. Bayat, R. K. Mungara, and G. Caire, "Achieving spatial scalability for coded caching via coded multipoint multicasting," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 227–240, Jan. 2019.

[36] Y. Lu, W. Chen, and H. V. Poor, "Coded joint pushing and caching with asynchronous user requests," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1843–1856, Aug. 2018.

[37] X. Wu, Q. Li, V. C. M. Leung, and P. C. Ching, "Joint fronthaul multicast and cooperative beamforming for cache-enabled cloud-based small cell networks: An MDS codes-aided approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4970–4982, Oct. 2019.

[38] B. Wang, Y. Jiang, F. Zheng, M. Bennis, X. Gao, and X. You, "Joint redundant MDS codes and cluster cooperation based coded caching in fog radio access networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[39] J. A. Bondy and Murty, *Graph theory with applications*. London: Macmillan, 1976.

Yanxiang Jiang (S'03-M'07-SM'18) received the B.S. degree in electrical engineering from Nanjing University, Nanjing, China, in 1999 and the M.S. and Ph.D. degrees in communications and information systems from Southeast University, Nanjing, China, in 2003 and 2007, respectively.

Dr. Jiang was a Visiting Scholar with the Signals and Information Group, Department of Electrical and Computer Engineering, University of Maryland at College Park, College Park, MD, USA, in 2014. He is currently an Associate Professor with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. His research interests are in the area of broadband wireless mobile communications, covering topics such as machine learning for wireless communications, edge intelligence, B5G and 6G mobile communications.

Bao Wang received the M.S. degree in communications and information systems from Southeast University, Nanjing, China.

His research interests include radio resource management and edge caching.

Fu-Chun Zheng obtained the BEng (1985) and MEng (1988) degrees in radio engineering from Harbin Institute of Technology, China, and the PhD degree in Electrical Engineering from the University of Edinburgh, UK, in 1992.

From 1992 to 1995, he was a post-doctoral research associate with the University of Bradford, UK, Between May 1995 and August 2007, he was with Victoria University, Melbourne, Australia, first as a lecturer and then as an associate professor in mobile communications. He was with the University of Reading, UK, from September 2007 to July 2016 as a Professor (Chair) of Signal Processing. He has also been a Distinguished Adjunct Professor with Southeast University, China, since 2010. Since August 2016, he has been with Harbin Institute of Technology (Shenzhen), China, as a Distinguished Professor. He has been awarded two UK EPSRC Visiting Fellowships - both hosted by the University of York (UK): first in August 2002 and then again in August 2006. Over the past two decades, Dr Zheng has also carried out many government and industry sponsored research projects - in Australia, the UK, and China. He has been both a short term visiting fellow and a long term visiting research fellow with British Telecom, UK. Dr Zhengs current research interests include ultra-dense networks (UDN), ultra-reliable low latency communications (URLLC), multiple antenna systems, green communications, and machine learning based communications.

He has been an active IEEE member since 1995. He was an editor (2001 C 2004) of IEEE Transactions on Wireless Communications. In 2006, Dr Zheng served as the general chair of IEEE VTC 2006-S, Melbourne, Australia (www.ieeevtc.org/vtc2006spring) - the first ever VTC held in the southern hemisphere in VTCs history of six decades. More recently he was the executive TPC Chair for IEEE VTC 2016-S, Nanjing, China (www.ieeevtc.org/vtc2016spring) - the first ever VTC held in mainland China.

Mehdi Bennis (S'07-AM'08-SM'15) received his M.Sc. degree in electrical engineering jointly from EPFL, Switzerland, and the Eurecom Institute, France, in 2002. He obtained his Ph.D. from the University of Oulu in December 2009 on spectrum sharing for future mobile cellular systems. Currently he is a professor at the University of Oulu and an Academy of Finland research fellow. His main research interests are in radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond. He has co-authored one book and published more than 200 research papers in international conferences, journals, and book chapters. He was the recipient of the prestigious 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the Journal of Wireless Communications and Networks, and the 2017 all-University of Oulu Award for Research.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3143888, IEEE Transactions on Wireless Communications

Xiaohu You (SM'11-F'12) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 1982, 1985, and 1988, respectively. Since 1990, he has been working with National Mobile Communications Research Laboratory at Southeast University, where he holds the ranks of professor and director. He is the Chief of the Technical Group of China 3G/B3G Mobile Communication R&D Project. His research interests include mobile communications, adaptive signal processing, and artificial neural networks, with applications to communications and biomedical engineering.

Dr. You was a recipient of the Excellent Paper Prize from the China Institute of Communications in 1987; the Elite Outstanding Young Teacher award from Southeast University in 1990, 1991, and 1993; and the National Technological Invention Award of China in 2011. He was also a recipient of the 1989 Young Teacher Award of Fok Ying Tung Education Foundation, State Education Commission of China.