Cache-Aided Radio Access Networks
with Partial Connectivity

Ahmed Roushdy†∗, Abolfazl Seyed Motahari‡, Mohammed Nafie§ and Deniz Gunduz†

† Information Processing and Communication Lab, Imperial College London
‡ Department of Computer Engineering, Sharif University of Technology
∗Wireless Intelligent Networks Center (WINC), Nile University
§Electronics and Communications Department, Cairo University

Email: \{ahmed.elkordy17, d.gunduz\}@imperial.ac.uk, motahari@sharif.edu, mnafie@ieee.org

Abstract

Centralized coded caching and delivery is studied for a partially-connected radio access network (RAN), whereby a set of $H$ edge nodes (ENs) (without caches), connected to a cloud server via orthogonal fronthaul links with limited capacity, serve a total of $K$ UEs over wireless links. The cloud server is assumed to hold a library of $N$ files, each of size $F$ bits; and each user, equipped with a cache of size $MF$ bits, is connected to a distinct set of $r$ ENs. The objective is to minimize the normalized delivery time (NDT), which refers to the worst case delivery latency when each user requests a single file from the library. Two coded caching and transmission schemes are proposed, called the MDS-IA and soft-transfer schemes. MDS-IA utilizes maximum distance separable (MDS) codes in the placement phase and real interference alignment (IA) in the delivery phase. The achievable NDT for this scheme is presented for $r = 2$ and an arbitrary cache size $M$, and also for an arbitrary value of $r$ when the cache capacity is above a certain threshold. The soft-transfer scheme utilizes soft-transfer of coded symbols to ENs that implement zero forcing (ZF) over the edge links. The achievable NDT for this scheme is presented for arbitrary $r$ and cache size $M$. The results indicate that the fronthaul capacity determines which scheme achieves a better performance in terms of the NDT, and the soft-transfer scheme becomes favorable as the fronthaul capacity increases.
Keywords— Coded caching, Partially connected interference networks, Interference management, Delivery latency.

I. INTRODUCTION

Proactively caching popular contents into user devices during off-peak traffic periods, by exploiting the increasingly abundant storage resources in wireless terminals, is a promising solution for the growing network traffic and latency for future communication networks [1]–[4]. A centralized coded proactive caching scheme was introduced by Maddah-Ali and Niesen in [5], where a single server serves multiple cache-enabled users over an error-free shared link; and it is shown to provide significant coding gains with respect to classical uncoded caching. Decentralized coded caching is considered in [6] and [7], where each user randomly stores some bits from each file independently of the other users.

More recently, the idea of coded caching has been extended to wireless radio access networks (RANs), where transmitters and/or receivers are equipped with cache memories. Cache-aided delivery over a noisy broadcast channel is considered in [8] and [9]. Cache-aided delivery from multiple transmitters is considered in [10]–[17]. It is shown in [10] that caches at the transmitters can improve the sum degrees of freedom (DoF) by allowing cooperation among transmitters for interference mitigation. In [11] and [18] this model is extended to a $K_T \times K_R$ network, in which both the transmitters and receivers are equipped with cache memories. An achievable scheme exploiting real interference alignment (IA) for the general $K_T \times K_R$ network is proposed in [12], which also considers decentralized caching at the users.

While the above works assume that the transmitter caches are large enough to store all the database, the fog-aided RAN (F-RAN) model [13] allows the delivery of contents from the cloud server to the edge-nodes (ENs) through dedicated fronthaul links. Coded caching for the F-RAN scenario with cache-enabled ENs is studied in [13]. The authors propose a centralized coded caching scheme to minimize the normalized delivery time (NDT), which measures the worst case delivery latency with respect to an interference-free baseline system in the high signal-to-noise ratio (SNR) regime. In [14], the authors consider a wireless fronthaul that enables coded multicasting. In [15], decentralized coded caching is studied for a RAN architecture with two ENs, in which both the ENs and the users have caches. In [16], this model is extended to an arbitrary number of ENs and users. We note that the models in [13]–[16] assume a fully connected interference network between the ENs and users. A partially connected RAN is studied in [17] from an online caching perspective.

If each EN is connected to a subset of the users through dedicated error free orthogonal links, the corresponding architecture is known as a combination network. Coded caching in a combination network is studied in [19]–[21]. In such networks, the server is connected to a set of $H$ relay nodes, which communicate to $K = \binom{H}{r}$ users, such that each user is connected to a distinct set of $r$ relay nodes, where $r$ is referred to as the receiver connectivity. The links are assumed to be error- and interference-free. The objective is to determine the minimax link load, defined as

This work was supported in part by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie action TactileNET (grant agreement No 690893), by the European Research Council (ERC) Starting Grant BEACON (grant agreement No 725731), and by a grant from the Egyptian Telecommunications Regulatory Authority.
the minimum achievable value of the maximum load among all the links (proportional to the download time) and over all possible demand combinations. Note that, although the delivery from the ENs to the users takes place over orthogonal links, that is, there are no multicasting opportunities as in \[5\], the fact that the messages for multiple users are delivered from the server to each relay through a single link allows coded delivery to offer gains similarly to \[5\]. The authors of \[20\] consider a class of combination networks that satisfy the resolvability property, which require \(H\) to be divisible by \(r\). A combination network in which both the relays and the users are equipped with caches is presented in \[21\]. For the case when there are no caches at the relays, the authors are able to achieve the same performance as in \[20\] without requiring the resolvability property.

In this paper we study the centralized caching problem in a RAN with cache-enabled user equipments (UEs), as depicted in Fig. 1. Our work differs from the aforementioned prior works \[13\]–\[16\] as we consider a partially connected interference channel from the ENs to the UEs, instead of a fully connected RAN architecture. This may be due to physical constraints that block the signals or the long distance between some of the EN-UE pairs.

The considered network topology from the server to the UEs, where ENs act as relay nodes for the UEs they serve, is similar to the combination network architecture; however, we consider interfering wireless links from the ENs to the UEs instead of dedicated links, and study the normalized delivery time in the high SNR regime. The authors in \[22\] study the NDT for a partially connected \((K + L - 1) \times K\) interference channel with caches at both the transmitters and the receivers, where each receiver is connected to \(L\) consecutive transmitters. Our work is different from \[22\], since we take into consideration the fronthaul links from the server to the ENs, and consider a network topology in which the number of transmitters (ENs in our model) is less than or equal to the number of receivers.

We formulate the minimum NDT problem for a given receiver connectivity \(r\). Then, we propose two centralized caching and delivery schemes; in particular, the MDS-IA scheme that we proposed in our previous work \[23\] and the soft-transfer scheme. The MDS-IA scheme exploits real IA to minimize the NDT for receiver connectivity of \(r = 2\). We then extend this scheme to an arbitrary receiver connectivity of \(r\) assuming a certain cache capacity. For this scheme, we show that increasing the receiver connectivity for the same number of ENs and UEs will decrease the NDT for the specific cache capacity region studied, while the reduction in the NDT depends on the fronthaul capacity. On the other, in the soft-transfer scheme the server delivers quantized channel input symbols to the ENs in order to enable them to implement zero-forcing transmission to the UEs to minimize the NDT for an arbitrary receiver connectivity and cache capacity. Our results show that the scheme that achieves a smaller NDT depends on the fronthaul capacity. The MDS-IA scheme achieves a smaller NDT when the fronthaul capacity is relatively limited, while the soft-transfer scheme performs better as the fronthaul capacity increases.

The rest of the paper is organized as follows. In Section II we introduce the system model and the performance measure. In Section III the main results of the paper are presented. The MDS-IA scheme is presented in Section IV while the soft-transfer scheme is introduced in Section V. The numerical results are presented in section VI. Finally, the paper is concluded in Section VII.
Fig. 1: RAN architecture with receiver connectivity $r = 2$, where $H = 5$ ENs serve $K = 10$ UEs.

A. Notation

We denote sets with calligraphic symbols and vectors with bold symbols. The set of integers $\{1, \ldots, N\}$ is denoted by $[N]$. The cardinality of set $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

II. SYSTEM MODEL AND PERFORMANCE MEASURE

A. System Model

We consider the $H \times K$ RAN architecture as illustrated in Fig. 1, which consists of a cloud serve and a set of $H$ ENs, $\mathcal{E} \triangleq \{\text{EN}_1, \ldots, \text{EN}_H\}$, that help the cloud server to serve the requests from a set of $K$ UEs, $\mathcal{U} \triangleq \{\text{UE}_1, \ldots, \text{UE}_K\}$. The cloud is connected to each ENs via orthogonal fronthaul links of capacity $C_F$ bits per symbol, where the symbol refers to a single use of the edge channel from the ENs to the UEs. The edge network from the ENs to the users is a partially connected interference channel, where $\text{UE}_k \in \mathcal{U}$ is connected to a distinct set of $r$ ENs, where $r < H$ is referred to as the receiver connectivity. The number of UEs is $K = \binom{H}{r}$, which means that $H \leq K$. In this architecture, $\text{EN}_i, i \in [H]$, is connected to $L = \binom{H-1}{r-1} = \frac{rK}{H}$ UEs.

The cloud server holds a library of $N$ files, $\mathcal{W} \triangleq \{W_1, \ldots, W_N\}$, each of size $F$ bits. We assume that the UEs request files from this library only. Each UE is equipped with a cache memory of size $MF$ bits, $0 \leq M \leq N$, while the ENs have no caches. We define two parameters, $t_U = \frac{M}{N}$ and $t_E = \frac{M}{N}$, where the former is the normalized cache capacity (per file) available across all the UEs, while the latter is the normalized cache capacity of the UEs connected to a particular edge node. We denote the set of UEs connected to $\text{EN}_i$ by $\mathcal{K}_i$, where $|\mathcal{K}_i| = L$, and the set of ENs connected to $\text{UE}_k$ by $\mathcal{N}_k$, where $|\mathcal{N}_k| = r$. We will use the function $\text{Index}(i,k) : [H] \times [K] \rightarrow [L] \cup \epsilon$, where $\epsilon$ represents the empty set.
which returns $\epsilon$ if UE$_k$ is not served by EN$_i$, and otherwise returns the relative order of UE$_k$ among the L UEs served by EN$_i$. For example, in Fig. 1, we have $\mathcal{K}_1 = \{1, 2, 3, 4\}$, $\mathcal{K}_3 = \{2, 5, 8, 9\}$ and

\[
\begin{align*}
\text{Index}(1, 2) &= 2, & \text{Index}(1, 3) &= 3, & \text{Index}(1, 5) &= \epsilon, \\
\text{Index}(3, 2) &= 1, & \text{Index}(3, 5) &= 2, & \text{Index}(3, 1) &= \epsilon.
\end{align*}
\]

The system operates in two phases: a placement phase and a delivery phase. The placement phase takes place when the traffic load is low, and the UEs are given access to the entire library $\mathcal{W}$. UE$_k$, $k \in [K]$, is then able to fill its cache using the library without any prior knowledge of the future demands or the channel coefficients. Let $Z_k$ denote the cache contents of UE$_k$ at the end of the placement phase. We consider centralized placement; that is, the cache contents of UEs, donated by $Z_1, \ldots, Z_K$, are coordinated jointly.

In the delivery phase, UE$_k$, $k \in [K]$, requests file $W_{d_k}$ from the library, $d_k \in [N]$. We define $d = [d_1, \ldots, d_K] \in [N]^K$ as the demand vector. Once the demands are received, the cloud server sends message $G_i = (G_i(t))_{t=1}^{T_E}$ of blocklength $T_F$ to EN$_i$, $i \in [H]$, via the fronthaul link. This message is limited to $T_FC_F$ bits to guarantee correct decoding at EN$_i$ with high probability. In this paper, we consider half-duplex ENs; that is, ENs start transmitting only after receiving their messages from the cloud server. This is called serial transmission in [13], and the overall latency is the sum of the latencies in the fronthaul and the edge connections. EN$_i$ has an encoding function that maps the fronthaul message $G_i$, the demand vector $d$, and the channel coefficients $H \triangleq \{h_{k,i}\}_{k \in [K], i \in [H]}$, where $h_{k,i}$ denotes the complex channel gain from EN$_i$ to UE$_k$, to a channel input vector $V_i = (V_i(t))_{t=1}^{T_E}$ of blocklength $T_E$, which must satisfy an average power constraint of $P$, i.e., $E\left[\frac{1}{T_E} V_i V_i^\dagger\right] \leq P$. UE$_k$ decodes its requested file as $\hat{W}_{d_k}$ by using its cache contents $Z_k$, the received signal $Y_k(t) = (Y_k(t))_{t=1}^{T_E}$, as well as its knowledge of the channel gain matrix $H$ and the demand vector $d$. We have

\[
Y_k(t) = \sum_{i \in N_k} h_{k,i} V_i(t) + n_k(t),
\]

where $n_k(t) \sim \mathcal{CN}(0, 1)$ denotes the independent additive complex Gaussian noise at the $k$th user. The channel gains are independent and identically distributed (i.i.d.) according to a continuous distribution, and remain constant within each transmission interval. Similarly to [10]–[14], we assume that perfect channel state information is available at all the terminals of network. The probability of error for a coding scheme, consisting of the cache placement, cloud encoding, EN encoding, and user decoding functions, is defined as

\[
P_e = \max_{d \in [N]^K} \max_{k \in [K]} P_e(\hat{W}_{d_k} \neq W_{d_k}),
\]

which is the worst-case probability of error over all possible demand vectors and all the users. We say that a coding scheme is feasible, if we have $P_e \to 0$ when $F \to \infty$, for almost all realizations of the channel matrix $H$.

**B. Performance Measure**

We will consider the normalized delivery time (NDT) in the high SNR regime [24] as the performance measure. Note that the capacity of the edge network scales with the SNR. Hence, to make sure that the fronthaul links do not constitute a bottleneck, we let $C_F = \rho \log P$, where $\rho$ is called the fronthaul multiplexing gain. For cache capacity
M and fronthaul multiplexing gain $\rho$ we say that $\delta(N, M, \rho)$ is an achievable NDT if there exists a sequence of feasible codes that satisfy

$$
\delta(N, M, \rho) = \lim_{P,F \to \infty} \sup (T_F + T_E) \log P.
$$

(3)

We additionally define the fronthaul NDT as

$$
\delta_F(N, M, \rho) = \lim_{P,F \to \infty} \sup T_F \log P,
$$

(4)

and the edge NDT as

$$
\delta_E(N, M, \rho) = \lim_{P,F \to \infty} \sup T_E \log P,
$$

(5)

such that the end-to-end NDT is the sum of the fronthaul and edge NDTs. We define the minimum NDT for a given $(N, M, \rho)$ tuple as

$$
\delta^*(N, M, \rho) = \inf \{ \delta(N, M, \rho) : \delta(N, M, \rho) \text{ is achievable} \}.
$$

III. MAIN RESULT

The main result of the paper is stated in the following two theorems.

**Theorem 1.** For an $H \times K$ partially-connected RAN architecture outlined above, with user cache capacity of $M$, fronthaul multiplexing gain $\rho \geq 0$, number of files $N \geq K$, and considering centralized cache placement, the following NDT is achievable by using the MDS-IA scheme for integer values of $t_E$:

$$
\delta_{MDS-IA}(N, M, \rho) = \left( \frac{L - t_E}{r} \right) \left[ \frac{r - 1}{L} + \frac{1}{t_E + 1} \left( 1 + \frac{1}{\rho} \right) \right]
$$

(6)

for a receiver connectivity of $r = 2$, or for arbitrary receiver connectivity when $t_E \geq L - 2$.

The NDT for non-integer $t_E$ values can be obtained as a linear combination of the NDTs corresponding to the nearest integer $t_E$ values through memory-sharing.

**Theorem 2.** For the same partially connected RAN architecture, the following NDT is achievable by using the soft-transfer scheme for integer values of $t_U$:

$$
\delta_{soft}(N, M, \rho) = (K - t_U) \left[ \frac{1}{\min\{H + t_U, R\}} + \frac{1}{H\rho} \right].
$$

(7)

The NDT for non-integer $t_U$ values can be obtained as a linear combination of the NDTs corresponding to the nearest integer $t_U$ values through memory-sharing.

**Remark.** From Theorem 1 Eqn. (6), when $r \geq 2$, the NDT achieved by the MDS-IA scheme is given by

$$
\delta_{MDS-IA}(N, M, \rho) = \begin{cases} 
\frac{2}{r} \left( \frac{r - 1}{L} + \frac{1}{L - 1} \left( 1 + \frac{1}{\rho} \right) \right), & t_E = L - 2 \\
\frac{L}{r} \left( 1 + \frac{1}{\rho r} \right), & t_E = L - 1 
\end{cases}
$$

Consider two different RAN architectures with $H$ ENs, denoted by RAN-A and RAN-B, with receiver connectivities $r_A$ and $r_B$, respectively, where $r_A + r_B = H$ and $r_A \geq r_B$. The two networks have the same number of UEs
Fig. 2: Comparison of the achievable NDT for a 7 × 21 RAN architecture with library $N = 21$ files for different receiver connectivity and fronthaul multiplexing gains.

$$K = \binom{r_A + r_B}{r_A} = \binom{r_A + r_B}{r_B},$$
but the number of UEs each EN connects to is different, and is given by $L_x = \frac{K}{H} r_x$, $x \in \{A, B\}$. We illustrate the achievable NDT performance of the MDS-IA scheme in a 7 × 21 RAN in Fig. 2 setting $r_A = 5$ and $r_B = 2$ for different fronthaul multiplexing gains. We observe from the figure that, with the same cache capacity $M$ the achievable NDT of network RAN-A is less than or equal to that of network RAN-B, and the gap between the two increases as the fronthaul multiplexing gain decreases. This suggests that the increased connectivity helps in reducing the NDT despite potentially increasing the interference as well, and the gap between the two achievable NDTs for RAN-A and RAN-B becomes negligible as the fronthaul multiplexing gain increases, i.e., $\rho \to \infty$.

IV. MDS-IA Scheme

In this section, we present the MDS-IA scheme for the partially-connected RAN architecture.

A. Cache Placement Phase

We use the cache placement algorithm proposed in [21], where the cloud server divides each file into $r$ equal-size non-overlapping subfiles. Then, it encodes the subfiles using an $(H, r)$ maximum distance separable (MDS) code [25]. The resulting coded chunks, each of size $F/r$ bits, are denoted by $f_n^i$, where $n$ is the index file, and $i \in [H]$ is the index of the coded chunk. EN$_i$ will act as an edge server for the encoded chunk $f_n^i$, $i \in [H]$. Note that, thanks to the MDS code, any $r$ encoded chunks are sufficient to reconstruct the whole file.

Each encoded chunk $f_n^i$ is further divided into $\binom{L-t_E}{t_E}$ equal-size non-overlapping pieces, each of which is denoted by $f_{n,T}^i$, where $T \subseteq [L]$, $|T| = t_E$. The pieces $f_{n,T}^i$, $\forall n$, are stored in the cache memory of UE$_k$ if $k \in K_i$ and Index$(i,k) \in T$; that is, the pieces of chunk $i$, $i \in [H]$, are stored by the $L$ UEs connected to EN$_i$. At the end of the placement phase, each user stores $N r \binom{L-t_E}{t_E-1}$ pieces, each of size $\frac{F}{r (t_E)}$ bits, which sum up to $MF$ bits, satisfying the memory constraint with equality. We will next illustrate the placement phase through an example.
TABLE I: Cache contents after the placement phase for the RAN scenario considered in Example 1, where $K = N = 10$, $r = 2$, $L = 4$, $t_E = 1$ and $M = \frac{5}{2}$.

| EN | 1,2 | 3,1 | 1,2 | 2,1 | 5,2 |
|---|---|---|---|---|---|
| X1 | $f_{1,1} + f_{2,1}$ | $f_{2,1} + f_{2,1}$ | $f_{2,1} + f_{2,1}$ | $f_{1,1} + f_{2,1}$ | $f_{2,1} + f_{2,1}$ |
| X2 | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ |
| X3 | $f_{1,1} + f_{4,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ |
| X4 | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ |
| X5 | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ | $f_{1,1} + f_{3,1}$ |

TABLE II: The data delivered from the cloud server to each EN for Example 1.

**Example 1.** Consider the partially connected RAN depicted in Fig. 1, where $H = 5$, $K = N = 10$, $r = 2$ and $L = 4$. The cloud server divides each file into $r = 2$ subfiles. These subfiles are then encoded using a $(5, 2)$ MDS code. As a result, there are 5 coded chunks, denoted by $f^i_n$, $n \in [10]$, $i \in [5]$, each of size $F/2$ bits. For $t_E = 1$, i.e., $M = N/L$, each encoded chunk $f^i_n$ is further divided into $(t_E) = 4$ pieces $f^i_n \cdot$, where $\mathcal{T} \subseteq [4]$ and $|\mathcal{T}| = t_E = 1$. Cache contents of each user are listed in TABLE I. Observe that each user stores two pieces of the encoded chunks of each file for a total of 10 files, i.e., $\frac{5}{2}F$ bits, which satisfies the memory constraint.

**B. Delivery Phase**

The delivery phase is carried out in two steps. The first step is the delivery from the cloud server to the ENs, and the second step is the delivery from the ENs to the UEs.

1) **Delivery from the cloud server to the ENs:** For each $(t_E + 1)$-element subset $S$ of $[L]$, i.e., $S \subseteq [L]$ and $|S| = t_E + 1$, the cloud server will deliver the following message to $EN_i$:

$$X^S_i \triangleq \bigoplus_{k \in K, \text{ Index}(i, k) \in S} f^i_{d_k \cdot S \setminus \text{Index}(i, k)}.$$  \hspace{1cm} (8)

Overall, for given $d$, the following set of messages will be delivered to $EN_i$

$$\{X^S_i : S \subseteq [L], |S| = t_E + 1\},$$  \hspace{1cm} (9)

which makes a total of $(L(t_E + 1)) \frac{F}{r(t_E) \rho}$ bits. The fronthaul NDT from the cloud server to the ENs is then given by

$$\delta_F(N, M, \rho) = \frac{L(t_E + 1)}{r(t_E) \rho} = \frac{L - t_E}{(t_E + 1)\rho}.$$  \hspace{1cm} (10)

The message to be delivered to each EN in Example 1 is given in TABLE I and we have $\delta_F(10, \frac{5}{2}, 1) = \frac{3}{4\rho}$.
Algorithm 1: Generator for $A$, $B$ and $C$ Matrices

1 $A = \{\}$, $B = \{\}$, $C = \{\}$, $g = 0$
2 FOR $k = 1, \ldots, K$
3 FOR $j = 1, \ldots, I$
4 $g = g + 1$
5 FOR $i = 1, \ldots, r$
6 $B_g \leftarrow [X_k(j, i) \ B_g]$
7 Find $J_i$: set of other UEs receiving the same interference signal $X_k(j, i)$, $|J_i| = (L - |S| - 1)$. Sort UEs in $J_i$ in ascending order.
8 FOR each user in $J_i$, find interference vector $x_k^q$, s.t. UE $k \in J_i$ and $X_k(j, i) \notin x_k^q$
9 $Q_i \leftarrow$ set of vectors $x_k^q$
10 END FOR
11 IF $|J_i| \geq 1$
12 FOR $R = 1, \ldots, |J_i|$
13 FOR $e = 1, \ldots, |Q_1(:, R)|$
14 FOR $c = 1, \ldots, |Q_2(:, R)|$
15 IF $Q_1(e, R) = Q_2(c, R)$
16 $B_g = [B_g \ Q_1(e, R)]$
17 Go to 21, i.e., next iteration of $R$.
18 END IF
19 END FOR
20 END FOR
21 END FOR
22 END IF
23 $C_g \leftarrow \bigcup_{k: \hat{S} \subseteq B_g, \text{where } |\hat{S}| = r} u_k$, for $\hat{S} \subseteq B_g$
24 FOR $e = 1, \ldots, |C_g|$
25 FOR $i = 1, \ldots, r$
26 $A_g \leftarrow [A_g \ h_{C_g(e), N_{C_g(e)}}(i)]$
27 END FOR
28 END FOR
29 Remove interference signals in $B_g$ from $(X_k)_{k=1}^K$
30 $\mathcal{J}_i = \{\}$, $Q_i = \{\}$ for $i = 1, \ldots, r$
31 END FOR
32 END FOR
where the dimensions of these matrices are $C_1$, $C_2$, $C_3$, $C_4$, $C_5$, $C_6$, $C_7$, $C_8$, $C_9$, $C_{10}$.

| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | $X_7$ | $X_8$ | $X_9$ | $X_{10}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $X_{1,1}^{2,3}$ | $X_{1,2}^{2,3}$ | $X_{1,3}^{2,3}$ | $X_{1,4}^{2,3}$ | $X_{1,5}^{2,3}$ | $X_{2,1}^{2,3}$ | $X_{2,2}^{2,3}$ | $X_{2,3}^{2,3}$ | $X_{2,4}^{2,3}$ | $X_{2,5}^{2,3}$ |
| $X_{3,1}^{2,4}$ | $X_{3,2}^{2,4}$ | $X_{3,3}^{2,4}$ | $X_{3,4}^{2,4}$ | $X_{3,5}^{2,4}$ | $X_{4,1}^{2,4}$ | $X_{4,2}^{2,4}$ | $X_{4,3}^{2,4}$ | $X_{4,4}^{2,4}$ | $X_{4,5}^{2,4}$ |
| $X_{5,1}^{2,4}$ | $X_{5,2}^{2,4}$ | $X_{5,3}^{2,4}$ | $X_{5,4}^{2,4}$ | $X_{5,5}^{2,4}$ | $X_{6,1}^{2,4}$ | $X_{6,2}^{2,4}$ | $X_{6,3}^{2,4}$ | $X_{6,4}^{2,4}$ | $X_{6,5}^{2,4}$ |

TABLE III: The interference matrices at the UEs of Example 1.

2) Delivery from the ENs to the UEs: $UE_k$, $k \in [K]$, is interested in the following set of messages:

$$\mathcal{M}_k = \bigcup_{i, S : i \in \mathcal{N}_k, S \subseteq [L], |S| = |t_k + 1|, \text{Index}(i,k) \in S} X_i^S,$$

(11)

where $|M_k| = r(L-1)$. On the other hand, the transmission of the following messages interfere with the transmissions of the messages in $\mathcal{M}_k$:

$$\mathcal{I}_k = \bigcup_{i, S : i \in \mathcal{N}_k, S \subseteq [L], |S| = |t_k + 1|, \text{Index}(i,k) \notin S} X_i^S.$$

(12)

Each $X_i^S \in \mathcal{I}_k$ causes interference at $L - |S|$ UEs, including $UE_k$. Hence, the total number of interfering signals at $UE_k$ from the ENs in $N_k = \mathcal{N}_k(q)$ is $rI$, where $I \triangleq \binom{L}{t_k + 1} - \binom{L-1}{t_k}$ is the number of interfering signals from each EN connected to $UE_k$.

We enumerate the ENs in $N_k$, $k \in [K]$, such that $N_k(q)$ is the q-th element in $N_k$ when they are ordered in ascending order. At $UE_k$, $k \in [K]$, we define the interference matrix $X_k$ to be an $I \times r$ matrix whose columns are denoted by $\{x_k^q\}_{q=1}^r$, where the q-th column $x_k^q$ represents the interference caused by a different EN in $N_k(q)$. For each column vector $x_k^q$, we sort the set of interfering signals $\mathcal{I}_k$ for $i = N_k(q)$ in ascending order. In Example 1, we have $N_1(1) = EN_1$, $N_1(2) = EN_2$, etc., and the interference matrices are shown in TABLE III.

We will use real IA, presented in [26], and extended to complex channels in [27], for the delivery of the ENs to the UEs to align each of the $r$ interfering signals in $\mathcal{I}_k$, one from each EN, in the same subspace. We define $A$, $B$ and $C$ to be the basis matrix, i.e., function of the channel coefficients, the data matrix and user matrix, respectively, where the dimensions of these matrices are $G \times r(\binom{L-|S|-1}{r})$, $G \times (r + L - |S| - 1)$ and $G \times (\binom{r + L - |S|-1}{r})$, respectively, where $G = \binom{L}{t_k + 1}$. We denote the rows of these matrices by $A_g$, $B_g$ and $C_g$, respectively, where $g \in [G]$. The row vectors $\{A_g\}_{g=1}^G$ are used to generate the set of monomials $G(A_g)^G_{\geq 1}$. Note that, the function $\mathcal{T}(u)$ defined in [10] corresponds to $G(A_g)$ in our notation. The set $G(A_g)^G_{\geq 1}$ is used as the transmission directions for the modulation constellation $\mathbb{Z}_Q$ [10] for the whole network. In other words, each row data vector $B_g$ will use the set $G(A_g)$ as the transmission directions of all its data to align all the $r$ interfering signals from $B_g$ in the same subspace at $UE_k \in C_g$, if these $r$ signals belong to $X_k$.

We next explain matrix $C$ more clearly. For each $S \subseteq B_g$, with $|S| = r$, there will be a user at which these data will be aligned in the same subspace, i.e., $|C_g| = (\binom{r + L - |S|-1}{r})$. The row $C_g$ consists of $UE_k$, where $S \in X_k$. 
We employ Algorithm 1 to obtain matrices $A$, $B$ and $C$ for a receiver connectivity of $r = 2$, and for arbitrary receiver connectivity when $t_E = L - 2$. In Example 1, the three matrices are given as follows:

$$
A = \begin{pmatrix}
h_{1,1} & h_{1,2} & h_{4,1} & h_{4,5} & h_{7,2} & h_{7,5} \\
h_{1,1} & h_{1,2} & h_{3,1} & h_{3,4} & h_{6,2} & h_{6,4} \\
h_{1,1} & h_{1,2} & h_{2,1} & h_{2,3} & h_{5,2} & h_{5,3} \\
h_{2,1} & h_{2,3} & h_{4,1} & h_{4,5} & h_{9,3} & h_{9,5} \\
h_{2,1} & h_{2,3} & h_{3,1} & h_{3,4} & h_{8,3} & h_{8,4} \\
h_{3,1} & h_{3,4} & h_{4,1} & h_{4,5} & h_{10,4} & h_{10,5} \\
h_{5,2} & h_{5,3} & h_{7,2} & h_{7,5} & h_{9,3} & h_{9,5} \\
h_{5,2} & h_{5,3} & h_{6,2} & h_{6,4} & h_{8,3} & h_{8,4} \\
h_{6,2} & h_{6,4} & h_{7,2} & h_{7,5} & h_{10,4} & h_{10,5} \\
h_{8,3} & h_{8,4} & h_{10,4} & h_{10,5} & h_{9,3} & h_{9,5}
\end{pmatrix},
$$

$$
B = \begin{pmatrix}
X_1^{2,3} & X_2^{2,3} & X_5^{3,4} \\
X_1^{2,4} & X_2^{2,4} & X_4^{3,4} \\
X_1^{3,4} & X_2^{3,4} & X_3^{3,4} \\
X_1^{3,3} & X_3^{2,4} & X_5^{2,4} \\
X_1^{1,4} & X_3^{2,4} & X_4^{2,4} \\
X_1^{1,2} & X_2^{1,3} & X_5^{2,3} \\
X_2^{1,3} & X_3^{1,4} & X_5^{1,4} \\
X_2^{1,4} & X_3^{1,4} & X_4^{1,4} \\
X_2^{1,2} & X_1^{1,3} & X_5^{1,3} \\
X_4^{1,2} & X_3^{1,2} & X_5^{1,2}
\end{pmatrix},
$$

$$
C = \begin{pmatrix}
UE_1 & UE_4 & UE_7 \\
UE_1 & UE_4 & UE_6 \\
UE_1 & UE_2 & UE_5 \\
UE_2 & UE_4 & UE_9 \\
UE_2 & UE_3 & UE_8 \\
UE_3 & UE_4 & UE_{10} \\
UE_5 & UE_7 & UE_9 \\
UE_5 & UE_6 & UE_8 \\
UE_6 & UE_7 & UE_{10} \\
UE_8 & UE_{10} & UE_9
\end{pmatrix}.
$$

Then, for each signal in $B_g$, we construct a constellation that is scaled by the monomial set $G(A_g)$, i.e, the signals $X_2^{2,4}$ in $B_2$ use the monomials in $G(A_2)$, resulting in the signal constellation

$$
\sum_{v \in G(A_g)} vZ_{Q}.
$$

(13)

Focusing on Example 1, we want to assess whether the interfering signals have been aligned, and whether the requested subfiles arrive with independent channel coefficients, the decodability is guaranteed. Starting with $u_1$, the received constellation corresponding to the desired signals $X_1^{1,2}, X_1^{1,3}, X_1^{1,4}, X_2^{1,2}, X_2^{1,3}$ and $X_2^{1,4}$ is given as follow

$$
C_D = h_{1,1} \sum_{v \in G(A_6)} vZ_{Q} + h_{1,1} \sum_{v \in G(A_1)} vZ_{Q} + h_{1,1} \sum_{v \in G(A_5)} vZ_{Q} \\
+ h_{1,2} \sum_{v \in G(A_6)} vZ_{Q} + h_{1,2} \sum_{v \in G(A_7)} vZ_{Q} + h_{1,2} \sum_{v \in G(A_8)} vZ_{Q}.
$$

(14)
The received constellation for the interfering signals $X_1^{2,3}X_2^{2,3}$, $X_1^{2,4}$, $X_2^{2,4}$, $X_2^{3,4}$ and $X_2^{3,4}$ is given by

$$C_I = h_{1,1} \sum_{v \in G(A_1)} vZ_Q + h_{1,2} \sum_{v \in G(A_1)} vZ_Q + h_{1,1} \sum_{v \in G(A_2)} vZ_Q + h_{1,2} \sum_{v \in G(A_2)} vZ_Q + h_{1,1} \sum_{v \in G(A_3)} vZ_Q + h_{1,2} \sum_{v \in G(A_3)} vZ_Q.$$  \hspace{1cm} (15)

Equation (15) proves that every two interfering signals, one from each EN, i.e., the first two terms in equation (15), have collapsed into the same sub-space. Also, since the monomials $G(A_1)$, $G(A_2)$ and $G(A_3)$ do not overlap and linear independence is obtained, the interfering signals will align in $I = 3$ different subspaces. We can also see in (14) that the monomials corresponding to the intended messages do not align, and rational independence is guaranteed (with high probability), and the desired signals will be received over 6 different subspaces. Since the monomials form different constellations, $C_D$ and $C_I$, whose terms are functions of different channel coefficients, we can assert that these monomials do not overlap. Hence, we can claim that IA is achieved Thus, our scheme guarantees that the desired signals at each user will be received in $r^L_{t_E}$ different subspaces, and each $r$ interfering signals will be aligned into the same subspace, i.e., one from each EN, resulting in a total of $I = \binom{L}{t_E + 1} - \binom{L - 1}{t_E - 1}$ interference subspaces.

When $t_E = L - 1$, the number of interference signals at each user is $I = 0$. Hence, we just transmit the constellation points corresponding to each signal. We are sure that the decodability is guaranteed since all channel coefficients are i.i.d. according to a continuous distribution.

UE$_k$ utilizes its cache content $Z_k$ to extract the pieces $f_{i,T}^1$, for $i \in N_k$ and $\text{Index}(i,k) \notin T$. Therefore, UE$_k$ reconstructs $f_{i,T}^1$ and decodes its requested file $W_k$. In Example 1, UE$_1$ utilizes its memory $Z_1$ in TABLE[I] to extract $f_{1,T}^1$, for $i = 1, 2$, and $T = \{2, 3, 4\}$. Hence, UE$_1$ reconstructs $f_{1}^1$ and $f_{1}^2$, and decodes its requested file $W_1$; and similarly for the remaining UEs. Thus, the edge NDT from ENs to the UEs is equal to $\delta_E(10, \frac{5}{2}, 1) = \frac{9}{8}$, while the total NDT is $\delta(10, \frac{5}{2}, 1) = \frac{3}{8} + \frac{9}{8}$. In the general case, the NDT from the ENs to the UEs by using the MDS-IA scheme is given by

$$\delta_E(N,M,\rho) = \frac{(L - 1)(r - 1) + (L - 1)}{r(L - 1)} = \frac{L - t_E}{r} \left( \frac{r - 1}{L} + \frac{1}{t_E + 1} \right).$$ \hspace{1cm} (16)

Together with the fronthaul NDT in (10), we obtain the end-to-end NDT as given in Theorem 1. The NDT achieved by the MDS-IA scheme for various system settings will be presented in section VI.

V. SOFT-TRANSFER SCHEME

In this section, we present a centralized coded caching scheme for the same partially-connected RAN architecture with receiver connectivity $r$, and $t_U \in [K]$. The soft-transfer of channel input symbols over fronthaul links is proposed in [28], where the cloud server implements ZF-beamforming and quantizes the encoded signal to be transmitted to each EN. Therefore, the fronthaul NDT is given as follow

$$\delta_{F-\text{soft}}(N,M,\rho) = \left(1 - \frac{t_U}{K}\right) \frac{K}{H\rho}.$$ \hspace{1cm} (17)
while the total NDT can be expressed as follows

$$\delta_{\text{soft}}(N,M,\rho) = \delta_{E-\text{Ideal}} + \delta_{F-\text{soft}}$$  \hspace{1cm} (18)$$

where $\delta_{E-\text{Ideal}}$ is the achievable edge NDT in an ideal system in which the ENs can acts as one big multi-antenna transmitter. This is equivalent to assuming either full caching of the whole library $W$ at all the ENs, or no fronthaul capacity limitations, i.e., $\rho \to \infty$. In such an ideal system, full cooperation is possible among the ENs for any user demand vector. We will provide a coding scheme that uses ZF for this ideal system to provide a general expression for $\delta_{E-\text{Ideal}}$.

### A. Cache Placement Phase

For any file $W_n$ in the library, $n \in [N]$, we partition it into $\binom{K}{t_U}$ equal-size subfiles, each of which is denoted by $W_{n,T}$, where $T \subseteq [K]$, $|T| = t_U$. The subfiles $W_{n,T}$, $\forall n$, are stored in the cache memory of UE$k$ if $k \in T$. At the end of the placement phase, each user stores $N\binom{K-1}{t_U-1}$ subfiles, each of size $F/\binom{K}{t_U}$ bits, which sum up to $MF$ bits, satisfying the cache capacity constraint with equality.

**Example 2.** Consider the $4 \times 6$ partially-connected RAN architecture with $H = 4$, $K = N = 6$, $r = 2$ and $L = 3$. For $t_U = 2$, file $W_n$, $\forall n \in [N]$, is divided into $\binom{6}{2} = 15$ disjoint subfiles $W_{n,T}$, where $T \subseteq [K]$ and $|T| = t_U = 2$. The size of each subfile is $\frac{F}{15}$ bits. Cache contents of each user are listed in TABLE IV. Observe that each user stores $6\binom{5}{1} = 30$ subfiles, each of size $\frac{F}{15}$ bits, which sum up to $2F$ bits, satisfying the memory constraint with equality.

### B. Delivery Phase

In the delivery phase, the receiver requests are revealed. Let $W_{d_k}$ denote the request of user UE$k$, $k \in [K]$. Then, the ENs need to deliver the subfiles in

$$\{W_{d_k,T} : k \notin T, T \subseteq [K], |T| = t_U\}$$  \hspace{1cm} (19)
TABLE V: Missing subfiles for each user’s request in Example 2. These subfiles must be delivered to the corresponding user within the delivery phase.

TABLE VI: The alternative representation of the missing subfiles of each user in Example 2, including the UEs at which each subfile will be zero-forced at.

to UE\(_k\), i.e., the subfiles of file W\(_{d_k}\) which have not been already stored in the cache of UE\(_k\). The total number of such subfiles is \(\binom{K}{t_U} - \binom{K-1}{t_U-1}\).

For Example 2, assuming, without loss of generality, that UE\(_k\) request W\(_k\), the missing subfiles of each user request, to be delivered in the delivery phase, are listed in TABLE V.

We first describe the delivery phase when K - H ≤ t\(_U\) ≤ K - 1. We will later consider the case t\(_U\) < K - H separately. Note that, when t\(_U\) = K the achievable NDT is equal to zero since each user can cache all the N files.

Case 1 (K - H ≤ t\(_U\) ≤ K - 1): We introduce an alternative representation for each subfile in (19), which will denote the UEs at which each of these subfiles will be zero-forced. In particular, we will denote subfile W\(_{d_k,T,\pi}\) by W\(_{d_k,T,\pi}\), where \(\pi \subseteq [K] \backslash \{(k) \cup T\}\), |\(\pi\)| = K - (t\(_U\) + 1), denotes the set of receivers at which this subfile will be zero-forced. The total number of subfiles intended for UE\(_k\) is \(\binom{K}{t_U} - \binom{K-1}{t_U-1}\) subfiles. TABLE VI shows this alternative representation for the missing subfiles for each user in Example 2.

All the ENs will transmit W\(_{d_k,T,\pi}\) by using the beamforming vector v\(_\pi\) ∈ \(\mathbb{R}^H\) to zero-force this subfile at the UEs in \(\pi\). We define the matrix H\(_\pi\) with dimensions K - (t\(_U\) + 1) × H to be the channel matrix from the ENs to the UEs in \(\pi\) and the set of ENs \(\mathcal{E}\). The beamforming vector v\(_\pi\) is designed as follows:

\[
v_\pi = \sum_{i=1}^{D} v_i,
\] (20)
Fig. 3: One step of the delivery phase for the RAN architecture in Example 2 with receiver connectivity \( r = 2 \), where \( H = 4 \) ENs serve \( K = 6 \) UEs.

where \( D = H - K + (t_U + 1) \) is the size of the null space of the matrix \( H_\pi \), while \( \mathbf{v}_i \) is the \( i \)th basis vector of this null space. The null space of matrix \( H_\pi \) with \( 1 \leq |\pi| \leq H - 1 \) always has a non-zero element since \( D \geq 1 \). Hence, the subfile \( W_{d_k, T, \pi} \) for any \( \pi \geq 1 \) can be always zero-forced at the UEs in \( \pi \). In Example 2, the size of the null space is \( D = 1 \).

In each step of the delivery phase, we transmit one subfile from each requested file, which means that we will have \( (K - t_U) \) steps in total. The transmitted set of subfiles at each step will be decoded by their intended receivers without any interference since each subfile \( W_{d_k, T, \pi} \) in this set is already cached at \( |T| = t_U \) UEs, \( T \subseteq \mathcal{U} \), and will be zero-forced at \( |\pi| = K - (t_U + 1) \) other UEs, \( \pi \subseteq \mathcal{U} \). Since \( T \cap \pi = \phi \) and \( |T| + |\pi| = |\mathcal{U}| - 1 \), each subfile will not cause any interference at the \( |K| - 1 \) undesired UEs. Hence, the edge NDT of the ideal system is given by

\[
\delta_{E-Ideal}(N, M, \rho) = \frac{K - t_U}{K}.
\] (21)

The end-to-end achievable NDT for the soft-transfer scheme is given by

\[
\delta_{soft}(N, M, \rho) = (K - t_U) \left( \frac{1}{K} + \frac{1}{H\rho} \right).
\] (22)

In Example 2, the delivery phase of the soft-transfer scheme consists of \( \binom{6}{2} - \binom{5}{1} = 10 \) steps. At each step all the ENs will cooperate to transmit \( K = 6 \) subfiles, one for each user, by acting as one transmitter with 4 antennas according to the ideal system assumption. In Fig. 3, we show one step of the delivery phase given that all the ENs cooperate to deliver the missing subfiles, i.e., the ideal system as discussed earlier. In this figure, we can see that each user will be able to decode its desired subfile without any interference, as the undesired subfiles are
either cached or zero-forced at this user. The edge NDT of this ideal system is $\delta_{\text{Ideal}} = \frac{2}{3}$, while the achievable end-to-end NDT for the soft-transfer scheme is $\delta_{\text{soft}}(6, 2, \rho) = \frac{2}{3} + \frac{2}{\rho}$.

Case 2 ($t_U < K - H$): In this case each subfile $W_{d_k, T}$ in (19) is partitioned into $\binom{K-(t_U+1)}{H-1}$ disjoints chunks of equal size, denoted by

$$W_{d_k, T} = \{W_{d_k, T, \pi, \pi'} : \pi \subseteq [K] \setminus (\{k\} \cup T), |\pi| = H - 1, \pi' \subseteq [K] \setminus (\{k\} \cup T \cup \pi)\},$$

(23)

where $\pi, |\pi| = H - 1$, is the set of receivers at which this chunk will be zero-forced, while $\pi_t, |\pi_t| = K - (H + t_U)$, is the set of receivers at which this chunk will cause interference, i.e., the set of receivers at which this chunk is neither cached nor zero-forced. The total number of chunks intended for user $UE_k$ is $\binom{K}{t_U} - \binom{K-1}{t_U-1} \binom{K-(t_U+1)}{H-1}$, while the size of each chunk is $|W_{d_k, T, \pi, \pi'}| = \frac{F}{\binom{K}{t_U} \binom{K-(t_U+1)}{H-1}}$ bits.

All the ENs $E$ transmit $W_{d_k, T, \pi, \pi'}$ by using the beamforming vector $v_{\pi} \in \mathbb{R}^H$ to zero-force this chunk at the UEs in set $\pi$. The matrix $H_{\pi}$ for this case is of dimensions $H - 1 \times H$. The beamforming vector $v_{\pi}$ is designed as follow

$$v_{\pi} = \text{Null}\{H_{\pi}\},$$

(24)

where $\text{Null}\{H_{\pi}\}$ is the null space of matrix $H_{\pi}$. The null space always exists and its size is 1.

In each step of delivery phase, we transmit the set of chunks that have the same $\pi_t$ and belong to different requested files, i.e., we transmit at most one chunk from each requested file at the same time. In other words, in each step we transmit the set of chunks that belong to different requested files but cause interference at the same set of receivers. The size of each transmitted set per step is $(H + t_U)$. This is because, the chunks that have the same $\pi_t$ do not belong to the set of requested files $\{W_{d_k} : k \in \pi_t\}$, with size $K - (H + t_U)$, and will only belong to the one of the remaining $(H + t_U)$ requested files, denoted by the set

$$\mathcal{R} = \{W_{d_k} : k \notin \pi_t\},$$

(25)

while we only transmit the chunks that belong to different requested files at the same time.

Each user is interested in $\binom{K}{t_U} - \binom{K-1}{t_U-1} \binom{K-(t_U+1)}{H-1}$ chunks, while the total number of requested files is $K$, and in every step we transmit only $(H + t_U)$ chunks. Hence, the total number of the steps is given by

$$\frac{\binom{K}{t_U} - \binom{K-1}{t_U-1} \binom{K-(t_U+1)}{H-1}}{H + t_U}. $$

(26)

We denote the set of UEs interested in the transmitted $(H + t_U)$ chunks that have the same $\pi_t$ and belong to different requested files by $UE_{\pi_t} = \{U \setminus \pi_t\}$, where $|U_{\pi_t}| = H + t_U$. For the chunk $W_{d_k, T, \pi, \pi'}$ requested by user $UE_k \in U_{\pi_t}$, we have $(T \cup \pi \cup \pi') \subseteq U$; and hence, $(T \cup \pi) \subseteq U_{\pi_t}$. The set of transmitted chunks will be decoded by the UEs in set $U_{\pi_t}$, without interference since each chunk $W_{d_k, T, \pi, \pi'}$ in this set is already cached at $|T| = t_U$ UEs, where $T \subseteq U_{\pi_t}$, and will be zero-forced at $|\pi| = H - 1$ UEs, $\pi \subseteq U_{\pi_t}$. Since $T \cap \pi = \phi$ and $|T| + |\pi| = |U_{\pi_t}| - 1$, ...
(a) Fronthaul multiplexing gain $\rho = 0.05$.

(b) Fronthaul multiplexing gain $\rho = 20$.

Fig. 4: Comparison of the achievable NDT for a $5 \times 10$ RAN architecture with library $N = 10$ files and receiver connectivity $r = 2$ for the MDS-IA scheme and the soft-transfer scheme.

which means that each chunk will not cause any interference at the $|\mathcal{U}_\pi| - 1$ undesired UEs in set $\mathcal{U}_\pi$. Hence, each user in this set $\mathcal{U}_\pi$ can decode its desired chunk, the edge NDT of the ideal system in this case is given by

$$\delta_{E-Ideal} = \frac{K - t_U}{K} \frac{K}{H + t_U}.$$  \hfill (27)

The achievable end-to-end NDT by using the soft-transfer scheme in case 2 is given by

$$\delta_{soft}(N, M, \rho) = (K - t_U) \left( \frac{1}{H + t_U} + \frac{1}{H\rho} \right)$$  \hfill (28)

VI. NUMERICAL RESULTS

In this section we compare the end-to-end latency achieved by the two proposed schemes. The NDTs achieved by the MDS-IA and soft-transfer schemes for two different fronthaul multiplexing gains are plotted in Fig. 4. We observe from Fig. 4 that the end-to-end NDT significantly reduces as $\rho$ increases and the NDT in Fig. 4-b is mainly dominated by the edge NDT. We also observe by comparing the two plots that, which of the two schemes perform better depends highly on the fronthaul multiplexing gain. In Fig. 4-a, with a low fronthaul multiplexing gain of $\rho = 0.05$, the MDS-IA scheme performs better than the soft-transfer scheme for almost all cache capacity values, while we observe in Fig. 4-b that the soft-transfer scheme outperforms the MDS-IA scheme when the multiplexing gain increases, i.e., $\rho = 20$. This is mainly because the edge latency of the soft transfer scheme is minimal as it is derived based on an ideal fully cooperative delivery from the ENs. Therefore, when the fronthaul links are of high capacity, the performance of the soft transfer scheme becomes nearly optimal. On the other hand, when the fronthaul links are the bottleneck, the latency can be reduced by delivering less information to the ENs, and this is achieved by the MDS-IA scheme.
VII. CONCLUSIONS

We have studied centralized caching and delivery over a partially-connected RAN with a specified network topology between the ENs and the UEs. We have proposed two schemes, namely the MDS-IA scheme and the soft-transfer scheme. The MDS-IA scheme exploits MDS coding to virtually store data at multiple ENs, and real IA to deliver these coded packets to the UEs over the edge network. We have derived the achievable NDT for this scheme for a receiver connectivity of \( r = 2 \); that is, when each UE can be served by a distinct pair of ENs, or for an arbitrary receiver connectivity when the user cache capacities are above a certain threshold. The derived NDT expression shows that, increasing the receiver connectivity for the same number of ENs and UEs will reduce the NDT for a specific cache capacity value, while the amount of reduction depends on the fronthaul value. This result is due to the fact that the size of the transmitted data through the fronthaul link for the network with higher connectivity is less compared to a network with more limited connectivity. We also consider the soft-transfer scheme which quantizes and transmits coded symbols to each of the ENs over the fronthaul links, in order to implement ZF over the edge network. We have derived the achievable NDT for this scheme as well. We have then compared the achievable NDT for these two schemes for different values of the fronthaul multiplexing gain. We have observed that MDS-IA scheme performs better when the fronthaul multiplexing gain is limited, while the soft-transfer scheme outperforms MDS-IA when the fronthaul multiplexing gain is high.

REFERENCES

[1] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, “Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution,” IEEE Commun. Mag., vol. 51, no. 4, pp. 142–149, Apr. 2013.
[2] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gunduz, “Wireless content caching for small cell and D2D networks,” IEEE J. Sel. Areas Commun., vol. 34, no. 5, pp. 1222–1234, May 2016.
[3] E. Bastug, M. Bennis, and M. Debbah, “Social and spatial proactive caching for mobile data offloading,” in 2014 IEEE International Conference on Communications Workshops (ICC), June 2014, pp. 581–586.
[4] P. Blasco and D. Gunduz, “Multi-armed bandit optimization of cache content in wireless infostation networks,” in Proc. IEEE Int’l Symp. on Inform. Theory (ISIT), Honolulu, HI, Jun. 2014, pp. 51–55.
[5] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” IEEE Trans. on Inform. Theory, vol. 60, no. 5, pp. 2856–2867, May 2014.
[6] ——, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” IEEE/ACM Trans. Netw., vol. 23, no. 4, August 2015.
[7] M. Amiri, Q. Yang, and D. Gunduz, “Decentralized caching and coded delivery with distinct cache capacities,” IEEE Trans. on Comms., vol. 65, no. 11, pp. 4657–4669, Nov 2017.
[8] M. M. Amiri and D. Gunduz, “Cache-aided data delivery over erasure broadcast channels,” in 2017 IEEE Int’l Conf. on Comms. (ICC), May 2017, pp. 1–6.
[9] S. S. Bidokhti, M. Wigger, and R. Timo, “Noisy broadcast networks with receiver caching,” IEEE Trans. on Inform. Theory, pp. 1–21, 2018.
[10] M. A. Maddah-Ali and U. Niesen, “Cache-aided interference channels,” in IEEE Int’l Symp. on Inform. Theory, June 2015, pp. 809–813.
[11] F. Xu, M. Tao, and K. Liu, “Fundamental tradeoff between storage and latency in cache-aided wireless interference networks,” *IEEE Trans. on Inform Theory*, vol. PP, no. 99, pp. 1–28, 2017.

[12] J. P. Roig, D. Gunduz, and F. Tosato, “ Interference networks with caches at both ends,” *IEEE Int’l Conf. on Comms.*, May 2017.

[13] A. Sengupta, R. Tandon, and O. Simeone, “Fog-aided wireless networks for content delivery: Fundamental latency tradeoffs,” *IEEE Trans. on Inform. Theory*, vol. 63, no. 10, pp. 6650–6678, Oct 2017.

[14] J. Koh, O. Simeone, R. Tandon, and J. Kang, “Cloud-aided edge caching with wireless multicast fronthauling in fog radio access networks,” in *IEEE Wireless Comms. and Networking Conf.*, Mar. 2017, pp. 1–6.

[15] A. M. Girgis, O. Ercetin, M. Nafie, and T. ElBatt, “Decentralized coded caching in wireless networks: Trade-off between storage and latency,” in *IEEE Int’l Symp. on Inform. Theory*, Jun. 2017, pp. 2443–2447.

[16] J. S. P. Roig, F. Tosato, and D. Gunduz, “Storage-latency trade-off in cache-aided fog radio access networks,” in *2018 IEEE Int’l Conf. on Comms. (ICC)*, May 2018, pp. 1–6.

[17] S. M. Azimi, O. Simeone, and R. Tandon, “Content delivery in fog-aided small-cell systems with offline and online caching: An information theoretic analysis,” *Entropy*, vol. 19, no. 7, 2017.

[18] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, “Fundamental limits of cache-aided interference management,” *IEEE Trans. on Inform. Theory*, vol. 63, no. 5, pp. 3092–3107, May 2017.

[19] M. Ji, M. F. Wong, A. M. Tulino, J. Llorca, G. Caire, M. Effros, and M. Langberg, “On the fundamental limits of caching in combination networks,” in *IEEE Int’l Workshop on Signal Proc. Advances in Wireless Comms.*, Jun. 2015, pp. 695–699.

[20] L. Tang and A. Ramamoorthy, “Coded caching for networks with the resolvability property,” in *IEEE Int’l Symp. on Inform. Theory*, July 2016, pp. 420–424.

[21] A. A. Zewail and A. Yener, “Coded caching for combination networks with cache-aided relays,” in *IEEE Int’l Symp. on Inform. Theory*, Jun. 2017, pp. 2433–2437.

[22] F. Xu and M. Tao, “Cache-aided interference management in partially connected wireless networks,” *ArXiv e-prints*, Aug. 2017.

[23] A. Roushdy, A. S. Motahari, M. Nafie, and D. Gunduz, “Cache-aided fog radio access networks with partial connectivity,” in *IEEE Wireless Comms. and Networking Conf.*, April 2018, pp. 1–6.

[24] J. Zhang and P. Elia, “Fundamental limits of cache-aided wireless bc: Interplay of coded-caching and csit feedback,” *IEEE Trans. on Inform. Theory*, vol. 63, no. 5, pp. 3142–3160, May 2017.

[25] S. Lin and D. J. Costello, *Error Control Coding. Second Edition*, 2004.

[26] A. S. Motahari, S. Oveis-Gharan, M. A. Maddah-Ali, and A. K. Khandani, “Real interference alignment: Exploiting the potential of single antenna systems,” *IEEE Trans. on Inform. Theory*, vol. 60, no. 8, pp. 4799–4810, Aug 2014.

[27] M. A. Maddah-Ali, “On the degrees of freedom of the compound miso broadcast channels with finite states,” in *IEEE Int’l Symp. on Informa. Theory*, Jun. 2010, pp. 2273–2277.

[28] O. Simeone, O. Somekh, H. V. Poor, and S. Shamai (Shitz), “Downlink multicell processing with limited-backhaul capacity,” *EURASIP Journal on Advances in Signal Proc.*., vol. 2009, no. 1, Jun 2009.

[29] J. Zhang, X. Lin, and X. Wang, “Coded caching under arbitrary popularity distributions,” in *Inf. Theory and Applications Workshop (ITA)*, Feb 2015, pp. 98–107.