2022 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-2159-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISIT50566.2022.9834669

Base Station-Assisted Cooperative Network Coding for Cellular Systems with Link Constraints

Suayb S. Arslan
MEF University
34396 Maslak, Istanbul, Turkey

Massoud Pourmandi
Boğaziçi University
34342 Bebek, Istanbul, Turkey

Elif Haytaoglu
Pamukkale University
20067 Denizli, Turkey

***Abstract*—We consider a novel distributed data storage/caching scenario in a cellular network, where multiple nodes may fail/depart simultaneously To meet reliability, we allow cooperative regeneration of lost nodes with the help of base stations allocated in a set of hierarchical layers[1]. Due to this layered structure, a symbol download from each base station has a different cost, while the link capacities between the nodes of the cellular system and the base stations are also constrained. Under such a setting, we formulate the fundamental trade-off with closed form expressions between repair bandwidth cost and the storage space per node. Particularly, the minimum storage as well as bandwidth cost points are formulated. Finally, we provide an explicit optimal code construction for the minimum storage regeneration point for a special set of system parameters.**

I. Introduction

In cellular systems, the data is usually cached in erasure coded form within the cell and in order to maintain the target reliability, we need to regenerate the unavailable content. A data of size F symbols is chopped into k fragments, erasure–encoded into n fragments and finally distributed across different n mobile nodes of a local cell where α symbols are stored by each node. A user can reconstruct the data by downloading all symbols that reside in any k local nodes. In the case of a node departure or permanent failure, either an instantaneous process repair the lost node contents or a *lazy* process may be adapted where no repair is initiated for a time duration before a *cooperative* regeneration phase, both by downloading γ symbols from other nodes. Dimakis *et al.* [1] considered a single node failure and formulated the fundamental trade-off between the data storage per node (α) versus the required bandwidth (γ). On the other hand, the cooperative regeneration is shown to reduce the required bandwidth dramatically [2], [3]. In a lazy cooperative repair, there are two phases. Supposing that there are t failed/departed nodes at the time of regeneration, empty nodes (newcomers) are regenerated by first contacting $d \leq n-t$ other live nodes to download $\beta \leq \alpha$ units of data to resurrect the lost data content (a total bandwidth of $\gamma = d\beta$). In the second phase, after each newcomer processes the downloaded data, the t newcomers exchange data among themselves, each by downloading β' units of data from the rest of the other $t-1$ newcomers.

In this paper, we consider a layered storage architecture, supported with multiple base stations (BSs) which cache the stored content in the nodes of the local cellular network. In particular, we formally introduce two novel concepts, namely *bandwidth cost* and *link capacity* in addition to cooperation. More specifically, we constrain the maximum link capacity limit to be $b_l\beta$ to be downloaded from the BS at the l-th layer. The bandwidth cost (γ_c) defines the price we pay per downloaded symbols from each BS.

Although there are similarities to previous rack-aware [4], [5] and clustered storage configurations [6] in which intra–rack/cluster and cross-rack/cluster bandwidths are introduced and tied together with a cost-like factor ϵ, the problem is more general in our case. More specifically, ϵ is introduced as a ratio of inter (cross) cluster bandwidth β_c and intra cluster repair bandwidth β_I per node [6]. On the other hand, we defined the cost notion more formally and assumed capacity-constrained communication channels. A later work [7] addressed the cooperative scenario for rack-aware regeneration in a limited context where cooperation is allowed only among nodes of different racks. Moreover, no BS or a back-up server is utilized in such a setting. Alternatively, the repair process can be handled in a centralized fashion to further reduce the bandwidth requirements. In [8], repairing multiple nodes in a centralized strategy is studied for the regenerating codes. Despite this strategy reduces the repair bandwidth in general, when the number of losses is less than k and more than one, it becomes impossible to achieve the minimum bandwidth for linear exact repair codes. Moreover such centralization may be the source of single point of failure and therefore constitutes a vulnerability. On the other hand, Calis *et al.* [9] considered a hierarchical storage with backup nodes in order to improve the overall communication cost. Furthermore, they compared distributed and centralized repair strategies [10]. However these studies did not take into account the cooperation. The consideration of BSs as helper nodes changes the problem dramatically and calls for distinct code constructions that would help us with cellular networks in which energy consumption and file popularity are important [11].

II. Motivation and System Model

A. Motivating Example

We consider a recurring example in the literature whereby out of four storage nodes, two nodes fail and we attempt cooperative regeneration of data in these nodes. In that setting, each node stores two packets where each packet is assumed to be 1MB ($F = 4$ MB). The first node stores A_1, A_2, second node stores B_1, B_2, third node contains A_1+B_1 and $2A_2+B_2$.

[1] *This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under grant number 119E235.

Finally, the fourth node stores $2A_1+B_1$ and A_2+B_2. Assume further that the second and fourth nodes fail. To recover both nodes the classical way and simultaneously, they need to contact the remaining alive nodes and download four packets per lost node (4MB) for the regeneration process. However, studies such as [2] and [3] showed that cooperation can help reduce the amount of data exchange to 3MB per node.

Let us assume a two-layer scenario in which different BSs are involved in the recovery process. We partition packets to half packets (0.5MB) and assume BS communication to be $w_1 > 1$ more costly compared to local communication. Also, communication with the satellite/cloud layer costs $w_2 > w_1 > 1$ whereby BS is able to provide at most r fraction of local downloaded data (β) and satellite/cloud layer stores the entire file. This constraint on the link capacity implies that the number of bytes that can be communicated with higher layer nodes is a linear function of that of local nodes (β)[2]. In that case, one packet transfer between local nodes and higher layers correspond to w_1 or w_2 packet transfers solely between local nodes subject to this limitation.

We provide an example scenario in Fig. 1 in which half packets for A_1 namely, A_1^1 and A_1^2 and for A_2 namely, A_2^1 and A_2^2 are summed before transmission to newcomers. The top newcomer initially resurrect $B_1^1 + B_1^2$ and downloads from the other newcomer $B_2^1 + B_2^2$. Finally by downloading B_1^2 from the satellite/cloud layer and B_2^2 from the BS, it would be able to complete the repair process. Similarly, the bottom newcomer initially computes $A_2^1 + B_2^1 + A_2^2 + B_2^2$, and through downloading $2A_1^1 + B_1^1 + 2A_1^2 + B_1^2$ from the other newcomer and the half packets $2A_1^2 + B_1^2$ from the satellite/cloud layer and $A_2^2 + B_2^2$ from the BS, respectively, it would be able to complete the repair process. Overall a total bandwidth cost of $3 + w_1 + w_2$ is experienced for the exact repair. In Table I, we compare the bandwidth cost per repaired node using this coding scheme (CoopLayer) based on the example in [3] (CoopLocal) with $r = 1$. We have also included no cooperation (NoCoopLocal) and a protocol which only contacts higher layers and download the node content directly from them without local assistance (FullLayer). Note that w_1 and w_2 are not applicable to methods in [1], [3] and [8]. As can be seen, the coding scheme we present downloads 0.5MB from the BS and 0.5MB from the satellite/cloud layer corresponding to $r = 1$ fraction of the local download ($\beta = 0.5$MB) and performs best among other schemes except “Centralized", which is not an option for distributed system design due to single point failure. In this study, we argue that the presented coding scheme uses both layers and is optimal for $r = 1$ for the weight values of Scenario 1 given in Table I. In this case, the cost of CoopLayer is the sum of 1 unit cost from the helper nodes, 0.5 unit cost from the cooperation phase, 0.55 from the BS and 0.85 from the satellite layers. However it turns out that in Scenario 2, using different weight distribution, it is optimal to use only the BS in the repair process. In other words, using $r = 1$ and smaller packetizations, we can show that there exists an optimal coding that can achieve a bandwidth cost of as low as $43/15 \approx 2.867$ (corresponds to $8/3 \approx 2.66$ MBs).

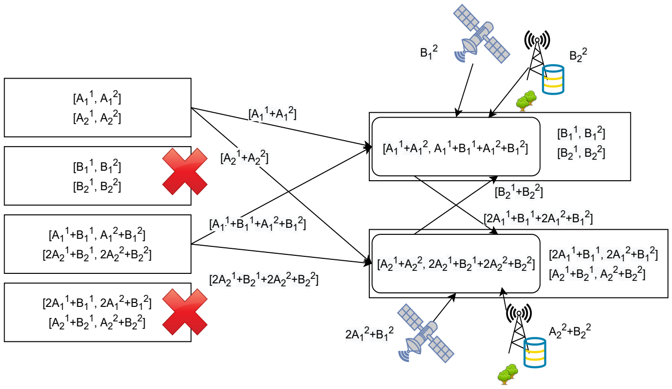
B_1^2
B_2^2
$[A_1^1, A_1^2]$
$[A_2^1, A_2^2]$
$[B_1^1, B_1^2]$
$[B_2^1, B_2^2]$
$[A_1^1+B_1^1, A_1^2+B_1^2]$
$[2A_2^1+B_2^1, 2A_2^2+B_2^2]$
$[2A_1^1+B_1^1, 2A_1^2+B_1^2]$
$[A_2^1+B_2^1, A_2^2+B_2^2]$
$[A_1^1+A_1^2]$
$[A_2^1+A_2^2]$
$[A_1^1+B_1^1+A_1^2+B_1^2]$
$[2A_2^1+B_2^1+2A_2^2+B_2^2]$
$[A_1^1+A_1^2, A_1^1+B_1^1+A_1^2+B_1^2]$
$[B_1^1, B_1^2]$
$[B_2^1, B_2^2]$
$[B_2^1+B_2^2]$
$[2A_1^1+B_1^1+2A_1^2+B_1^2]$
$[A_2^1+A_2^2, 2A_2^1+B_2^1+2A_2^2+B_2^2]$
$[2A_1^1+B_1^1, 2A_1^2+B_1^2]$
$[A_2^1+B_2^1, A_2^2+B_2^2]$
$2A_1^2+B_1^2$
$A_2^2+B_2^2$

Fig. 1. Cooperative regeneration of multiple failed nodes with the help of a BS and a satellite/cloud layer.

B. System Model and Information Flow Diagram

We consider a local cell with n storage nodes and multiple upper layers serving as back-up cluster nodes, each with size n_l with $l = 1, .., M$, such that communicating a symbol with one of the servers at the l-th layer are assigned a cost factor w_l. Higher layers are assumed to be in possession of the data being repaired at the expense of a higher download cost. For simplicity, we shall assume $n_l = 1$ and that clusters are stacked according to non-descending costs, therefore cluster with label 0 would correspond to local cluster and cluster with label M is the cluster that has the highest cost factor. We further assume each link to have limited capacity $b_l \beta$ and hence the number of symbols that can be downloaded from the l-th layer BS (β_l'') is constrained i.e., $\beta_l'' \leq b_l \beta$. This could be due to scarcity of resources or service level agreements.

Similar to previous analyses, in our study we will resort to information flow graphs to characterize the trade-off region, which are graphical representations of a network in which some information sources are multicast to a set of destination sinks through intermediate nodes. An information flow graph can be described by three types of nodes namely *source nodes*, *intermediate nodes* and *sink nodes*. In general, for the case of single source multiple sink nodes problem, achievable capacity of the network is characterized through max-flow min-cut theorem [12], [1]. In general case of multiple source nodes, achievable capacity is still an open problem. As a pioneering work, Alswede *et al.* [12] characterized the min-cut max-flow

	β	w_1	w_2	NoCoopLocal [1]	CoopLocal [3]	CoopLayer	FullLayer	Cenralized [8]
Scenario 1	0.5 (MB)	1.1	1.7	4	3	2.9	3.1	2
Scenario 2	2/3 (MB)	1.3	2.1	4	3	2.867	3.667	2

TABLE I
BANDWIDTH COST PER NEWCOMER FOR COOPERATIVE, LOCAL AND BS-ASSISTED COOPERATIVE REGENERATION SCHEMES ($r = 1$).

[2] This constraint will be a parameter of our work later on.

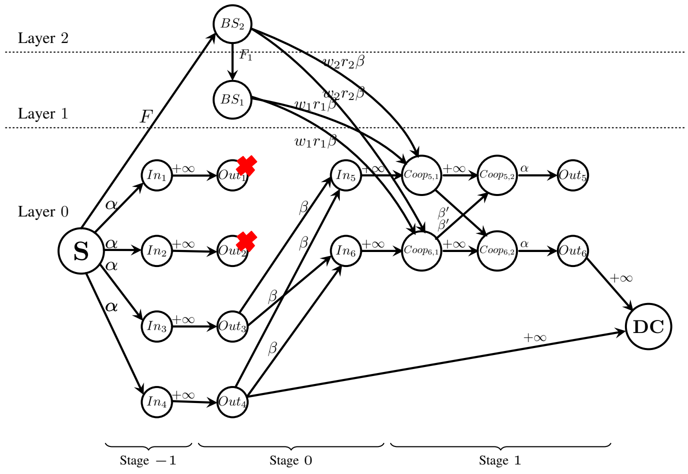

Fig. 2. Information flow diagram for $M = 2$, $n_1 = n_2 = 1$. For short-hand notation, only subscripts are used. For instance, $x^0_{In_1}, x^1_{Coop_{5,1}}$ are denoted by $In_1, Coop_{5,1}$. A fraction of file F, namely F_1 is stored in BS_1.

with network coding and proved that linear network codes are sufficient to achieve the min-cut bound in a multicast problem with a single source. In the context of a distributed storage system, a directed acyclic graph called an *information flow graph* $\mathcal{G}$ is used. As mentioned before, $\mathcal{G}$ consists of a source $\mathbf{S}$, multiple intermediary nodes, and one or more data collectors ($\mathbf{DC}$), which only contact any k nodes for the full reconstruction of the stored file in the network.

We characterize our proposed information flow graph with $(n, k, d, t, \alpha, \gamma, \{b_l\}, \{w_l\})$ tuple using M distinct layers, unlike in [3]. An example graph is illustrated in Fig. 2 for $M = 2$ case. Besides, for a given code parameters $(n, k, d, t, \alpha, \gamma)$, we denote all possible graphs with $\mathcal{G}(n, k, d, t; \alpha, \beta, \beta', \{\beta_l''\}_{l=1}^M)$. Data file $\mathcal{F}$ of size F is represented by the source node $\mathbf{S}$ at stage $s = -1$. Initial nodes are represented by pairs of vertices $(x^i_{In_j}, x^i_{Out_j})$ at stage $s = 0$, where i and j are the time index and device label introduced at time index i, respectively. Secondary nodes are represented by tuple $(x^i_{In_j}, x^i_{Coop_{j,1}}, x^i_{Coop_{j,2}}, x^i_{Out_j})$ and data collectors are represented by single node both with stage $s \geq 1$.

In our repair configuration, the initial data repair stage involves contacting d other local live nodes in the same cell and downloading β symbols from each through edges $x^k_{Out_j} \to x^i_{In_{j'}}$ for $k < i$ and $j \neq j'$. In the next phase, the newcomer contacts M above layers (since $n_l = 1$, we contact a server in each layer) and download β_l'' symbols from the l-th layer whereby each layer only stores $F_l \geq b_l\beta$ symbols for some $b_l \in \mathbb{R}$ and j-th node in l-th layer is associated with cost $w_l \geq 1$ per symbol download. In the final phase, all newcomers undergo a joint local cooperation by downloading β' symbols from at most $t - 1$ other failed nodes within the same local cell through the edges $x^i_{Coop_{j,1}} \to x^i_{Coop_{j',2}}$ for $j \neq j'$. The newcomer stores α symbols, and this is represented by a directed edge from $x^i_{Coop_{j,2}}$ to $x^i_{Out_j}$ with capacity α. To successfully reconstruct the file, a data collector is connected to k alive "out" nodes with distinct indices, but not necessarily from the same stage, by k infinite-capacity edges. We finally realize that most of the previous information flow diagram descriptions would be a special case of this general description.

C. A Fundamental Trade-off

Without loss of generality, we shall assume each layer contains a single base station. Moreover, similar to previous works [13], we introduce auxiliary variables r_l to be able to express β_l'' in terms of β i.e., $\beta_l'' = r_l\beta$. We characterize the link constraints by bounding the r_l i.e. $0 \leq r_l \leq b_l$ for some fixed $b_l \in \mathbb{R}$. Based on the regeneration description, the total repair bandwidth cost per failed node (γ_c) is obtained as follows,

$$\gamma_c(\mathbf{s}) = d\beta + (t-1)\beta^{'} + \sum_{l=1}^{M} s_l w_l r_l \beta \tag{1}$$

where for some $\rho \in \{0, 1, \ldots, M\}$ and entries s_l, which defines the number of used BSs with

$$\mathbf{s}_\rho = \left\{ (\underbrace{1, ..., 1}_{\rho}, \underbrace{0, ..., 0}_{M-\rho}) : \sum_{l=1}^{M} s_l = \rho \right\}. \tag{2}$$

The minimum cut of the flow diagram G induces the following constraint on the file size

$$\min_{r_l, \mathbf{u} \in P} \left\{ u_0\alpha + \sum_{j=1}^{g} u_j (d' - \sum_{i=0}^{j-1} u_i)\beta + u_j\,(t - u_j)\,\beta' \right\} \geq F \tag{3}$$

where $P = \{\mathbf{u} = (u_i)_{0 \leq i < g} : 1 \leq u_i \leq t$ and $\sum_i^g u_i = k\}$ and $d' = d + \sum_{l=1}^{M} s_l r_l$. Here, u_i is the number of nodes contacted in each repair group of size t during the repair and g is the number of repairing stages. Note that unlike d, d' is not necessarily an integer. Therefore, slightly modifying the steps given in [3] we can obtain the file size constraint in (3), which can further be expressed for $g = 0, ..., k$ as,

$$F \leq \begin{cases} \Phi_{g,\mathbf{s}_\rho} + g(\beta/2 - \beta') & \text{if } \beta \geq 2\beta', \\ \Phi_{g,\mathbf{s}_\rho} + \psi_{g,t}(\beta/2 - \beta') & \text{otherwise} \end{cases} \tag{4}$$

where $\Phi_{g,\mathbf{s}} = \alpha(k-g) + g\beta(d' - k + \frac{g}{2}) + \beta' g t$ used as a shorthand notation and $\psi_{g,t} = \lfloor g/t \rfloor t^2 + (g - \lfloor g/t \rfloor t)^2$.

For a given file size F and system parameters $\alpha, d, k, t, \mathbf{w} = (w_l)_{1 \leq l \leq M}, \mathbf{b} = (b_l)_{1 \leq l \leq M}$, the bandwidth cost-storage trade-off is the solution to the following constrained optimization problem

$$\min_{\mathbf{s}, r_1, \ldots, r_M, \beta, \beta'} \left\{ d\beta + (t-1)\beta' + \sum_{l=1}^{M} s_l w_l r_l \beta \right\} \tag{5}$$

subject to the conditional constraint in (4) and

$$0 \leq \beta, \beta' \leq \frac{F}{d} \tag{6}$$

$$b_l \in \mathbb{R}, r_l \in [0, b_l], \quad l = 1, ..., M \tag{7}$$

$$\mathbf{s}_\rho = [s_1, ..., s_M], s_l \in \{0, 1\} \tag{8}$$

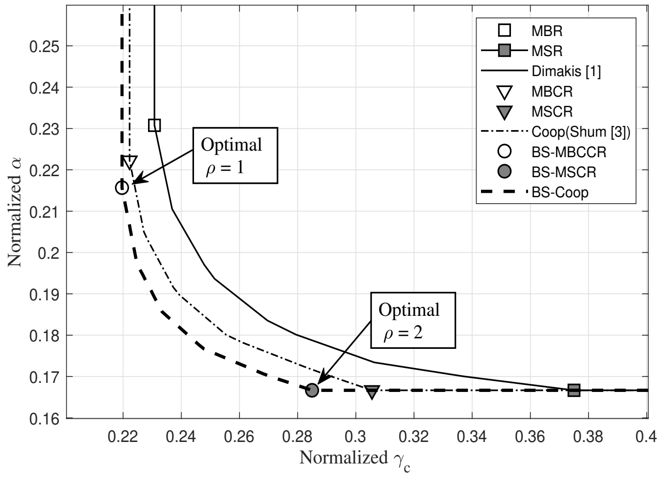

Fig. 3. Storage versus repair bandwidth cost trade off for $k = 6$, $d = 9$, $t = 3$, $\mathbf{b} = [1, 0.75, 0.5, 0.25]$ and $\mathbf{w} = [1.2, 1.4, 1.8, 1.84]$.

Due to space constraints, we present a solution of the optimization problem using a few selected parameters in Fig. 3. As can be seen the new trade-off curve, which uses different number of BSs at different operating points, allows better achievable region as compared to two major past works.

III. BS-MSCR AND BS-MBCCR OPERATING POINTS

Two operating points of this trade-off curve are of special importance namely, minimum storage and bandwidth cost operating points.

Theorem 3.1. *For a given set values of $r_1, ..., r_M$ in a BS-assisted cooperative repair scenario, the minimum storage and minimum repair bandwidth cost regeneration, namely BS-MSCR and BS-MBCCR points on the tradeoff curve can be characterized as*

$$(\gamma_{\textit{BS-MSCR}}, \alpha_{\textit{BS-MSCR}}) = \left(\frac{F(d+\sum\limits_{l=1}^{M} s_i w_i r_i + t - 1)}{k(d+\sum\limits_{l=1}^{M} s_i r_i + t - k)}, \frac{F}{k}\right) \tag{9}$$

$$(\gamma_{\textit{BS-MBCCR}}, \alpha_{\textit{BS-MBCCR}}) =$$
$$\left(\frac{F(2(d+\sum\limits_{l=1}^{M} s_i w_i r_i) + t - 1)}{k(2(d+\sum\limits_{l=1}^{M} s_i r_i) + t - k)}, \frac{F(2(d+\sum\limits_{l=1}^{M} s_i r_i) + t - 1)}{k(2(d+\sum\limits_{l=1}^{M} s_i r_i) + t - k)}\right) \tag{10}$$

Proof. At the BS-MBCCR operating point, similar to the reasoning given in [3], for a given set of values of $r_1, ..., r_M$, the minimum repair bandwidth cost is the intersection between $\beta = 2\beta'$ and $\beta - \beta'$ planes corresponding to the constraints given in (4) for $g = k$ and $\alpha \geq d\beta + (t-1)\beta' + \sum\limits_{l=1}^{M} s_l r_l$. To find the intersection, we substitute $\beta = 2\beta'$ in the $\beta - \beta'$ plane to get

$$\alpha(k-k) + k\beta\left(d + \sum_{l=1}^{M} s_l r_l - k + \frac{k+1}{2}\right) + \frac{k\beta}{2}(t-1) = F \tag{11}$$

which subsequently leads to

$$\beta = \frac{2F}{k\left(2(d+\sum\limits_{l=1}^{M} s_l r_l) + t - k\right)}. \tag{12}$$

Thus, by substituting β and $\beta' = \beta/2$ in the expression of $\gamma_c(\mathbf{s})$, the result follows. As for the BS-MSCR operating point, let $g = lt + q$, where l is an integer and $0 \leq q \leq t$. Note that $l = 0$ corresponds to the minimum storage point i.e., $\alpha = \frac{F}{k}$. Taking into this consideration, for a given set of values of $r_1, ..., r_M$, the intersection between $\beta = \beta'$ with $\beta - \beta'$ plane subject to the constraint in (4) gives us the BS-MSCR point. □

Note that in order to calculate BS-MSCR and BS-MBCCR operating points, we still need to identify the set of values $\{r_l\}$. Next theorem demonstrates that minimum bandwidth cost is indeed attained at the upper bounds.

Theorem 3.2. *For the given values of ρ, $\mathbf{s}_\rho = [\underbrace{1, \ldots, 1}_{\rho}, \underbrace{0, \ldots, 0}_{M-\rho}]$ and $0 \leq r_l \leq b_l$ for $l = 1, ..., M$, satisfying*

$$\sum_{l=1}^{\rho} w_l r_l \leq \bar{w}_t \sum_{l=1}^{\rho} r_l \tag{13}$$

with $\bar{w}_t \in \left\{\frac{d+t-1}{d+t-k}, \frac{2d+t-1}{2d+t-k}\right\}$, repair bandwidth cost $\gamma_c \in \{\gamma_{\textit{BS-MSCR}}, \gamma_{\textit{BS-MBCCR}}\}$ is minimized at the upper bounds $[b_1, ..., b_\rho]$.

Proof. In BS-MBCCR operating point, let us assume that we happen to use ρ number of base stations associated with the cost factors $\{w_1, ..., w_\rho\}$. In other words,$\{r_1, ..., r_\rho\}$ are non-zero values satisfying the condition

$$\sum_{l=1}^{\rho} w_l r_l \leq \frac{2d+t-1}{2d-k+t} \sum_{l=1}^{\rho} r_l. \tag{14}$$

Note that in order for BSs to reduce the overall bandwidth cost, we would need to satisfy $\gamma_c(\mathbf{s}_\rho) \leq \gamma_c(\mathbf{0})$ which leads to equation (14). Here $\mathbf{0}$ denotes the all-zero vector. Accordingly, inequality (14) can be rewritten as follows:

$$\sum_{l=1}^{\rho} w_l r_l = \tau \bar{w}_t \sum_{l=1}^{\rho} r_l \tag{15}$$

for some τ satisfying $0 \leq \tau \leq 1$. If we reformulate $\gamma_c(\mathbf{s}_\rho)$ in terms of ρ, repair bandwidth cost can be rewritten as,

$$\gamma_c(\mathbf{s}_\rho) = \frac{\bar{w}_t}{k} \frac{2d - k + t + 2\tau \sum\limits_{l=1}^{\rho} r_l}{2d - k + t + 2\sum\limits_{i=1}^{\rho} r_i}. \tag{16}$$

Algorithm 1 Optimal Number of BSs (ρ_{min})

```
function OptBSNumCal($k, d, t, M, \mathbf{b}, \mathbf{w}, p_t$)
    $\rho_{min} \leftarrow 0$
    for $i = 1 : M$ do
        $\bar{d} \leftarrow d + \sum_{l=1}^{i-1} w_l b_l$ , $\bar{b} \leftarrow d + \sum_{l=1}^{i-1} b_l$    ▷ $\mathbf{b} = \{b_l\}$
        $\bar{w}_t \leftarrow p_t \left(\frac{\bar{d}+t-1}{\bar{b}+t-k}\right) + (1-p_t)\left(\frac{2\bar{d}+t-1}{2\bar{b}+t-k}\right)$ ▷ $p_t \in \{0,1\}$
        if $w_i > \bar{w}_t$ then    ▷ $\mathbf{w} = \{w_l\}$
            break;
        end if
        $\rho_{min} \leftarrow \rho_{min} + 1$
    end for
    return $\rho_{min}$
```

Note that equation (16) is inversely related to $\sum_{l=1}^{\rho} r_l$ (it can be easily shown by taking the derivative). In other words, in order to minimize $\gamma_c(\mathbf{s}_\rho)$, it is sufficient to maximize the term $\sum_{l=1}^{\rho} r_l$. A similar proof following the same line of reasoning can be given for BS-MSCR point. □

The implication of the result of Theorem 3.2 is that we can simply obtain BS-MSCR and BS-MBCCR points by replacing r_i with b_i as,

$$(\gamma_{\text{BS-MSCR}}, \alpha_{\text{BS-MSCR}}) = \left(\frac{F(d + \sum_{l=1}^{\rho_{\text{BS-MSCR}}} w_i b_i + t - 1)}{k(d + \sum_{l=1}^{\rho_{\text{BS-MSCR}}} b_i + t - k)}, \frac{F}{k} \right) \quad (17)$$

$$(\gamma_{\text{BS-MBCCR}}, \alpha_{\text{BS-MBCCR}}) = \left(\frac{F(2(d + \sum_{l=1}^{\rho_{\text{BS-MBCCR}}} w_i b_i) + t - 1)}{k(2(d + \sum_{l=1}^{\rho_{\text{BS-MBCCR}}} b_i) + t - k)}, \frac{F(2(d + \sum_{l=1}^{\rho_{\text{BS-MBCCR}}} b_i) + t - 1)}{k(2(d + \sum_{l=1}^{\rho_{\text{BS-MBCCR}}} b_i) + t - k)} \right) \quad (18)$$

In Algorithm 1, we provide a linear time search algorithm to find $\rho_{\text{BS-MSCR}}$ and $\rho_{\text{BS-MBCCR}}$ i.e., the optimal number of contributing BSs, ρ_{min} for both BS-MSCR and BS-MBCCR operating points. We use p_t to indicate whether the results corresponds to BS-MSCR ($p_t = 1$) or BS-MBCCR $p_t = 0$).

IV. An explicit construction for Exact BS-Assisted MSCR code

As mentioned in some of the previous studies [3], exact repair has a number of advantages, including lower maintenance when compared to functional repair. Exact MSR and MSCR code constructions for a given set of code parameters have already been demonstrated [14]. In this section, We propose a family of regeneration codes for BS-assisted cooperative repair with parameters ρ and $d = k \leq n-t$. We'll also assume $bl = 1$ for the sake of simplicity. We formulate our construction using multiple maximum distance separable (MDS) codes of length n with dimension k, consisting of symbols from $GF(p^q)$ with prime p and $n \leq p^q$ for a given positive integer q, similar to earlier work. The simplified architecture we give in this subsection is a special instance of the code presented in subsection II.A.

The same set of $k \times n$ generator matrices[3] $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n]$ are used in our construction methodology, where $\mathbf{g}_i$ represents the i-th column of $\mathbf{G}$. The file of size F is partitioned into $k(t+\rho)$ number of chunks whose symbols are selected from $GF(p^q)$. The chunks are restructured as $(t+\rho) \times k$ message matrices $\mathbf{M}$. Therefore, we multiply the matrices i.e., $\mathbf{MG}$ and for $j = 1, 2, \ldots, n$ we distribute the j-th column of the output to j-th node to store. Let us the notation $\mathbf{m}_1^T, \mathbf{m}_2^T, \ldots, \mathbf{m}_{t+\rho}^T$ to denote each row of $\mathbf{M}$. In this case, we would have k message symbols in each row put into a vector form $\mathbf{m}_i^T = [m_1 \ m_2 \ \ldots \ m_k]$, each $m_i \in GF(p^m)$ encoded into the codeword $\mathbf{m}_i^T \mathbf{G}$. In such a setting, j-th node would store $\mathbf{m}_i^T \mathbf{g}_j$, for $i = 1, 2, \ldots, t+\rho$.

Assuming nodes with indices $j_1, j_2, \ldots, j_t$ become permanently unavailable. In the initial phase of the regeneration, the l-th newcomer (j_l) connects to any $d = k$ remaining available nodes, say $\pi_l(1), \pi_l(2), \ldots, \pi_l(k)$ and downloads $\mathbf{m}_l^T \mathbf{g}_{\pi_l(1)}, \mathbf{m}_l^T \mathbf{g}_{\pi_l(2)}, \ldots, \mathbf{m}_l^T \mathbf{g}_{\pi_l(k)}$ to successfully compute/reconstruct $\mathbf{m}_l^T$. In the subsequent cooperative phase, newcomer indexed by j_l computes $\mathbf{m}_l^T \mathbf{g}_{j_h}$ to send to the newcomer indexed by j_h with $h \neq l$. In this cooperative phase, a total of $t-1$ chunks are exchanged by each newcomer. After the cooperative phase, l-th newcomer indexed by j_l has t chunks namely $\mathbf{m}_h^T \mathbf{g}_{j_l}$ for $h = 1, \ldots, t$. In the final regeneration phase, we download the remaining ρ chunks from the available ρ base stations each providing one chunk of information. We may notice that the last two phases of the regeneration are exchangeable. We also note that a chunk of information is of size $\frac{F}{k(t+\rho)}$ symbols and the total number of chunks downloaded for each newcomer is $d + \rho + t - 1$ achieving the minimum bandwidth possible.

V. Conclusions and Future work

For cellular networks, the cooperative node repair procedure of a distributed data storage application is studied while BSs can actively participate in the data regeneration process. We've included the concept of bandwidth cost and link capacities in our problem formulation since symbol download between local nodes and BSs can be costly and limited at times. Furthermore, we have defined appropriate file size limitations and numerically solved the resulting optimization problem. In addition, we have found closed form expressions at the minimum storage bandwidth operating points in our setting and conjecture that such expressions can be found for each operating point in the trade-off curve. Our results indicate that a better bandwidth cost-storage space trade-off may be possible in the presence of BSs. We also realize that the cost notion can be applied to storage as well and a natural extension to this work would be to characterize the bandwidth cost-storage cost trade-off. We leave different code constructions for general parameters and link capacity settings as a future work.

[3] Any $k \times k$ submatrix of $\mathbf{G}$ is invertible by construction. Thus, any data collector connecting a subset of k nodes would be able to reconstruct the file.

REFERENCES

[1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[2] A. Kermarrec, N. Le Scouarnec, and G. Straub, “Repairing multiple failures with coordinated and adaptive regenerating codes,” in *2011 International Symposium on Networking Coding*, 2011, pp. 1–6.

[3] K. W. Shum and Y. Hu, “Cooperative regenerating codes,” *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7229–7258, 2013.

[4] Z. Chen and A. Barg, “Explicit constructions of msr codes for clustered distributed storage: The rack-aware storage model,” *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 886–899, 2019.

[5] H. Hou, P. P. C. Lee, K. W. Shum, and Y. Hu, “Rack-aware regenerating codes for data centers,” *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 4730–4745, 2019.

[6] J. Sohn, B. Choi, S. W. Yoon, and J. Moon, “Capacity of clustered distributed storage,” *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 81–107, 2019.

[7] S. Gupta and V. Lalitha, “Rack-aware cooperative regenerating codes,” in *2020 International Symposium on Information Theory and Its Applications (ISITA)*. IEEE, 2020, pp. 264–268.

[8] M. Zorgui and Z. Wang, “Centralized multi-node repair regenerating codes,” *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4180–4206, 2019.

[9] G. Calis and O. O. Koyluoglu, “On the maintenance of distributed storage systems with backup node for repair,” in *2016 International Symposium on Information Theory and Its Applications (ISITA)*. IEEE, 2016, pp. 46–50.

[10] G. Calis, S. Shivaramaiah, O. O. Koyluoglu, and L. Lazos, “Repair strategies for mobile storage systems,” *IEEE Transactions on Cloud Computing*, 2019.

[11] J. Pääkkönen, C. Hollanti, and O. Tirkkonen, “Device-to-device data storage with regenerating codes,” in *International Workshop on Multiple Access Communications*. Springer, 2015, pp. 57–69.

[12] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[13] G. Calis and O. O. Koyluoglu, “On the maintenance of distributed storage systems with backup node for repair,” in *2016 International Symposium on Information Theory and Its Applications (ISITA)*, 2016, pp. 46–50.

[14] M. Ye and A. Barg, “Explicit constructions of high-rate mds array codes with optimal repair bandwidth,” *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.