This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

Index Coding Algorithms: Cooperative Caching and Delivery for F-RANs

Salwa Mostafa, *Student Member, IEEE*, Chi Wan Sung, *Senior Member, IEEE*, Terence H. Chan, *Member, IEEE* and Guangping Xu, *Member, IEEE*

***Abstract*—In a Fog Radio Access Network (F-RAN), fog access points (F-APs) are equipped with caches that can store popular files during off-peak hours. Besides, they are densely deployed to have overlapping radio coverage so that requested files can be delivered cooperatively using beamforming. The bottleneck of the network is typically in the bandwidth-limited wireless fronthaul, which connects a cloud server to the F-APs. This work studies index coding design for cooperative caching and delivery in F-RAN to minimize fronthaul traffic and transmit energy. Index coding algorithms are designed considering the cached content at the F-APs and the possibility of beamforming in the access network under coded and uncoded caching schemes. An optimal polynomial-time index coding algorithm for uncoded and repetition caching and an efficient heuristic for Maximum Distance Separable (MDS) coded caching are designed, and their superior performance is verified by simulations. The study is further extended to consider the tradeoff between the traffic load of the fronthaul link and the transmit energy consumed in the access network. At the expense of more fronthaul traffic, beamforming opportunities can be increased, significantly reducing energy consumption. Algorithms to achieve the tradeoff are crafted, and simulation results show that uncoded caching well balances the tradeoff.**

***Index Terms*—cooperative caching, index coding, beamforming, fronthaul traffic, transmit energy.**

I. INTRODUCTION

Fog radio access network (F-RAN) is considered an emerging network architecture for 5G and beyond, as it boosts the network capacity and energy efficiency. In F-RANs, a cloud server is equipped with high storage, computing, and signal processing capabilities. The cloud is connected to densely deployed fog access points (F-APs) via wireless fronthaul links allowing joint processing and cooperation among multiple F-APs. The F-APs are equipped with caching and computing capabilities to bring the network functions close to mobile users. Proactive caching stores popular files on the local cache of F-APs during off-peak hours during the *cache placement phase* to reduce the fronthaul traffic load and energy consumption during peak traffic hours during the *content delivery phase*. Since the F-APs have limited cache space and the fronthaul link is bandwidth limited, efficient utilization of those scarce resources is important [1], [2].

Coded caching has shown a great advantage in reducing traffic load over shared broadcast links. It carefully stores the files in the caches during the cache placement phase and exploits *coded multicasting* opportunity during the content delivery phase using index coding [3]–[5]. A fundamental study is performed in [6] considering a basic system model in which a server connects to multiple cache-enabled users over an error-free shared wireless link. Subsequently, different caching schemes have been proposed and investigated [7]–[12] under the same model. Further extension is made in [13]–[16], which considers multiple servers known as *femtocaching* [17]. For femtocaching, since a user is often connected to multiple servers, it is possible to consider the cache of those servers together as a cumulative cache to serve users. This approach is called *cooperative caching*, which is proved to be an effective caching technique [18]–[20].

Due to the dense deployment of F-APs and the high signal processing capability of the cloud, beamforming has attracted the attention of academia and industry as a delivery scheme to minimize the transmit energy [21]. However, integrating beamforming in cache-enabled networks is a challenging problem since not only the channel condition has to be taken into consideration but the cached content of F-APs as well. In general, the more F-APs involve in beamforming, the lower the transmit energy. Nevertheless, suppose the requested file is not available at all those F-APs. In that case, it needs to be fetched from the cloud server, which significantly increases the signaling overhead and payload data sharing over the fronthaul. As a result, there exists a tradeoff between fronthaul traffic and transmit energy under beamforming [22]–[24]. Joint design of content placement, base station clustering and beamforming is considered in [25], [26], and multicasting over fronthaul to support beamforming is considered in [27]–[31]. Nevertheless, the studies mentioned above use naive multicasting without network coding and mainly focus on beamforming vector design.

In this paper, we consider the case where a user is connected to multiple F-APs so that cooperative caching and beamforming can be used. Index coding is used in the fronthaul to support coded multicasting. We have performed a preliminary

This work was supported in part by the City University of Hong Kong (Project 7005448), and the National Natural Science Foundation of China (61971309). Salwa Mostafa is with the University of Oulu, Oulu, Finland, and the Faculty of Electronic Engineering, Menofia University, Menofia, Arab Republic of Egypt. Chi Wan Sung is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China (e-mail: smostafa3-c@my.cityu.edu.hk, albert.sung@cityu.edu.hk). Terence. H. Chan is with the Institute for Telecommunications Research, University of South Australia, Adelaide, Australia (e-mail: Terence.chan@unisa.edu.au). Guangping Xu is with the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin 300384, China (e-mail: xugp@email.tjut.edu.cn). This work was done when Salwa Mostafa was studying for her Ph.D. at the City University of Hong Kong. (*Corresponding author: Chi Wan Sung.*)

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

study of the interplay between caching, index coding, and beamforming in [32], showing that beamforming can be used to minimize the fronthaul traffic via careful cache placement and delivery schemes design. We extend our previous work by addressing the following three important problems.

- Beamforming requires the availability of the same content on more than one F-AP to serve the users cooperatively. For this reason, existing caching schemes typically cache the same content in some F-APs to facilitate the use of beamforming. In this study, we show that caching different contents enable more effective use of index coding in the fronthaul link and beamforming in the access network. To realize the idea, we propose uncoded caching and MDS coded caching based on file subpacketization. Our simulation results show that they outperform random caching [33] and probabilistic caching [34], which are widely used together with beamforming in the literature.
- In the literature, index coding algorithms are designed to minimize the number of broadcast packets by exploiting the side information at the cache of the receivers [35]. When beamforming is used in the access network of the F-RAN, the design of index coding needs to take into account the access channel conditions as well, which differs from the classical model of index coding. In this work, we design an optimal polynomial-time index coding algorithm for uncoded caching and a well-performed heuristic for MDS coded caching.
- An overall system design typically involves more than one design objective. In this work, a framework is established for analyzing the tradeoff between traffic load of the fronthaul and transmit energy in the access network. Due to the discrete nature of the index coding problem, convex optimization techniques, which are widely used for obtaining performance tradeoffs in wireless networks, cannot be applied. To solve this problem, the *dummy node insertion* methodology from graph theory is applied to obtain the Pareto optimal tradeoff curve for uncoded caching in polynomial time, avoiding exhaustive search.

The rest of this paper is organized as follows. Section II states our system model and our cache placement and content delivery schemes. In Section III, we state our objectives and formulate the beamforming subproblem. Index coding algorithms for minimizing the fronthaul traffic with minimum transmit energy is discussed in Section IV. In Section V, the tradeoff between fronthaul traffic and transmit energy is investigated. Section VI provides our simulation model and results. We conclude the paper in Section VII.

II. SYSTEM MODEL

A. Network Model

Consider an F-RAN, which consists of a cloud server (CS), M cache-enabled single-antenna F-APs operating at the same frequency band and having encoding/decoding functionality, and N users having single-antenna devices. Denote the index sets of the F-APs and users by $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$ and $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$, respectively. The cloud server has a library $\mathcal{W} = \{W^{(1)}, W^{(2)}, \ldots, W^{(F)}\}$ of F popular files, each of B bits. Each user requests a file[1] from $\mathcal{W}$ with a probability according to a file popularity distribution, $\rho_1, \rho_2, \ldots, \rho_F$, where $\sum_{f=1}^{F} \rho_f = 1$. The cloud server is connected to the F-APs via a wireless *broadcast* fronthaul link. Typically, this link is bandwidth limited. To avoid overwhelming this link, a cache space of C bits is allocated to each F-AP, where $C < FB$. Furthermore, we assume that it supports noise-free transmissions of constant bit rate since we target delay-sensitive applications such as video-on-demand services. Note that traffic load reduction in the fronthaul can be translated into energy saving.

The communication channels between F-APs and users are called the access network. We assume that the frequency band for the access network is disjoint from that for the fronthaul link, so there is no interference between them. We consider a time-slotted system adopting the block fading channel model, where the channel gains stay constant at a specific time slot. The cloud server schedules the transmission from the F-APs to the users over the time slot to avoid interference. Each user is served with a transmission rate of R Mbps. The channel gain between F-AP m and user n is denoted by $h_{nm} \in \mathbb{C}$ for $n \in \mathcal{N}$ and $m \in \mathcal{M}$. The channel gains are normalized such that the noise power at each receiver is equal to 1. Let the $N \times M$ matrix $\mathbf{H} \triangleq [h_{nm}]$ be the normalized channel gain matrix. The channel is assumed power limited. Each F-AP is subject to a peak power constraint of P. Given a fixed modulation and coding scheme, a target signal-to-noise ratio (SNR), γ must be met.

B. Cache Placement Schemes

We consider three caching schemes. Under each scheme, a file, $W^{(f)}$, to be cached, is partitioned into $k \leq M$ subfiles, $W_1^{(f)}, W_2^{(f)}, \ldots, W_k^{(f)}$, of equal size.

- *Uncoded Caching* ($k = M$): Each F-AP m stores the subfile $W_m^{(f)}$, for $m \in \mathcal{M}$. All the subfiles are also stored in the cloud.
- *Repetition Caching* ($k = \frac{M}{2}$, M is even): Each F-AP m stores the subfile $W^{(f)}_{(m-1 \bmod k)+1}$, for $m \in \mathcal{M}$. All the subfiles are also stored in the cloud.
- *MDS Coded Caching* ($k \leq M$): The k subfiles are encoded using an $(M+k, k)$ MDS code to obtain $M+k$ blocks, $X_1^{(f)}, X_2^{(f)}, \ldots, X_{M+k}^{(f)}$. The first M blocks are for caching, each of which is placed in one F-AP. They are also stored in the cloud. In addition, the remaining k of them, denoted by $\mathcal{Z} \triangleq \{X_{M+1}^{(f)}, X_{M+2}^{(f)}, \ldots, X_{M+k}^{(f)}\}$, are stored only in the cloud, but not in the F-APs.

For each scheme, the data cached in an F-AP for a file is called a *block*, which can be an uncoded subfile or an MDS-coded block. The block size for uncoded caching, repetition caching, and MDS coded caching are B/M, $2B/M$, and B/k bits, respectively. A user has to receive k independent blocks to obtain an intended file. In general, the cache space may not be large enough to cache all files in the F-APs. In that case, files are cached according to *Most Popular First* (MPF),

---

[1] If a user requests more than one file, his requests can be scheduled in different time blocks.

i.e., one after another in descending order of their popularity. (If there are two or more files of the same popularity, the tie is broken arbitrarily.) If the remaining space is not enough to cache a whole block, part of each block is cached, which fills up the available cache space. For example, assume that $C_r < B/M$ bits remain in the cache of each F-AP, which is not sufficient to cache a whole block of the next file. Then, a partial block of size C_r bits will be cached in each F-AP.

C. Content Delivery Schemes

During the delivery phase, each user requests a file. The objective is to serve all users efficiently. For the wireless fronthaul, the performance measure is the *traffic load* measured by the number of bits transmitted. For the access network, the performance measure is the *total transmit energy*, which is the sum of the energy used to transmit the packets to serve all users. If a user cannot obtain enough information from his associated F-APs, the cloud server has to broadcast a certain number of fixed-length packets through the fronthaul link. Index coding is incorporated over the fronthaul link to minimize the fronthaul traffic in which the coded packets are assumed to be instantly decodable.

Definition 1. A coded packet is said to be *instantly decodable* at a receiver if the receiver can decode a new block from the coded packet and the packets already received.

Instantly decodable packets have great advantages in reducing delay and decoding complexity. Therefore, we assume instant decodability at both the F-APs and the users. More precisely, an F-AP can either forward a packet or compute over the received packet and its cached block and send the results to one or more of its associated users. Afterward, the F-AP discards the received packet before processing the next packet from the fronthaul link. No extra buffering is required at F-APs. Furthermore, cache replacement is not considered in this work, assuming the popularity distribution is time-invariant over the peak hours of a day. Under our caching schemes, each F-AP caches a unique block from each cached file. Instantly decodable coded packets are allowed if only pairwise index coding is performed. Thus, we assume packets can be transmitted over the fronthaul link, *without coding* or *with pairwise index coding* in the form of $L_i \oplus L_j$, which represents bitwise XOR between two blocks L_i and L_j. We allow both *intra-file* and *inter-file* index coding, where two blocks are originated from the same file and different files, respectively.

The requested files need to be delivered to the users with a target SNR; thus, we consider direct transmission from an associated F-AP to a user if the F-AP can deliver the cached block with the target SNR. Otherwise, beamforming is performed from the associated F-APs to deliver a block with the target SNR. Beamforming requires the same data block to be available on the F-APs cooperate in the beamforming transmission. Pairwise index coding allows at most two F-APs to be able to decode any coded packet broadcast over the fronthaul link. Thus, at most two F-APs can form beamforming to deliver any block after decoding to the users. Let p_m be the transmit power of F-AP m. If $|h_{nm}|^2 \geq \frac{\gamma}{P}$, the link is said to be *strong*, as F-AP m alone can deliver packets successfully to user n with power $p_m = \frac{\gamma}{|h_{nm}|^2}$. The user is said to be strongly associated with the F-AP. If $\frac{\gamma}{4P} \leq |h_{nm}|^2 < \frac{\gamma}{P}$, the link is said to be *weak* because F-AP m alone is unable to deliver packets to user n, but it can transmit by beamforming with another F-AP m' that has the same packets. To see this, let the two F-APs align their signals in phase, so the received SNR, Γ_m, is given by

$$\Gamma_m = (|h_{nm}|\sqrt{p_m} + |h_{nm'}|\sqrt{p_{m'}})^2 \geq (\sqrt{p_m} + \sqrt{p_{m'}})^2 \frac{\gamma}{4P}.$$

The target SNR can be met, for example, by $p_m = p_{m'} = P$. Thus, the user is said to be weakly associated with F-AP m and also with F-AP m'. As a result, if $|h_{nm}|^2 < \frac{\gamma}{4P}$, there is no way for F-AP m to serve user n, and the corresponding link between F-AP m and user n is said to be *missing*. The association matrix between users and F-APs is represented by an $N \times M$ ternary matrix $\boldsymbol{A}$, where its component, a_{nm}, equals 0, 1, or 2 if the link between user n and F-AP m is missing, weak, or strong, respectively.

In the access network, as mentioned above, a packet can be delivered to a user by direct transmission over a single strong link or by beamforming via any two (strong/weak) links. If a packet can be delivered to a user in more than one way, the one with the least energy is most desirable. Recall that to obtain an intended file, a user has to receive k independent blocks. Due to instant decodability, each of them is obtained either directly from an uncoded packet or from an instantly decodable packet. Therefore, the transmit energy for a particular user is the sum of the transmit energy for k packets. The energy for transmitting a packet is equal to the transmit power of an F-AP, or the total transmit power of two F-APs if beamforming is used, times the packet duration.

III. PROBLEM FORMULATION

The primary objective of this paper is to minimize the traffic load over the fronthaul link, since it is bandwidth limited, and the reduction of fronthaul traffic load can shorten delivery delay, which improves user experience. In general, multiple solutions minimize the fronthaul load. Among those solutions, we want to find the one that minimizes the total transmit energy in the access network for energy-saving purpose. To minimize the fronthaul traffic load, the association matrix and the cached content need to be taken into consideration in the design of index-coded transmissions over the fronthaul link, which will be discussed in the next section. To minimize the total transmit energy over the access channel, beamforming is used to deliver packets whenever possible due to its additional power gain. The corresponding power minimization problem can be formulated as follows, where p_1 and p_2 are the transmit power values of the two F-APs under concern, h_1 and h_2 are their normalized channel gains to a particular user:

$$\min \; p_1 + p_2 \tag{1}$$

$$\begin{aligned} \text{s.t} \;\; & C1: |h_1|\sqrt{p_1} + |h_2|\sqrt{p_2} - \sqrt{\gamma} \geq 0, \\ & C2: P - p_1 \geq 0, \; C3: P - p_2 \geq 0, \qquad (2) \\ & C4: p_1 \geq 0, \; C5: p_2 \geq 0, \end{aligned}$$

where $C1$ guarantees the target SNR is met, while $C2$ to $C5$ are power constraints.

Lemma 1. *The power minimization problem in (1) is convex.*

Proof. Since the objective function and the last four constraints are all linear, we only need to check whether the region constrained by C_1 is convex or not. Define $f(p_1,p_2) \triangleq \sqrt{\gamma} - |h_1|\sqrt{p_1} - |h_2|\sqrt{p_2}$. Its Hessian is given by

$$\nabla^2 f = \begin{bmatrix} \frac{1}{4}|h_1|p_1^{-3/2} & 0 \\ 0 & \frac{1}{4}|h_2|p_2^{-3/2} \end{bmatrix}.$$

Since both diagonal entries are positive, the Hessian is positive definite, and so f is convex. Hence, the problem is convex. $\square$

Theorem 2. *Given* $\mathbf{h} \triangleq (h_1,h_2)$ *and* $|h_1| \geq |h_2|$, *the optimum solution to the power minimization problem in (1), if feasible, is given by* $p_T(h_1,h_2) \triangleq p_1^* + p_2^*$, *where*

$$p_1^* = \min\Big\{\frac{|h_1|^2\gamma}{\|\mathbf{h}\|^4}, P\Big\}, \;\; p_2^* = \begin{cases} \frac{(\sqrt{\gamma}-|h_1|\sqrt{P})^2}{|h_2|^2} & \text{if } p_1^* = P, \\ \frac{|h_2|^2\gamma}{\|\mathbf{h}\|^4} & \text{otherwise.} \end{cases}$$

Proof. Since the sum power is to be minimized, it is obvious that $C1$ must hold with equality and can be treated as an equality constraint. Define the Lagrangian

$$\mathcal{L}(p_1,p_2,\lambda,\mu_1,\mu_2,\eta_1,\eta_2)$$
$$\triangleq \sum_{i=1}^{2}\Big(-p_i + \mu_i(p_i - P) - \eta_i p_i\Big) + \lambda\Big(\sum_{i=1}^{2}|h_i|\sqrt{p_i} - \sqrt{\gamma}\Big),$$

where λ is the Lagrange multiplier for $C1$, and $\mu_1, \mu_2, \eta_1, \eta_2$ are the non-negative Lagrange multipliers for the other four constraints.

By Lemma 1, the problem is convex. Given any feasible instance, it is easy to see that Slater's condition always holds, except when there is only one single feasible point, $p_1 = p_2 = P$, which must also be optimal. For all other cases, the following Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient to identify the optimal solution:

$$|h_1|\sqrt{p_1} + |h_2|\sqrt{p_2} = \sqrt{\gamma}, \tag{3}$$

$$1 + \mu_i - \eta_i + \frac{\lambda|h_i|}{2\sqrt{p_i}} = 0 \;\text{ for } i = 1,2, \tag{4}$$

$$\sum_{i=1}^{2}\mu_i(p_i - P) + \eta_i p_i = 0. \tag{5}$$

Suppose $C2$ to $C5$ are all inactive, i.e., $\mu_i = \eta_i = 0$ for $i = 1,2$. Solving (3) and (4), we obtain the following roots for the power values:

$$(\tilde{p}_1, \tilde{p}_2) \triangleq \frac{\gamma}{\|\mathbf{h}\|^4}(|h_1|^2, |h_2|^2),$$

which always satisfies $C4$ and $C5$. Since $|h_1| \geq |h_2|$, we have $\tilde{p}_1 \geq \tilde{p}_2$. If $C2$ is not violated, neither is $C3$, and thus $(\tilde{p}_1, \tilde{p}_2)$ is feasible and is thus optimal. Otherwise, consider the case where $C2$ is active, i.e., $p_1' = P$. Substituting it into (3), we obtain $p_2' = \frac{(\sqrt{\gamma}-|h_1|\sqrt{P})^2}{|h_2|^2}$. The statement is obtained by combining the above cases. $\square$

Based on the above result, we define a function $p_{\text{BF}}(h_1,h_2)$ to represent the power required for two F-APs to transmit successfully to a user with normalized link gains, h_1 and h_2, respectively. For notation simplicity, we also include the case where one link is strong while the other link is missing, so the function represents the power required for one of the F-APs to transmit directly to the user via its strong link. The function is defined as follows:

$$p_{\text{BF}}(h_1,h_2) \triangleq \begin{cases} p_T(h_{\max}, h_{\min}) & \text{if } h_{\min} \geq \frac{\gamma}{4P}, \\ \frac{\gamma}{|h_{\max}|^2} & \text{if } h_{\min} < \frac{\gamma}{4P}, h_{\max} \geq \frac{\gamma}{P}, \\ \infty & \text{otherwise,} \end{cases}$$

where $h_{\max} \triangleq \max(h_1,h_2)$ and $h_{\min} \triangleq \min(h_1,h_2)$.

IV. INDEX CODING ALGORITHMS

This section considers the fronthaul traffic minimization problem for different caching schemes under both intra-file and inter-file index coding. The intra-file coding for uncoded caching, repetition caching and MDS coded caching are discussed in subsections IV-A, IV-B, and IV-C, respectively. In subsection IV-D, we extend our proposed solution to inter-file coding. Note that intra-file coding is preferable in latency-critical applications, where the requested files have a delay tolerance constraint. The reason is that the intra-file coded packets broadcast over the fronthaul link are useful to all users requesting that file and allow them to reconstruct the requested file faster. We assume each user is connected with at least one strong link or two weak links, for otherwise, it cannot receive anything from the network, and the problem is clearly infeasible. Since under intra-file coding, different files are treated independently, it suffices to describe the transmissions for one single file, $W^{(f)}$. For simplicity, we omit the superscript and simply write it as W. We denote the set of users requesting the file by $\mathcal{N}$, where $|\mathcal{N}| = N$. Our aim is to design algorithms for minimizing the fronthaul traffic with minimum transmit energy over the access channel for the three caching schemes.

A. Uncoded Caching

Recall that $\mathbf{A}$ is the $N \times M$ association matrix. Let $A[i,j]$ be the $N \times 2$ submatrix of $\mathbf{A}$ obtained by preserving only columns i and j of $\mathbf{A}$. The following concept is important:

Definition 2. A pair of distinct subfiles, i and j, denoted by (i,j), is said to be a *potential coded group* if the sum of each row of $A[i,j]$ is greater than or equal to two. Furthermore, (i,j) and (k,l) are said to be *disjoint* if $i \notin \{k,l\}$ and $j \notin \{k,l\}$.

Note that (i,j) and (j,i) refer to the same coded packet, $W_i \oplus W_j$. It is easy to see that if $W_i \oplus W_j$ is broadcast over the fronthaul, all users are able to decode both W_i and W_j via the F-APs by receiving (i) either W_i or W_j, and (ii) $W_i \oplus W_j$ [2].

---

[2] Note that we send $W_i \oplus W_j$ because the coded packet $W_i \oplus W_j$ is broadcast over the fronthaul link and can be delivered to each user via the two F-APs with the best channel gains to him. In this case, the transmit energy is equal to or lower than sending the other subfile W_i or W_j from the two F-APs i and j.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

To ensure instant decodability, a user should receive either W_i or W_j *before* receiving $W_i \oplus W_j$. Note that after broadcasting $W_i \oplus W_j$ over the fronthaul, both F-APs i and j have both subfiles so that either subfile can be delivered. The two F-APs can use beamforming to transmit either subfile cooperatively to user n, or an F-AP can transmit alone if the other link is missing. The total energy to deliver one of the subfiles to all users is given by

$$E_1(i,j) = \frac{B}{kR} \sum_{n \in \mathcal{N}} p_{\text{BF}}(h_{n,i}, h_{n,j}). \tag{6}$$

Next, $W_i \oplus W_j$ needs to be delivered to all users. If user n has at least two links, either weak or strong, user n can obtain the packet $W_i \oplus W_j$ through beamforming between any two F-APs to which he is associated with, and can minimize the energy usage by choosing the two F-APs, denoted by $\xi_1(n)$ and $\xi_2(n)$, that have the highest normalized channel gains to himself. If user n has only one strong link, he receives the packet $W_i \oplus W_j$ from the F-AP with the strong link directly. Hence, the minimum energy required to deliver $W_i \oplus W_j$ to all users is given by

$$E_2 = \frac{B}{kR} \sum_{n \in \mathcal{N}} p_{\text{BF}}(h_{n,\xi_1(n)}, h_{n,\xi_2(n)}). \tag{7}$$

Note that E_2 does not depend on the indices i and j. Therefore, the total transmit energy associated with any potential coded group can be computed as follows:

$$E(i,j) = E_1(i,j) + E_2. \tag{8}$$

Since E_2 is fixed, it is enough to distinguish the transmit energy associated with the potential coded group (i,j) by its *partial transmit energy* defined by $E_1(i,j)$.

Remark 1. The coded packet $W_i \oplus W_j$ is broadcast over the fronthaul link; all F-APs can cooperate to transmit the coded packet to a user. Nevertheless, that involves more signaling overhead and CSI information. Throughout the paper, we assume that at most two F-APs cooperate to transmit it. However, our framework can be easily generalized to involve more than two F-APs to deliver any coded packet or an uncoded packet broadcast over the fronthaul link[3].

The above information can be represented by a weighted graph $G(\mathcal{V}, \mathcal{E})$, where the vertices in $\mathcal{V}$ represent the packets, the edges in $\mathcal{E}$ represent the potential coded groups. A non-negative weight w_{ij} is assigned to each edge (i,j) by $w_{ij} \triangleq \max_{(i,j)\in\mathcal{E}} E_1(i,j) - E_1(i,j)$. Note that the first term is only a constant ensuring all weights are non-negative. An edge with a larger weight is preferable, since the corresponding potential coded group requires a smaller partial transmit energy. This transforms energy minimization into weight maximization.

In graph theory, a *matching* in G is a subset of edges without common vertices, and a *maximum cardinality matching* is a matching that has the maximum number of edges. We are interested in the following problem:

Problem: PACKETPAIRING
Instance: A weighted graph $G(\mathcal{V}, \mathcal{E})$ with $w_{ij} \geq 0$ for all $(i,j) \in \mathcal{E}$.
Objective: Find a maximum cardinality matching which has maximum sum weight.

It is related to the well-known maximum weighted matching problem, which needs to find a matching with maximum sum weight, as shown below:

Proposition 3. PACKETPAIRING *can be reduced to maximum weighted matching and be solved in polynomial time.*

Proof. Let Q be a value greater than $\sum_{(i,j)\in\mathcal{E}} w_{ij}$. The reduction can be done by adding Q to each weight w_{ij} to obtain w'_{ij} for all $(i,j) \in \mathcal{E}$. We claim that a maximum weighted matching in G with weights w'_{ij}s must have maximum cardinality. Suppose $\mathcal{S}$ is a maximum weighted matching and there exists another matching $\mathcal{S}'$ where $|\mathcal{S}'| > |\mathcal{S}|$. The sum weight of the matching $\mathcal{S}$ is equal to

$$\sum_{(i,j)\in\mathcal{S}} w'_{ij} = |\mathcal{S}|Q + \sum_{(i,j)\in\mathcal{S}} w_{ij} < (|\mathcal{S}|+1)Q \leq \sum_{(i,j)\in\mathcal{S}'} w'_{ij},$$

which leads to a contradiction. Since maximum weighted matching can be solved in polynomial time, PACKETPAIRING can also be solved in polynomial time[4]. ☐

Now we describe our proposed algorithm, which minimizes the fronthaul traffic load with minimum transmit energy. It consists of two phases. In the first phase, we identify those F-APs which have strong links to all users and denote them by the set $\mathcal{M}'$. Since the subfiles cached in $\mathcal{M}'$ are possible to be delivered directly without incurring fronthaul traffic, we focus on the delivery of the remaining subfiles. To minimize the number of packets required to be transmitted over the fronthaul, a weighted graph $G(\mathcal{V}, \mathcal{E}_{\mathcal{V}})$ is constructed, where $\mathcal{V} = \mathcal{M} \setminus \mathcal{M}'$ is the vertex set, and $\mathcal{E}_{\mathcal{V}}$ is defined as the edge set which consists of $(i,j) \in \mathcal{V}^2$ if (i,j) is a potential coded group. The weight of edge (i,j) is defined by $\max_{(i,j)\in\mathcal{E}_{\mathcal{V}}} E_1(i,j) - E_1(i,j)$. Given the weighted graph G, we solve PACKETPAIRING to obtain a matching $\mathcal{C}$, which contains the set of matched edges in $\mathcal{E}_{\mathcal{V}}$ (i.e., the chosen combination of potential coded groups to be broadcast), and let $\mathcal{U}$ be the set of unmatched vertices in $\mathcal{V}$. We denote such a procedure by $(\mathcal{C}, \mathcal{U}) := \text{MATCHING}(G)$. If either $\mathcal{M}'$ or $\mathcal{U}$ is empty, the algorithm skips the second phase and returns $\mathcal{C} \cup \mathcal{U}$, which represents the set of coded and uncoded packets to be broadcast.

In the second phase, if $|\mathcal{U}| < |\mathcal{M}'|$, a set of dummy nodes $\mathcal{D}$ of cardinality $|\mathcal{M}'| - |\mathcal{U}|$ and a set of dummy edges $\mathcal{E}'(\mathcal{M}') \triangleq \mathcal{D} \times \mathcal{M}'$ are defined. Otherwise, $\mathcal{D}$ is defined as the empty set. The graph $G'(\mathcal{M} \cup \mathcal{D}, \mathcal{E}_{\mathcal{M}} \cup \mathcal{E}')$ is then constructed, where $\mathcal{E}_{\mathcal{M}}$ is the edge set which consists of $(i,j) \in \mathcal{M}^2$ if (i,j) is a potential coded group. The weight assigned to each edge (i,j) in $\mathcal{E}_{\mathcal{M}}$ is $\max_{(i,j)\in\mathcal{E}_{\mathcal{M}}} E_1(i,j) - E_1(i,j)$ while that assigned

---

[3] Since each F-AP under our three caching schemes stores an uncoded subfile/MDS coded block from each cached file, the instantaneous decodability condition restricts the index coding to be done between two uncoded subfiles/MDS coded blocks only even if more F-APs are incorporated in forming the beamforming. Therefore, involving more F-APs in beamforming does not change our proposed coding scheme. However, the transmitting energy can be further minimized.

[4] The MATLAB code implemented the algorithm for finding the maximum cardinality matching which has maximum sum weight can be found in [36].

Algorithm 1: Fronthaul Transmissions for Uncoded Caching

```
Input  : A set of F-APs $\mathcal{M}$, a set of users $\mathcal{N}$, an
         association matrix $\mathbf{A}$, a partial transmit
         energy function $E_1$.
Output: A set of packets $\mathcal{P}$.
1: Let $\mathcal{M}' := \{m \in \mathcal{M} \mid a_{nm} = 2 \; \forall n \in \mathcal{N}\}$;
2: Construct $G(\mathcal{M} \setminus \mathcal{M}', \mathcal{E}_{\mathcal{M}\setminus\mathcal{M}'})$, where the weight of
   edge $(i,j)$ is $w_{ij}$;
3: Find $(\mathcal{C},\mathcal{U}) :=$ MATCHING$(G)$;
4: if $\mathcal{U}$ and $\mathcal{M}'$ are both non-empty then
5:     $(G',\mathcal{E}') :=$ ADDDUMMY$(G, (|\mathcal{M}'| - |\mathcal{U}|)^+, \mathcal{M}')$;
6:     Find $(\mathcal{C},\mathcal{U}) :=$ MATCHING$(G')$;
7:     $\mathcal{C} := \mathcal{C} \setminus \mathcal{E}'$;
8: end if
9: return $\mathcal{C} \cup \mathcal{U}$;
```

to each edge in $\mathcal{E}'$ is a very large number, denoted by ∞. Note that G' is a supergraph of G. To simplify our notation, given a weighted graph $G(\mathcal{M}, \mathcal{E}_{\mathcal{M}})$ and a vertex subset $\mathcal{M}' \subseteq \mathcal{M}$, we denote the following operation by ADDDUMMY$(G, d, \mathcal{M}')$:

1) the addition of d dummy nodes, and
2) the addition of an infinite-weight edge from each dummy node to each vertex in the subset $\mathcal{M}' \subseteq \mathcal{M}$.

It returns the new graph and the set of dummy edges, i.e.,

$$(G', \mathcal{E}') := \textsc{AddDummy}(G, (|\mathcal{M}'| - |\mathcal{U}|)^+, \mathcal{M}'),$$

where $(x)^+ \triangleq \max(x, 0)$.

Afterwards, we solve PACKETPAIRING to find $(\mathcal{C},\mathcal{U}) :=$ MATCHING(G'). The algorithm then returns $\mathcal{U} \cup (\mathcal{C} \setminus \mathcal{E}')$, where $\mathcal{C} \setminus \mathcal{E}'$ contains all the matched edges except the dummy ones.

The pseudo-code of the whole algorithm is shown in Algorithm 1, in which the if-block corresponds to the second phase. The number of packet transmissions is given by the cardinality of $\mathcal{P}$, where $\mathcal{P}$ is the output of the algorithm. Note that the transmission order of the packets in $\mathcal{P}$ is not important because the packets obtained by Algorithm 1 in $\mathcal{P}$ are disjoint potential coded groups and *independent* uncoded packets. Note that an uncoded packet is said to be independent in $\mathcal{P}$ if it does not participate in any coded packets in $\mathcal{P}$. The order of transmissions of those packets thus has no effect on instant decodability at F-APs or energy consumption in the access network. For a coded packet in $\mathcal{P}$ broadcast over the fronthaul link, the required transmit energy in the access network is given by (8). For an uncoded packet in $\mathcal{P}$, it is delivered to all users with total energy E_2 via beamforming between the two F-APs that have the highest normalized channel gain to each user. For an uncoded packet, W_m that is delivered directly from the cache of F-AP m without involving the fronthaul link, the total energy consumption in the access network is given by

$$E_0(m) = \frac{B}{kR} \sum_{n \in \mathcal{N}} \frac{\gamma}{|h_{nm}|^2}. \tag{9}$$

Example 1. Consider seven F-APs which caches a file $W \triangleq \{W_1, W_2, \ldots, W_7\}$ of size 100 Mbits using uncoded caching. Each F-AP has a peak power constraint P of 2 W. The target SNR, γ, equals 16, and the transmission rate, R, over the access channel is 100 Mbps. Two users who request the file are associated with the F-APs, with the normalized channel gain matrix $\mathbf{H}$. Thus, the association matrix can be computed as in $\mathbf{A}$

$$\mathbf{H} = \begin{bmatrix} 2 & 3 & 1 & 4 & 8 & 9 & 10 \\ 3 & 4 & 5 & 1 & 9 & 8 & 11 \end{bmatrix} \mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 0 & 2 & 2 & 2 \end{bmatrix}.$$

Algorithm 1 finds the set of coded packets as follows. In the first phase, we have $\mathcal{M} = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathcal{M}' = \{5, 6, 7\}$. A weighted graph $G(\mathcal{V}, \mathcal{E}_{\mathcal{V}})$ is constructed, where $\mathcal{V} = \{1, 2, 3, 4\}$ and $\mathcal{E}_{\mathcal{V}} = \{(1, 2)\}$. The weight assigned to the edge $(1, 2)$ is equal to 0. Since there is only one single edge in the graph, the maximum cardinality matching is obviously $\mathcal{C} = \{(1, 2)\}$. The result of the first phase of Algorithm 1 is shown in Fig. 1a. Since $0 < 2 = |\mathcal{U}| < |\mathcal{M}'| = 3$, the algorithm enters the second phase. The graph $G'(\mathcal{M} \cup \mathcal{D}, \mathcal{E}_{\mathcal{M}} \cup \mathcal{E}')$, with $\mathcal{D} = \{d\}$ and $\mathcal{E}' = \{(d, 5), (d, 6), (d, 7)\}$, is constructed, as shown in Fig. 1b. Given G', the maximum cardinality matching with maximum sum weight $\mathcal{C} = \{(1, 2), (3, 6), (4, 7), (d, 5)\}$ is found (Fig. 1b). There are no unmatched vertices. The algorithm returns $\{(1, 2), (3, 6), (4, 7)\}$ and halts. It means that $W_1 \oplus W_2$, $W_3 \oplus W_6$, and $W_4 \oplus W_7$ should be broadcast over the fronthaul link, and the six subfiles should be delivered to the users as described previously. Note that W_5 is not sent over the fronthaul link and is delivered to the users via the strong links from the cache of F-AP 5 with transmit energy $E_0(5) = 540$ mJ. The energy consumption of each coded packet can be obtained by subtracting the corresponding edge weight from $E_2 + \max_{(i,j)\in\mathcal{E}} E_1(i, j) = 235 + 784 = 1019$ mJ. As a result, the total energy consumption is given by $E_{total} = E(1, 2) + E(3, 6) + E(4, 7) + E_0(5) = 1019 + (1019 - 379) + (1019 - 430) + 540$ mJ $= 2.8$ J.

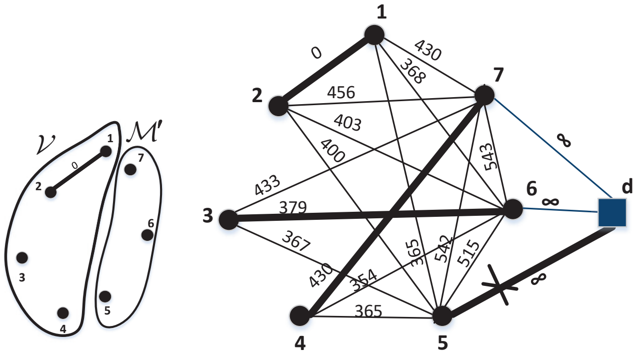
Next, we analyze the time complexity of Algorithm 1. Identifying the subfiles in Step 1 requires $O(NM)$. Finding all potential coded groups and computing the corresponding weight to construct the graph G in Step 2 requires $O(NM^2)$. The PACKETPAIRING problem in Step 3 can be solved in $O(NM^{2.5})$ by Micali-Vazirani Algorithm [37] and finding the unmatched vertices $\mathcal{U}$ takes $O(M)$. The overall time complexity of Algorithm 1 is, therefore, $O(NM^{2.5})$.

Lastly, we prove that Algorithm 1 is optimal. The proof is divided into several lemmas.

Lemma 4. *There always exists an optimal solution which does not contain a cycle of three or more coded packets, i.e.,* $W_{k_1} \oplus W_{k_2}, W_{k_2} \oplus W_{k_3}, \ldots, W_{k_n} \oplus W_{k_1}$, *where* $k_1, k_2, \ldots, k_n$ *are distinct and* $n \geq 3$.

Proof. If there is an optimal solution which contains such a cycle, the n coded packets can be replaced by the corresponding n uncoded subfiles, i.e, $W_{k_1}, W_{k_2}, \ldots, W_{k_n}$. Regardless of the original transmit sequence of the n coded packets, we assume that in the new solution, the uncoded packets are transmitted one by one (in any order) before all other packets. This ensures that the instant decodability of subsequent packets will not

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976



(a) (b)

Figure 1: The graphs and outputs of Algorithm 1 in (a) the first phase and (b) the second phase. The weight of each edge is measured in millijoules (mJ). In both phases, $\max_{(i,j)\in\mathcal{E}_{\mathcal{V}}} E_1(i,j) = \max_{(i,j)\in\mathcal{E}_{\mathcal{M}}} E_1(i,j) = E_1(1,2) = 784$ mJ and $E_2 = 235$ mJ. The maximum cardinality matching with maximum sum weight is indicated by thick solid lines.

be violated. We claim that the new solution is feasible and minimizes the fronthaul traffic with minimum transmit energy. First, the new solution is feasible because each user, having at least two weak links or a strong link, can receive the n uncoded subfiles as well as in the original solution. Next, as the new solution has exactly the same number of packets transmitted as the original solution, it also minimizes the fronthaul traffic. Finally, we need to show that the new solution minimizes total transmit energy without increasing fronthaul traffic. In the original solution, a user must obtain W_{k_i}, for $i = 1, 2, \ldots, n$, from exactly one received packet due to instant decodability. The packet may or may not be one of the n coded packets in the cycle. In any case, the energy for delivering such a packet would not be strictly less than that for delivering the subfile W_{k_i} directly in the new solution, as the latter can be achieved with minimum energy by beamforming between the two F-APs with highest normalized channel gains to the user. Hence, the new solution, containing no cycle, is optimal. □

Lemma 5. *There exists an optimal solution which uses only disjoint potential coded groups and independent uncoded packets.*

Proof. By Lemma 4, there is an optimal solution, say $\mathcal{P}^*$, that has no cycle of coded packets. Assume that $\mathcal{P}^*$ contains a maximal chain of $\tau > 1$ coded packets, i.e., $\{(l_1, l_2), (l_2, l_3), \ldots, (l_\tau, l_{\tau+1})\} \subseteq \mathcal{P}^*$. Note that $(l_1, l_j) \notin \mathcal{P}^*$ for any $j \neq 2$, for otherwise the chain is not maximal.

First, we show that there is an optimal solution in which the uncoded packets are independent. Suppose l_i and (l_i, l_j) both belong to $\mathcal{P}^*$. Replacing (l_i, l_j) by l_j in $\mathcal{P}^*$ will not increase the total energy because W_{l_i} is broadcast over the fronthaul and therefore can be delivered to any user with minimum energy via cooperative beamforming by the two F-APs that have the strongest channels. Therefore, we can assume that the uncoded packets in $\mathcal{P}^*$ are independent, and in particular, $l_1, l_2 \notin \mathcal{P}^*$.

Next, we show that there is an optimal solution which does not contain a chain of coded packets. Since $l_1 \notin \mathcal{P}^*$ and $(l_1, l_j) \notin \mathcal{P}^*$ for any $j \neq 2$, after $W_{l_1} \oplus W_{l_2}$ is broadcast over the fronthaul link, every user must be able to obtain W_{l_1}. A user can obtain W_{l_1} either from F-AP l_1 alone via a strong link, or from F-AP l_1 and F-AP l_2 via beamforming. In either case, the user can obtain W_{l_2} with minimum energy if $W_{l_1} \oplus W_{l_2}$ is delivered via beamforming by the two F-APs that have the strongest channel to the user. This argument applies to all users. Therefore, we can obtain another optimal solution from $\mathcal{P}^*$ by replacing (l_2, l_3) by l_3. The new optimal solution has a maximal chain of length $\tau - 1$. Repeat the argument to reduce the length of the maximal chain until $\tau - 1 = 1$. The chain then no longer exists, and we are done. The whole argument can be applied again if $\mathcal{P}^*$ contains more than one maximal chain. As a result, we can assume that $\mathcal{P}^*$ contains no chain of coded packets.

Last, we prove by contradiction that the coded packets in $\mathcal{P}^*$ are all potential coded groups. Assume $(l_i, l_j) \in \mathcal{P}^*$ is not a potential coded group. Following its definition, there must exist a user who cannot obtain W_{l_i} or W_{l_j}, since $l_i, l_j \notin \mathcal{P}^*$ and they do not get involved in any other coded packets in $\mathcal{P}^*$. That means, $\mathcal{P}^*$ is infeasible, which leads to a contradiction. □

Theorem 6. *Algorithm 1 minimizes the fronthaul traffic with minimum transmit energy.*

Proof. First, we show that phase 1 of Algorithm 1 produces a solution that minimizes the fronthaul traffic. Since we do not care about the transmit energy in this phase, it is clear that those F-APs with strong links to all users can first deliver their cached subfiles without using the fronthaul link. Step 1 in Algorithm 1 puts the indices of those F-APs in $\mathcal{M}'$. By Lemma 5, there is an optimal solution which can be represented by the union of unmatched vertices and a matching of the graph constructed in Step 2. Since the fronthaul traffic can be reduced by forming more coded packets, an optimal solution is represented by the union of a maximum cardinality matching and the remaining unmatched vertices, which can be determined by Step 3. If $\mathcal{M}'$ is empty, the solution produced in phase 1 is optimal, since the procedure MATCHING produces the maximum cardinality matching with minimum total energy. If $\mathcal{M}'$ is non-empty but $\mathcal{U}$ is empty, considering $\mathcal{M}'$ in the matching process will increase $|\mathcal{P}|$, resulting in a larger amount of fronthaul traffic.

If both $\mathcal{M}'$ and $\mathcal{U}$ are non-empty, the algorithm enters phase 2. Note that in the graph G', there is an edge (u, v), for all $u \in \mathcal{M}'$ and $v \in \mathcal{V}$. We divide the proof of this part into two cases.

First, consider the case where $|\mathcal{U}| \geq |\mathcal{M}'|$. Compared with G used in Step 3, there are $|\mathcal{M}'|$ vertices added to G', so there are at most $|\mathcal{M}'|$ more edges obtained from Step 6. It is easy to see that the edges obtained from Step 6 are a superset of that obtained from Step 3, with $|\mathcal{M}'|$ new edges being a maximum weight matching in the subgraph with edge set $\mathcal{U} \times \mathcal{M}'$. Since the new coded packets are obtained by XOR between the subfiles indexed by $\mathcal{U}$ and the subfiles indexed by $\mathcal{M}'$, the amount of fronthaul traffic is the same as in phase 1,

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

which is minimized. The solution obtained is optimal since the total energy is minimized with the given cardinality of matching.

Next, consider the case where $|\mathcal{U}| < |\mathcal{M}'|$. The idea is the same as the previous case except that the edges in $(\mathcal{V} \setminus \mathcal{U}) \times \mathcal{M}'$ should not be selected by Step 6. To ensure that it would not occur, $|\mathcal{M}'| - |\mathcal{U}|$ dummy vertices are added and connected to all vertices in $|\mathcal{M}'|$. Since Step 6 needs to output a maximum cardinality matching, the edges incident on the dummy vertices must be selected. The same argument then goes as in the previous case, and the proof is completed. ☐

B. Repetition Caching

Since our primary objective is to minimize the fronthaul traffic, we first ignore the issue of transmit energy. To find the minimum fronthaul traffic under repetition caching, we reduce it to the problem for uncoded caching by defining a new $N \times M/2$ association matrix $\mathbf{A}'$, whose entries are defined by $a'_{n,m} = \min(a_{n,m} + a_{n,m+M/2}, 2)$, for all $n \in \mathcal{N}$ and $m \in \mathcal{M}$. It means that the link in the new association matrix between user n and F-AP m is strong if either of the corresponding links in the original matrix is strong or both are weak. If exactly one of the corresponding links in the original matrix is weak, the link is weak. It is missing if both of the corresponding links are missing. It is clear that the new association matrix is equivalent to the original association matrix in the sense that the same number of fronthaul traffic is required for successful file delivery. Thus, the first phase of Algorithm 1 can be applied to find the minimum fronthaul traffic.

Now we consider how the transmitted energy can be minimized. For any broadcast coded packet $W_i \oplus W_j$, the F-APs in $\mathcal{M}_{i,j} = \{i, i + M/2, j, j + M/2\}$ can decode it and have both W_i and W_j. The transmit energy of either W_i or W_j can be minimized by choosing the two F-APs from the subset $\mathcal{M}_{i,j}$ that have the highest normalized channel gain to user n, indexed by $\zeta_1(i, j, n)$ and $\zeta_2(i, j, n)$, to transmit either subfile cooperatively via beamforming. Thus, the partial transmit energy associated with a potential coded group (i, j) for repetition caching can be defined as follows:

$$E_1(i, j) = \frac{B}{kR} \sum_{n \in \mathcal{N}} p_{\text{BF}}(h_{n,\zeta_1(i,j,n)}, h_{n,\zeta_2(i,j,n)}).$$

For any broadcast (uncoded or coded) packet, it can be delivered with energy E_2 defined in (7) via beamforming between the two F-APs that have the highest normalized channel gain to each user. For an uncoded packet W_m delivered directly from the cache of the F-APs, it is stored repeatedly in F-APs m and $m+M/2$, and can be delivered via beamforming with transmit energy

$$E_0(m) = \frac{B}{kR} \sum_{n \in \mathcal{N}} p_{\text{BF}}(h_{n,m}, h_{n,m+M/2}).$$

Since E_2 is fixed for any coded packet (i, j), it is enough to distinguish the transmit energy associated with the potential coded group (i, j) by its *partial transmit energy* defined by $E_1(i, j)$. After the reduction, Algorithm 1 can be applied with input $\boldsymbol{A}'$ and partial transmit energy function $E_1(i, j)$ to minimize the fronthaul traffic with minimum transmit energy for repetition caching.

Remark 2. For repetition caching, since the coded packet $W_i \oplus W_j$ is broadcast over the fronthaul link, the F-APs in $\mathcal{M}_{i,j}$ can cooperate to transmit either W_i or W_j to a user. Nevertheless, that involves more signaling overhead and CSI information. Since our main focus in this paper is on index coding design, we assume that at most two F-APs cooperate to transmit it. However, our framework can be easily generalized to involve the four F-APs to deliver either W_i or W_j.

C. MDS Coded Caching

For MDS coded caching, each user requires any k MDS coded blocks to reconstruct his requested file. Unlike the case of uncoded caching, the minimum number of packets that need to be sent over the fronthaul is difficult to characterize. Let $\mathcal{P}^*$ be an optimal solution. Given an association matrix, we can find a lower and an upper bound on $|\mathcal{P}^*|$. For $n \in \mathcal{N}$, let s_n and w_n be the number of strong links and the number of weak links associated with user n. After downloading the blocks from the caches of s_n F-APs via his strong links, user n requires $r_n = \max(k - s_n, 0)$ more blocks to recover the file. We then have the following result:

Theorem 7. *For any requested file cached using an $(M + k, k)$ MDS coded caching, the minimum number of broadcast packets over the fronthaul link is bounded by*

$$\max_{n \in \mathcal{N}} \left(r_n - \left\lfloor \frac{\min(r_n, w_n)}{2} \right\rfloor \right) \leq |\mathcal{P}^*| \leq \max_{n \in \mathcal{N}} r_n. \tag{10}$$

Proof. The upper bound is obvious since all users can be satisfied by sending any $\max_{n \in \mathcal{N}} r_n$ packets from $\mathcal{Z}$. The lower bound is a single-user bound, obtained by considering each user independently. First, consider the case where $r_n \leq w_n$. Since there are sufficient weak links, user n can be satisfied by broadcasting $\lfloor r_n/2 \rfloor$ XOR packets, and one additional packet from $\mathcal{Z}$ if r_n is an odd number. In other words, $\lceil r_n/2 \rceil$ packets are needed. Next, consider the case where $r_n > w_n$. There are not enough weak links, so at most $\lfloor w_n/2 \rfloor$ XOR packets can be broadcast to deliver $2\lfloor w_n/2 \rfloor$ blocks. The remaining blocks can be chosen from $\mathcal{Z}$ and delivered by broadcasting $r_n - 2\lfloor w_n/2 \rfloor$ packets. In total, $r_n - \lfloor w_n/2 \rfloor$ packets are needed to satisfy the user. Combining the two cases, we obtain that $r_n - \lfloor \min(r_n, w_n)/2 \rfloor$ packets are needed for user n. The tightest bound is obtained by taking the maximum among the N users. ☐

We describe our proposed heuristic algorithm, which minimizes the fronthaul traffic with minimum transmit energy. Without loss of generality, assume that all users in $\mathcal{N}$ cannot reconstruct the requested file via their strong links, for otherwise, those users who are able to do so could be ignored in our consideration.

First, the number of additional distinct coded packets, r_n, required for each user is computed from the given association matrix. Next, we identify the F-APs that have strong links to

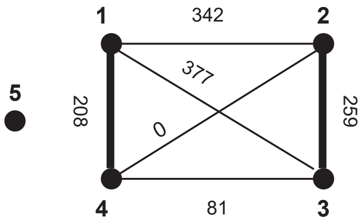


Figure 2: The graphs and outputs of Algorithm 2. The weight of each edge is measured in millijoules (mJ). The $\max_{(i,j)\in\mathcal{E}_\mathcal{V}} E_1(i,j) = E_1(2,3) = 970$ mJ and $E_2 = 410$ mJ. The maximum cardinality matching with maximum sum weight is indicated by thick solid lines.

all users and exclude them from $\mathcal{M}$ to obtain $\mathcal{V}$. Afterwords, we construct the weighted graph $G(\mathcal{V},\mathcal{E}_\mathcal{V})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}_\mathcal{V}$ consisting of $(i,j) \in \mathcal{V}^2$ if (i,j) is a potential coded group. The weight of each edge (i,j) is defined by $w_{ij} \triangleq \max_{(i,j)\in\mathcal{E}_\mathcal{V}} E_1(i,j) - E_1(i,j)$, where $E_1(i,j)$ is computed by (11). We then solve the PACKETPAIRING problem to obtain $(\mathcal{C},\mathcal{U}) := \textsc{Matching}(G)$.

The coded packets (i,j)'s in $\mathcal{C}$ are ordered in descending order of their associated weights. The algorithm then simulates the broadcast of those packets following the above order.

The value of r_n for each $n \in \mathcal{N}$ is updated after each simulated transmission. If all coded packets in $\mathcal{C}$ have been used while some users are still not satisfied, the blocks from the set $\mathcal{Z}$ are then chosen until all users obtain k unique blocks. The algorithm then checks if the cardinality of $\mathcal{P}$ is greater than or equal to the upper bound stated in (10). If so, the algorithm replaces $\mathcal{P}$ by $\max_{n\in\mathcal{N}} r_n$ elements from the set $\mathcal{Z}$. The pseudo-code is stated in Algorithm 2.

The coded packets in $\mathcal{C}$ are delivered with transmit energy $E(i,j)$ computed by (8) in a similar way as uncoded caching. The packets transmitted from $\mathcal{Z}$ are delivered with transmit energy E_2 computed by (7) via beamforming between the two F-APs that have the highest normalized channel gain to each user. The blocks delivered directly from the F-APs caches without involving fronthaul traffic are delivered with energy $E_0(m)$ defined by (9). The following example demonstrates the effectiveness of our proposed algorithm in minimizing the fronthaul traffic with minimum transmit energy.

Example 2. Consider five F-APs which caches a file $W^{(f)} \triangleq \{W_1^{(f)}, W_2^{(f)}, W_3^{(f)}\}$ of size 100 Mbits by an $(8,3)$ MDS code. Each F-AP has a peak power constraint of 2 W. The target SNR, γ, equals 16, and the transmission rate over the access channel is 100 Mbps. A user who requests the file is associated with the F-APs with normalized channel gains $\tilde{\mathbf{h}} = (5, 3.5, 4, 2, 8)$. The first four links are weak, while the last is strong. The user requires $r_1 = 2$ more packets to reconstruct the requested file. The possible potential coded packets and their associated weights are shown in Fig. 2. The output of Step 4 is $\mathcal{C} = \{(2,3),(1,4)\}$ and $\mathcal{U} = \emptyset$. The user can be satisfied by broadcasting either coded packet. The packet $(2,3)$ should be chosen, since it has lower transmit energy. The user can construct the requested file with total transmit energy $E_{total} = E(2,3) + E_0(5) = 1.79$ Joule.

Algorithm 2: Fronthaul Transmission for MDS Coded Caching

```
Input  : A set of F-APs $\mathcal{M}$, a set of users $\mathcal{N}$, an
         association matrix $\mathbf{A}$, a set of MDS coded
         packets $\mathcal{Z}$, a partial transmit energy function
         $E_1$.
Output: A set of packets $\mathcal{P}$.
 1: Let $r_n$ be the extra number of packets required by
    user $n$ for $n \in \mathcal{N}$ and $r_{\max} := \max_{n\in\mathcal{N}} r_n$;
 2: Let $\mathcal{V} := \mathcal{M} \setminus \{m \in \mathcal{M} \mid a_{nm} = 2\ \forall n \in \mathcal{N}\}$;
 3: Construct the weighted graph $G(\mathcal{V},\mathcal{E}_\mathcal{V})$;
 4: Find $(\mathcal{C},\mathcal{U}) := \textsc{Matching}(G)$;
 5: Sort the elements of $\mathcal{C}$ in descending order of their
    weights, and enqueue them to a queue $\mathcal{Q}$ in that order;
 6: while $r_n > 0$ for some $n$ do
 7:    if $\mathcal{Q}$ is non-empty then
 8:       Dequeue $c$ from $\mathcal{Q}$ and put it in $\mathcal{P}$;
 9:       Update $r_n$ for all $n$, assuming $c$ is broadcast;
10:    else
11:       Move any $\max_n r_n$ elements from $\mathcal{Z}$ to $\mathcal{P}$;
12:       Let $r_n := 0$ for all $n$;
13:    end if
14: end while
15: if $|\mathcal{P}| \geq r_{\max}$ then
16:    Replace $\mathcal{P}$ by $r_{\max}$ elements from $\mathcal{Z}$;
17: end if
18: return $\mathcal{P}$;
```

Note that if the coded packet $(1,4)$ is broadcast, the user retrieves the file with total transmit energy $E_{total} = E(1,4) + E_0(5) = 1.84$ Joules. Thus, ordering the coded packets in $\mathcal{C}$ in Step 5 reduces the transmit energy. Besides, if packets from the set $\mathcal{Z}$ are transmitted without considering the potential coded packets, two packets are required to satisfy the user. Hence, sending potential coded packets reduces the fronthaul traffic.

Corollary 8. *Algorithm 2 is a 2-approximation algorithm for minimizing fronthaul traffic under MDS coded caching.*

Proof. By Theorem 7, we have

$$|\mathcal{P}^*| \geq \max_{n\in\mathcal{N}} \left(r_n - \left\lfloor \frac{\min(r_n, w_n)}{2} \right\rfloor\right) \geq \max_{n\in\mathcal{N}} \frac{r_n}{2} \geq \frac{|\mathcal{P}|}{2},$$

where the last inequality follows from Steps 15 to 17 of Algorithm 2. Hence, the solution obtained is within a constant 2 of the optimal value. ☐

Now we analyze the time complexity of Algorithm 2. Steps 1 and 2 each requires $O(NM)$. Finding all potential coded groups in Step 3 requires $O(NM^2)$. The PACKETPAIRING problem in Step 4 can be solved in $O(M^{2.5})$. Sorting in Step 5 takes $O(M \log M)$. The while loop in Step 6 executes $O(M)$ times, and Step 9 requires $O(N)$. The overall time complexity of Algorithm 2 is, therefore, $O(NM^{2.5})$.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

D. Inter-file Coding

This subsection shows how our proposed solution can be generalized to include inter-file coding for uncoded caching. Since we only allow pairwise coding, inter-file coding is performed between any pair of files. Let the two groups of users requesting the two files be denoted by $\mathcal{N}$ and $\mathcal{N}'$, respectively. The association matrix $\boldsymbol{B}$ is constructed by including all users in the two groups. Each column of the association matrix represents the index of the subfiles cached from all files on each of the F-AP. The coded packets allowed between the two files are the potential coded groups defined below:

Definition 3. For any pair of files, a pair of distinct subfiles with indices, i and j, denoted by (i, j), is said to be a *potential coded group* if the sum of each row of $B[i, j]$ is greater than or equal to two. Furthermore, (i, j) and (k, l) are said to be *disjoint* if $i \notin \{k, l\}$ and $j \notin \{k, l\}$.

For an inter-file coded packet, the energy consumption of a potential coded group (i, j) is given by

$$E_3(i, j) = \frac{B}{kR} \sum_{n \in \mathcal{N} \cup \mathcal{N}'} p_{\text{BF}}(h_{n,i}, h_{n,j}), \tag{11}$$

which is the energy of delivering the two uncoded subfiles (belonging to two different files) indexed by i and j after decoding via beamforming or direct transmission to the users from the F-APs i and j. For each user group, find the set of subfiles that has strong links with all users in that group. The union of the two sets of subfiles is denoted by $\mathcal{R}'$. Let the set of all other subfiles, which require fronthaul transmissions, be denoted by $\mathcal{R}$. To minimize the fronthaul traffic load, we consider both intra-file and inter-file potential coded groups. We construct a weighted graph $G(\mathcal{R}, \mathcal{E}_{\mathcal{R}})$ in which the set of vertices represents the subfiles in $\mathcal{R}$ and the set of edges $\mathcal{E}_{\mathcal{R}}$ represents all potential coded group. The energy associated with an intra-file and inter-file potential coded group is defined by $E(i, j) = E_1 + E_2$ and $E(i, j) = E_3$, respectively. A non-negative weight w_{ij} is assigned to each edge (i, j) by $w_{ij} \triangleq \max_{(i,j) \in \mathcal{E}} E(i, j) - E(i, j)$. Afterward, steps 3-9 from Algorithm 1 are applied to find the transmission packets with the set $\mathcal{M}'$ replacing by $\mathcal{R}'$.

Example 3. Six F-APs cache two files $(W^{(1)}, W^{(2)}) \triangleq \{W_1^{(1)}, W_2^{(1)}, \ldots, W_6^{(1)}, W_1^{(2)}, W_2^{(2)}, \ldots, W_6^{(2)}\}$ using uncoded caching. The system parameters are $B = 100$ Mbits, $P = 2$ W, $\gamma = 16$, and $R = 100$ Mbps. Users 1 and 3 request $W^{(1)}$ while users 2 and 4 request $W^{(2)}$. The normalized channel gain matrix and the association matrix are given, respectively, by

$$\mathbf{H} = \begin{bmatrix} 2 & 3 & 5 & 8 & 8 & 10 \\ 4 & 2 & 9 & 6 & 11 & 1 \\ 3 & 4 & 5 & 11 & .5 & 12 \\ 5 & 4 & 8 & 2 & 9 & 10 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 1 & 2 & 0 & 2 \\ 1 & 1 & 2 & 1 & 2 & 2 \end{bmatrix}.$$

We have $\mathcal{R}' = \{W_4^{(1)}, W_6^{(1)}, W_3^{(2)}, W_5^{(2)}\}$ and $\mathcal{R} = \{W_1^{(1)}, W_2^{(1)}, W_3^{(1)}, W_5^{(1)}, W_1^{(2)}, W_2^{(2)}, W_4^{(2)}, W_6^{(2)}\}$. The graph $G(\mathcal{R}, \mathcal{E}_{\mathcal{R}})$ is shown in Fig. 3b, where the edge weights are omitted. By Steps 3-9 of Algorithm 1, four coded packets are needed as shown in the figure. In contrast, intra-file coding requires six coded packets as shown in Fig. 3a, which shows the advantage of incorporating inter-file coding.

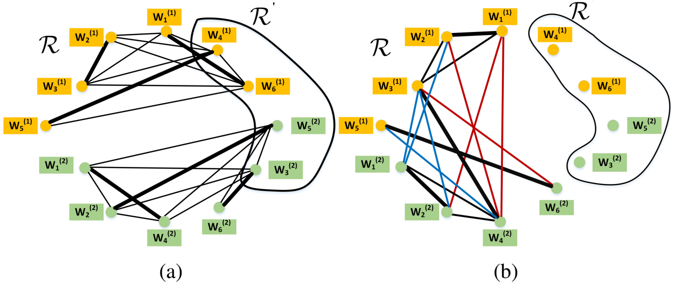


Figure 3: (a) Intra-file coding and (b) Inter-file coding. The maximum cardinality matching with maximum sum weight is indicated by thick solid lines.

When there are more than two requested files, we consider all possible potential coded groups from each pair of files and then apply Algorithm 1. For repetition and MDS coded caching, the same idea can be applied. The details are omitted due to space limitation.

V. TRADEOFF CURVES BETWEEN FRONTHAUL TRAFFIC AND TRANSMIT ENERGY

In this section, we consider the graph of transmit energy against fronthaul traffic and find the curve that characterizes their tradeoff for each caching scheme. We focus on one single file, and the fronthaul traffic load is normalized by the file size, B. For an uncached file, the whole file needs to be sent over the fronthaul link. Thus, the normalized fronthaul traffic load can only be equal to one, and the tradeoff curve degenerates to one single point. For this reason, it suffices to consider only cached files. For each cached file, the tradeoff curve is defined by the $k + 1$ feasible discrete points whose x-coordinates are $|\mathcal{P}|/k, (|\mathcal{P}| + 1)/k, \ldots, 1$, where $\mathcal{P}$ is the set of packets obtained from Algorithms 1 and 2 for uncoded and MDS coded caching, respectively. Our objective is to find the combination of packets that minimizes the total transmit energy at each discrete point with fronthaul traffic load t, where $t = |\mathcal{P}|, |\mathcal{P}| + 1, \ldots, k$. The case where $t = |\mathcal{P}|$ has been considered, so we focus on the other cases in this section.

A. Uncoded Caching

Let G be the graph whose vertex set is $\mathcal{M}$ and the edge set contains all the potential coded groups. As before, let $\mathcal{M}' \subset \mathcal{M}$ be the subset of F-APs which have strong links to all users. An F-AP in $\mathcal{M}'$ can form a potential coded group with any other F-AP in $\mathcal{M}$, so its degree in G is $M - 1$. Note that if $\mathcal{M}'$ is empty, then all subfiles, either uncoded or pairwise index coded, must be sent over the fronthaul, and thus we must have $|\mathcal{P}| > |\mathcal{M}|/2$.

Our algorithm is divided into the following two cases, depending on whether the value of t is greater than $\lfloor |\mathcal{M}|/2 \rfloor$, which is the maximum number of disjoint coded packets that can be sent over the fronthaul.

Algorithm 3: Fronthaul Transmission of t Packets for Uncoded Caching

```
Input  : A set of F-APs $\mathcal{M}$, a set of users $\mathcal{N}$, an
         association matrix $\mathbf{A}$, a partial transmit
         energy function $E_1$, number of broadcast
         packets $t$.
Output: A set of $t$ packets, $\mathcal{P}$.
 1: if $t \le \lfloor |\mathcal{M}|/2 \rfloor$ then
 2:     Let $\mathcal{M}' := \{m \in \mathcal{M} \mid a_{nm} = 2\ \forall n \in \mathcal{N}\}$;
 3:     $(G', \mathcal{E}') :=$ AddDummy$(G, |\mathcal{M}| - 2t, \mathcal{M}')$;
 4:     Find $(\mathcal{C},\mathcal{U}) :=$ Matching$(G')$;
 5: else
 6:     $(G', \mathcal{E}') :=$ AddDummy$(G, 2t - |\mathcal{M}|, \mathcal{M})$;
 7:     Find $(\mathcal{C},\mathcal{U}) :=$ Matching$(G')$;
 8:     $\mathcal{U} := \{m \in \mathcal{M} : m$ is incident to some $e \in \mathcal{C} \cap \mathcal{E}'\}$;
 9: end if
10: return $(\mathcal{C} \setminus \mathcal{E}') \cup \mathcal{U}$;
```

1) $t \le \lfloor |\mathcal{M}|/2 \rfloor$: Note that $\mathcal{M}'$ must be non-empty. Since *pairwise* index coding is used, at most $2t$ subfiles can be involved in the packets broadcast over the fronhaul. Therefore, we add $|\mathcal{M}|-2t$ dummy nodes, each of which is connected to all vertices in $\mathcal{M}'$. The resultant graph is denoted by G'. The PACKETPAIRING problem is solved to obtain the matching $(\mathcal{C},\mathcal{U}) :=$ MATCHING(G'). The result is returned after excluding those coded packets involving dummy nodes.
2) $t > \lfloor |\mathcal{M}|/2 \rfloor$: A total of $2t - |\mathcal{M}|$ dummy nodes are added, each of which is connected to all vertices in $\mathcal{M}$. The resultant graph is denoted by G'. Then, the PACKETPAIRING problem is solved to obtain the matching $(\mathcal{C},\mathcal{U}) :=$ MATCHING(G'), where $\mathcal{U}$ must be empty. The subfiles matched with the dummy nodes are sent uncoded. In other words, they are added to $\mathcal{U}$ and their dummy edges are excluded from $\mathcal{C}$. The result is then returned.

The pseudo-code is stated in Algorithm 3, which has a time complexity of $O(NM^{2.5})$.

Example 2 (revisited). According to Algorithm 1, $|\mathcal{P}| = 3$, and the corresponding total transmit energy is 2.79 J. This gives the first point of the tradeoff curve, i.e., $t = 3$. The last point $t = 7$ is trivial since all subfiles should be broadcast without coding. The total transmit energy is 1.64 J. The remaining points, $t = 4, 5, 6$, can be obtained by Algorithm 3. Since $t \ge \lfloor |\mathcal{M}|/2 \rfloor = 3$, the second condition is true for all the three points. We omit the details for $t = 4$ and 6, and use only $t = 5$ for illustration. Three dummy nodes, d_1, d_2 and d_3, are added, each of which is connected to all vertices in $\mathcal{M}$. Step 7 of Algorithm 3 outputs $\mathcal{C} = \{(2,7),(5,6),(1,d_1),(3,d_2),(4,d_3)\}$ and $\mathcal{U} = \emptyset$. Step 10 returns $\{(2,7),(5,6),1,3,4\}$, which means $W_2 \oplus W_7$, $W_5 \oplus W_6$, W_1, W_3, and W_4 should be broadcast. The corresponding total transmit energy is 1.77 J. The tradeoff curve is shown in Fig. 4.

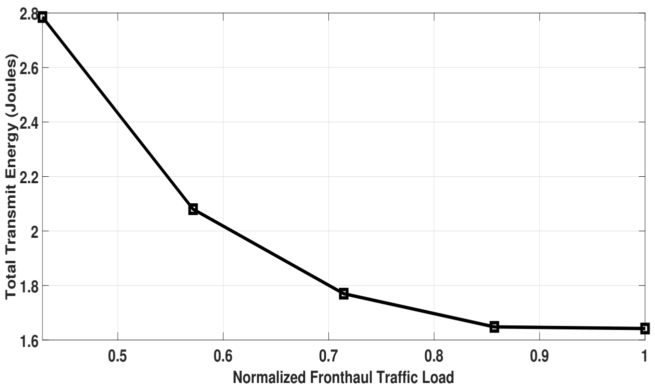


Figure 4: The tradeoff between fronthaul traffic and total transmit energy in Example 2 ($M = 7$, $N = 2$ and $F = 1$).

Theorem 9. *Given a feasible value of fronthaul traffic load t, Algorithm 3 outputs a set of t packets which minimizes the total transmit energy.*

Proof. It can be proved that there exists an optimal solution that uses only disjoint potential coded groups and independent uncoded packets. The idea is the same as that in Lemmas 4 and 5, so we omit the details.

First, consider the case where $t \le \lfloor |\mathcal{M}|/2 \rfloor$. Let $t^* = |\mathcal{P}|$ be the minimum traffic load under uncoded caching as determined by Algorithm 1, and thus $t^* < t$. Since *pairwise* index coding is used, at most $2t^*$ subfiles are involved in $\mathcal{P}$. Therefore, the number of F-APs that have strong links to all users is bounded below by $|\mathcal{M}'| \ge M - 2t^* > M - 2t$. Since a vertex in $\mathcal{M}'$ is connected to every other nodes in G', in Step 4, $\mathcal{U}$ must be empty, and $\mathcal{C}$ must contain t edges in the original graph G and $|\mathcal{M}| - 2t$ dummy edges. The energy is guaranteed to be minimized by MATCHING.

Next, consider the case where $t > \lfloor |\mathcal{M}|/2 \rfloor$. Since the traffic load is at most t, we have $|\mathcal{C}| + |\mathcal{U}| \le t$. It is obvious that for any pair of i and j, sending W_i and W_j individually consumes less energy than sending $W_i \oplus W_j$. Therefore, to achieve optimality, we must have $|\mathcal{C}| + |\mathcal{U}| = t$. Furthermore, we claim that a subfile cached in an F-AP that has strong links to all users should be involved in the packets broadcast over the fronthaul. Let such an F-AP have index i. By (11) and (9), we have $E_0(i) \ge E_1(i,j)$ for any j, which implies $E_2 + E_0(i) \ge E(i,j)$. That means, sending $W_i \oplus W_j$ consumes less energy than sending W_j alone, which proves our claim. Since all subfiles should be involved and all packets are disjoint, we have $2|\mathcal{C}| + |\mathcal{U}| = |\mathcal{M}|$. Hence, the optimal solution should take the form of $|\mathcal{C}| = |\mathcal{M}| - t$ and $|\mathcal{U}| = 2t - |\mathcal{M}|$. It can be determined by adding $2t - |\mathcal{M}|$ dummy nodes and then solved by MATCHING. ☐

B. Repetition Caching and MDS Coded Caching

For repetition caching, the problem stated in Section III can be reduced to the uncoded caching problem by defining a new association matrix $\boldsymbol{A}$ and the corresponding transmit energy values as explained before in subsection IV-B. Its tradeoff curve can then be obtained by Algorithm 3.

For MDS coded caching, the tradeoff curve for each cached file can be obtained by first ordering the potential coded

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

Table I
SIMULATION PARAMETERS

Parameters	Value
Cell radius (R_c)	400 m
Number of F-APs (M)	10 F-APs
Number of Users (N)	20 users
F-APs Peak Power (P)	2 W
F-APs Transmission Rate (R)	Based on Shannon's capacity formula
Target SNR (γ)	$16-22$ dB
Path loss at distance d Km	$140.7+36.7\log_{10}d$, dB
System Bandwidth	2 MHz
Noise Figure	6 dB
Noise Power Spectral Density	-174 dBm/Hz
Number of Files (F)	10 files
Distribution Skewness (β)	1
File Size (B)	100 Mbits
Cache Size (C)	100 Mbits

packets, if any, in the set $\mathcal{P}$ ascendingly based on their associated weights in a queue. Then, at each increment of t, we replace a potential coded packet in the queue starting from the one with the lowest associated weight (i.e., highest transmit energy) by two packets from the set $\mathcal{Z}$. If the queue is empty, a new packet from the set $\mathcal{Z}$ is added, which increases the number of broadcast packets by one and is used to replace a subfile that could originally be delivered to all users via their strong links. Note that the obtained tradeoff curve is not guaranteed to be Pareto optimal for MDS coded caching.

Note that the same idea for intra-file coding explained above can be extended to find the tradeoff under inter-file coding. We apply Algorithm 3 with the set $\mathcal{M}$ and $\mathcal{M}'$ being replaced by $\mathcal{R}$ and $\mathcal{R}'$, respectively.

VI. SIMULATION MODEL AND RESULTS

We consider a single circular cell of radius R_c m with a cloud server located at its center. The F-APs and the users are randomly distributed according to a homogeneous Poisson point process over the whole cell. The signal attenuation from an F-AP to a user is obtained according to the distance-based path loss model (shown in Table I) in the 3GPP standard [38]. Each user requests file f from the library according to the Zipf distribution $p_f = \frac{f^{-\beta}}{\sum_{i=1}^{F} i^{-\beta}}$, where $\beta \geq 0$ is the distribution skewness. The simulation parameters are listed in Table I. All simulation results in this section are averaged over 10,000 random realizations.

Random caching and probabilistic caching are implemented for comparison with our proposed caching schemes. Under random caching, each F-AP picks a file uniformly at random from the library to store until its cache is full. Under probabilistic caching, each F-AP picks a file randomly according to the file popularity distribution until its cache is full. Note that index coding is not used for fronthaul transmissions, since under these two schemes the whole file is either cached or not.

To investigate the effect of the link condition between the F-APs and the users, we vary the target SNR, γ. We assume ideal coding is used so that the transmission rate is given by Shannon's capacity formula, $R = W\log(1+\gamma)$. As can be seen in Fig. 5, as the target SNR γ increases, there are less strong and weak links and more missing links. A user is said to be in an outage if he is unable to obtain his requested file. In other words, the user does not have at least a strong link or two weak links with beamforming and at least a strong link without beamforming. Fig. 5 also shows that beamforming (labeled as "BF") reduces outage probability significantly especially for high target SNR.

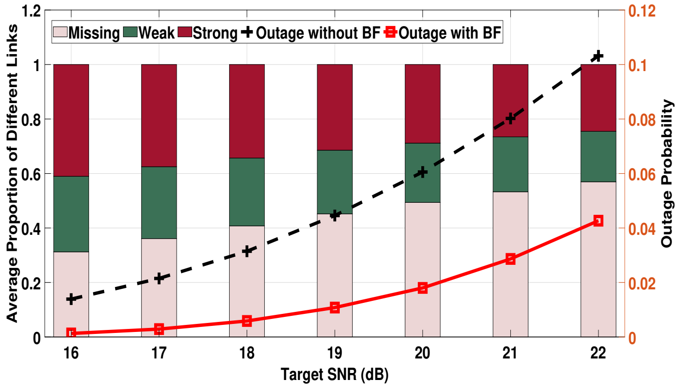


Figure 5: Average proportion of different links and outage probability in the F-RAN versus target SNR.

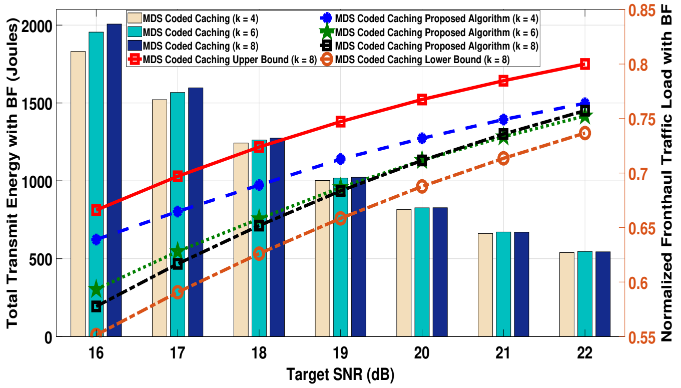


Figure 6: The performance of MDS coded caching with different values of k, and its comparison with the upper and lower bounds on fronthaul traffic.

We evaluate the performance of our proposed caching schemes in minimizing the fronthaul traffic load by varying the target SNR. For MDS coded caching, the subpacketization

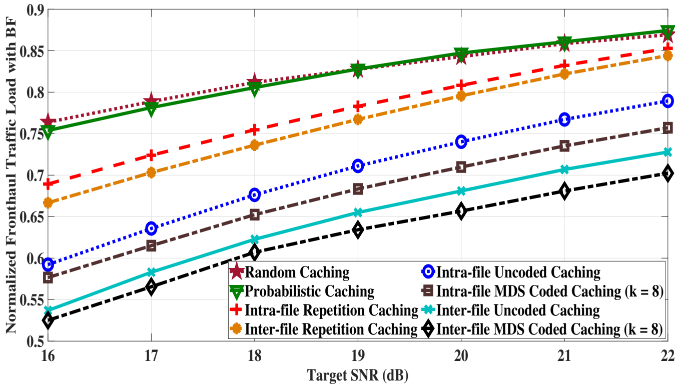


Figure 7: Normalized fronthaul traffic load with BF versus target SNR.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

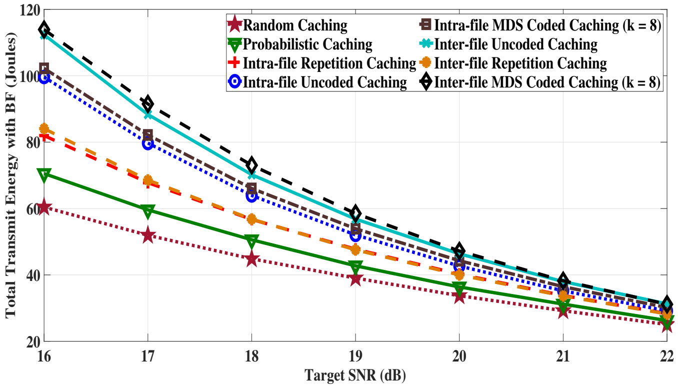


Figure 8: Total transmit energy with BF versus target SNR.

value, k, is a parameter to be chosen empirically. As can be seen from Fig. 6, MDS coded caching with $k = 8$ gives reasonably good performance in fronthaul traffic over a wide range of target SNR at the expense of slightly higher energy consumption. For the rest of our study, we focus on the case where $k = 8$. Fig. 6 also compares the performance of our proposed heuristics for MDS coded caching with the upper and lower bounds. As can be seen, it performs reasonably close to the lower bound, verifying its effectiveness as a heuristic.

The performance results of other caching schemes, as well as MDS coded caching, are shown in Fig. 7. As the target SNR increases, the fronthaul traffic loads of all schemes, obviously, increase as there are more missing links. As expected, inter-file coding reduces the fronthaul traffic load compared to intra-file coding for all caching schemes. The reason is that inter-file coding allows more index coded packets to be formed and transmitted over the fronthaul link compared to intra-file coding. It can be seen that MDS coded caching outperforms all other schemes. This is because MDS coded caching requires fewer blocks to reconstruct the requested file. In addition, independent MDS coded blocks can be broadcast over the fronthaul if needed, which provides new information to all users. The second best scheme is uncoded caching. It outperforms repetition caching because it splits the files into a larger number of subfiles, allowing more files to be cached and more index coded packets to be formed, which in turn reduces the fronthaul traffic load. All our proposed three schemes outperform probabilistic caching and random caching because they cache the most popular files first. At low target SNR, it can be observed that probabilistic caching gives better performance than random caching. At low target SNR, the users connect to more F-APs. Under probabilistic caching, popular files are more likely to be cached, thus giving lower fronthaul traffic. As the target SNR increases, the users connect to fewer F-APs and the advantage of probabilistic caching diminishes.

Next, we calculate the total transmit energy under each caching scheme. Fig. 8 shows the total transmit energy corresponding to the fronthaul traffic load in Fig. 7. As explained before, as the target SNR increases, the fronthaul traffic load increases. The files sent over the fronthaul links are delivered with minimum transmit energy via beamforming between the best channel gains to each user. Thus, the transmit energy decreases as the target SNR increases. In general, the higher the fronthaul traffic load for a caching scheme, the lower its total transmit energy. The two packets included in any inter-file coded packet are delivered via beamforming from the two F-APs that can decode it. For any intra-file coded packet, the coded packet is transmitted via beamforming between the two F-APs with the best channel to each user, and one of the packets after decoding is delivered from the two F-APs that can decode the intra-file coded packet. Thus, the transmit energy for inter-file coding is higher than intra-file coding.

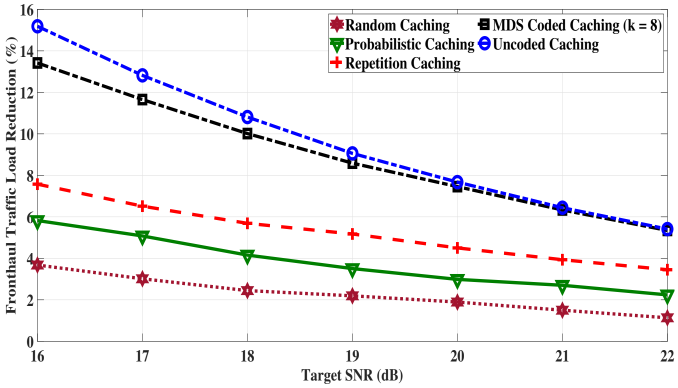


Figure 9: Fronthaul traffic load reduction by BF versus target SNR for intra-file coding.

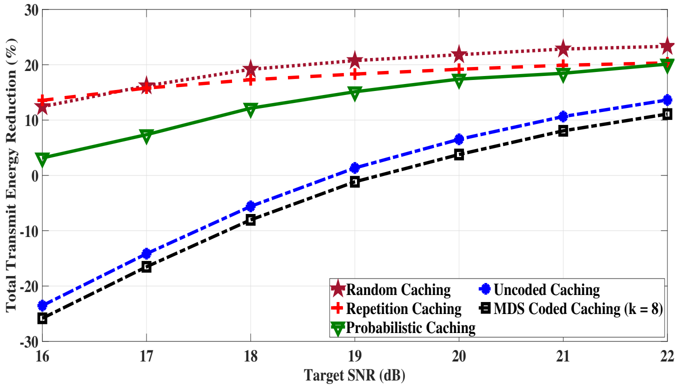


Figure 10: Total transmit energy reduction by BF versus target SNR for intra-file coding.

To examine the advantage of incorporating beamforming into a delivery scheme, Fig. 9 and 10 show, respectively, the percentage reductions in fronthaul traffic load and total transmit energy by beamforming. Under probabilistic, random, and repetition caching, some files (or subfiles) are replicated on the caches of F-APs. When requested, they can be delivered via beamforming even though the corresponding links may be weak, thus reducing fronthaul traffic load. For uncoded and MDS coded caching, distinct blocks are cached. When beamforming is used, more index coded packets can be formed and multicast to F-APs that have either strong or weak links. Without beamforming, only strong links can be used. In summary, beamforming allows weak links to be used, thus reducing the fronthaul traffic for all schemes. Besides, the

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

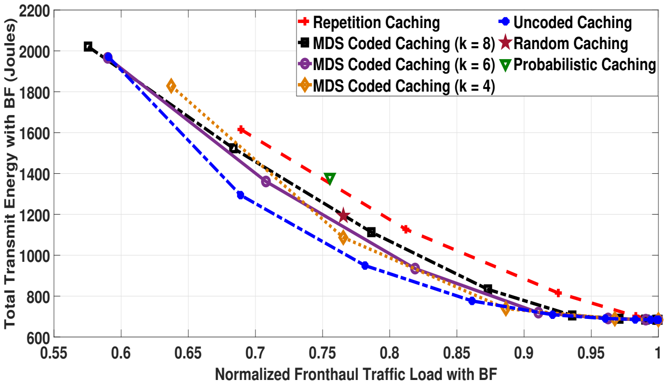


Figure 11: Tradeoff between fronthaul traffic load and total transmit energy with BF for intra-file coding, with $\gamma = 16$ dB.

gain is more significant for uncoded and MDS coded caching because the advantage of index coding is more prominent.

Fig. 10 shows that when the target SNR is 19 dB or above, all schemes have energy savings if beamforming is used. The reason is that all packets broadcast over the fronthaul can be delivered to each user via beamforming between the two strongest channels. The percentage reductions of uncoded and MDS caching are less significant than the other three schemes because they have a larger reduction in fronthaul traffic. When the target SNR is below 19 dB, somewhat surprising, uncoded and MDS caching, with beamforming, consume more energy than their counterparts without beamforming. This is due to the existence of more weak links as can be seen from Fig. 5, which allows an increased use of pairwise index coding and beamforming to reduce fronthaul traffic at the expense of higher energy consumption.

Fig. 11 shows the tradeoff curve between the normalized fronthaul traffic and the total transmit energy at the target SNR of 16 dB. Since intra-file index coding is used over the fronthaul, the tradeoff curve for each file is obtained separately and then added up. Uncoded caching gives superior performance in balancing the tradeoff because it uses the cache space in the most efficient manner by storing all the ten files in the library. Although it is unlikely that a user has strong links to all F-APs to obtain all the subfiles, index coding provides an energy-efficient means to deliver those missing subfiles. MDS coded caching is less space-efficient due to its storage redundancy, so in general, it performs worse than uncoded caching. When $k = 8$, MDS coded caching outperforms uncoded caching when minimizing fronthaul traffic is the primary objective, since the fronthaul link is not used so often as only eight blocks are needed for the reconstruction of the requested file. Nevertheless, if k is too small, the redundancy is too high that many files are not cached, which incurs a higher fronthaul traffic load. When $k = 4$, many independent coded blocks from the set $\mathcal{Z}$ are delivered, which significantly reduces the transmit energy. Repetition caching gives the worst performance among our schemes due to its low storage efficiency. For the same reason, probabilistic caching also performs poorly. Random caching has comparable fronthaul traffic load but lower energy consumption. Note that probabilistic and random caching lack the flexibility of achieving a tradeoff between fronthaul traffic and transmit energy.

VII. CONCLUSIONS

Index coding, edge caching, and transmit beamforming are well-known techniques for improving the energy and spectral efficiencies of a wireless network. While each of them has been extensively studied, their integration into an overall system design for F-RANs is less well understood. This work is an attempt to study how they should work together. Three representative caching schemes, namely, uncoded caching, repetition caching, and MDS caching, are considered in this study. Somewhat surprising, beamforming not only reduces transmitting energy in the access network but also plays a role in minimizing the traffic load in the fronthaul link. The reason is that it enables an effective use of index coding. By transmitting one XOR-coded packet to two F-APs, beamforming can be used to deliver two packets to end-users. It can be envisaged that such a feature can be generalized to distributed beamforming with more F-APs.

If fronthaul traffic minimization is the only objective, MDS caching outperforms uncoded caching and repetition schemes, which is expected since it creates redundancy efficiently, reducing the need for using the fronthaul. Our study, however, unveils the hidden phenomenon that naive uncoded caching, in fact, strikes a better balance between traffic load in the fronthaul link and transmit energy in the access network. The reason is that it stores independent information in the F-APs, creating more index coding opportunities allowed by beamforming. We designed a polynomial-time algorithm based on graph matching to achieve Pareto optimal tradeoff for this particularly interesting scenario. We hope that this study can shed some light on the overall design of the F-RAN and motivate further research.

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, 2016.

[2] M. Peng and K. Zhang, "Recent advances in fog radio access networks: Performance analysis and radio resource allocation," *IEEE Access*, vol. 4, pp. 5003–5009, 2016.

[3] M. A. R. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," in *IEEE INFOCOM Workshops*, pp. 1–4, 2008.

[4] M. Dai, K. W. Shum, and C. W. Sung, "Data dissemination with side information and feedback," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 4708–4720, 2014.

[5] A. Douik, S. Sorour, T. Y. Al-Naffouri, and M.-S. Alouini, "Instantly decodable network coding: From centralized to device-to-device communications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1201–1224, 2017.

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

[7] S. Sorour and S. Valaee, "An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 869–878, 2010.

[8] C. Zhan, V. C. Lee, J. Wang, and Y. Xu, "Coding-based data broadcast scheduling in on-demand broadcast," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3774–3783, 2011.

[9] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions On Networking*, vol. 23, no. 4, pp. 1029–1040, 2014.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. 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/TVT.2022.3194976

[10] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1146–1158, 2016.

[11] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3923–3949, 2017.

[12] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281–1296, 2017.

[13] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching and coded multicasting: Multiple groupcast index coding," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 881–885, 2014.

[14] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "On the average performance of caching and coded multicasting with random demands," in *11th International Symposium on Wireless Communications Systems (ISWCS)*, pp. 922–926, 2014.

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

[16] Y. Guo, S. Mostafa, J. Zou, and C. W. Sung, "A linear-time grouping algorithm for F-RANs with index coding and cache-aided NOMA," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2021.

[17] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

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

[19] S. Mostafa, C. W. Sung, and G. Xu, "Code rate maximization of cooperative caching in ultra-dense networks," in *IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, 2019.

[20] S. Mostafa, C. W. Sung, T. H. Chan, and G. Xu, "Cooperative caching for ultra-dense Fog-RANs: Information optimality and hypergraph coloring," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3652 – 3663, 2021.

[21] M. Peng, *Fog Radio Access Networks (F-RAN): Architectures, Technologies, and Applications*. Springer Nature.

[22] X. Peng, J.-C. Shen, J. Zhang, and K. B. Letaief, "Joint data assignment and beamforming for backhaul limited caching networks," in *IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pp. 1370–1374, 2014.

[23] E. Chen and M. Tao, "User-centric base station clustering and sparse beamforming for cache-enabled cloud RAN," in *IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1–6, 2015.

[24] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6118–6131, 2016.

[25] M.-M. Zhao, Y. Cai, M.-J. Zhao, and B. Champagne, "Joint content placement, RRH clustering and beamforming for cache-enabled cloud-RAN," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2019.

[26] M.-M. Zhao, Y. Cai, M.-J. Zhao, B. Champagne, and T. A. Tsiftsis, "Improving caching efficiency in content-aware C-RAN-based cooperative beamforming: A joint design approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4125–4140, 2020.

[27] H. Zhou, M. Tao, E. Chen, and W. Yu, "Content-centric multicast beamforming in cache-enabled cloud radio access networks," in *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2015.

[28] B. Hu, C. Hua, C. Chen, and X. Guan, "Multicast beamforming for wireless backhaul with user-centric clustering in cloud-RANs," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2016.

[29] X. Chen, M. Zhao, and Y. Cai, "Energy efficient content-centric beamforming in multicast fog radio access network," in *9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2017.

[30] X. Peng, Y. Shi, J. Zhang, and K. B. Letaief, "Layered group sparse beamforming for cache-enabled green wireless networks," *IEEE Transactions on Communications*, vol. 65, no. 12, pp. 5589–5603, 2017.

[31] R. Sun, Y. Wang, N. Cheng, L. Lyu, S. Zhang, H. Zhou, and X. Shen, "QoE-driven transmission-aware cache placement and cooperative beamforming design in cloud-RANs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 636–650, 2019.

[32] S. Mostafa, C. W. Sung, T. H. Chan, and G. Xu, "The interplay between index coding, caching, and beamforming for fog radio access networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, 2020.

[33] W. Wen, Y. Cui, F.-C. Zheng, S. Jin, and Y. Jiang, "Random caching based cooperative transmission in heterogeneous wireless networks," *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 2809–2825, 2018.

[34] Y. Chen, M. Ding, J. Li, Z. Lin, G. Mao, and L. Hanzo, "Probabilistic small-cell caching: Performance analysis and optimization," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4341–4354, 2016.

[35] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479–1494, 2011.

[36] D. Saunders, "Weighted maximum matching in general graphs, https://www.mathworks.com/matlabcentral/fileexchange/42827-weighted-maximum-matching-in-general-graphs," 2021.

[37] S. Micali and V. V. Vazirani, "An $o(\sqrt{|V|}|e|)$ algorithm for finding maximum matching in general graphs," in *21st Annual Symposium on Foundations of Computer Science (SFCS)*, pp. 17–27, 1980.

[38] "Further advancements for E-UTRA physical layer aspects (Release 9), 3GPP standard TS 36.814," Mar. 2010.

Salwa Mostafa received the B.S. and M.S. degrees in Electronics and Communication Engineering from the Faculty of Electronic Engineering, Menoufia University, Egypt, in 2012 and 2015, respectively. She received her Ph.D. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong in 2022. Her research interests include wireless communication, optical communication, edge caching and computing, index coding, distributed learning, reinforcement learning, and non-orthogonal multiple access (NOMA).

Chi Wan Sung (Senior Member, IEEE) was born in Hong Kong. He received the B.Eng., M.Phil., and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 1993, 1995, and 1998, respectively. He worked as an Assistant Professor at The Chinese University of Hong Kong in 1999, and then joined the faculty at City University of Hong Kong in 2000. He is now the Associate Head (Undergraduate Programmes) of the Department of Electrical Engineering. His current research interest is on the interplay between coding, communications, and networking, with emphasis on algorithm design and complexity analysis. He was an Associate Editor of the Transactions on Emerging Telecommunications Technologies (ETT) from 2013 to 2016, and he is currently on the Editorial Boards of ETRI Journal and Electronics Letters.

Guangping Xu received the B.S. degree in computer science from Tianjin University of Technology, China, in 2000, and the M.Sc. and Ph.D. degrees in computer science from Nankai University, China, in 2005 and 2009, respectively. He is currently an associate professor of Tianjin University of Technology, China. His research interests include distributed storage systems and algorithm optimization.

Terence H. Chan completed his PhD in Feb 2001. He was an assistant Professor at The Chinese University of Hong Kong in 2001. From Feb 2002 to Jun 2004, he was a Post-doctoral Fellow at the Department of Electrical and Computer Engineering at the University of Toronto. In 2004, he became an assistant professor at the Department of Computer Science in University of Regina, Canada. He joined Institute for Telecommunications Research (ITR) at University of South Australia as a Senior Research Fellow in 2006. He is now an Associate Professor in ITR