Trade-off for Heterogeneous Distributed Storage Systems between Storage and Repair Cost

K. G. Benerjee and M. K. Gupta

aDhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India
e-mail: *mankg@computer.org

Received August 13, 2019; revised October 2, 2020; accepted December 30, 2020

Abstract—We consider heterogeneous distributed storage systems (DSSs) having flexible reconstruction degree, where each node in the system has nonuniform repair bandwidth and nonuniform storage capacity. In particular, a data collector can reconstruct the file using some \( k \) nodes in the system and, for a node failure, the system can be repaired by some set of active nodes. Using min-cut bound, we investigate the fundamental trade-off between storage and repair costs for our model of the heterogeneous DSS. Further, the problem is formulated as bi-objective optimization linear programming problem for various heterogeneous DSSs. For some DSSs, it is shown that the calculated min-cut bound is tight.

Key words: cloud storage, codes for distributed storage, heterogeneous distributed storage system, information flow, trade-off between storage cost and repair cost.

DOI: 10.1134/S0032946021010026

1. INTRODUCTION

Cloud storage is a distributed storage system (DSS) in which information is stored on distinct nodes as encoded packets in a redundant manner. One can retrieve the file by contacting certain nodes in the system. In the case of a node failure, it can be repaired using other nodes in the system. For such DSSs, one has to optimize various parameters in the system such as storage capacity, repair bandwidth, availability, reliability, security, and scalability. Such DSSs are used by many commercial systems like Facebook, Yahoo!, IBM, Amazon, and Microsoft Windows Azure system [1–4].

In homogeneous DSSs (where each node has same storage capacity and same repair degree) [5], encoded data packets of a file with size \( B \) are distributed among \( n \) nodes (each having storage capacity \( \alpha \)) such that connecting any \( k \) \((<n)\) nodes, one can retrieve the whole file. In the case of any arbitrary node failure, the system is repaired by downloading \( \beta \) packets from any \( d \) \((<n)\) active nodes, called helper nodes [5]. In these systems, one can provide reliability by simply replicating or encoding the message data packets. In the case of simple replication, storage minimization is inefficient. On the other hand, encoding of data packets using erasure MDS (maximum distance separable) codes leads to inefficiency for bandwidth minimization during node repair process. To optimize these conflicting parameters, in a seminal work [5] regenerating codes are introduced. In [6, 7], dependency between node storage capacity \( \alpha \) and repair bandwidth \( d\beta \) is analyzed by plotting trade-off curves for regenerating codes. All points on the trade-off curve can be obtained by linear network codes over finite fields [8,9]. In the trade-off curve, by minimizing both parameters in different order, minimum bandwidth regenerating (MBR) codes and minimum storage regenerating (MSR) codes are obtained [6]. The trade-off between storage and repair bandwidth for exact-repair is studied in [10]. In [11], a cut-capacity lower bound on repair bandwidth for a special
In the heterogeneous DSS each node has flexible storage capacity $\alpha_i$, $i = 1, 2, \ldots, n$, and repair bandwidth. In the system, the flexible reconstruction degree, for a data collector, is $k_t$, $t \in \{1, 2\}$. Repair degree for an arbitrary node failure $U_i$ is also nonuniform. A failed node $U_i$ is repaired by some $d_i^{(t)}$ nodes.

Flexible setting is calculated for a homogeneous DSS. In a nice survey [12], an overview of some existing results and repair models on DSSs are explored. Recently, in [13], the trade-off between storage capacity and repair bandwidth is investigated for exact repair linear regenerating codes for $k = d = n - 1$.

Heterogeneous DSSs are more close to real world scenarios, where characterization of all storage nodes in various aspects are not necessarily uniform due to geographical environment, storage devices cost, etc. Many such heterogeneous DSS have been studied recently [14–16]. In [17–19], storage allocation problem is investigated to maximize the probability of successful recovery. For heterogeneous DSS, repair cost can be reduced by allowing helper nodes to encode codewords of other nodes [20]. In [21–25], authors investigated the trade-off between storage capacity and repair bandwidth for the generalized regenerating codes and have shown that each point on the curve is achievable. In the generalized regenerating code, the set of all nodes is divided into two partitions such that every node in each partition has uniform parameters $(\alpha_i, d_i, \beta_i)$ for $i = 1, 2$. In [26], a capacity bound is calculated of a heterogeneous DSS having nonuniform repair bandwidth and constant repair degree. In [27], a capacity bound is calculated for heterogeneous DSSs with nonuniform repair bandwidth where node repair is done by some specific helper nodes. In [28], trade-off between system storage cost and system repair cost is investigated for heterogeneous DSSs with nonuniform storage and repair cost. In [29], selective regenerating codes are proposed and trade-off between storage per node and repair bandwidth is drawn for the regenerating codes, where selective regenerating codes are regenerating codes in which helper nodes of a failed node are chosen in such a smart way that it reduces repair bandwidth. In [30, 31], trade-off between node storage capacity and bandwidth is analyzed for exact repair. In [32], constructions for interior points of the normalized trade-off are given. In [33], bounds on exact repair for regenerating codes are improved. In [34], authors talked about improvement of regenerating codes by smart selection of helper nodes for an arbitrary node failure.

In this work, we consider a heterogeneous DSS, where a file of size $B$ is distributed among $n$ nodes each with different storage capacities. File reconstruction is done in a flexible manner where at any time instant, data collector can reconstruct the file by connecting some $k_t$ number of nodes for some $t$. Hence, for a given file, the reconstruction degree $k_t$ is flexible. On the other hand, in case
A model of such a heterogeneous DSS is shown in Fig. 1. A data collector reconstructs a distributed file by connecting $k_1$ nodes if the recovered packets are some functions of the lost packets, then the repair is exact repair. If the recovered packets are exact copies of lost packets, then it is called functional repair. A failed node can be done in two ways, exact repair and functional repair. If the recovered packets can be repaired exactly by downloading packets $x_1$, $x_2$, $x_3$, and $x_4$, then it is called exact repair. On the other hand, if the recovered packets are some functions of the lost packets, then the repair is functional repair. A model of such a heterogeneous DSS is shown in Fig. 1. A data collector reconstructs a distributed file by connecting $k_1$ nodes, $t \in \{1, 2\}$. In addition, a failed node $U_i$ is repaired by some $d_i^{(t)}$ nodes.

An example of such a heterogeneous DSS is considered in Fig. 2. In this system, a file $B$ is divided into 4 message information packets $x_1$, $x_2$, $x_3$, and $x_4$. The message information packets are encoded into 11 packets by taking linear combination of message information packets as $y_1 = x_1$, $y_2 = x_2$, $y_3 = x_3$, $y_4 = x_1 + x_2$, $y_5 = x_4$, $y_6 = x_1 + x_2$, $y_7 = x_1$, $y_8 = x_3$, $y_9 = x_2 + x_4$, $y_{10} = x_2$, and $y_{11} = x_1 + x_4$. The encoded packets $y_m$, $m = 1, 2, \ldots, 11$, are distributed on the 5 nodes such that packets $y_1$ and $y_2$ are stored on node $U_1$, packets $y_3$ and $y_4$ are distributed on node $U_2$, packets $y_5$ and $y_6$ are on node $U_3$, packets $y_7$, $y_8$, and $y_9$ are on node $U_4$, and the remaining two packets are on node $U_5$. Clearly, the node storage capacity is $\alpha_i = 2$ for $i = 1, 2, 3, 5$ and $\alpha_4 = 3$. In this example, if node $U_5$ fails, then it can be repaired by downloading packets $y_7$ and $y_9$ from node $U_4$. Since the recovered packets are functions of lost packets, it is functional repair. On the other hand, node $U_5$ can be repaired exactly by downloading packets $y_1$, $y_2$, and $y_5$ from nodes $U_1$ and $U_3$ and solving $y_{10} = y_2$ and $y_{11} = y_1 + y_5$.

An $(n, k, d)$ DSS is a storage system with $n$ nodes, reconstruction degree $k$ and repair degree $d$ such that

- each node contains a fraction of data file information,
- any data collector can reconstruct the complete file information by downloading packets from $k$ ($< n$) nodes, and
- a failed node is repaired by some $d$ helper nodes.

Fig. 2. A file with size 4 units ($= B$) is divided into 11 encoded packets over a field $\mathbb{F}_q$. These packets are distributed among 5 ($= n$) nodes in such a way that any data collector can download the whole file by contacting at most 3 ($= k$) nodes. In this heterogeneous DSS, $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) = (2, 2, 2, 3, 2)$. A failed node can be repaired by at most 2 ($= d$) nodes. Functional and exact repairs are shown for the node failure of $U_5$ with the help of surviving sets $S^{(2)}_5 = \{U_4\}$ and $S^{(3)}_5 = \{U_1, U_3\}$, where a surviving set is a set of some helper nodes. Surviving set $S^{(3)}_5$ is not considered in Table 1, since $S^{(1)}_5 \subseteq S^{(3)}_5$. of any node failure $U_i$, $i \in \{1, 2, \ldots, n\}$, it can be repaired by downloading packets from some $d_i^{(t)}$ nodes. Hence, the repair degree $d_i^{(t)}$ is also flexible with respect to the number of nodes. Repair of a failed node can be done in two ways, exact repair and functional repair. If the recovered packets in repair process are exact copies of lost packets, then it is called exact repair. On the other hand, if the recovered packets are some functions of the lost packets, then the repair is functional repair. A model of such a heterogeneous DSS is shown in Fig. 1. A data collector reconstructs a distributed file by connecting $k_t$ nodes, $t \in \{1, 2\}$. In addition, a failed node $U_i$ is repaired by some $d_i^{(t)}$ nodes.
Contribution. In this paper, we have calculated min-cut bound for the considered heterogeneous DSS. For such a heterogeneous DSS, we have established a bi-objective optimization linear programming problem subject to min-cut bound. The solutions of the LP problem are plotted as a trade-off curve between system storage and repair cost. In a heterogeneous DSS, system storage cost and system repair cost are average costs to store and repair unit information data on a node respectively. We have plotted some trade-off curve and compared it with trade-off for heterogeneous DSSs as considered in [28] and homogeneous DSSs as investigated in [7]. Some specific cases are investigated for the established bi-objective optimization problem. The computational complexity to calculate parameters, for the fundamental bound achieving codes, is very high. So, we consider a special case by choosing constant repair traffic and reconstruction degree. Further we calculate a relation on parameters which achieve the bounds. For the particular parameters, the optimal codes (the fundamental bound achieving codes) are constructed using the graph.

Organization. The paper is organized as follows. Section 2 collects the required preliminary concepts and describes our model. Section 3 investigates the min-cut bound for our model. Under the constraints of the min-cut bound, we also establish a bi-objective linear optimization problem to plot the trade-off curve between storage and repair costs per node. In Section 4, for a specific heterogeneous DSS, the fundamental bound is analyzed. Finally Section 5 concludes the paper with general remarks.

2. PRELIMINARIES

In this paper, we focus our attention to the heterogeneous DSS with nonuniform parameters. For any heterogeneous DSS, let us define a set $A_i = \{U_i : i \in I \subset \{1, 2, \ldots, n\}\}$ as a collection of the nodes having sufficient packets to retrieve the complete file. The set $A_i$ is called a reconstruction set and $|I| = |A_i| = k_i$. If the set $A = \{A_1, A_2, \ldots, A_i, \ldots\}$ is a collection of all distinct reconstruction sets for any heterogeneous DSS, then the set $A$ will be finite. Hence, $\exists \omega \in \mathbb{N}$ such that $|A| = \omega$. Now, for the heterogeneous DSS, the reconstruction degree $k_i$ is the size of the reconstruction set $A_i$, i.e., $k_i = |A_i|$, and $k = \max\{k_i : i \in \{1, 2, \ldots, \omega\}\}$. For example, as given in Fig. 2, $A = \{A_i : i = 1, 2, \ldots, 7\}$, where $A_1 = \{U_1, U_2, U_3\}$, $A_2 = \{U_1, U_2, U_5\}$, $A_3 = \{U_1, U_4\}$, $A_4 = \{U_2, U_4\}$, $A_5 = \{U_2, U_5\}$, $A_6 = \{U_3, U_4\}$, and $A_7 = \{U_4, U_5\}$. Thus, $(k_1, k_2, k_3, k_4, k_5, k_6, k_7) = (3, 3, 2, 2, 2, 2, 2)$, and therefore $k = 3$.

In the heterogeneous DSS, if a node $U_i$, $i \in \{1, 2, \ldots, n\}$, fails, then some active nodes, called helper nodes, downloads required packets and generate a new node say $U_i'$. The new node $U_i'$ takes place of the failed node $U_i$, and the system is repaired. In particular, the set of those helper nodes is called a surviving set for the failed node $U_i$. Note that there may exist more than one surviving sets for any failed node. For a node $U_i$, let the number of distinct surviving sets be $N_i$. Now, a surviving set with index $\ell$ is denoted by $S_i^{(\ell)} = \{U_j : j \in J \subset \{1, 2, \ldots, n\} \setminus \{i\}\}$, where $\ell \in \{1, 2, \ldots, \tau_i\}$ [27]. If a failed node $U_i$ is repaired by nodes of the surviving set $S_i^{(\ell)}$, then the repair degree of the node $U_i$ is $d_i^{(\ell)} = |J| = |S_i^{(\ell)}|$. For the heterogeneous DSS considered in Fig. 2, surviving sets are listed in Table 1. In this example, one can see that if the node $U_5$ fails, then it can be repaired by connecting nodes $U_1$ and $U_3$ or nodes $U_1$ and $U_4$. Hence, surviving sets for the node $U_5$ are $S_5^{(1)} = \{U_1, U_3\}$ and $S_4^{(2)} = \{U_1, U_4\}$ as listed in Table 1. Also note that, in the table, we have chosen those surviving sets which are not a super set of any other surviving set for the same node failure. In particular, this condition ensures active participation of each node of any surviving set during the repair process.

For any failed node $U_i$, if the system is repaired by nodes of a specific surviving set $S_i^{(\ell)}$, then the number of information packets downloaded by node $U_j \in S_i^{(\ell)}$ will be given by $\beta(U_i, U_j, S_i^{(\ell)}) > 0$. Note that the positive value for $\beta(U_i, U_j, S_i^{(\ell)})$ ensures active participation of each helper node of
the surviving set $S_i^{(\ell)}$. For example, in Fig. 2, to repair node failure $U_4$, all packets from the node $U_2 \in S_i^{(2)}$ and the node $U_5 \in S_i^{(2)}$ are downloaded and, from the downloaded packets, the missing packets $y_7, y_8,$ and $y_9$ can be obtained as $y_7 = y_4 - y_{10}, y_8 = y_3,$ and $y_9 = y_{11} + 2 y_{10} - y_4$. Hence, $\beta(U_4, U_5, S_i^{(2)}) = 2$ and $\beta(U_4, U_2, S_i^{(2)}) = 2$.

Now, formally we define the following. For a failed node $U_i$ and surviving set $S_i^{(\ell)}$,

$$\beta_i^{(\ell)} = (\beta(U_i, U_{j_1}, S_i^{(\ell)}), \beta(U_i, U_{j_2}, S_i^{(\ell)}), \ldots, \beta(U_i, U_{j_m}, S_i^{(\ell)}))$$

is the vector of number of packets downloaded from helper nodes of the surviving set $S_i^{(\ell)} = \{U_{j_1}, U_{j_2}, \ldots, U_{j_m}\}$ of size $m$, i.e., $m = d_i^{(\ell)} = |S_i^{(\ell)}|$. For a failed node $U_i$, we define $\beta_i = (\beta_i^{(1)}, \beta_i^{(2)}, \ldots, \beta_i^{(\tau_i)})$ as the vector of all $\beta_i^{(\ell)}$, and therefore, $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$.

If a failed node $U_i, i \in \{1, 2, \ldots, n\}$, is repaired by nodes of a surviving set $S_i^{(\ell)}$, then the repair bandwidth (denoted by $\gamma(U_i, S_i^{(\ell)})$) for the node $U_i$ is the total number of packets downloaded by every nodes of the surviving set $S_i^{(\ell)}$, i.e.,

$$\gamma(U_i, S_i^{(\ell)}) = \sum_{j: U_j \in S_i^{(\ell)}} \beta(U_i, U_j, S_i^{(\ell)}).$$

For example, in Fig. 2, if node $U_4$ fails and is repaired by nodes of the surviving set $S_i^{(2)} = \{U_2, U_5\}$, then the repair bandwidth for the node $U_4$ is $\gamma(U_4, S_i^{(2)}) = \beta(U_4, U_2, S_i^{(2)}) + \beta(U_4, U_5, S_i^{(2)}) = 2 + 2 = 4$ units.

Remark 1. In this paper, single node failure is considered, because simultaneous multi-node failures can be assumed as a sequence of single node failure.

Now, a heterogeneous DSS can be formally defined as follows.

Definition 1. A heterogeneous DSS with parameters $(n, k, d, \alpha, \beta, B)$ is a storage system with $n$ nodes such that

PROBLEMS OF INFORMATION TRANSMISSION Vol. 57 No. 1 2021
- Each node $U_i$, $i = 1, 2, \ldots, n$, contains a fraction of data file information encoded into data of size $\alpha_i$;
- Any data collector can reconstruct the complete file information by downloading packets from some $k_t$ nodes ($k_t$ is the size of reconstruction set $A_t$ with the index $t \in \{1, 2, \ldots, \omega\}$); and
- A failed node $U_i$ is repaired by downloading $\sum_{U_j \in S_i^{(\ell)}} \beta(U_i, U_j, S_i^{(\ell)})$ packets from helper nodes from the surviving set $S_i^{(\ell)}$, $\ell \in \{1, 2, \ldots, \tau_i\}$; where

$$
\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n),
$$

$$
\mathbf{d} = (d_1, d_2, \ldots, d_n), \quad d_i = (|S_i^{(1)}|, |S_i^{(2)}|, \ldots, |S_i^{(\tau_i)}|),
$$

$$
\beta_i^{(\ell)} = (\beta(U_i, U_{j_1}, S_i^{(\ell)}), \beta(U_i, U_{j_2}, S_i^{(\ell)}), \ldots, \beta(U_i, U_{j_m}, S_i^{(\ell)})), \quad m = |S_i^{(\ell)}|,
$$

$$
\beta_i = (\beta_i^{(1)}, \beta_i^{(2)}, \ldots, \beta_i^{(\tau_i)}), \quad \beta = (\beta_1, \beta_2, \ldots, \beta_n).
$$

To plot the trade-off curve between storage capacity and repair bandwidth for any homogeneous DSS, in [7] an optimization problem is solved with a constraint of the min-cut bound between the parameters. The bound is calculated by analyzing information flow graph for the homogeneous DSS [7]. In the similar manner, one can obtain a min-cut bound and plot a trade-off curve for any heterogeneous DSS. In this paper, the min-cut bound is obtained using the information flow graph as described in [27, 28], where the information flow graph is an acyclic weighted directed graph $G = (V, E)$ with vertex set $V$ and edge set $E$.

For any heterogeneous DSS with a given reconstruction set $A_t = \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_k_t}\}$, the information flow graph $G$ is shown in Fig. 3, where all the nodes $U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_k_t}$ are considered to fail one by one in the same order and those failed nodes are repaired by surviving sets $S_i^{(\ell_1)}, S_i^{(\ell_2)}, \ldots, S_i^{(\ell_{\lambda_k_t})}$ for $\ell_i \in \{1, 2, \ldots, \tau_{\lambda_i}\}$. For any reconstruction set $A_t$, the graph is divided into $k_t + 3$ steps, starting from label $-1$ to label $k_t + 1$. In step label $-1$, the graph contains source node “s,” and in the step label $k_t + 1$, it contains data collector node “D.” A typical node $U_{\lambda_i}$, $i = 1, 2, \ldots, n$, in the heterogeneous DSS is mapped to a pair of vertices “In$_{\lambda_i}$” and “Out$_{\lambda_i}$” in $V$ such that both vertices are connected with an edge, i.e., $\{\text{In}_i, \text{Out}_i\}$ $\in$ $E$, where $\lambda_i$ are permuted indices on nodes. The storage capacity $\alpha_{\lambda_i}$ of node $U_{\lambda_i}$ is mapped with $w(\text{In}_i, \text{Out}_i)$, where $w(\text{In}_i, \text{Out}_i)$ is the weight associated with the edge $\{\text{In}_i, \text{Out}_i\}$ $\in$ $E$. In the graph $G$ as given in Fig. 3, at step label $0$ there are $2n$ vertices named $\text{In}_i$ and $\text{Out}_i$ associated with nodes $U_{\lambda_i}$, $i \in \{1, 2, \ldots, n\}$.

In the heterogeneous DSS, a failed node $U_{\lambda_i}$, $i \in \{1, 2, \ldots, n\}$, is repaired by generating a new node $U'_{\lambda_i}$. In the information flow graph, the node $U'_{\lambda_i}$ is mapped to a new pair of nodes $\text{In}'_{\lambda_i}$ and $\text{Out}'_{\lambda_i}$ such that $\{\text{In}'_{\lambda_i}, \text{Out}'_{\lambda_i}\}$ $\in$ $E$ with $w(\text{In}'_{\lambda_i}, \text{Out}'_{\lambda_i}) = \alpha_{\lambda_i}$. In each step label $j$, $j = 1, 2, \ldots, k_t$, the flow graph contains one pair of nodes $\text{In}'_{\lambda_j}$ and $\text{Out}'_{\lambda_j}$. As is shown in Fig. 3, in the heterogeneous DSS, the system is repaired for the node failure $U_{\lambda_i}$ by downloading $\beta(U_{\lambda_i}, U_{\mu_j}, S_{(\ell_k)}^{(\lambda_1)})$ amount of data from the node $U_{\mu_j} \in S_{(\ell_k)}^{(\lambda_1)} = \{U_{\mu_j} : j = 1, 2, \ldots, (\ell_k - 1)\}$, where $\mu_j$ is some permutation on indices of nodes. For a particular system repair, each downloading process maps by one distinct edge that joins one ‘Out’ node from some previous step label to the node $\text{In}'_{\lambda_j}$. Thus, if $\mu_j \in \{1, 2, \ldots, \lambda_i-1\}$, then the corresponding helper node has been repaired after failing earlier. Such downloading processes are represented by edges $(\text{Out}'_{\mu_j}, \text{In}'_{\lambda_j})$ $\in$ $E$, and these edges are represented as descending arrows in the central part of the information flow graph as given in Fig. 3. For the other case, if the node $\text{Out}'_{\mu_j}$ does not fall among nodes $\{\text{Out}'_{\lambda_1}, \ldots, \text{Out}'_{\lambda_{i-1}}\}$, then consider the node $\text{Out}_{\mu_j}$ from step label $0$ to join with the node $\text{In}'_{\lambda_j}$ (i.e., $(\text{Out}_{\mu_j}, \text{In}'_{\lambda_j})$ $\in$ $E$) with the weight $w(\text{Out}_{\mu_j}, \text{In}'_{\lambda_j}) = \beta(U_{\lambda_i}, U_{\mu_j}, S_{(\ell_k)}^{(\lambda_1)})$. These cases are represented by ascending arrows in the central part of the information flow graph as given in Fig. 3. In graph $G$, exactly one node failure is considered in each step label.
A data collector $D$ connects all $k_t$ nodes of the set $A_t = \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{k_t}}\}$. In the information flow graph as given in Fig. 3, the data collector $D$ connects nodes $Out'_{\lambda_j}$, $j = 1, 2, \ldots, k_t$, from step label 1 to step label $k_t$ and downloads a certain data file, then $(Out'_{\lambda_j}, D) \in E$ such that $w(Out'_{\lambda_j}, D) \rightarrow \infty$. For the heterogeneous DSS in Fig. 2, an example of the information flow graph is shown in Fig. 4. In particular, a data collector is connected with the nodes of $A_1 = \{U_1, U_2, U_3\}$. In the information flow graph, if the nodes fail in the order of $U_1$, $U_2$, and $U_3$, then it can be repaired by nodes of $S_1^{(1)}$, $S_2^{(1)}$, and $S_3^{(1)}$ respectively. Note that only one packet $y_1 + y_2$ (the sum of the stored packets on the node $U_1$) is downloaded from the node $U_1$ to repair the failed node $U_2$, so $\beta(U_2, U_1, S_2^{(1)}) = 1$.

In [7, 26–28], min-cut bound is calculated by analyzing flow passes through the source node $s$ to the data collector node $D$ across the information flow graph for any DSS. In the similar manner, the flow analysis is done for the heterogeneous DSS as considered in this paper. Hence, one can define the flow across the information flow graph as follows.

**Definition 2** (information flow). A function $f : E \rightarrow [0, \infty) \subset \mathbb{R}$ is called an information flow or simply a flow on an information flow graph $G = (V, E)$ if

1. (capacity constraint): For each edge $(x, y) \in E$, $f((x, y)) \leq c((x, y))$, where $c((x, y)) = w(x, y)$ and $c((x, y))$ is the capacity of the edge $(x, y)$, and
2. (flow conservation constraint): For each vertex $y \in V \setminus \{s, t\}$,

$$
\sum_{x : (x, y) \in E} f((x, y)) = \sum_{z : (y, z) \in E} f((y, z)).
$$

More details and examples on the flow function can be found in [35, 36].

For a given information flow graph $G = (V, E)$, the value of the flow delivered to a data collector $D$ is defined as the total amount of flow passes through the edges $(x, D) \in E$ for all possible $x \in V$. 

**PROBLEMS OF INFORMATION TRANSMISSION**  Vol. 57  No. 1  2021
For the heterogeneous DSS as considered in Fig. 2, an information flow graph is shown for a specific data collector that connects with nodes of the reconstruction set $A_1 = \{U_1, U_2, U_3\}$. The particular information flow graph is plotted for the surviving sequence $\langle S_1^{(1)}, S_2^{(1)}, S_3^{(1)} \rangle$, where, for any heterogeneous DSS, the surviving sequence is defined in Definition 4.

For networks, the maximum possible value of the flow delivered to $D$ is governed by the min-cut max-flow theorem [35–37]. According to the min-cut max-flow theorem, the maximum possible value of the flow, $\maxflow(s, D)$, passing from source $s$ to a specific data collector $D$ across the network is equal to the minimum of the cut-capacity $\minflow(s, D)$, where

$$\minflow(s, D) = \min_{\cut(X, \overline{X})} \text{cut-capacity}(X, \overline{X}).$$

Note that $\text{cut}(X, \overline{X})$ represents the set of all edges having one end vertex in the set $X$ and another end vertex in the set $\overline{X}$ such that removing all those edges will improve the number of components in the graph $G = (V, E)$. Here cut-capacity $(X, \overline{X})$ is the sum of capacities of all edges in $\text{cut}(X, \overline{X})$.

A specific data collector $D$ which connects nodes $U_{\lambda_1}$ of the set $A_t \in A$ has $|A_t|! \prod_{i=1}^{\lambda_1} \tau_{\lambda_i}$ distinct information flow graphs. For every information flow graph $G = (V, E)$, $D$ can retrieve the whole file $B$, so

$$B \leq \min_{G} \maxflow(s, D).$$

From the min-cut max-flow theorem, the total flow through a cut is equal to the capacity of the cut. Thus,

$$B \leq \min_{G} \min \text{cut-capacity}(s, D). \quad (1)$$

In [7], for an information flow graph, flow analysis is done by taking a topological order of failed nodes that are also connected with data collector to retrieve the whole file. In this paper, we are defining some sequences of nodes and corresponding surviving sets for our model to analyze the flow. The definitions are as follows.
Definition 3 (node sequence set). A set of all possible sequences of nodes in a reconstruction set $\mathcal{A}_t \in \mathcal{A}$ is called a node sequence set and denoted by $\mathcal{A}(\mathcal{A}_t) = \{(U_{\lambda_i})_{i=1}^{[\mathcal{A}_t]} : U_{\lambda_i} \in \mathcal{A}_t\}$, where $(U_{\lambda_i})_{i=1}^{[\mathcal{A}_t]}$ represents a sequence of distinct nodes of a set $\mathcal{A}_t \in \mathcal{A}$. Clearly, $|\mathcal{A}(\mathcal{A}_t)| = |\mathcal{A}_t|!$.

For example, in Fig. 2, $\mathcal{A}(\mathcal{A}_3) = \{(U_1, U_4), (U_4, U_1)\}$, where $\mathcal{A}_3 = \{U_1, U_4\}$.

Definition 4 (surviving sequence). For any reconstruction set $\mathcal{A}_t \in \mathcal{A}$, one can define sequences of surviving sets $S_{\lambda_i}^{(\ell)}$, $i = 1, 2, \ldots, |\mathcal{A}_t|$, $\ell = 1, 2, \ldots, \tau_{\lambda_i}$, such that $U_{\lambda_i} \in \mathcal{A}_t$. A surviving sequence associated with a node sequence $(U_{\lambda_i})_{i=1}^{[\mathcal{A}_t]} \in \mathcal{A}(\mathcal{A}_t)$ can be denoted by $(S_{\lambda_i}^{(\ell)})_{i=1}^{[\mathcal{A}_t]}$.

In Fig. 2, $(S_1^{(3)}, S_4^{(2)})$ is one of the possible surviving sequence among ten surviving sequences for the node sequence $(U_1, U_4)$.

Definition 5 (surviving sequences set). The set of all surviving sequences associated with a node sequence $(U_{\lambda_i})_{i=1}^{[\mathcal{A}_t]}$ is defined as the surviving sequence set $S((U_{\lambda_i})_{i=1}^{[\mathcal{A}_t]}) = \{(S_{\lambda_i}^{(\ell)})_{i=1}^{[\mathcal{A}_t]} : \ell \in \{1, 2, \ldots, \tau_{\lambda_i}\}\}$. Clearly, $|S((U_{\lambda_i})_{i=1}^{[\mathcal{A}_t]})| = \left(\prod_{i=1}^{[\mathcal{A}_t]} \tau_{\lambda_i}\right)!$.

For example, in Fig. 2, observe that $\{(S_1^{(\ell_1)}, S_4^{(\ell_2)}) : \ell_1 = 1, 2, 3, 4, 5 \text{ and } \ell_2 = 1, 2\} = S((U_1, U_4))$ etc.

In [28], a trade-off curve between system storage cost and system repair cost was given for any heterogeneous DSS with uniform reconstruction degree. Similarly one can give a trade-off curve between system storage cost and system repair cost for the heterogeneous DSS model considered in our paper. For our model, we define system storage cost, node repair cost, and system repair cost as follows.

Definition 6 (system storage cost). The total amount of the cost $C_s(\alpha)$ to store unit data in any heterogeneous DSS $(n, k, d, \alpha, \beta, B)$ is called the system storage cost, where the storage amount vector is $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, the storage cost vector is $s = (s_1, s_2, \ldots, s_n)$, storage capacity of node $U_i$ is $\alpha_i$, and the amount $s_i$ is the cost to store unit information data in node $U_i$, $i = 1, 2, \ldots, n$. Clearly,

$$C_s(\alpha) = \frac{1}{B} \sum_{j=1}^{n} s_j \alpha_j.$$  

Definition 7 (node repair cost). The average amount of the cost to repair a node $U_i$, $i \in \{1, 2, \ldots, n\}$, in any heterogeneous DSS $(n, k, d, \alpha, \beta, B)$ is called the node repair cost $r(\beta_i)$ associated with repair cost vector $r = (r_1, r_2, \ldots, r_n)$; i.e.,

$$r(\beta_i) = \frac{1}{B \tau_i} \sum_{\ell=1}^{\tau_i} \sum_{j: U_j \in S_i^{(\ell)}} r_j \beta(U_i, U_j, S_i^{(\ell)}),$$

where $r_j$ is the cost to download unit amount of data from node $U_j$ during the repair process. Clearly, the node repair vector is $r(\beta) = (r(\beta_1), r(\beta_2), \ldots, r(\beta_n))$.

Definition 8 (system repair cost). The system repair cost $C_r(\beta)$ is the total amount of the cost to repair all nodes in any heterogeneous DSS $(n, k, d, \alpha, \beta, B)$. Mathematically,

$$C_r(\beta) = \sum_{j=1}^{n} r(\beta_j).$$

Now, in the next section, we give some results and analysis for the min-cut bound for the model of the heterogeneous DSS considered in this paper.
3. RESULTS

For our model, it is shown that minimum possible value of flexible reconstruction degree is lower bound of cardinality of any cut set which separates source node and data collector node. For the heterogeneous DSS min-cut bound is calculated in Theorem 1. Using that min-cut bound, it is shown that file size should be lower bound of min-cut bound for the heterogeneous DSS. Using the particular bound as constraint, a bi-objective optimization linear programming problem is formulated to minimize system storage cost and system repair cost for the considered heterogeneous model. A family of solutions is calculated for the optimization problem by substituting some numerical values of system parameters. The numerical parameter is plotted the trade-off curve between system storage cost and system repair cost. The curve is compared with trade-off curve for homogeneous DSS [7] and trade-off curve for heterogeneous DSS [28].

In a heterogeneous DSS, information delivered to data collector \( D \) depends on minimum of cut-capacity(\( s, D \)). Theorem 1 gives the lower bound of min cut-capacity(\( s, D \)).

**Theorem 1** (min-cut bound). For a given heterogeneous DSS with a data collector \( D \) associated with flexible reconstruction degree \( k_t \), we have min cut-capacity(\( s, D \)) \( \geq \Omega \), where

\[
\Omega = \min_{A_t \in \mathcal{A}} \min_{\{U_{\lambda_i}\}_{i=1}^{k_t}} \sum_{i=1}^{k_t} \min\left\{ \alpha_{\lambda_i}, \min_{(S_{\lambda_i}^{(\ell)})_{i=1}^{k_t}} \sum_{j} \beta(\{U_{\lambda_i}, U_{\mu_j}, S_{\lambda_i}^{(\ell)}\}) \right\}
\]

for \( |A_t| = k_t \), \( \{U_{\lambda_i}\}_{i=1}^{k_t} \in \mathcal{A}(A_t) \), \( (S_{\lambda_i}^{(\ell)})_{i=1}^{k_t} \in \mathcal{S}(\{U_{\lambda_i}\}_{i=1}^{k_t}) \), \( \ell_i \in \{1, 2, \ldots, \tau \} \), and the index \( \mu_j \) is associated with the node \( U_{\mu_j} \in S_{\lambda_i}^{(\ell)} \setminus \{U_{\lambda_1}, \ldots, U_{\lambda_{i-1}}\} \). In this expression, the remaining used notation has common meaning as defined earlier.

**Proof.** Consider a heterogeneous DSS \( (n, k, d, \alpha, \beta, B) \) with a given reconstruction set \( A_t = \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{k_t}}\} \). If all the nodes \( U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{k_t}} \) fail one by one according to the order of nodes in the given node sequence \( \{U_{\lambda_i}\}_{i=1}^{k_t} \), then, for \( \ell_i \in \{1, 2, \ldots, \tau \} \), those failed nodes will be repaired by surviving sets \( S_{\lambda_i}^{(\ell_1)}, S_{\lambda_i}^{(\ell_2)}, \ldots, S_{\lambda_i}^{(\ell_{k_t})} \). Thus, the repair process follows the same order, i.e., the order of surviving sets in the given surviving sequence \( (S_{\lambda_i}^{(\ell_i)})_{i=1}^{k_t} \). For the heterogeneous DSS, consider an information flow graph \( G = (V, E) \). Each information flow graph \( G = (V, E) \) has a source node \( s \), a data collector node \( D \), where the data collector \( D \) is connected with all nodes of the reconstruction set \( A_t \), and therefore, the reconstruction degree is \( k_t \). In the heterogeneous DSS, a failed node \( U_i \) can be repaired by nodes of some surviving set \( S_{\lambda_i}^{(\ell_i)} \), where \( \ell_i \in \{1, 2, \ldots, \tau \} \).

Let \( X \subseteq V \), \( \overline{X} = V \setminus X \), \( s \in X \), and \( D \in \overline{X} \) be such that some nonempty subset \( \text{cut}(X, \overline{X}) \subseteq \mathcal{E} \) exists. Now, if \( X = V \setminus \{D\} \), then cut-capacity\( (X, \overline{X}) \) \( \to \infty \). Similarly, if \( X = \{s\} \), then again cut-capacity\( (X, \overline{X}) \) \( \to \infty \). Hence, min cut-capacity\( (X, \overline{X}) \) would be obtained by all those \( \text{Out}_i \in \overline{X} \) and \( \text{In}_i \in X \), since it will give a finite cut-capacity\( (X, \overline{X}) \), where \( i \in \{1, 2, \ldots, n\} \) and \( j \in \{1, 2, \ldots, k_t\} \).

An information flow graph \( G = (V, \mathcal{E}) \) is a directed acyclic graph, so it can be represented in a topological order of its vertices. For the topological order, sequences of node failures and corresponding sequences of surviving sets are arranged by using definitions as given in previous sections. For that, assume that a data collector \( D \) connects with all nodes of the set \( A_t \in \mathcal{A} \) and reconstructs the file \( B \). The set \( \mathcal{A}(A_t) \) is the collection of all possible sequences of nodes of \( A_t \in \mathcal{A} \). A sequence \( \{U_{\lambda_i}\}_{i=1}^{k_t} \in \mathcal{A}(A_t) \) represents the order of node failures of a specific set \( A_t \). Recall the set of all possible surviving sequences \( (S_{\lambda_i}^{(\ell_i)})_{i=1}^{k_t} \) associated with a node sequence \( \{U_{\lambda_i}\}_{i=1}^{k_t} \) is \( \mathcal{S}(\{U_{\lambda_i}\}_{i=1}^{k_t}) \).

For a specific node sequence \( \{U_{\lambda_i}\}_{i=1}^{k_t} \) with a specific surviving sequence \( (S_{\lambda_i}^{(\ell_i)})_{i=1}^{k_t} \), one can analyze the following.
For \( \text{Out'}_{\lambda_1} \in \overline{\mathcal{X}} \) associated with the first node in node sequence \( \langle U_{\lambda_1} \rangle_{i=1}^{k_t} \), the following two cases are possible.

- If \( \text{In'}_{\lambda_1} \in \mathcal{X} \), then \( (\text{In'}_{\lambda_1}, \text{Out'}_{\lambda_1}) \in \text{cut}(\mathcal{X}, \overline{\mathcal{X}}) \). Hence, \( \alpha_{\lambda_1} \) will contribute in cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)).

- If \( \text{In'}_{\lambda_1} \in \overline{\mathcal{X}} \), then edges \( (\text{Out}_\mu, \text{In'}_{\lambda_1}) \in \text{cut}(\mathcal{X}, \overline{\mathcal{X}}) \), where \( U_\mu \in S_{\lambda_1}^{(\ell_{\lambda_1})} \) and \( S_{\lambda_1}^{(\ell_{\lambda_1})} \in \langle S_{\lambda_1}^{(\ell_{\lambda_1})} \rangle_{i=1}^{k_t} \) for any \( \ell_{\lambda_1} \in \{1, 2, \ldots, \tau_{\lambda_1} \} \). Hence, this case contributes in cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) by

\[
\sum_{\mu_j: U_\mu \in S_{\lambda_1}^{(\ell_{\lambda_1})}} \beta(U_{\lambda_1}, U_\mu, S_{\lambda_1}^{(\ell_{\lambda_1})}).
\]

Thus, the contribution in min cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) supported by the node \( U_{\lambda_1} \) is

\[
\min \left\{ \alpha_{\lambda_1}, \sum_{\mu_j: U_\mu \in S_{\lambda_1}^{(\ell_{\lambda_1})}} \beta(U_{\lambda_1}, U_\mu, S_{\lambda_1}^{(\ell_{\lambda_1})}) \right\}.
\]

If a node \( U_p, p = 1, 2, \ldots, n, \) fails in the system, then all nodes of some surviving set \( S_p^{(\ell_p)} \) will generate a new node \( U'_p \) with the same characteristic.

For the remaining part of the proof, we use the notation \( U_p, p \in \{1, 2, \ldots, n\} \), in place of \( U'_p \), since characteristics of both nodes \( U_p \) and \( U'_p \) are the same and one of them appears at instant.

In general to compute contribution in min cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) by node \( U_{\lambda_i} \in \langle U_{\lambda_i} \rangle_{i=1}^{k_t} \), assume \( \text{Out'}_{\lambda_i} \in \overline{\mathcal{X}} \). Now, the following two cases are possible.

- If \( \text{In'}_{\lambda_i} \in \mathcal{X} \), then \( (\text{In'}_{\lambda_i}, \text{Out'}_{\lambda_i}) \in \text{cut}(\mathcal{X}, \overline{\mathcal{X}}) \). Hence, \( \alpha_{\lambda_i} \) will contribute in cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)).

- If \( \text{In'}_{\lambda_i} \in \overline{\mathcal{X}} \), then all possible edges \( (\text{Out}_\mu, \text{In'}_{\lambda_i}) \) such that \( U_\mu \in S_{\lambda_i}^{(\ell_{\lambda_i})} \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{i-1}}\} \) and \( S_{\lambda_i}^{(\ell_{\lambda_i})} \in \langle S_{\lambda_i}^{(\ell_{\lambda_i})} \rangle_{i=1}^{k_t} \) for any \( \ell_{\lambda_i} \in \{1, 2, \ldots, \tau_{\lambda_i} \} \) will contribute in cut(\( \mathcal{X}, \overline{\mathcal{X}} \)). Edges \( (\text{Out}_{\lambda_j}, \text{In'}_{\lambda_i}) \) associated with node \( U_\mu \in S_{\lambda_i}^{(\ell_{\lambda_i})} \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{i-1}}\} \) are newly investigated from step label 0 for cut(\( \mathcal{X}, \overline{\mathcal{X}} \)). Edges \( (\text{Out'}_{\lambda_m}, \text{In'}_{\lambda_i}) \) must be excluded, because they have investigated earlier at step label \( m \), where \( m \in \{1, 2, \ldots, i-1\} \), such that \( U_{\lambda_m} \in S_{\lambda_i}^{(\ell_{\lambda_i})} \). Hence, this case contributes in cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) by

\[
\sum_{\mu_j: U_\mu \in S_{\lambda_i}^{(\ell_{\lambda_i})} \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{i-1}}\}} \beta(U_{\lambda_i}, U_\mu, S_{\lambda_i}^{(\ell_{\lambda_i})}).
\]

Thus, the contribution in min cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) by node \( U_{\lambda_i} \) is

\[
\min \left\{ \alpha_{\lambda_i}, \sum_{\mu_j: U_\mu \in S_{\lambda_i}^{(\ell_{\lambda_i})} \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{i-1}}\}} \beta(U_{\lambda_i}, U_\mu, S_{\lambda_i}^{(\ell_{\lambda_i})}) \right\}.
\]

If a data collector \( D \) connects with each of the nodes \( U_{\lambda_i} \in \mathcal{A}_t, i \in \{1, 2, \ldots, k_t\} \), then for a specific node sequence \( \langle U_{\lambda_i} \rangle_{i=1}^{k_t} \) associated with a specific surviving sequence \( \langle S_{\lambda_i}^{(\ell_{\lambda_i})} \rangle_{i=1}^{k_t} \) the contribution in min cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) is

\[
\sum_{i=1}^{k_t} \min \left\{ \alpha_{\lambda_i}, \sum_{\mu_j: U_\mu \in S_{\lambda_i}^{(\ell_{\lambda_i})} \{U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{i-1}}\}} \beta(U_{\lambda_i}, U_\mu, S_{\lambda_i}^{(\ell_{\lambda_i})}) \right\}.
\]

Now, for a given reconstruction set \( \mathcal{A}_t \), one can find min cut-capacity(\( s, D \)) for a specific \( D \) by taking the minimum among all possible cut-capacity(\( \mathcal{X}, \overline{\mathcal{X}} \)) which is calculated for all possible
node sequences \( (U_{\lambda_i})_{i=1}^{k_t} \) among all possible associated surviving sequences \( (S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t} \), i.e.,
\[
\min\left(\min_{(U_{\lambda_i})_{i=1}^{k_t}} \sum_{U_{\lambda_i} \in A_t} \min_{(S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t}} \sum_{\alpha_i} \min_{\mu_j, U_{\mu_j} \in S_{\lambda_i}^{(\ell_{\lambda_i}) \setminus \{U_{\lambda_1},...,U_{\lambda_{i-1}}\}}} \beta\left(U_{\lambda_i}, U_{\mu_j}, S_{\lambda_i}^{(\ell_{\lambda_i})}\right)\right). \tag{3}
\]

But the node storage capacity index \( \lambda_i \) is governed by nodes of the node sequence. Therefore, the value given in (3) is equal to
\[
\min\left(\sum_{U_{\lambda_i} \in A_t} \min_{(S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t}} \sum_{\alpha_i} \min_{\mu_j, U_{\mu_j} \in S_{\lambda_i}^{(\ell_{\lambda_i}) \setminus \{U_{\lambda_1},...,U_{\lambda_{i-1}}\}}} \beta\left(U_{\lambda_i}, U_{\mu_j}, S_{\lambda_i}^{(\ell_{\lambda_i})}\right)\right).
\]

Hence, for a given heterogeneous DSS, the cut-capacity \( (\mathcal{X}, \mathcal{F}) \) is
\[
\min_{A_t \in A} \left(\sum_{(U_{\lambda_i})_{i=1}^{k_t}} \min_{U_{\lambda_i} \in A_t} \min_{(S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t}} \sum_{\alpha_i} \min_{\mu_j, U_{\mu_j} \in S_{\lambda_i}^{(\ell_{\lambda_i}) \setminus \{U_{\lambda_1},...,U_{\lambda_{i-1}}\}}} \beta\left(U_{\lambda_i}, U_{\mu_j}, S_{\lambda_i}^{(\ell_{\lambda_i})}\right)\right),
\]
where \(|A_t| = k_t\), \( (U_{\lambda_i})_{i=1}^{k_t} \in A_t\), and \( (S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t} \in B((U_{\lambda_i})_{i=1}^{k_t})\).

The bound is tight, since the min-cut bound is calculated by taking the minimum value of all possible cut bounds. In other words, the min-cut bound is calculated for all possible reconstruction sets with all possible node sequences \( (U_{\lambda_i})_{i=1}^{k_t} \) associated with all possible surviving sequences \( (S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t} \). Hence, there exists at least one surviving sequence, say \( (S_{\lambda_i}^{(\ell_{\lambda_i})})_{i=1}^{k_t} \), associated with node sequence, say \( (U_{\lambda_i})_{i=1}^{k_t} \), for which the inequality holds with equality; i.e., the min-cut bound in Theorem 1 is tight. \( \triangle \)

**Remark 2.** For a given heterogeneous DSS, if an arbitrary data collector connects each node \( U_{\lambda_j} \) from the subset \( A_t \in A \), then the total number of possible information flow graphs is given by
\[
\sum_{A_t \in A} \left(|A_t|! \prod_{j=1}^{k_t} \tau_{\lambda_j}\right).
\]
In particular, for a specific information flow graph, the total number of computational comparisons is \(|A_t|\). Therefore, the total number of computational comparisons for a heterogeneous DSS is
\[
\sum_{A_t \in A} \left(|A_t|! \prod_{j=1}^{k_t} \tau_{\lambda_j}\right).
\]
Hence, one can say that the time complexity to calculate min-cut bound for a heterogeneous DSS is
\[
O\left(|A_t|! \prod_{j=1}^{k_t} \tau_{\lambda_j}\right).
\]

From Theorem 1, one can calculate the minimum requirement of storage node capacity and repair bandwidth to store a file with size \( B \). In other words, the upper bound of stored file with size \( B \) is given by the following lemma.

**Lemma 1.** If a file with size \( B \) is stored in a given heterogeneous DSS \( (n, k, d, \alpha, \beta, B) \), then
\[
B \leq \Omega, \tag{4}
\]
where \( \Omega \) is given in equation (2) and the remaining used notation has common meaning as defined earlier.

**Proof.** Any arbitrary data collector node \( D \) must be able to reconstruct the whole file with size \( B \). Hence, the maximum information flow value delivered to any data collector should be
at least $B$. Now the proof follows from the min-cut max-flow theorem, Theorem 1, and inequality (1).

The min cut-capacity $(s, D)$ for the information flow graph as shown in Fig. 4 will be

\[
\min \{ \alpha_1, \beta(U_1, U_2, S^{(1)}_1) \} + \min \{ \alpha_2, \beta(U_2, U_4, S^{(1)}_2) \} + \min \{ \alpha_3, \beta(U_3, U_4, S^{(1)}_3) \} = 2 + 1 + 1 = 4
\]

units. Now, one can frame an optimization problem to find the minimum system storage cost and system repair cost under the constraint that the maximum possible information delivered to a data collector node $D$ is at least $B$.

**Problem 1.**

Minimize: $[C_s(\alpha), C_r(\beta)]$

subject to

Inequality (4);

\[
\alpha_i \geq 0;
\]

\[
\beta(U_i, U_j, S^{(\ell)}_i) \geq 0;
\]

where $i = 1, 2, \ldots, n$, $\ell = 1, 2, \ldots, \tau$, and $U_j \in S^{(\ell)}_i$ for some $j \in \{1, 2, \ldots, n\} \setminus \{i\}$.

**Remark 3.** One can find the trade-off curve between repair cost and storage cost by the optimization Problem 1 for the exact or functional repair using the surviving sets as the collection of those helper nodes which help to repair failed nodes as exact or functional respectively.

Optimum values for both objective functions of the bi-objective optimization Problem 1 are plotted as a trade-off curve between $C_s(\alpha)$ and $C_r(\beta)$. In this paper, the optimization Problem 1 is solved by the weighted sum method for some numeric examples. Some specific cases for optimization Problem 1 are analyzed in the following subsection.

**3.1. Some Specific Cases**

The heterogeneous DSS considered in our work can be reduced to the following cases under some specific constraints.

(1) (uniform reconstruction): If an arbitrary data collector can retrieve the file by downloading data from exactly $k$ nodes for any combination out of $n$ nodes, then the constraint inequality (4) for the optimization Problem 1 has an additional property $k_t = k$ for $t = 1, 2, \ldots, \omega$.

**Problem 2.**

Minimize: $[C_s(\alpha), C_r(\beta)]$

subject to

\[
B \leq \min \{ \alpha_{\lambda_i} \} + \sum_{i=1}^{k} \min \{ \beta(U_{\lambda_i}, U_{\mu_j}, S^{(\ell)}_{\lambda_i}) \} ;
\]

where $\mu_j$ is the index of the node $U_{\mu_j} \in S^{(\ell)}_{\lambda_i} \setminus \{U_{\lambda_1}, \ldots, U_{\lambda_{i-1}}\}$ for $i = 1, 2, \ldots, k$.

(2) (uniform repair degree): For a heterogeneous DSS, let a node failure can be repaired by any $d$ nodes out of the remaining $n - 1$ nodes. Under the particular assumption, the constraint inequality (4) for the optimization Problem 1 reduces to the following.
\textbf{Problem 3.} 

Minimize: $[C_s(\alpha), C_r(\beta)]$

subject to

$$B \leq \min_{A_t \in \mathcal{A}} \sum_{t=1}^{k_t} \min_{U_{\lambda_t} \in \mathcal{A}_t} \left\{ \alpha_{\lambda_t}, \sum_{\mu_j} \beta \left( U_{\lambda_t}, U_{\mu_j}, S_{\lambda_t}^{(\ell_t)} \right) \right\};$$

$$0 \leq \alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n;$$

$$1 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{k_t} \leq n;$$

where $\mu_j$ is the index of the node $U_{\mu_j} \in S_{\lambda_t}^{(\ell_t)} \setminus \{ U_{\lambda_1}, \ldots, U_{\lambda_{t-1}} \}$ such that $\{ U_{\lambda_1}, \ldots, U_{\lambda_{t-1}} \} \subset S_{\lambda_t}^{(\ell_t)}, i = 1, 2, \ldots, k_t$, with some $j \in \{1, 2, \ldots, d\}$.

In this case $|S_{m}^{(f_m)}| = d$, $\tau_m = \left( \frac{n - 1}{d} \right)$, for $m = 1, 2, \ldots, n$. Here, \text{min cut-capacity}(s, D) will be given by the node sequence $\langle U_{\lambda_i} \rangle_{i=1}^{k_t} \in \mathcal{A}(\mathcal{A}_t)$ associated with a surviving sequence $\langle S_{\lambda_t}^{(\ell_t)} \rangle_{i=1}^{k_t}$ such that $\alpha_{\lambda_1} \leq \alpha_{\lambda_2} \leq \ldots \leq \alpha_{\lambda_{k_t}}$ and $\{ U_{\lambda_1}, U_{\lambda_2}, \ldots, U_{\lambda_{t-1}} \} \subset S_{\lambda_t}^{(\ell_t)}$.

(3) (uniform repair download amount): In this case we assume that the downloaded amount from any arbitrary helper node to repair the system is constant, say $\beta$. Hence, the optimization Problem 1 under the restriction has additional properties as $\beta(U_{i}, U_{j}, S_{t}^{(\ell_i)}) = \beta$, $\beta \geq 0$, $i = 1, 2, \ldots, n$, all possible $j \in \{1, 2, \ldots, n\} \setminus \{i\}$, and $\ell_i = 1, 2, \ldots, \tau_i$.

\textbf{Problem 4.} 

Minimize: $[C_s(\alpha), C_r(\beta)]$

subject to

$$B \leq \min_{A_t \in \mathcal{A}} \min_{U_{\lambda_t} \in \mathcal{A}_t} \sum_{t=1}^{k_t} \min_{(S_{\lambda_t}^{(\ell_t)})_{i=1}^{k_t}} \left\{ \alpha_{\lambda_t}, \min \left\{ S_{\lambda_t}^{(\ell_t)} \setminus \{ U_{\lambda_1}, \ldots, U_{\lambda_{t-1}} \} \right\} / \beta \right\};$$

$$0 \leq \alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n; \beta \geq 0.$$

(4) (homogeneous DSS): A heterogeneous DSS reduces to a homogeneous DSS if characteristics of the parameters are uniform. Hence, assume that effective reconstruction degree for any data collector is $k$ and storage capacity of each node is $\alpha$. In addition, let a node failure can be repaired by \textit{any} $d$ nodes out of remaining $n - 1$ nodes by downloading $\beta$ packets from each helper node. Under these restrictions, the constraint inequality (4) for the optimization Problem 1 reduces to the following.

\textbf{Problem 5.} 

Minimize: $[C_s(\alpha), C_r(\beta)]$

subject to

$$B \leq \sum_{i=1}^{k} \min \{ \alpha, (d - i - 1)\beta \};$$

$$\alpha \geq 0;$$

$$\beta \geq 0.$$

(5) (other): In this paper, the considered heterogeneous DSS model can be reduced into some more specific DSS by applying some appropriate restrictions on constraints. For example, a heterogeneous DSS with uniform reconstruction and uniform repair degree (case 1 and 2 respectively) collectively reduces to a heterogeneous DSS as investigated in [28].
One can find solutions of the bi-objective optimization Problem 1 for some numerical values and plot the solution as a trade-off curve for the same. One can compare the trade-off curve with the trade-off curve for the existing heterogeneous DSS investigated in [28]. Hence, in the next section we are calculating some optimum solutions for numerical parameters for our model and comparing it with the homogeneous model [7] and heterogeneous model [28].

3.2. Numerical Work

For the optimization Problem 1, LP problems with single objective function is solved. The single objective function is calculated by taking linear combination of the two objective functions of the optimization Problem 1. Then such LP problems are solved by taking distinct linear combination factors between $10^{-3}$ and $10^3$. Plotting the trade-off and solving LP problems are done with the help of MATLAB and lp_solve [38].

In Fig. 5, four trade-off curves are plotted between the system repair cost $C_r$ and system storage cost $C_s$ for the respective DSSs. In particular, in Fig. 5, one curve is plotted for a homogeneous DSS as investigated in [7], another one is drawn for a heterogeneous DSS as investigated in [28], and the remaining two curves are plotted for two heterogeneous DSSs as studied in this paper. In particular, one of the remaining two curves has the minimum effective reconstruction degree $k_{\text{min}}$ as 2 and the other has the maximum effective reconstruction degree $k_{\text{max}}$ as 2. For all the considered DSSs, the common parameters are as follow: $n = 4$, $B = 1$ unit, $s = (1 \ 10 \ 100)$, and $r = (10 \ 1 \ 1 \ 1)$. For the homogeneous DSS and heterogeneous DSS studied in [28], we have reconstruction degree $k = 2$ and repair degree $d = 3$. Both remaining heterogeneous DSSs have surviving sets $S^{(1)}_1 = \{U_2, U_3, U_4\}$, $S^{(1)}_2 = \{U_1, U_4\}$, $S^{(1)}_3 = \{U_1, U_2\}$, and $S^{(1)}_4 = \{U_2, U_3\}$.

In Fig. 5, one can see that our heterogeneous DSS model has more optimum system storage and repair cost than the homogeneous DSS studied in [7]. Although the characteristics of our heterogeneous model and the heterogeneous model investigated in [28] are different, we obtained
some more optimum points for our model as in Fig. 5. It is shown in Section 3.1 that one can find the heterogeneous DSS considered in [28] by taking some restrictions on our model.

Remark 4. Note the scaling of an arbitrary file size $B$ to 1 leads to a respective integer solution that does not necessarily scale to some integer. In the particular trade-off curves, noninteger solutions of the bi-objective optimization problem 1 can also be considered.

Now consider the example as given in Fig. 2. In this example, the average reconstruction degree is 2.286 in this heterogeneous DSS. The average repair degrees for the nodes are $(2, 2, 2, 2.5, 2)$, and 2.1 is the mean of all average node repair degrees. In the case of node failure for the DSS given in that example, repair bandwidths are calculated in Table 1. Thus, for storage cost vector $s = (1, 1, 1, 1, 1)$ and repair cost vector $r = (1, 1, 1, 1, 1)$, the system storage cost and system repair cost are 2.750 unit and 3.667 unit. Therefore, system storage per node and system repair per node are 0.550 unit and 0.733 unit. Now, the trade-off for the homogeneous DSS with reconstruction degree 2 and repair degree 2 is plotted in the graph given in Fig. 6. The graph in the figure is the comparison of the example (Fig. 2) of a heterogeneous DSS with the homogeneous DSS with $k = 2$ and $d = 2$.

4. ANALYSIS OF THE BOUND

4.1. Model

In a heterogeneous DSS, a file is divided into encoded packets, and the encoded packets are distributed among $n$ distinct nodes $U_i$, $i = 1, 2, \ldots, n$, such that each node has storage capacity $\alpha_i$ and repair degree $d_i$. A user can reconstruct the file by downloading data from any $k$ ($< n$) nodes. If a node $U_i$ fails, then a data collector will download $\beta$ packets from specific $d_i$ nodes out of the remaining $n - 1$ nodes. The particular $d_i$ nodes are called helper nodes for the failed node $U_i$. 

Fig. 6. Trade-off curve between storage per node and repair bandwidth per node for the homogeneous DSS vs. storage per node and repair bandwidth per node for the example of a heterogeneous DSS in Fig. 2.
In such a case, the repair bandwidth for a node \( U_i \) is \( \gamma_i = d_i \beta \). Note that Lemmas 2 and 3 are derived for such heterogeneous DSSs.

An example of such a heterogeneous DSS is illustrated in Fig. 7. In this example, a file with size 3 (= \( B \)) is stored in an \( (n = 6, k = 2) \) heterogeneous DSS with repair traffic \( \beta \) as 1. In the particular DSS, the node storage capacity \( \alpha_i \) is 2, 2, 3, 2, 2 for \( i = 1, 2, 3, 4, 5, 6 \) (see Fig. 7). Note that \( \alpha_i = \gamma_i = d_i \) for \( i = 1, 2, 3, 4, 5, 6 \). In [27], the heterogeneous DSS is mapped with an acyclic directed graph called an information flow graph. Analyzing the min-cut of the information flow graph, a fundamental bound on the file size \( B \) is computed for such a heterogeneous DSS. The bound (a special case of Lemma 1) is described in the following lemma.

**Lemma 2** (fundamental bound). For the heterogeneous DSS with \( n \) nodes and reconstruction degree \( k \), the file size \( B \) must satisfy the following inequality:

\[
B \leq \min_{\langle S_{\lambda_i}^{(\ell_i)} \rangle_{i=1}^n} \left\{ \sum_{i=1}^k \min_{\lambda_i} \left\{ \alpha_{\lambda_i} \left| S_{\lambda_i}^{(\ell_i)} \setminus \left( \bigcup_{j=0}^{i-1} \{ U_{\lambda_j} \} \right) \right| \beta \right\} \right\},
\]

where \( \{ U_{\lambda_0} \} = \emptyset, 0 \leq j < i \leq k, S_{\lambda_i}^{(\ell_i)} \in \langle S_{\lambda_i}^{(\ell_i)} \rangle_{i=1}^n, \) and \( \ell_i \in \{1, 2, \ldots, \tau_{\lambda_i} \} \).

**Proof.** For the considered heterogeneous DSSs, any set \( A \subset \{ U_1, U_2, \ldots, U_n \} \) of size \( k \) is a reconstruction set. Therefore, any sequence of length \( k \) of nodes will be the node sequence for the heterogeneous DSS. Therefore, any sequence of length \( k \) over surviving sets will also be the surviving sequence for the heterogeneous DSS. Hence, the proof follows from Lemma 1. \( \triangle \)

In [27], it is shown that there exist a code which achieves the fundamental bound for such an \( (n, k) \) heterogeneous DSS. Hence, one can get the optimal codes by reducing parameters which meet the fundamental bound. In the next section, parameters for the optimal codes are computed by minimizing the node storage capacity and repair bandwidth.

**4.2. Conditions for Optimality**

Consider an \( (n, k) \) heterogeneous DSS with \( \tau_i \) surviving sets \( S_i^{(\ell_i)} \) and repair degree \( |S_i^{(\ell_i)}| = d_i \), \( \ell_i = 1, 2, \ldots, \tau_i, i = 1, 2, \ldots, n \). If \( \alpha_i > |S_i^{(\ell_i)}|\beta \), then the failed node \( U_i \) cannot be repaired, so \( \alpha_i \leq |S_i^{(\ell_i)}|\beta \) for each \( i \) and \( \ell_i \). For optimality, \( \alpha_i = d_i \beta \). Hence, for constant repair traffic \( \beta \), the node

---

**Fig. 7.** A file is divided into 3 (= \( B \)) distinct coded packets \( x_1, x_2, \) and \( x_3 \) over a field \( \mathbb{F}_q \). These three packets are encoded into thirteen distinct packets and distributed in a heterogeneous DSS with six nodes and reconstruction degree 2.
storage capacity $\alpha_i$ and repair degree $d_i$ are proportional to each other. Consider $c_i \in (0, 1) \subset \mathbb{R}$ such that $\sum_{i=1}^{n} c_i = 1$ and $\frac{c_i}{\alpha_i} = \frac{c_j}{\alpha_j}$ for $1 \leq i < j \leq n$. Hence, $\alpha_i = c_i \sum_{i=1}^{n} \alpha_i = c_i \alpha^*, i = 1, 2, \ldots, n$. Thus, the parameters $\alpha_i$ and $c_i$ are proportional to each other. Again, $k$ is the reconstruct degree, so $B \leq \sum_{i \in K} \alpha_i = \sum_{i \in K} c_i \alpha^*$ for any arbitrary set $K \subset \{1, 2, \ldots, n\}$ such that $|K| = k$. Hence, $B \leq \sum_{i=1}^{k} c_i \alpha^*$ for $c_1 \leq c_2 \leq \ldots \leq c_n$. For the optimum case, one can reduce $\alpha^*$ up to $\alpha^\text{min}$ such that

$$B = \sum_{i=1}^{k} c_i \alpha^\text{min} \implies \alpha^\text{min} = B \left( \sum_{j=1}^{k} c_j \right)^{-1}.$$  

Similarly, for a fixed proportional factor $\alpha^\text{min}$, one can minimize the repair traffic $\beta$ such that the bound given in Lemma 2 holds with equality. For a specific surviving sequence $(S_{\lambda_i}^{(\ell_i)})_{i=1}^{n}$ with sufficiently large repair traffic $\beta$, the inequality $\alpha_{\lambda_i} \leq \left| S_{\lambda_i}^{(\ell_i)} \setminus \left( \bigcup_{j=0}^{i-1} \{U_{\lambda_j}\} \right) \right| \beta$ holds for each $i = 1, 2, \ldots, k$. If we choose $\beta = \beta^\text{min}$ such that

$$\beta^\text{min} = \max_{(S_{\lambda_i}^{(\ell_i)})_{i=1}^{n}} \left\{ \max_{1 \leq i \leq k} \left\{ \alpha_{\lambda_i} \left| S_{\lambda_i}^{(\ell_i)} \setminus \left( \bigcup_{j=0}^{i-1} \{U_{\lambda_j}\} \right) \right|^{-1} \right\} \right\},$$

then $\beta^\text{min}$ is the minimum value of the repair traffic $\beta$ which ensures

$$\left| S_{\lambda_i}^{(\ell_i)} \setminus \left( \bigcup_{j=0}^{i-1} \{U_{\lambda_j}\} \right) \right| \beta^\text{min} \geq \alpha_{\lambda_i}$$

for each $\lambda_i$ of an arbitrary surviving sequence. Formally, the results can be summarized in the following lemma.

**Lemma 3.** Consider the heterogeneous DSS with $n$ nodes, reconstruction degree $k$, and surviving sets $S_i^{(\ell_i)}$, $i = 1, 2, \ldots, n$, $\ell_i = 1, 2, \ldots, \tau_i$ for some $\tau_i \in \mathbb{Z}$. A family of codes with $\alpha_i = c_i \alpha^\text{min} = d_i \beta^\text{min}$ and $\beta = \beta^\text{min}$ achieves the fundamental bound (Lemma 2), where $\alpha^\text{min}$ and $\beta^\text{min}$ can be calculated by equations (5) and (6).

## 5. CONCLUSION

In this paper, we proposed a model of a heterogeneous DSS with nonuniform reconstruction degree, storage node capacity, and repair bandwidth. In particular, a file can be reconstructed using a certain set of nodes, and the system is repaired for any failed node by contacting some set of helper nodes. For such a heterogeneous DSS, the fundamental trade-off curve between the system repair cost and system storage cost is investigated. To plot the trade-off curve, a bi-objective optimization problem is formulated with the constraints of the min-cut bound and nonnegative parameters of the heterogeneous DSS. The bi-objective optimization problem is solved by the weighted sum method for some numerical values of the parameters of the heterogeneous model. Analyzing the trade-off curve, we observed some more optimum points than the existing heterogeneous model [28]. The considered model is close to a real world scenario. Our heterogeneous model is flexible enough to mold it into any existing heterogeneous or homogeneous DSS by considering appropriate restrictions. It would be an interesting task to construct codes achieving the optimum points on the trade-off curve.

**ACKNOWLEDGMENTS**

The authors would like to thank the anonymous reviewers for their careful reading of the manuscript, which has improved the presentation.
ADDITIONAL INFORMATION

A preliminary version of the paper is available at https://arxiv.org/abs/1503.02276.

REFERENCES

1. Amazon Elastic Compute Cloud (Amazon EC2). Web Service, Jan. 2013. Available at http://aws.amazon.com/ec2/.

2. Huang, C., Simitci, H., Xu, Y., Ogus, A., Calder, B., Gopalan, P., Li, J., and Yekhanin, S., Erasure Coding in Windows Azure Storage, in Proc. 2012 USENIX Annu. Technical Conf. (USENIX ATC’12), Boston, MA, June 13–15, 2012, pp. 15–26.

3. Microsoft SkyDrive Live. Online Storage Service, Jan. 2013. Available at https://skydrive.live.com/.

4. Sathiamoorthy, M., Asteris, M., Papailiopoulos, D., Dimakis, A.G., Vadali, R., Chen, S., and Borthakur, D., XORing Elephants: Novel Erasure Codes for Big Data, Proc. VLDB Endow., 2013, vol. 6, no. 5, pp. 325–336. https://doi.org/10.14778/2535573.2488339

5. Dimakis, A.G., Godfrey, P.B., Wu, Y., Wainwright, M.J., and Ramchandran, K., Network Coding for Distributed Storage Systems, in Proc. 26th IEEE Annu. Joint Conf. on Computer Communications (INFOCOM’2007), Anchorage, AK, USA, May 6–12, 2007, pp. 2000–2008. https://doi.org/10.1109/INFCOM.2007.232

6. Dimakis, A.G., Godfrey, P.B., Wu, Y., Wainwright, M.J., and Ramchandran, K., Network Coding for Distributed Storage Systems, IEEE Trans. Inform. Theory, 2010, vol. 56, no. 9, pp. 4539–4551. https://doi.org/10.1109/TIT.2010.2054295

7. Wu, Y., Dimakis, A., and Ramchandran, K., Deterministic Regenerating Codes for Distributed Storage, in Proc. 45th Annu. Allerton Conf. on Communication, Control, and Computing, Monticello, IL, USA, Sept. 26–28, 2007, vol. 1, pp. 242–249.

8. Wu, Y., Existence and Construction of Capacity-Achieving Network Codes for Distributed Storage, in Proc. 2009 IEEE Int. Symp. on Information Theory (ISIT’2009), Seoul, Korea, June 28 – July 3, 2009, pp. 1150–1154. https://doi.org/10.1109/ISIT.2009.5206008

9. Wu, Y., Existence and Construction of Capacity-Achieving Network Codes for Distributed Storage, IEEE J. Sel. Areas Commun., 2010, vol. 28, no. 2, pp. 277–288. https://doi.org/10.1109/JSAC.2010.100217

10. Goparaju, S., El Rouayheb, S., and Calderbank, R., New Codes and Inner Bounds for Exact Repair in Distributed Storage Systems, in Proc. 2014 IEEE Int. Symp. on Information Theory (ISIT’2014), Honolulu, HI, USA, June 29–July 4, 2014, pp. 1036–1040. https://doi.org/10.1109/ISIT.2014.6874990

11. Shah, N.B., Rashmi, K.V., and Kumar, P.V., A Flexible Class of Regenerating Codes for Distributed Storage, in Proc. 2010 IEEE Int. Symp. on Information Theory (ISIT’2010), Austin, TX, USA, June 13–18, 2010, pp. 1943–1947. https://doi.org/10.1109/ISIT.2010.5513353

12. Dimakis, A.G., Ramchandran, K., Wu, Y., and Suh, C., A Survey on Network Codes for Distributed Storage, Proc. IEEE, 2011, vol. 99, no. 3, pp. 476–489. https://doi.org/10.1109/JPROC.2010.2096170

13. Prakash, N. and Krishnan, M.N., The Storage-Repair-Bandwidth Trade-off of Exact Repair Linear Regenerating Codes for the Case d = k = n − 1, https://arXiv.org/abs/1501.03983v2 [cs.IT], 2015.

14. Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Weimer, W., Wells, C., and Zhao, B., OceanStore: An Architecture for Global-Scale Persistent Storage, ACM SIGPLAN Notices, 2000, vol. 35, no. 11, pp. 190–201. https://doi.org/10.1145/356989.357007

15. Bianchi, G. and Melen, R., Performance and Dimensioning of a Hierarchical Video Storage Network for Interactive Video Services, Eur. Trans. Telecommun., 1996, vol. 7, no. 4, pp. 349–358. https://doi.org/10.1002/ett.4460070407
16. Pawar, S., El Rouayheb, S., Zhang, H., Lee, K., and Ramchandran, K., Codes for a Distributed Caching Based Video-on-Demand System, in Conf. Rec. 46th Asilomar Conf. on Signals, Systems and Computers (ASILOMAR’2011), Pacific Grove, CA, USA, Nov. 6–9, 2011, pp. 1783–1787. https://doi.org/10.1109/ACSSC.2011.6190328

17. Ntranos, V., Caire, G., and Dimakis, A.G., Allocations for Heterogeneous Distributed Storage, https://arXiv.org/abs/1202.1596 [cs.IT], 2012.

18. Li, Z., Ho, T., Leong, D., and Yao, H., Distributed Storage Allocation for Heterogeneous Systems, in Proc. 51st Annu. Allerton Conf. on Communication, Control, and Computing, Monticello, IL, USA, Oct. 2–4, 2013, pp. 320–326. https://doi.org/10.1109/Allerton.2013.6736541

19. Leong, D., Dimakis, A.G., and Ho, T., Distributed Storage Allocations, IEEE Trans. Inform. Theory, 2012, vol. 58, no. 7, pp. 4733–4752. https://doi.org/10.1109/TIT.2012.2191135

20. Gerami, M., Xiao, M., and Skoglund, M., Optimal-Cost Repair in Multi-hop Distributed Storage Systems, in Proc. 2011 IEEE Int. Symp. on Information Theory (ISIT’2011), St. Petersburg, Russia, July 31 – Aug. 5, 2011, pp. 1437–1441. https://doi.org/10.1109/ISIT.2011.6033777

21. Akhlaghi, S., Kiani, A., and Ghanavati, M.R., Cost-Bandwidth Tradeoff in Distributed Storage Systems, Comput. Commun., 2010, vol. 33, no. 17, pp. 2105–2115. https://doi.org/10.1016/j.comcom.2010.07.022

22. Akhlaghi, S., Kiani, A., and Ghanavati, M.R., A Fundamental Trade-off between the Download Cost and Repair Bandwidth in Distributed Storage Systems, in Proc. 2010 IEEE Int. Symp. on Network Coding (NetCod’2010), Toronto, ON, Canada, June 9-11, 2010, pp. 97–102. https://doi.org/10.1109/NETCOD.2010.5487685

23. Yu, Q., Shum, K.W., and Sung, C.W., Minimization of Storage Cost in Distributed Storage Systems with Repair Consideration, in Proc. 2011 IEEE Global Telecommunications Conf. (GLOBECOM’2011), Houston, TX, USA, Dec. 5–9, 2011, pp. 2931–2935. https://doi.org/10.1109/GLOCOM.2011.6133729

24. Pernas, J., Yuen, C., Gastón, B., and Pujol, J., Non-homogeneous Two-Rack Model for Distributed Storage Systems, in Proc. 2013 IEEE Int. Symp. on Information Theory (ISIT’2013), Istanbul, Turkey, July 7–12, 2013, pp. 1237–1241. https://doi.org/10.1109/ISIT.2013.6620424

25. Gastón, B., Pujol, J., and Villanueva, M., A Realistic Distributed Storage Systems That Minimizes Data Storage and Repair Bandwidth, in Proc. 2006 Data Compression Conf. (DCC’2006), Snowbird, UT, USA, Mar. 20–22, 2013, p. 491. https://doi.org/10.1109/DCC.2013.72

26. Ernvall, T., El Rouayheb, S., Hollanti, C., and Poor, H.V., Capacity and Security of Heterogeneous Distributed Storage Systems, in Proc. 2013 IEEE Int. Symp. on Information Theory (ISIT’2013), Istanbul, Turkey, July 7–12, 2013, pp. 1247–1251. https://doi.org/10.1109/ISIT.2013.6620426

27. Benerjee, K.G. and Gupta, M.K., On Heterogeneous Regenerating Codes and Capacity of Distributed Storage Systems, https://arXiv.org/abs/1402.3801, [cs.IT], 2014.

28. Yu, Q., Shum, K.W., and Sung, C.W., Tradeoff between Storage Cost and Repair Cost in Heterogeneous Distributed Storage Systems, Trans. Emerg. Telecommun. Technol., 2015, vol. 26, no. 10, pp. 1201–1211. https://doi.org/10.1002/ett.2887

29. Kiani, A. and Akhlaghi, S., Selective Regenerating Codes, IEEE Commun. Lett., 2011, vol. 15, no. 8, pp. 854–856. https://doi.org/10.1109/LCOMM.2011.061611.102271

30. Senthooor, K., Sasidharan, B., and Kumar, P.V., Improved Layered Regenerating Codes Characterizing the Exact-Repair Storage-Repair Bandwidth Tradeoff for Certain Parameter Sets, in Proc. 2015 IEEE Information Theory Workshop (ITW’2015), Jerusalem, Israel, Apr. 26–May 1, 2015, pp. 224–228. https://doi.org/10.1109/ITW.2015.7133121

31. Sasidharan, B., Senthooor, K., and Kumar, P.V., An Improved Outer Bound on the Storage-Repair-Bandwidth Tradeoff of Exact-Repair Regenerating Codes, in Proc. 2014 IEEE Int. Symp. on Information Theory (ISIT’2014), Honolulu, HI, USA, June 29 – July 4, 2014, pp. 2430–2434. https://doi.org/10.1109/ISIT.2014.6875270
32. Sasidaran, B. and Kumar, P.V., On the Interior Points of the Storage-Repair Bandwidth Tradeoff of Regenerating Codes, in Proc. 51st Annu. Allerton Conf. on Communication, Control, and Computing, Monticello, IL, USA, Oct. 2–4, 2013, pp. 788–795. https://doi.org/10.1109/Allerton.2013.6736605

33. Duursma, I.M., Outer Bounds for Exact Repair Codes, https://arXiv.org/abs/1406.4852 [cs.IT], 2014.

34. Ahmad, I. and Wang, C.C., When and by How Much Can Helper Node Selection Improve Regenerating Codes?, in Proc. 52nd Annu. Allerton Conf. on Communication, Control, and Computing, Monticello, IL, USA, Sept. 30 – Oct. 3, 2014, pp. 459–466. https://doi.org/10.1109/ALLERTON.2014.7028491

35. Ahlswede, R., Cai, N., Li, S.-Y.R., and Yeung, R.W., Network Information Flow, IEEE Trans. Inform. Theory, 2000, vol. 46, no. 4, pp. 1204–1216. https://doi.org/10.1109/18.850663

36. Elias, P., Feinstein, A., and Shannon, C., A Note on the Maximum Flow Through a Network, IEEE Trans. Inform. Theory, 1956, vol. 2, no. 4, pp. 117–119. https://doi.org/10.1109/TIT.1956.1056816

37. Ford, L.R., Jr. and Fulkerson, D.R., Maximal Flow through a Network, Canad. J. Math., 1956, vol. 8, pp. 399–404. https://doi.org/10.4153/CJM-1956-045-5

38. lp_solve (mathematical optimization software). A Mixed Integer Linear Programming (MILP) Solver. Version 5.5.2.0, 2011. Available at http://lpsolve.sourceforge.net/5.5/.