

Joint Redundant MDS Codes and Cluster Cooperation Based Coded Caching in Fog Radio Access Networks

Bao Wang^{1,2,3}, Yanxiang Jiang^{1,2,3,*}, Fu-Chun Zheng^{1,4}, Mehdi Bennis⁵, Xiqi Gao¹, and Xiaohu You¹

¹National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China.

²State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

³Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, 865 Changning Road, Shanghai 200050, China

⁴School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China

⁵Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland.

*E-mail: yxjiang@seu.edu.cn

Abstract—In this paper, we investigate maximum-distance separable (MDS) codes and cluster based coded caching in fog radio access networks (F-RANs). In order to minimize the fronthaul rate, multicast opportunities need to be constructed at the cloud server. Firstly, a redundant MDS codes based coded placement scheme is proposed to provide redundant coded packets and symmetrical cache contents. The redundant coded packets can be used to construct multicast opportunities for requests on the same file. Furthermore, based on the symmetrical cache contents, we propose a cluster cooperation based coded delivery scheme, which can induce considerable multicast opportunities between any two clusters regardless of whether the requests are on the same file or not. Finally, by utilizing the redundant coded packets and the symmetry of cache contents, a joint redundant MDS codes and cluster cooperation based coded caching policy is proposed to minimize the fronthaul rate. Simulation results show that our proposed policy can provide 30% savings of the fronthaul rate compared to the MDS-based uncoded delivery policy.

Index Terms—Coded caching, fog radio access networks, maximum-distance separable code, fronthaul rate.

I. INTRODUCTION

As video consumption has become the mainstream of demands, heavy traffic are brought to the peak periods of wireless networks, which often leads to network congestion and poor quality of services (QoS). In order to alleviate this, fog radio access networks (F-RANs) have raised widespread attention [1], [2]. 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 [3]. Furthermore, coded caching was firstly proposed in [4] and [5] 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 [6] and [7], a decentralized asynchronous coded caching scheme was proposed, which establishes the relationship between the coded-multicasting contents in asynchronous and synchronous coded caching. The coded data placement using maximum-distance separable (MDS) code was utilized in [8], which reduces the delay in the scene of user movement.

The use of MDS codes in heterogeneous wireless networks was studied in [9]–[11]. The authors in [9] formulated the op-

timal MDS encoding caching scheme as a convex optimization to minimize the backhaul rate. A caching strategy combining file splitting with MDS encoding was proposed in [10], which can improve the overall energy efficiency. By designing the number of MDS coded content cached at each small base station, the authors in [11] were committed to minimizing the power consumption. A deep reinforcement learning method was proposed in [12], where each F-AP in F-RANs obtains the number of MDS coded packets for caching by autonomous learning.

However, the papers mentioned above mainly focus on designing the cache placement phase by the MDS codes, and 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 [13], which uses the scheme in [4] to construct multicast messages and further encode the multicast messages by MDS codes. In [14], 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 was proposed in [15], which constructs multicast opportunities for requests on the same file by the redundancy of MDS codes. Nevertheless, the benefits of MDS codes in fronthaul multicast and cooperative transmissions have not been well unleashed yet [15].

Motivated by the aforementioned facts, we propose a joint redundant MDS codes and cluster cooperation based coded caching policy. In order to fully utilize the property of MDS codes, the cache content of each F-AP should be elaborately designed. Therefore, we propose a redundant MDS codes based coded placement scheme. By storing extra redundant coded packets at the cloud server instead of F-APs, the cloud server is able to construct multicast messages for requests on the same file. Furthermore, by using the cache contents between any two clusters, we propose a cluster cooperation based coded delivery scheme, which can construct considerable multicast messages for all kinds of requests. In our proposed policy, the cloud server can benefit from the property of MDS codes and the cache contents between any two

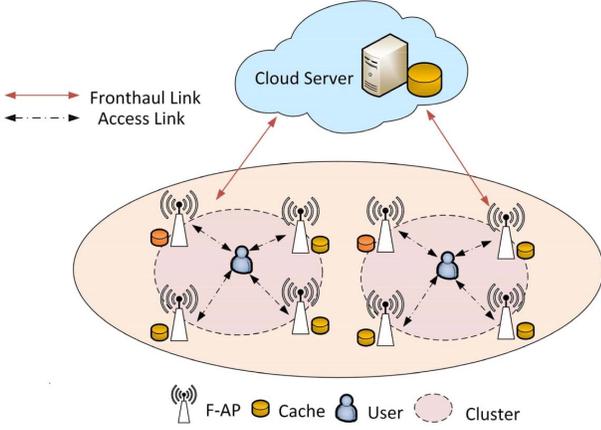


Fig. 1. Illustration of the F-RANs with $K = 8$, $U = C = 2$, $S = 4$.

clusters. As a result, the fronthaul rate can be significantly decreased. To the best of our knowledge, this is the first work to investigate the employment of MDS codes in constructing multicast messages for both requests on the same file and different files.

The rest of this paper is organized as follows. In Section II, the system model is presented. Our proposed joint redundant MDS codes and cluster cooperation based coded caching policy is presented in Section III. Simulation results are shown in Section IV. 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 cloud server. The cloud server has access to a library of N files $W_1, W_2, \dots, W_n, \dots, W_N$. The size of each file is F bits. Let $\mathcal{N} = \{1, 2, \dots, n, \dots, N\}$ denote the index set of N files. Let $\mathcal{K} = \{1, 2, \dots, k, \dots, K\}$ 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 MF bits for $0 < M < N$. C clusters are formed based on the geographical situation of K F-APs. Each cluster contains $S = \lfloor \frac{K}{C} \rfloor$ F-APs and only serves a single user at a time. It is easy to see that $K \geq CS$. Let $\mathcal{C} = \{1, 2, \dots, c, \dots, C\}$ denote the index set of C clusters. Let C_c denote the c th cluster, and we assume that the F-APs in C_c are indexed by $\{S(c-1) + 1, \dots, Sc\} \subset \mathcal{K}$. U users connect the F-APs through access links, where $U \leq C$. Let $\mathcal{U} = \{1, 2, \dots, u, \dots, U\}$ denote the index set of U users. Without loss of generality, we assume that user u is served by C_u , where $0 < u \leq U \leq C$.

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

$$p_n = \frac{1/n^\alpha}{\sum_{n=1}^N 1/n^\alpha}, \quad (1)$$

where α is the parameter of the Zipf distribution. Let $\mathbf{p} = [p_1, p_2, \dots, p_n, \dots, p_N]^T$ denote the popularity vector of N files. We assume that the popularity of N files are in a descending order.

Firstly, we use MDS codes to encode the caching files. Specifically, 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 $W_{n,1}, W_{n,2}, \dots, W_{n,K+l}$, where $K+l \geq L$. The l redundant coded packets are used to construct multicast opportunities for requests on the same file in the delivery phase. The size of each coded packet is the same as the 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. The cache contents of K F-APs are different from each other. Note that the cache size of each F-AP is MF bits, and $N^t \leq ML$ is needed to satisfy the cache constraint.

In the delivery phase, U users initiate requests for files in the library. We refer to multiple users request the same file and different files as *consistent requests* and *inconsistent requests*, respectively. Let $\mathbf{d} = [d_1, d_2, \dots, d_u, \dots, d_U]^T$ denote the request vector of U users, where $d_u \in \mathcal{N}$ is the request of user u . Based on \mathbf{d} , U users can be classified into $\mathcal{U} = \{\mathbb{U}_1, \mathbb{U}_2, \dots, \mathbb{U}_n, \dots, \mathbb{U}_N\}$, where $\mathbb{U}_n = \{u \mid d_u = n, \forall u \in \mathcal{U}\}$ denotes the index set of users who request W_n . Furthermore, $\mathcal{U}^c = \{\mathbb{U}_n \mid |\mathbb{U}_n| \geq 2, \forall \mathbb{U}_n \in \mathcal{U}\}$ and $\mathcal{U}^i = \{\mathbb{U}_n \mid |\mathbb{U}_n| = 1, \forall \mathbb{U}_n \in \mathcal{U}\}$ are used to represent the index set of users for consistent requests and inconsistent requests, respectively. Similarly, \mathcal{C} can be classified into $\mathcal{C} = \{\mathbb{C}_1, \mathbb{C}_2, \dots, \mathbb{C}_n, \dots, \mathbb{C}_N\}$, where $\mathbb{C}_n = \{C_u \mid \forall u \in \mathbb{U}_n\}$ denotes the set of serving clusters whose users request W_n . Furthermore, $\mathcal{C}^c = \{\mathbb{C}_n \mid |\mathbb{C}_n| \geq 2, \forall \mathbb{C}_n \in \mathcal{C}\}$ and $\mathcal{C}^i = \{\mathbb{C}_n \mid |\mathbb{C}_n| = 1, \forall \mathbb{C}_n \in \mathcal{C}\}$ are used to represent the set of serving clusters for consistent requests and inconsistent requests, respectively. In addition, \mathbb{C}_n^c and \mathbb{C}_n^i are used to indicate the category of \mathbb{C}_n .

In C_u , user u receives S distinct coded packets of W_{d_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 needs other $L - S$ coded packets to recover the desirable file. In this case, the cloud server randomly selects an F-AP in 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 C_u instead of the cloud server transmits the coded packets to the corresponding F-AP in C_u . Let R^f denote the fronthaul rate, which is defined as the size of messages that the cloud server transmits to all clusters for satisfying all requests. For conventional schemes based on the MDS codes, $R^f = U \cdot (L - S) \cdot \frac{F}{L}$.

The objective of this paper is to find a feasible coded caching policy to minimize the fronthaul rate by utilizing the optimality of MDS codes in terms of redundancy-reliability tradeoff and the cache contents between any two clusters.

Algorithm 1 Redundant MDS Codes Based Coded Placement Scheme

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

III. PROPOSED JOINT REDUNDANT MDS CODES AND CLUSTER COOPERATION BASED CODED CACHING POLICY

In this section, we first propose the redundant MDS codes based coded placement scheme and show the construction of multicast messages based on the l redundant coded packets. Then, we propose the cluster cooperation based coded delivery scheme and show the construction of multicast messages based on the cache contents between any two clusters. Finally, the joint redundant MDS codes and cluster cooperation based coded caching policy is proposed to reduce the fronthaul rate.

A. Proposed Redundant MDS Codes Based Coded Placement Scheme

In order to construct multicast opportunities, motivated by [4], the cache content of each F-AP should be dedicated designed. The redundant MDS codes based coded placement scheme is presented in Algorithm 1. According to Algorithm 1, each caching file is encoded into $K + l$ coded packets. K coded packets are placed in K F-APs and the l redundant coded packets are stored in the cloud server without caching in any F-AP. It is easy to see that the k th coded packet of W_n is placed in F-AP k and we refer to this correspondence as the *symmetry of cache contents* in F-APs. The construction of multicast messages based on the l redundant coded packets is presented as follows.

Consider $\mathbb{U}_n \in \mathcal{U}^c$ and $\mathbb{C}_n \in \mathcal{C}^c$, users in \mathbb{U}_n request the same file W_n . If W_n is cached in F-APs, according to Algorithm 1, the l redundant coded packets are different from any coded packets cached in K F-APs, so they are useful to all clusters in \mathbb{C}_n . If a single cluster cannot provide its serving user with L coded packets of W_n , i.e., $S < L$, it is easy to see that at most l multicast messages useful to $|\mathbb{C}_n|$ clusters can be constructed.

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

Algorithm 2 Cluster Cooperation Based Coded Delivery Scheme

```
Initialize  $\mathcal{C}^a, \mathbf{d}, S, L, n_a$ .
2: Randomly pair the clusters in  $\mathcal{C}^a$ .
if  $L - n_a \leq S$  then
4:    $n_r = L - n_a$ .
else
6:    $n_r = S$ .
end if
8: for  $\{C_i, C_j\} \subset \mathcal{C}^a$  do
   Generate the index sets of F-APs in  $C_i$  and  $C_j$ ,
10:   $C_i : \{S(i-1) + 1, S(i-1) + 2, \dots, Si\}$ 
    $C_j : \{S(j-1) + 1, S(j-1) + 2, \dots, Sj\}$ .
12:  for  $s = 1, 2, \dots, n_r$  do
   Cloud server transmits  $W_{d_j, S(i-1)+s} \oplus W_{d_i, S(j-1)+s}$  to
    $C_i$  and  $C_j$ .
14:  end for
   while  $n_a + n_r < L$  do
16:    Cloud server separately transmits  $L - n_a - n_r$  of the
     $K + l - n_a - n_r$  coded packets of  $W_n$  to  $C_i$  and  $C_j$ .
   end while
18: end for
```

B. Proposed Cluster Cooperation Based Coded Delivery Scheme

According to Algorithm 1, each F-AP has coded packets of the N^l most popular files and the cache contents of 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 symmetry of cache contents between two clusters is presented as follows.

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

The cluster cooperation based coded delivery scheme is presented in Algorithm 2. Note that $\mathcal{C}^a \subset \mathcal{C}$ is any subset of \mathcal{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. $L - n_a \leq S$, the cloud server transmits $L - n_a$ of the S coded packets to satisfy the requests served by \mathcal{C}^a . When the S multicast messages are not enough to satisfy the requests, i.e., $L - n_a > S$, the cloud server transmits the S

multicast messages firstly, and each user served by C^a needs other $L - n_a - S$ coded packets to recover the desirable file. Remind that each caching file is encoded into $K + l$ coded packets with $K + l \geq L$, there are $K + l - n_a - S$ distinct coded packets stored in the cloud server. Then, the requests of user i and user j are satisfied by separately transmitting $L - n_a - S$ of the remaining $K + l - n_a - S$ coded packets to C_i and C_j .

By using the symmetry of cache contents between two clusters, our proposed cluster cooperation based coded delivery scheme can construct multicast messages for both consistent and inconsistent requests, which is crucial for reducing the fronthaul rate.

Remark 1: If there exists a cluster cannot be paired in C^a , the required coded packets of the cluster are directly transmitted by the cloud server.

Remark 2: For consistent requests, $C_n \in C^c$, the multicast messages constructed by the l redundant coded packets and the cache contents between two clusters are useful to $|C_n|$ clusters and 2 clusters, respectively. It is easy to see that when $|C_n| = 2$, the contributions of two kinds of multicast messages to the reduction in fronthaul rate are the same. Therefore, when the requests are highly consistent, i.e., $|C_n| \geq 3$, the multicast messages constructed by the l redundant coded packets can further reduce the fronthaul rate.

C. Proposed Joint Redundant MDS Codes and Cluster Cooperation 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 serving user. In order to fully utilize the multicast messages constructed 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 $U_n \in \mathcal{U}^c$ and $C_n \in C^c$, users in U_n request the same file W_n . If W_n is not cached in any F-AP, i.e., $n > N^t$, the cloud server transmits a multicast message W_n with F bits to the clusters in C_n . However, $n \leq N^t$ is the typical case, where the requested file is cached in F-APs. In this case, each user in U_n can receive S coded packets of W_n from the corresponding cluster. According to the property of MDS codes, $L - S$ other coded packets are needed to recover W_n . As mentioned previously, the l redundant coded packets are useful to all clusters in C_n . So when the number of required coded packets is less than or equal to the amount of the l redundant coded packets, i.e., $L - S \leq l$, $L - S$ of the l redundant coded packets are transmitted to all clusters in C_n . Then, the requests on W_n can be satisfied. When the l redundant coded packets are not enough to recover W_n , i.e., $L - S > l$, the cloud server transmits the l redundant coded packets to the clusters in C_n , and each user in U_n needs other $L - S - l$ distinct coded packets to recover W_n . Then, Algorithm 2 is used to transmit the remaining coded packets.

2) *Inconsistent Requests:* Consider U^i and C^i , users in U^i all request different files. The requests for uncached files are satisfied by directly transmitting the uncoded files to the

Algorithm 3 Joint Redundant MDS Codes and Cluster Cooperation Based Coded Caching Policy

Placement phase: Execute Algorithm 1.
 Delivery phase:
 3: Initialize $d, C, \mathcal{U}, S, L, l, N^t$.
 Generate $\mathcal{U}^c, \mathcal{U}^i, C^c, C^i$.
for $C_n \in C^c$ **do**
 6: **if** $n \leq N^t$ **then**
 if $L - S \leq l$ **then**
 Cloud server transmits $L - S$ out of l coded packets of W_n to the clusters in C_n .
 9: **else**
 Cloud server transmits l coded packets of W_n to the clusters in C_n ,
 $n_a = S + l, C^a = C_n$,
 12: Execute Algorithm 2.
 end if
 15: **end if**
 18: $C^a = \emptyset$.
for $C_n \in C^i$ **do**
 21: **if** $n \leq N^t$ **then**
 $C^a = C^a \cup C_n$
 24: **end if**
 27: **end for**
 Cloud server transmits W_n to the cluster in C_n .
end for
 $n_a = S, C^a = C^a$,
 27: Execute Algorithm 2.

clusters in C^i . The requests for cached files are satisfied similar to consistent requests. Specifically, each cluster in C^i can provide its serving user with S coded packets of the desirable file. Then, each user in U^i needs other $L - S$ coded packets to recover the desirable file. The remaining coded packets are transmitted by Algorithm 2.

The detailed description of our proposed joint redundant MDS codes and cluster cooperation based coded caching policy is presented in Algorithm 3. The fronthaul rate of our proposed policy is as follows:

$$R^f = \begin{cases} \left(\left\lceil \frac{\sum_{n=1}^{N^t} |C_n^c|}{2} \right\rceil \cdot \min(S, L - S) + \sum_{n=1}^{N^t} |C_n^c| \cdot \max(0, L - 2S) \right) \cdot \frac{F}{L} \\ + \left(\sum_{n=1}^{N^t} \mathbb{I}_{C^c}(C_n) \cdot (L - S) \cdot \frac{F}{L} + \left(\sum_{n=N^t+1}^N \mathbb{I}_{C^c}(C_n) \cdot F \right) \right) & l \geq L - S \\ \left(\left\lceil \frac{\sum_{n=1}^{N^t} |C_n^c|}{2} \right\rceil \cdot \min(S, L - S - l) + \sum_{n=1}^{N^t} |C_n^c| \cdot \max(0, L - 2S - l) \right) \cdot \frac{F}{L} \\ + \left(\left\lceil \frac{\sum_{n=1}^{N^t} |C_n^c|}{2} \right\rceil \cdot \min(S, L - S) + \sum_{n=1}^{N^t} |C_n^c| \cdot \max(0, L - 2S) \right) \cdot \frac{F}{L} \\ + \left(\sum_{n=1}^{N^t} \mathbb{I}_{C^c}(C_n) \cdot l \cdot \frac{F}{L} + \left(\sum_{n=N^t+1}^N \mathbb{I}_{C^c}(C_n) \cdot F \right) \right) & 0 \leq l \leq L - S \end{cases}$$

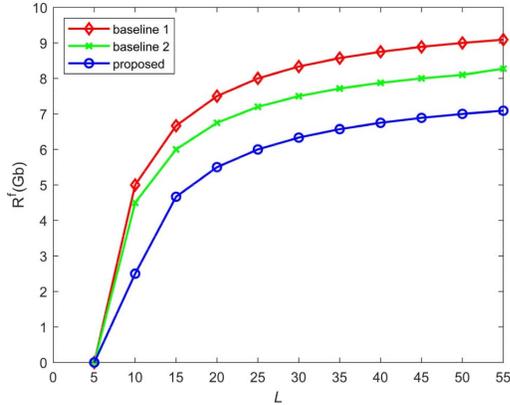


Fig. 2. Fronthaul rate versus the number of file fragment L with $K = 50$, $l = 5$.

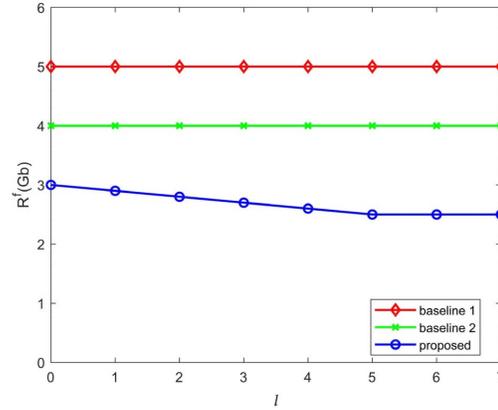


Fig. 4. Fronthaul rate versus the number of the redundant coded packets l with $K = 50$, $L = 10$.

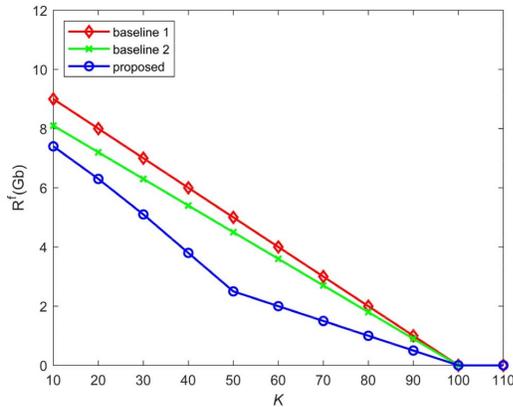


Fig. 3. Fronthaul rate versus the number of F-APs K with $L = 10$, $l = 5$.

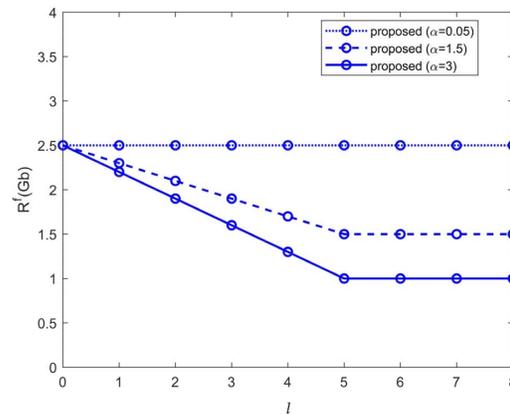


Fig. 5. Fronthaul rate versus with the number of the redundant coded packets l with $K = 50$, $L = 10$ and different α .

Where $\mathbb{I}_x(A)$ is the indicator function, i.e., $\mathbb{I}_x(A) = 1$ if $x \in A$; otherwise $\mathbb{I}_x(A) = 0$. By fully utilizing the property of MDS codes and the cache contents between any two clusters, our proposed policy can construct considerable multicast messages, which contributes a lot to the reduction in the fronthaul rate.

IV. SIMULATION RESULTS

In this section, the performance of our proposed joint redundant MDS codes and cluster cooperation based coded caching policy is evaluated via simulations. In order to investigate the performance of our proposed policy and exiting MDS-based coded caching policies. We adopt two different MDS-based coded caching policies as baselines.

Baseline 1 refers to the traditional MDS-based policy which uses MDS codes to perform cache placement whereas no multicast messages are constructed in delivery phase. Baseline 2 refers to the policy in [15], which only constructs multicast messages for consistent requests by using the redundancy of the MDS codes. In our simulations, the cache placement of all policies are performed by Algorithm 1 and users request files

according to \mathbf{p} . The parameters are set as follows: $F = 1$ Gb, $M = 20$, $N = 100$, $N^t = 80$, $U = C = 10$, $\alpha = 0.5$.

In Fig. 2, we depict how the number of file fragment, i.e., L affects the fronthaul rate of each policy with $K = 50$, $l = 5$. It is easy to see that $S = 5$, which indicates that there are 5 F-APs in each cluster. It can be observed that when $L = S$, the fronthaul rate of all policies are 0. The reasons are that the requested files are cached in F-APs and S clustered F-APs are able to provide their serving users with required L coded packets. It can also be observed that when L is larger, the gaps among three policies are large, which reveals that our proposed policy can construct more multicast messages than baseline 2, whereas baseline 1 cannot construct multicast messages. These observations verify that depending on the l redundant coded packets and the cache contents between any two clusters, our proposed policy can construct considerable multicast messages compared to two baselines.

In Fig. 3, we show the effect of the number of F-APs, i.e., K , on the fronthaul rate of each policy with $L = 10$, $l = 5$. As shown, our proposed policy can greatly reduce the fronthaul

rate compared to two baselines. It can be observed that with K increasing, the fronthaul rates of all policies reduce. The reason is that the local cluster can provide its serving user with more coded packets. Specifically, as K increases, we can observe that there is a turning point in the fronthaul rate of our proposed policy, which reveals that the multicast messages constructed by the l redundant coded packets are fully utilized. It can also be observed that when S is close to L , the fronthaul rates of all policies become 0. The reason is that most of the required coded packets are transmitted by the local cluster.

In Fig. 4, we show the effect of the number of the redundant coded packets, i.e., l , on the fronthaul rate of each policy with $L = 10$, $K = 50$. As shown, our proposed policy outperforms the two baselines in reducing the fronthaul rate. It can be observed that with l increasing, the fronthaul rates of baseline 1 and baseline 2 are both a constant. The reasons are that baseline 1 cannot construct multicast messages and K of the $K + l$ coded packets are enough for baseline 2 to satisfy all consistent requests regardless of l . It can also be observed that with l increasing, the fronthaul rate of our proposed policy decreases to a certain point and stays stable, which reveals that in addition to the multicast messages constructed by the cache contents between two clusters, our proposed policy can further construct multicast messages for consistent requests by the l redundant coded packets. This also reveals that when l reaches the certain point, the multicast messages constructed by the l redundant coded packets are fully utilized.

In Fig. 5, we show the effect of the redundant coded packets, i.e., l , on the fronthaul rate of our proposed policy with $L = 10$, $K = 50$ and different α . It can be observed that our proposed policy performs better when α is large, which verifies that when multiple users request the same file, the l redundant coded packets can further reduce the fronthaul rate. It can also be observed that when $\alpha = 0.05$, the fronthaul rate is a constant, which reveals that the number of users requesting the same file is smaller than three.

V. CONCLUSIONS

In this paper, we have proposed a joint redundant MDS codes and cluster cooperation based coded caching policy in F-RANs. Specifically, we have proposed a redundant MDS codes based coded placement scheme, which can provide redundant coded packets and symmetrical cache contents for the constructions of multicast messages. In addition, we have proposed a cluster cooperation based coded delivery scheme, which can fully utilize the cache contents between any two clusters to construct multicast messages for all kinds of requests. With the number of file fragment increasing, the multicast messages constructed by the redundant coded packets and the cache contents between any two clusters can be fully utilized. When the requests are highly consistent, the redundant coded packets can further reduce the fronthaul rate. Simulation results have shown that our proposed policy yields

a significant decrease in the fronthaul rate of approximately 30% compared to the MDS-based uncoded delivery policy.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grant 61971129, the Natural Science Foundation of Jiangsu Province under Grant BK20181264, the National Key R&D Program of China under Grant 2018YFB1801103, the Research Fund of the State Key Laboratory of Integrated Services Networks (Xidian University) under Grant ISN19-10, and the Research Fund of the Key Laboratory of Wireless Sensor Network & Communication (Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences) under Grant 2017002.

REFERENCES

- [1] 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, July 2016.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] —, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.
- [6] 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.
- [7] 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*, pp. 1–12, Jan. 2019.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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, 2019.
- [12] 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 (IC-CC Workshops)*, Aug. 2018, pp. 309–313.
- [13] 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.
- [14] 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.
- [15] 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*, pp. 1–13, 2019.