US009271229B2

(12) **United States Patent**
Gong et al.

(10) **Patent No.: US 9,271,229 B2**
(45) **Date of Patent: Feb. 23, 2016**

(54) **METHODS, SYSTEMS, AND MEDIA FOR PARTIAL DOWNLOADING IN WIRELESS DISTRIBUTED NETWORKS**

(71) Applicants: **Chen Gong**, La Jolla, CA (US); **Xiaodong Wang**, Ramsey, NJ (US)

(72) Inventors: **Chen Gong**, La Jolla, CA (US); **Xiaodong Wang**, Ramsey, NJ (US)

(73) Assignee: **The Trustees of Columbia University in the City of New York**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 69 days.

(21) Appl. No.: **13/948,123**

(22) Filed: **Jul. 22, 2013**

(65) **Prior Publication Data**

US 2014/0022970 A1 Jan. 23, 2014

Related U.S. Application Data

(60) Provisional application No. 61/674,264, filed on Jul. 20, 2012.

(51) **Int. Cl.**

H04W 52/02	(2009.01)
H04L 29/08	(2006.01)
H04W 28/08	(2009.01)

(52) **U.S. Cl.**

CPC ***H04W 52/0203*** (2013.01); *H04L 67/1023* (2013.01); *H04W 28/085* (2013.01)

(58) **Field of Classification Search**

None

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,271,480	A *	6/1981	Vinot	712/300
5,355,372	A *	10/1994	Sengupta et al.	370/367
5,598,435	A *	1/1997	Williams	375/261
5,685,010	A *	11/1997	Yoda	712/28
6,073,189	A *	6/2000	Bounsall et al.	710/52
7,576,884	B2 *	8/2009	Kitahara et al.	358/1.16
7,640,357	B2 *	12/2009	Kirov et al.	709/233
2002/0059439	A1 *	5/2002	Arroyo et al.	709/230
2002/0112102	A1 *	8/2002	Tarui et al.	710/60
2003/0065808	A1 *	4/2003	Dawson	709/232
2003/0115275	A1 *	6/2003	Toga	709/206
2004/0100941	A1 *	5/2004	Lim et al.	370/349
2004/0120332	A1 *	6/2004	Hendel	370/411
2004/0249956	A1 *	12/2004	Tanimoto	709/227
2005/0018702	A1 *	1/2005	Chen et al.	370/431
2005/0132078	A1 *	6/2005	Kumar et al.	709/230
2005/0268125	A1 *	12/2005	Ohneda et al.	713/300
2006/0013179	A1 *	1/2006	Yamane	370/338

(Continued)

OTHER PUBLICATIONS

Cadambe, V.R., et al, "Minimum Repair Bandwidth for Exact Regeneration in Distributed Storage", In Proceedings of the 2010 IEEE Wireless Network Coding Conference (WiNC), Boston, MA, US, Jun. 21, 2010, pp. 1-6.

(Continued)

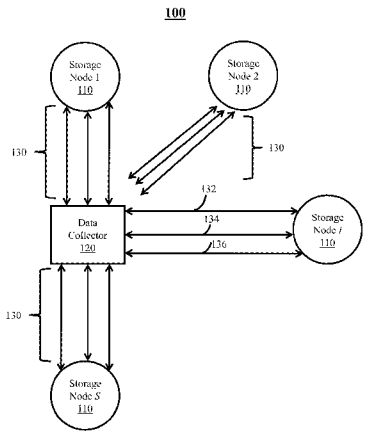
Primary Examiner — Dung B Huynh

(74) *Attorney, Agent, or Firm* — Byrne Poh LLP

(57) **ABSTRACT**

Methods, systems, and media for partial downloading in wireless distributed networks are provided. In some embodiments, methods for selecting numbers of symbols to be transmitted on a plurality of channels are provided, the methods comprising: for each of the plurality of channels, calculating using a hardware processor an increase in power that will be used by that channel if it transmits a symbol; selecting one of the plurality of channels with the smallest increase in power using the hardware processor; and allocating the symbol to the one of the plurality of channels using the hardware processor.

19 Claims, 14 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

2007/0014397	A1*	1/2007	Ukeda et al.	380/30
2008/0014981	A1*	1/2008	Venkatachalam	455/528
2008/0039058	A1*	2/2008	Ray	455/414.3
2008/0248752	A1*	10/2008	Ishino	455/62
2009/0005109	A1*	1/2009	Nishio et al.	455/550.1
2009/0279847	A1*	11/2009	Kinoshita et al.	386/83
2010/0011145	A1*	1/2010	Carver et al.	710/310
2010/0325339	A1*	12/2010	Ogawa et al.	711/103
2011/0161712	A1*	6/2011	Athalye et al.	713/340
2011/0197092	A1*	8/2011	Burklin et al.	714/15
2011/0205898	A1*	8/2011	Ichiki et al.	370/235
2011/0289351	A1*	11/2011	Rashmi et al.	714/6.32
2012/0030356	A1*	2/2012	Fletcher	709/226
2013/0024561	A1*	1/2013	Imai	709/224
2013/0182575	A1*	7/2013	McLean et al.	370/237

OTHER PUBLICATIONS

Dimakis, A.G., et al., "Network Coding for Distributed Storage Systems", In IEEE Transactions on Information Theory, vol. 56, No. 9, Sep. 2010, pp. 4539-4551.

Koetter, R., et al., "On a Theory of Network Equivalence", In Proceedings of the IEEE Information Theory Workshop on Networking and Information Theory (ITW '09), Volos, GR, Jun. 10-12, 2009, pp. 326-330.

Rashmi, K.V., et al, "Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage", Oct. 6, 2009, pp. 1-7, available at: http://arxiv.org/pdf/0906.4913.pdf.

Rashmi, K.V., et al., "Explicit and Optimal Exact-Regenerating Codes for the Minimum-BandWidth in Distributed Storage", Technical Report TR-PME-2010-5, Indian Institute of Science, Bangalore, IN, Feb. 8, 2010, pp. 1-25.

Rashmi, K.V., et al., "Optimal Exact-Regerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction", Jan. 20, 2011, pp. 1-20, available at: http://arxiv.org/pdf/1005.4178v2.pdf.

Shah, N.B., et al., "Explicit Codes Minimizing Repair Bandwidth for Distributed Storage", Sep. 5, 2009, pp. 1-11, available at: http://arxiv.org/pdf/0908.2984.pdf.

Shah, N.B., et al., "Flexible Class of Regenerating Codes for Distributed Storage", In Proceedings of the 2010 IEEE International Symposium on Information Theory (ISIT '10), Austin, TX, US, Jun. 13-18, 2010, pp. 1943-1947.

Suh, C. and Ramachandran, K., "Exact-Repair MDS Code Construction Using Interference Alignment", In IEEE Transactions on Information Theory, vol. 57, No. 3, Mar. 2011, pp. 1425-1442.

Suh, C. and Ramachandran, K., "Exact-Repair MDS Codes for Distributed Storage Using Interference Alignment", Apr. 15, 2010, pp. 1-35, available at: http://arxiv.org/pdf/1001.0107.pdf.

Wu, Y. and Dimakis, A.G., "Reducing Repair Traffic for Erasure Coding-Based Storage via Interference Alignment", In Proceedings of the 2009 IEEE International Conference on Symposium on Information Theory (ISIT '09), Seoul, KR, Jun. 28-Jul. 3, 2009, pp. 2276-2280.

Wu, Y., "Existence and Construction of Capacity-Achieving Network Codes for Distributed Storage", In IEEE Journal on Selected Areas in Communications, vol. 28, No. 2, Feb. 2010, pp. 277-288.

* cited by examiner

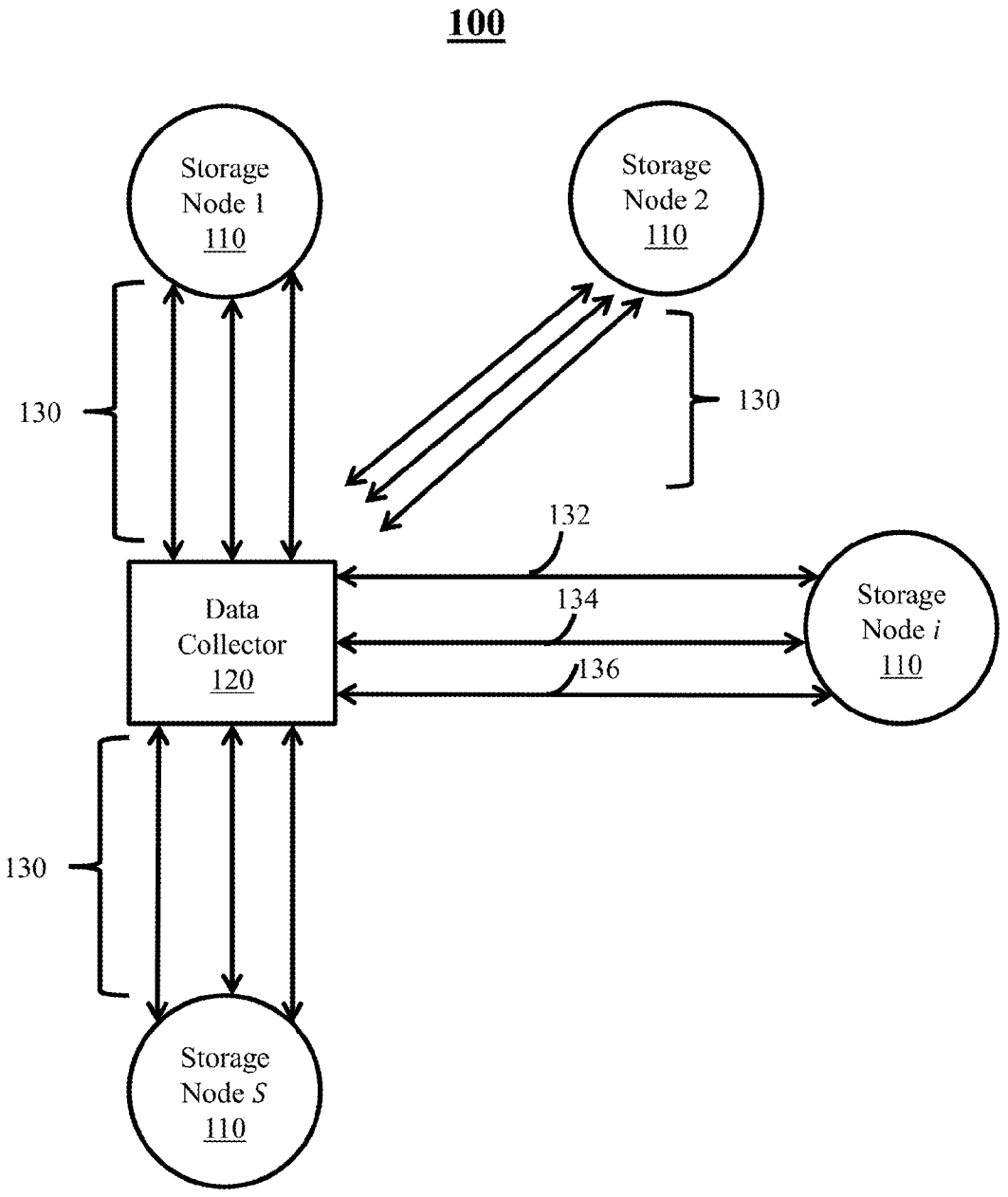

100
Storage Node 1 110
Storage Node 2 110
130
130
132
134
136
Data Collector 120
Storage Node i 110
130
Storage Node S 110

FIG. 1

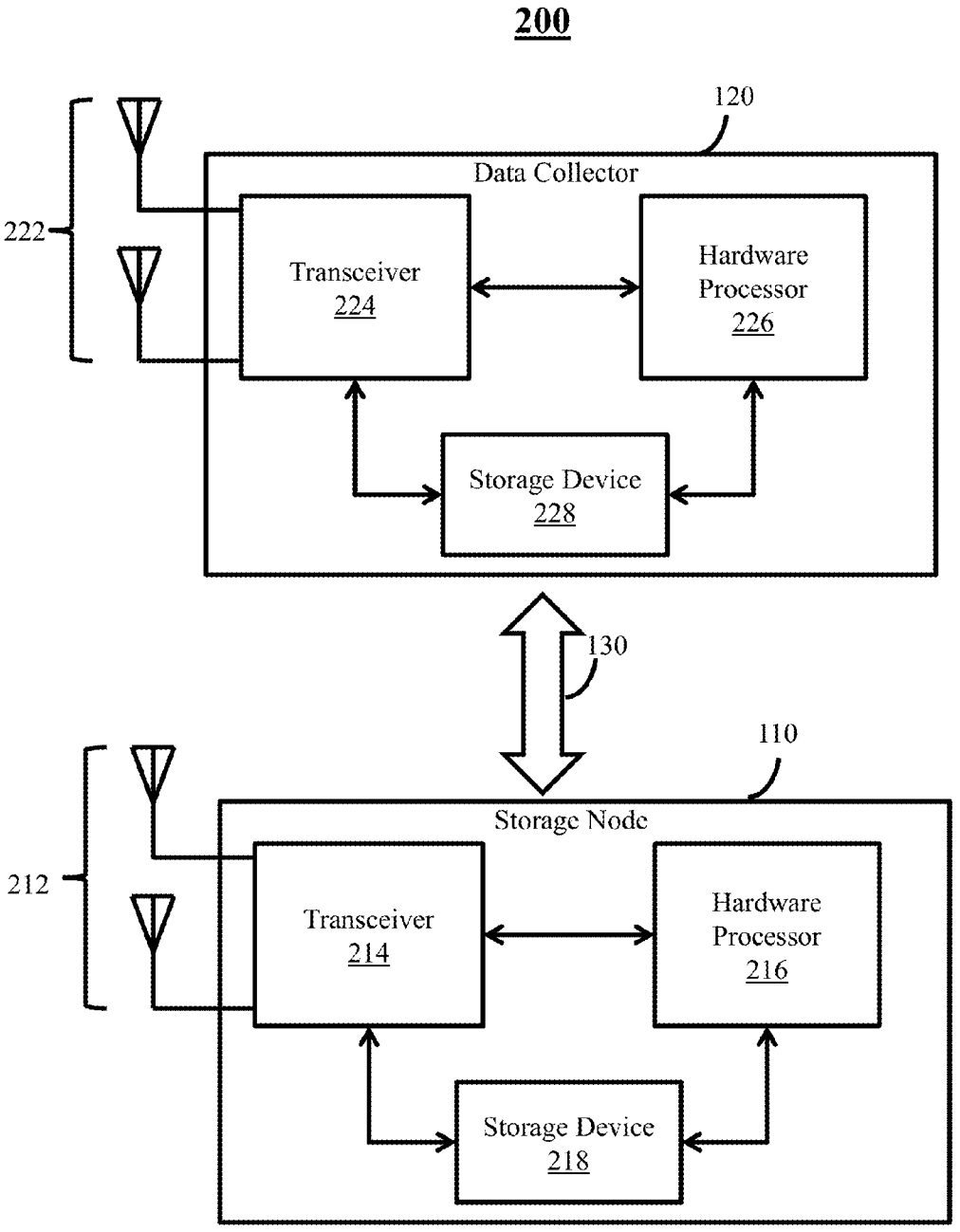

200
120
Data Collector
222
Transceiver
224
Hardware
Processor
226
Storage Device
228
130
110
Storage Node
212
Transceiver
214
Hardware
Processor
216
Storage Device
218

FIG. 2

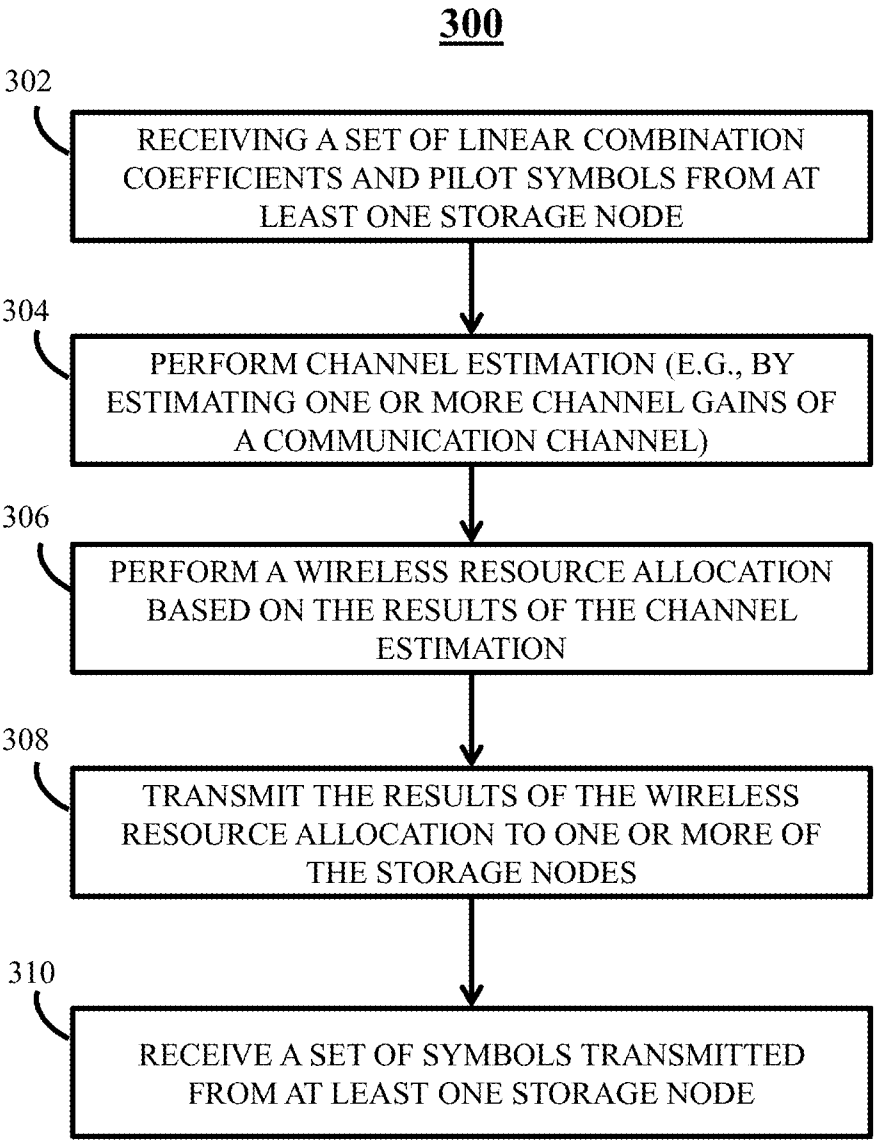

300
302
RECEIVING A SET OF LINEAR COMBINATION COEFFICIENTS AND PILOT SYMBOLS FROM AT LEAST ONE STORAGE NODE
304
PERFORM CHANNEL ESTIMATION (E.G., BY ESTIMATING ONE OR MORE CHANNEL GAINS OF A COMMUNICATION CHANNEL)
306
PERFORM A WIRELESS RESOURCE ALLOCATION BASED ON THE RESULTS OF THE CHANNEL ESTIMATION
308
TRANSMIT THE RESULTS OF THE WIRELESS RESOURCE ALLOCATION TO ONE OR MORE OF THE STORAGE NODES
310
RECEIVE A SET OF SYMBOLS TRANSMITTED FROM AT LEAST ONE STORAGE NODE

FIG. 3

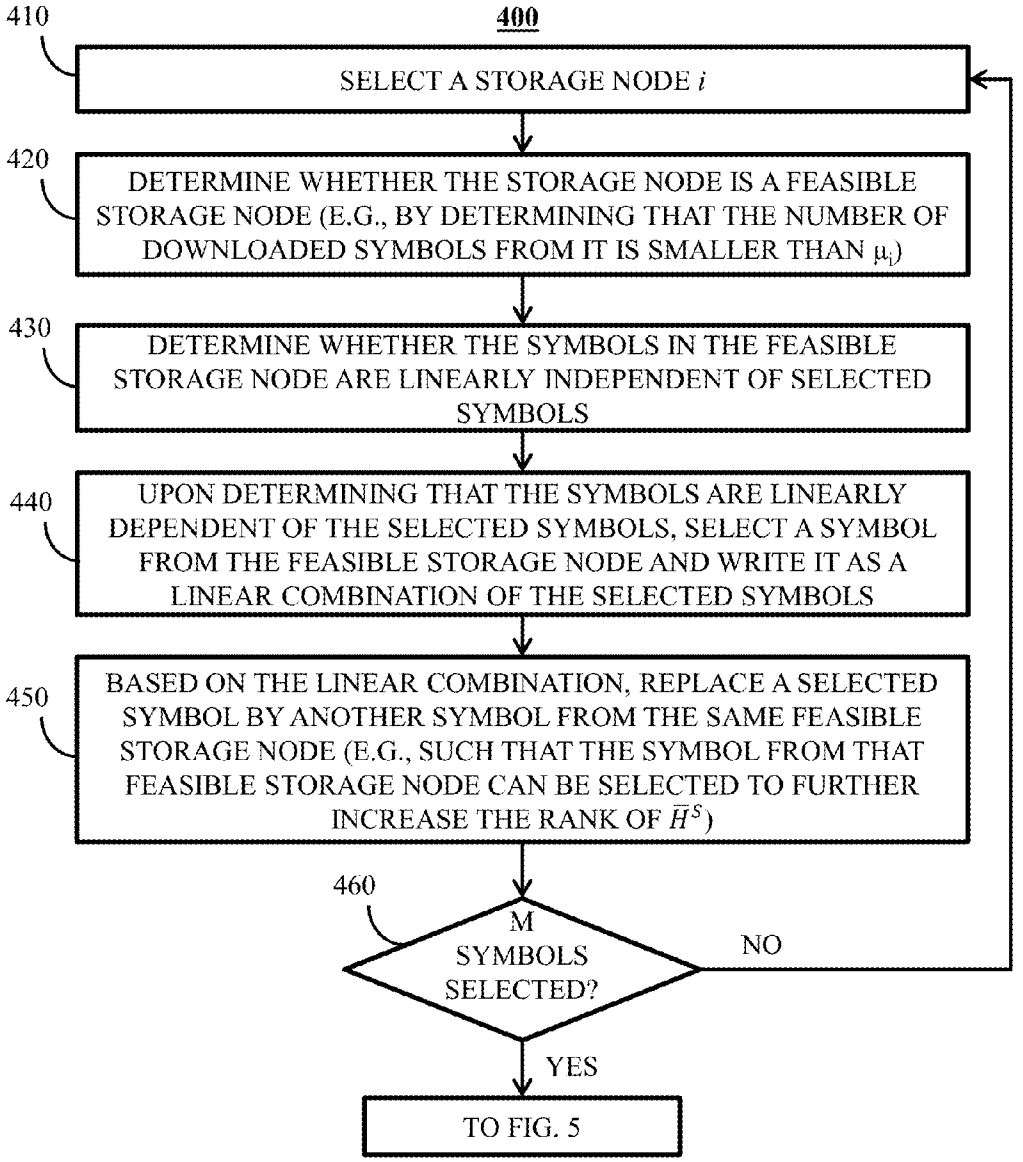

400
410
SELECT A STORAGE NODE i
420
DETERMINE WHETHER THE STORAGE NODE IS A FEASIBLE STORAGE NODE (E.G., BY DETERMINING THAT THE NUMBER OF DOWNLOADED SYMBOLS FROM IT IS SMALLER THAN μ_i)
430
DETERMINE WHETHER THE SYMBOLS IN THE FEASIBLE STORAGE NODE ARE LINEARLY INDEPENDENT OF SELECTED SYMBOLS
440
UPON DETERMINING THAT THE SYMBOLS ARE LINEARLY DEPENDENT OF THE SELECTED SYMBOLS, SELECT A SYMBOL FROM THE FEASIBLE STORAGE NODE AND WRITE IT AS A LINEAR COMBINATION OF THE SELECTED SYMBOLS
450
BASED ON THE LINEAR COMBINATION, REPLACE A SELECTED SYMBOL BY ANOTHER SYMBOL FROM THE SAME FEASIBLE STORAGE NODE (E.G., SUCH THAT THE SYMBOL FROM THAT FEASIBLE STORAGE NODE CAN BE SELECTED TO FURTHER INCREASE THE RANK OF $\bar{H}^S$)
460
M SYMBOLS SELECTED?
NO
YES
TO FIG. 5

FIG. 4

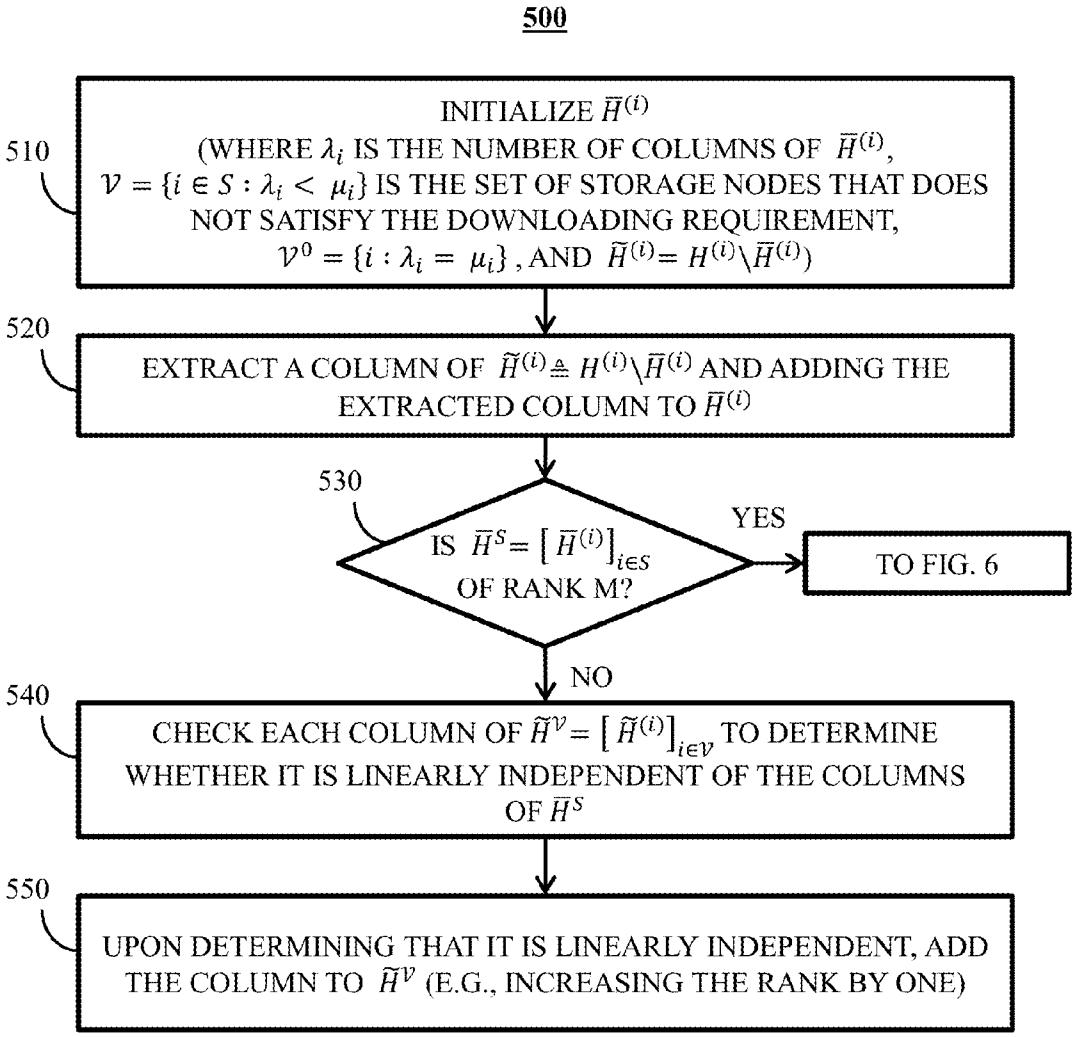

500
510
INITIALIZE $\bar{H}^{(i)}$
(WHERE λ_i IS THE NUMBER OF COLUMNS OF $\bar{H}^{(i)}$,
$\mathcal{V} = \{i \in S : \lambda_i < \mu_i\}$ IS THE SET OF STORAGE NODES THAT DOES NOT SATISFY THE DOWNLOADING REQUIREMENT,
$\mathcal{V}^0 = \{i : \lambda_i = \mu_i\}$, AND $\tilde{H}^{(i)} = H^{(i)} \backslash \bar{H}^{(i)}$)
520
EXTRACT A COLUMN OF $\tilde{H}^{(i)} \triangleq H^{(i)} \backslash \bar{H}^{(i)}$ AND ADDING THE EXTRACTED COLUMN TO $\bar{H}^{(i)}$
530
IS $\bar{H}^S = \left[\bar{H}^{(i)}\right]_{i \in S}$ OF RANK M?
YES
TO FIG. 6
NO
540
CHECK EACH COLUMN OF $\tilde{H}^{\mathcal{V}} = \left[\tilde{H}^{(i)}\right]_{i \in \mathcal{V}}$ TO DETERMINE WHETHER IT IS LINEARLY INDEPENDENT OF THE COLUMNS OF $\bar{H}^S$
550
UPON DETERMINING THAT IT IS LINEARLY INDEPENDENT, ADD THE COLUMN TO $\tilde{H}^{\mathcal{V}}$ (E.G., INCREASING THE RANK BY ONE)

FIG. 5

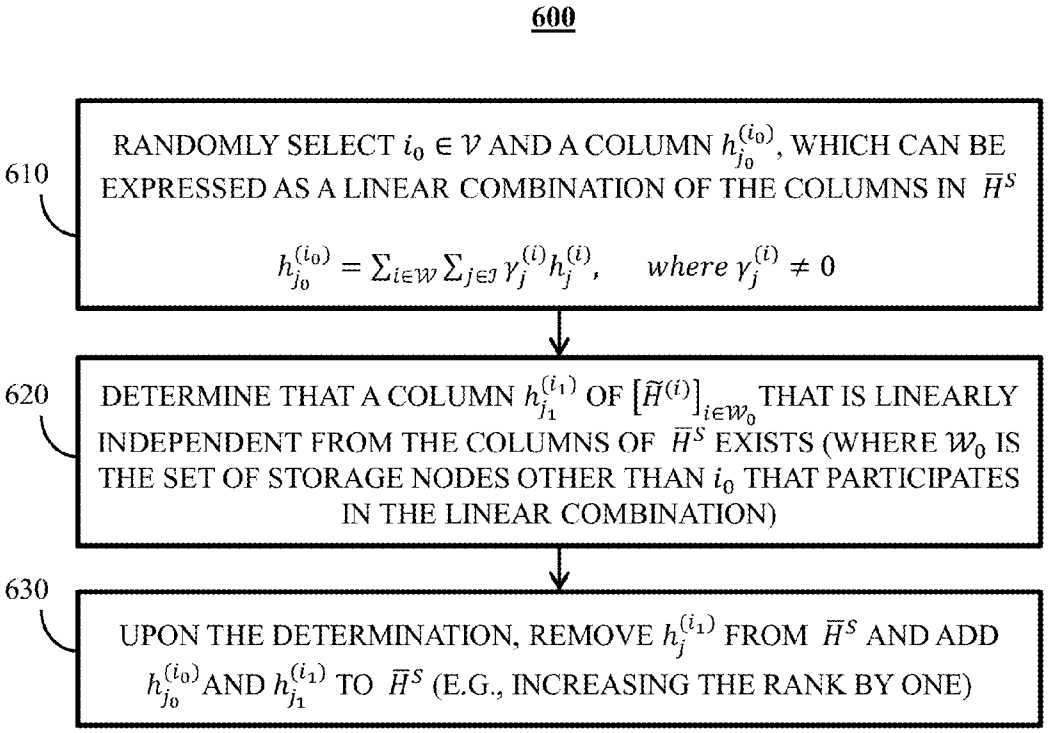

600
610
RANDOMLY SELECT $i_0 \in \mathcal{V}$ AND A COLUMN $h_{j_0}^{(i_0)}$, WHICH CAN BE EXPRESSED AS A LINEAR COMBINATION OF THE COLUMNS IN $\bar{H}^S$
$h_{j_0}^{(i_0)} = \sum_{i \in \mathcal{W}} \sum_{j \in \mathcal{J}} \gamma_j^{(i)} h_j^{(i)}, \quad where\ \gamma_j^{(i)} \neq 0$
620
DETERMINE THAT A COLUMN $h_{j_1}^{(i_1)}$ OF $[\widetilde{H}^{(i)}]_{i \in \mathcal{W}_0}$ THAT IS LINEARLY INDEPENDENT FROM THE COLUMNS OF $\bar{H}^S$ EXISTS (WHERE $\mathcal{W}_0$ IS THE SET OF STORAGE NODES OTHER THAN i_0 THAT PARTICIPATES IN THE LINEAR COMBINATION)
630
UPON THE DETERMINATION, REMOVE $h_j^{(i_1)}$ FROM $\bar{H}^S$ AND ADD $h_{j_0}^{(i_0)}$ AND $h_{j_1}^{(i_1)}$ TO $\bar{H}^S$ (E.G., INCREASING THE RANK BY ONE)

FIG. 6

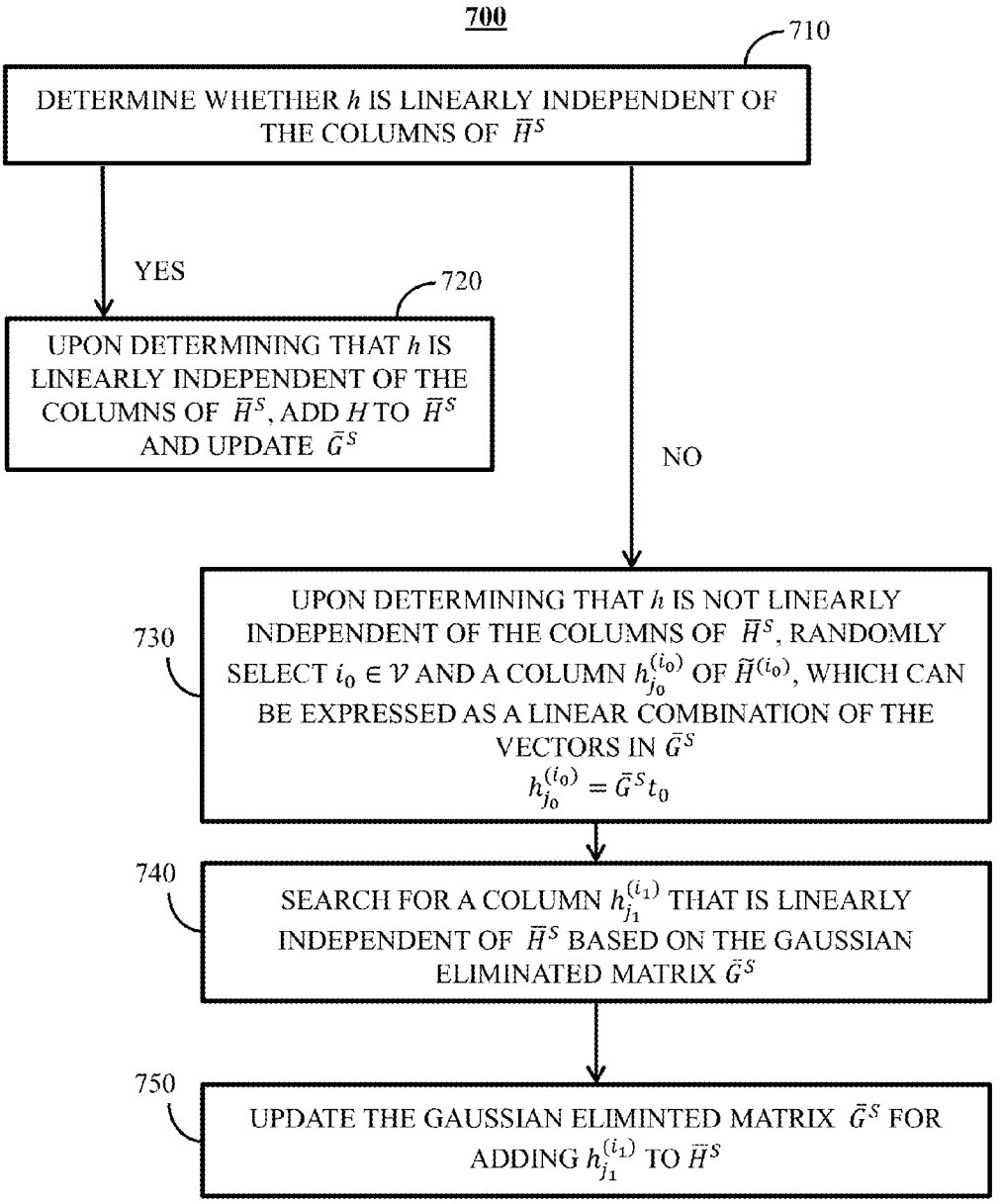

700
710
DETERMINE WHETHER h IS LINEARLY INDEPENDENT OF THE COLUMNS OF $\bar{H}^S$
YES
720
UPON DETERMINING THAT h IS LINEARLY INDEPENDENT OF THE COLUMNS OF $\bar{H}^S$, ADD H TO $\bar{H}^S$ AND UPDATE $\bar{G}^S$
NO
730
UPON DETERMINING THAT h IS NOT LINEARLY INDEPENDENT OF THE COLUMNS OF $\bar{H}^S$, RANDOMLY SELECT $i_0 \in \mathcal{V}$ AND A COLUMN $h_{j_0}^{(i_0)}$ OF $\tilde{H}^{(i_0)}$, WHICH CAN BE EXPRESSED AS A LINEAR COMBINATION OF THE VECTORS IN $\bar{G}^S$
$h_{j_0}^{(i_0)} = \bar{G}^S t_0$
740
SEARCH FOR A COLUMN $h_{j_1}^{(i_1)}$ THAT IS LINEARLY INDEPENDENT OF $\bar{H}^S$ BASED ON THE GAUSSIAN ELIMINATED MATRIX $\bar{G}^S$
750
UPDATE THE GAUSSIAN ELIMINTED MATRIX $\bar{G}^S$ FOR ADDING $h_{j_1}^{(i_1)}$ TO $\bar{H}^S$

FIG. 7

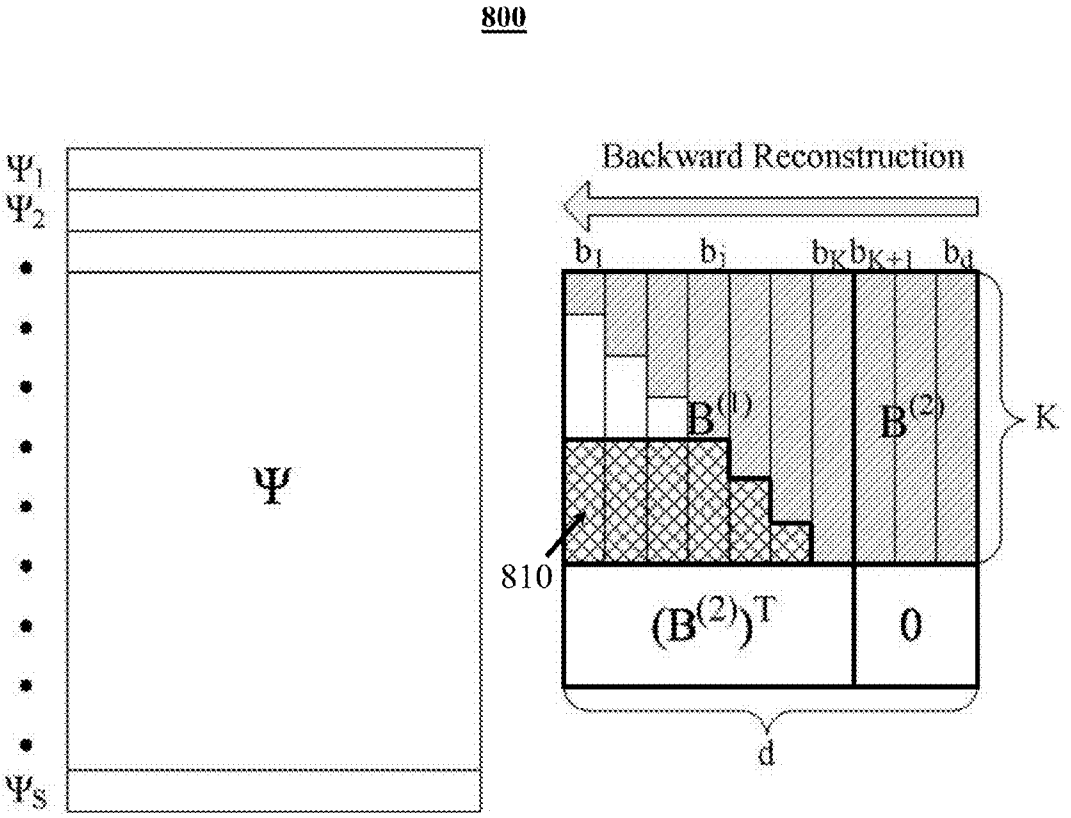

800
Ψ1
Ψ2
Ψ
ΨS
Backward Reconstruction
b1
bi
bK
bK+1
bd
B(1)
B(2)
K
810
(B(2))T
0
d

FIG. 8

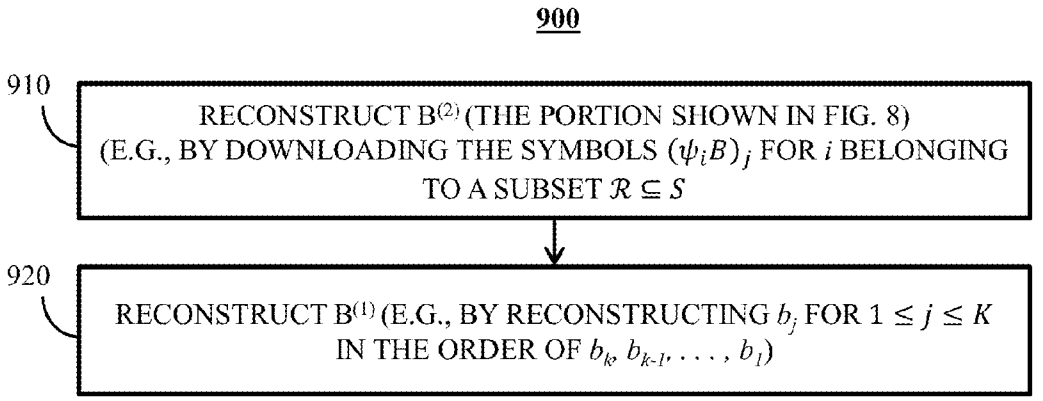

900
910
RECONSTRUCT $B^{(2)}$ (THE PORTION SHOWN IN FIG. 8) (E.G., BY DOWNLOADING THE SYMBOLS $(\psi_i B)_j$ FOR i BELONGING TO A SUBSET $\mathcal{R} \subseteq S$
920
RECONSTRUCT $B^{(1)}$ (E.G., BY RECONSTRUCTING b_j FOR $1 \leq j \leq K$ IN THE ORDER OF $b_k, b_{k-1}, \ldots, b_1$)

FIG. 9

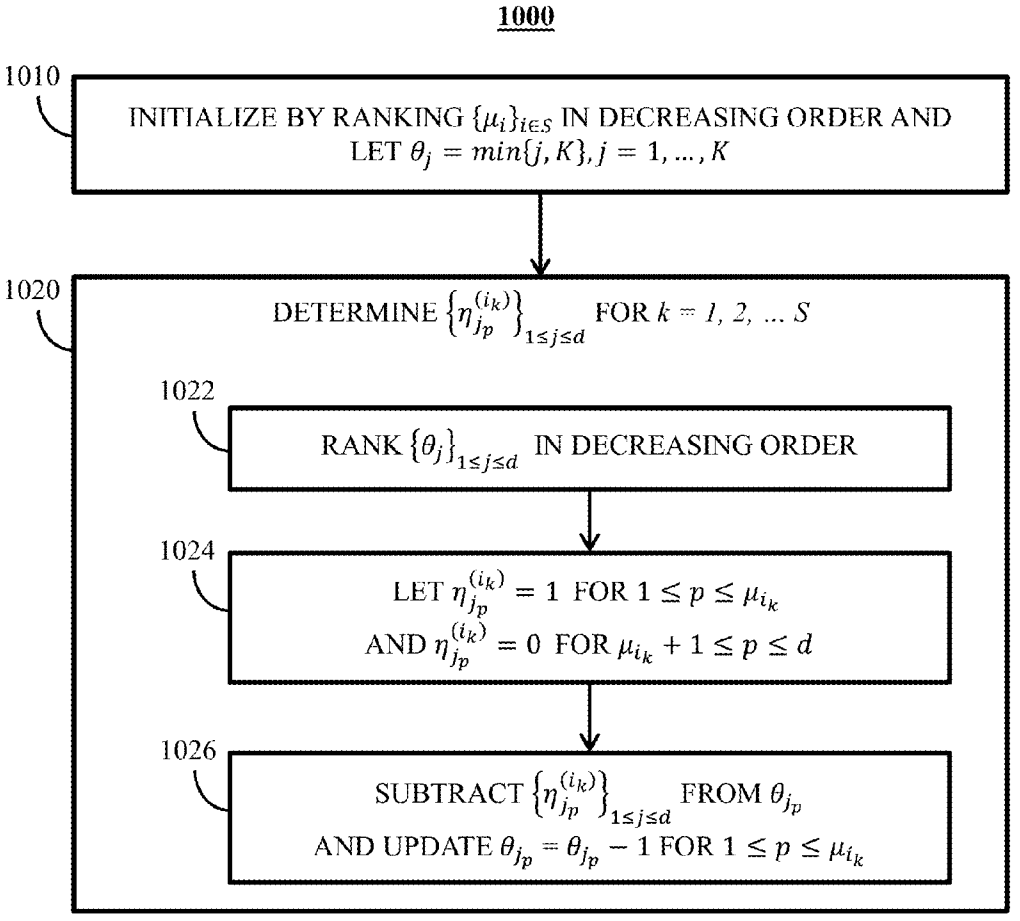

1000
1010
INITIALIZE BY RANKING $\{\mu_i\}_{i\in S}$ IN DECREASING ORDER AND LET $\theta_j = min\{j, K\}, j = 1, \ldots, K$
1020
DETERMINE $\left\{\eta_{j_p}^{(i_k)}\right\}_{1\le j\le d}$ FOR $k = 1, 2, \ldots S$
1022
RANK $\{\theta_j\}_{1\le j\le d}$ IN DECREASING ORDER
1024
LET $\eta_{j_p}^{(i_k)} = 1$ FOR $1 \le p \le \mu_{i_k}$
AND $\eta_{j_p}^{(i_k)} = 0$ FOR $\mu_{i_k} + 1 \le p \le d$
1026
SUBTRACT $\left\{\eta_{j_p}^{(i_k)}\right\}_{1\le j\le d}$ FROM θ_{j_p}
AND UPDATE $\theta_{j_p} = \theta_{j_p} - 1$ FOR $1 \le p \le \mu_{i_k}$

FIG. 10

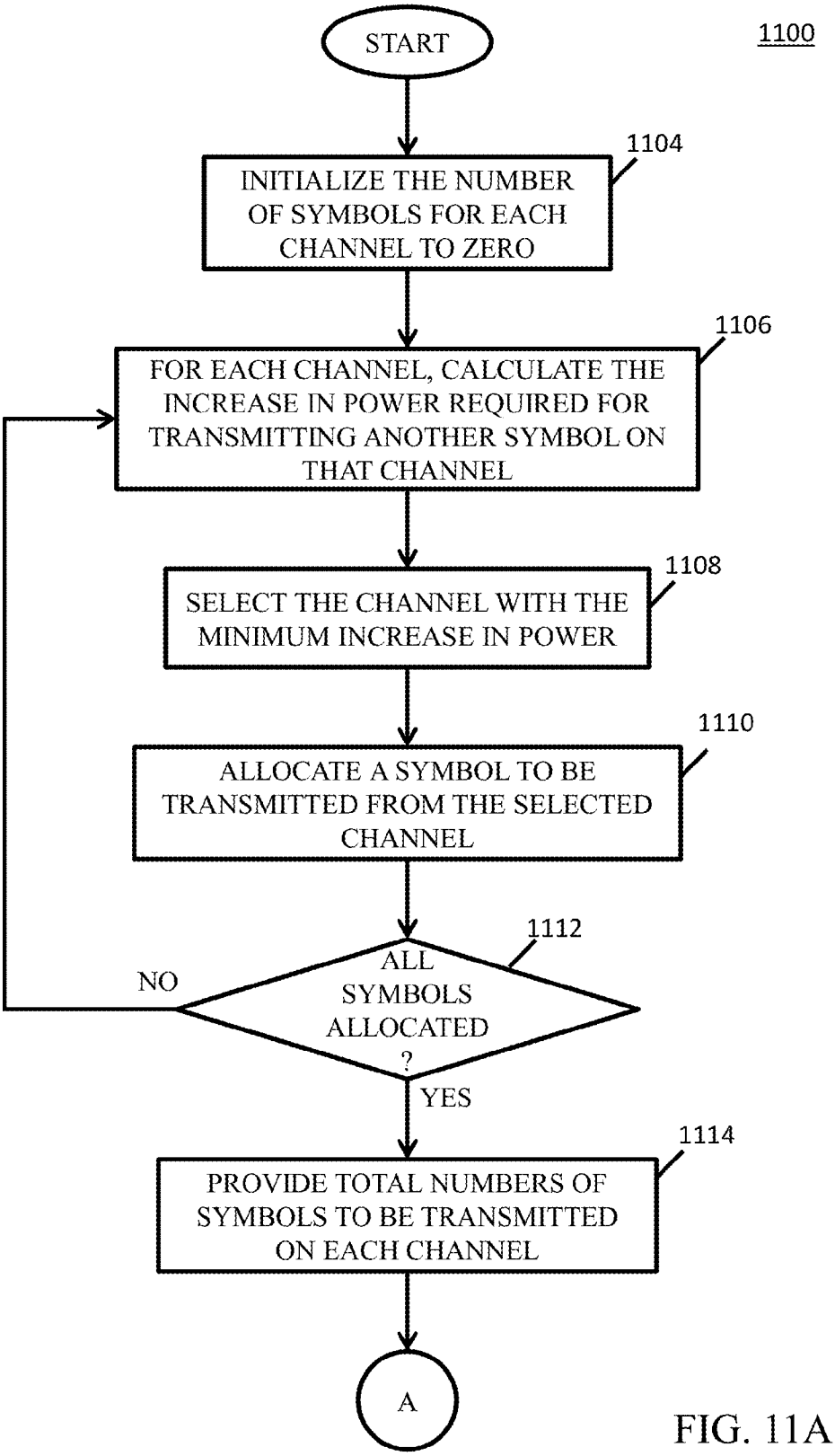

1100
START
1104
INITIALIZE THE NUMBER OF SYMBOLS FOR EACH CHANNEL TO ZERO
1106
FOR EACH CHANNEL, CALCULATE THE INCREASE IN POWER REQUIRED FOR TRANSMITTING ANOTHER SYMBOL ON THAT CHANNEL
1108
SELECT THE CHANNEL WITH THE MINIMUM INCREASE IN POWER
1110
ALLOCATE A SYMBOL TO BE TRANSMITTED FROM THE SELECTED CHANNEL
1112
ALL SYMBOLS ALLOCATED ?
NO
YES
1114
PROVIDE TOTAL NUMBERS OF SYMBOLS TO BE TRANSMITTED ON EACH CHANNEL
A

FIG. 11A

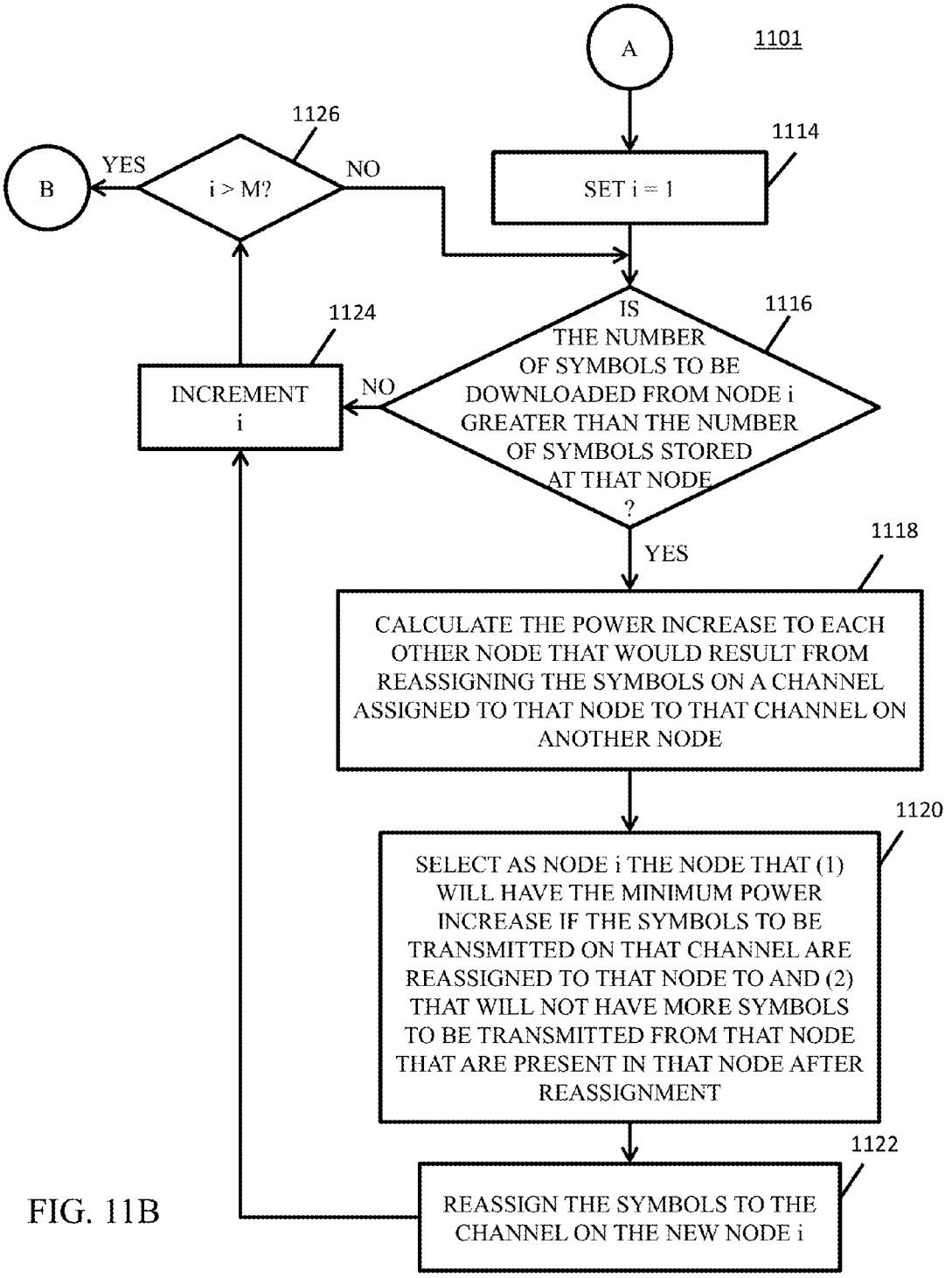

A
1101
SET i = 1
1114
IS THE NUMBER OF SYMBOLS TO BE DOWNLOADED FROM NODE i GREATER THAN THE NUMBER OF SYMBOLS STORED AT THAT NODE ?
1116
NO
INCREMENT i
1124
i > M?
1126
YES
B
YES
CALCULATE THE POWER INCREASE TO EACH OTHER NODE THAT WOULD RESULT FROM REASSIGNING THE SYMBOLS ON A CHANNEL ASSIGNED TO THAT NODE TO THAT CHANNEL ON ANOTHER NODE
1118
SELECT AS NODE i THE NODE THAT (1) WILL HAVE THE MINIMUM POWER INCREASE IF THE SYMBOLS TO BE TRANSMITTED ON THAT CHANNEL ARE REASSIGNED TO THAT NODE TO AND (2) THAT WILL NOT HAVE MORE SYMBOLS TO BE TRANSMITTED FROM THAT NODE THAT ARE PRESENT IN THAT NODE AFTER REASSIGNMENT
1120
REASSIGN THE SYMBOLS TO THE CHANNEL ON THE NEW NODE i
1122

FIG. 11B

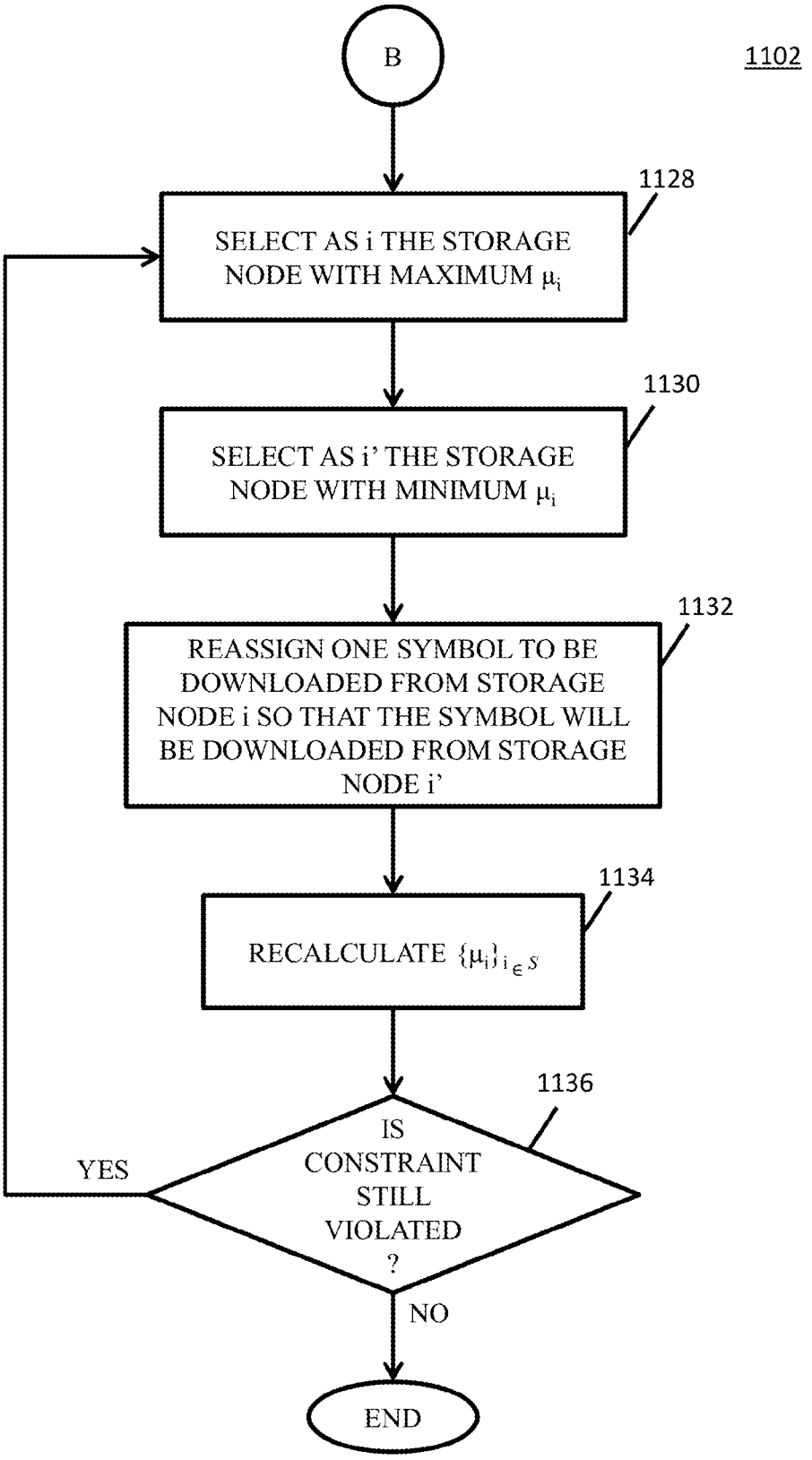

B
1102
1128
SELECT AS i THE STORAGE NODE WITH MAXIMUM μ_i
1130
SELECT AS i' THE STORAGE NODE WITH MINIMUM μ_i
1132
REASSIGN ONE SYMBOL TO BE DOWNLOADED FROM STORAGE NODE i SO THAT THE SYMBOL WILL BE DOWNLOADED FROM STORAGE NODE i'
1134
RECALCULATE $\{\mu_i\}_{i \in S}$
1136
IS CONSTRAINT STILL VIOLATED ?
YES
NO
END

FIG. 11C

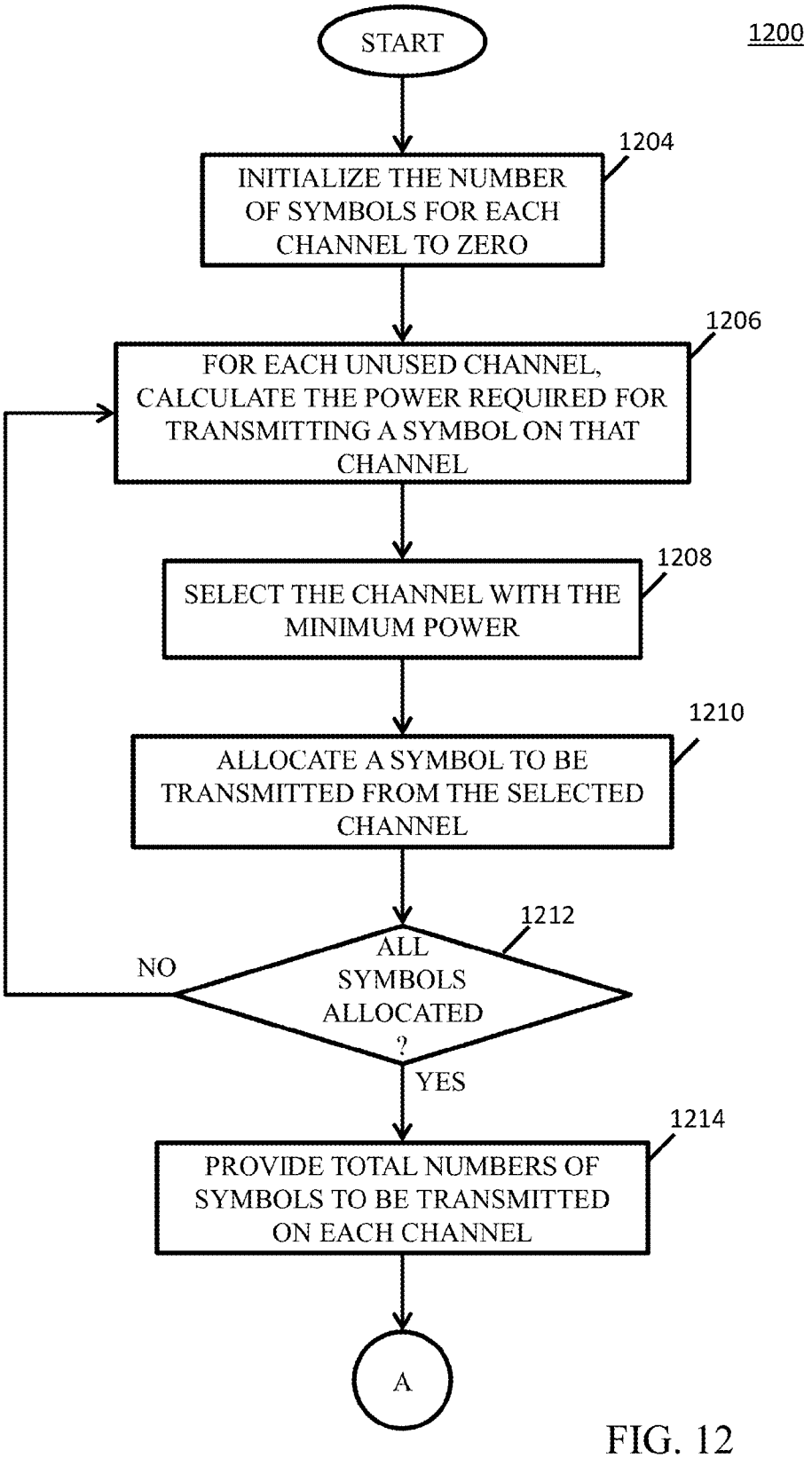

1200
START
1204
INITIALIZE THE NUMBER OF SYMBOLS FOR EACH CHANNEL TO ZERO
1206
FOR EACH UNUSED CHANNEL, CALCULATE THE POWER REQUIRED FOR TRANSMITTING A SYMBOL ON THAT CHANNEL
1208
SELECT THE CHANNEL WITH THE MINIMUM POWER
1210
ALLOCATE A SYMBOL TO BE TRANSMITTED FROM THE SELECTED CHANNEL
1212
ALL SYMBOLS ALLOCATED ?
NO
YES
1214
PROVIDE TOTAL NUMBERS OF SYMBOLS TO BE TRANSMITTED ON EACH CHANNEL
A

FIG. 12

METHODS, SYSTEMS, AND MEDIA FOR PARTIAL DOWNLOADING IN WIRELESS DISTRIBUTED NETWORKS

CROSS REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Patent Application No. 61/674,264, filed Jul. 20, 2012, which is hereby incorporated by reference herein in its entirety.

STATEMENT REGARDING GOVERNMENT FUNDED RESEARCH

This invention was made with government support under Grant No. CCF-0726480 awarded by the National Science Foundation (NSF) and Grant No. N00014-08-1-0318 awarded by the Office of Naval Research (ONR). The government has certain rights in the invention.

TECHNICAL FIELD

Methods, systems, and media for partial downloading in wireless distributed networks are provided.

BACKGROUND

Distributed storage systems are generally used to store data in a distributed manner to provide reliable access to the stored data. For example, a data file of size M can be divided into k fragments, each of size M/k. Each of the k fragments can be encoded and stored in a storage node of a distributed storage system. In such an example, the original data file can be recovered from a set of k encoded fragments. However, conventional approaches to reconstructing data stored in a distributed storage network have limited performance, especially for wireless distributed storage networks. For example, such approaches generally include downloading all of the symbols from a subset of the storage nodes. Such a full-downloading approach becomes inefficient in a wireless network, where wireless channels may not offer sufficient bandwidths for full downloading (e.g., due to channel fading). Moreover, full-downloading suffers from power constraints of the wireless network.

SUMMARY

In accordance with some embodiments of the disclosed subject matter, methods, systems, and media for partial downloading in wireless distributed networks are provided.

In some embodiments, methods for selecting numbers of symbols to be transmitted on a plurality of channels are provided, the methods comprising: for each of the plurality of channels, calculating using a hardware processor an increase in power that will be used by that channel if it transmits a symbol; selecting one of the plurality of channels with the smallest increase in power using the hardware processor; and allocating the symbol to the one of the plurality of channels using the hardware processor.

In some embodiments, systems for selecting numbers of symbols to be transmitted on a plurality of channels are provided, the systems comprising: at least one hardware processor that: for each of the plurality of channels, calculates an increase in power that will be used by that channel if it transmits a symbol; selects one of the plurality of channels with the smallest increase in power; and allocates the symbol to the one of the plurality of channels.

In some embodiments, non-transitory computer-readable media containing computer-executable instructions that, when executed by a processor, cause the processor to perform a method for selecting numbers of symbols to be transmitted on a plurality of channels are provide, the method comprising: for each of the plurality of channels, calculating an increase in power that will be used by that channel if it transmits a symbol; selecting one of the plurality of channels with the smallest increase in power; and allocating the symbol to the one of the plurality of channels.

BRIEF DESCRIPTION OF THE DRAWINGS

Various objects, features, and advantages of the disclosed subject matter can be more fully appreciated with reference to the following detailed description of the disclosed subject matter when considered in connection with the following drawings, in which like reference numerals identify like elements.

FIG. **1** is a generalized schematic diagram of an example of a distributed storage system in accordance with some embodiments of the disclosed subject matter.

FIG. **2** is a generalized schematic diagram of an example of a data collector and a storage node of FIG. **1** that can be used in accordance with some embodiments of the disclosed subject matter.

FIG. **3** is a flow chart of an example of a process for partial downloading in accordance with some embodiments of the disclosed subject matter.

FIGS. **4-6** are flow charts of an example of processes for determining whether a condition is sufficient for μ-reconstructability for the minimum-storage regenerating (MSR) point in accordance with some embodiments of the disclosed subject matter.

FIG. **7** is a flow chart of an example of a recursive symbol selection process for a partial downloading scheme for the MSR point in accordance with some embodiments of the disclosed subject matter.

FIG. **8** is an illustrative example of data reconstruction for the coding scheme at the minimum-bandwidth regenerating (MBR) point in accordance with some embodiments of the disclosed subject matter.

FIG. **9** is a flow chart of an example of a process for a partial downloading scheme that includes performing a backward reconstruction in accordance with some embodiments of the disclosed subject matter.

FIG. **10** is a flow chart of an example of a symbol selection process for a partial downloading scheme for the MBR point in accordance with some embodiments of the disclosed subject matter.

FIG. **11**A is a flow chart of an example of a process for minimizing power during reconstruction transmission in accordance with some embodiments of the disclosed subject matter.

FIG. **11**B is a flow chart of an example of a process for adjusting a transmission allocation for the MSR point in accordance with some embodiments of the disclosed subject matter.

FIG. **11**C is a flow chart of an example of a process for adjusting a transmission allocation for the MBR point in accordance with some embodiments of the disclosed subject matter.

FIG. **12** is a flow chart of an example of a process for minimizing power during regeneration transmission in accordance with some embodiments of the disclosed subject matter.

DETAILED DESCRIPTION

In accordance with various embodiments, as described in more detail below, mechanisms (e.g., systems, methods, media, etc.) for partial downloading in wireless distributed networks are provided. Such mechanisms can be used in a variety of applications. For example, the mechanisms can be used to retrieve and/or reconstruct data stored in a distributed storage system. As another example, the mechanisms can be used to perform node regeneration in some embodiments in which one or more storage nodes in a distributed storage system fail.

In some embodiments, the mechanisms can be implemented in a distributed storage network including a data collector and multiple storage nodes. The data collector can communicate with one or more of the storage nodes through any suitable communication channels. For example, in some embodiments, the storage nodes can be connected to the data collector through multiple orthogonal wireless channels.

In some embodiments, the mechanisms can store data in the storage nodes in a distributed manner. For example, a data block containing M symbols can be stored in S storage nodes. In a more particular example, each of the S storage nodes can store α symbols that can be used to reconstruct the data block. In some embodiments, each of the α symbols can be generated from one or more of the M symbols based on a set of linear combination coefficients.

In some embodiments, the data collector can reconstruct the data block from one or more of the storage nodes. For example, the data reconstruction can be performed based on a partial downloading scheme. In a more particular example, the data collector can download a suitable set of symbols from each storage node. In some embodiments, the data collector can determine the number of symbols to be downloaded from each channel and storage node based on suitable channel and power allocation schemes. In some embodiments, the data collector can also select the set of symbols to be downloaded based on one or more suitable symbol selection schemes. For example, the set of symbols can be selected from α symbols stored in a given storage node based on linear dependences among the symbols. As another example, the set of symbols can be selected based on a recursive algorithm. As another example, the data reconstruction can be performed based on a full downloading scheme. In a more particular example, the data collector can download α symbols for a subset of the storage nodes.

In some embodiments, when a storage node fails or leaves the distributed storage system, the mechanisms can regenerate the symbols stored in the failed node (e.g., α symbols) and create a new storage node. For example, the symbols can be regenerated by downloading a suitable number of symbols from one or more of the surviving storage nodes. In some embodiments, for example, the new storage node can download β symbols from each of a set of d surviving storage nodes. In such an example, a total number of $d\beta$ symbols can be downloaded from the surviving storage nodes for node regeneration.

In some embodiments, these mechanisms can be implemented in a wireless cloud storage network, where a large number of users need to download data symbols with limited bandwidth. In such a network, a large amount of data that needs to be downloaded by a mobile device can be obtained by partially downloading data symbols from storage nodes for data reconstruction and node regeneration. This can be performed, for example, while conserving power and bandwidth of the wireless cloud storage network.

It should be noted that, as used herein, for a family of matrices $\{H^{(i)}\}_{i\in\mathcal{A}}$ with the same number of rows, $H^{\mathcal{A}} \triangleq [H^{(i)}, i\in\mathcal{A}]$ can be the matrix obtained by horizontally concatenating $H^{(i)}$ for $i\in\mathcal{A}$, e.g., $H^{\mathcal{A}}=[H^{(1)}|H^{(2)}]$ for $\mathcal{A}=\{1,2\}$. In some embodiments, $H_0 \subseteq H$ can denote that H_0 is a submatrix of H by extracting columns of H. In some embodiments, $H_0 \subset H$ can denote that $H_0 \subseteq H$ and $H_0 \neq H$. In some embodiments, for $H_0 \subseteq H$, $H \backslash H_0$ can be the submatrix of H that includes all columns of H but not in H_0. In some embodiments, span(H) can be the space spanned by the columns of H. In some embodiments, for two spaces Q_0 and Q, $Q_0 \subseteq Q$ can denote that Q_0 is a subspace of Q. In some embodiments, $Q_0 \subset Q$ can denote that $Q_0 \subseteq Q$ and $Q_0 \neq Q$. In some embodiments, if $H_0 \subseteq H$, $\text{span}(H_0) \subseteq \text{span}(H)$. In some embodiments, rank(H) can be the column rank of the matrix H that equals to the dimension of space span(H).

Turning to FIG. **1**, a generalized schematic diagram of an example **100** of a distributed storage system in accordance with some embodiments of the disclosed subject matter is shown.

As illustrated, system **100** can include multiple storage nodes **110**, each of which is capable of storing a suitable amount of data. For example, system **100** can store a data block in S storage nodes **110** in a distributed manner. In a more particular example, the data block can contain M symbols that can be denoted as:

$$s=[s_1,s_2,\ldots,s_M]^T \tag{1}$$

In such an example, each storage node **100** can store α symbols that are generated based on one or more of the M symbols. In some embodiments, each of the α symbols may be a packet of subsymbols in a field GF(q) and may contain B bits. More particularly, for example, each storage node **110** (e.g., storage node i) can store a linear combination of data symbols that can be denoted as:

$$m^{(i)}=[m_1^{(i)},\ldots,m_\alpha^{(i)}], \tag{2}$$

where i denotes the index of the storage nodes. In some embodiments, each of the symbols given in Equation 2 can be obtained based on a set of the M symbols as follows:

$$m_j^{(i)}=\Sigma_{k=1}^{M} h_{kj}^{(i)} s_k = s^T h_j^{(i)}, 1\leq j\leq\alpha, \tag{3}$$

where the coefficients $h_j^{(i)}=[h_{1j}^{(i)}, h_{2j}^{(i)},\ldots,h_{Mj}^{(i)}]^T \in GF(q)^M$, for $1\leq j\leq\alpha$. In some embodiments, an encoding matrix can be defined as follows:

$$H^{(i)} \triangleq [h_{1j}^{(i)}, h_{2j}^{(i)},\ldots,h_{Mj}^{(i)}] \in GF(q)^{M\times\alpha}. \tag{4}$$

In some embodiments, equation (2) can be converted into the following format based on equations (3) and (4):

$$(m^{(i)})^T = s^T H^{(i)} \text{ for } i\in S. \tag{5}$$

As shown in FIG. **1**, system **100** can also include one or more data collectors **120**. In some embodiments, data collector **120** can perform data reconstruction by reconstructing the data stored in system **100** based on a suitable set of symbols stored in storage nodes **110**. For example, to reconstruct the data block defined in equation (1), data collector **120** can download a set of symbols from one or more of storage nodes **110**. Data collector **120** can then reconstruct the original data contained in the data block based on the encoding matrices associated with the storage nodes (e.g., $\{H^{(i)}\}_{i\in S}$).

In a more particular example, data collector **120** can perform data reconstruction based on a partial downloading scheme by downloading a set of symbols from one or more of storage nodes **110**. More particularly, for example, data collector **120** can perform channel and power allocation and determine the number of symbols downloaded from each

storage node **110**. Data collector **120** can then download a suitable number of symbols from each storage node **110** (e.g., downloading μ_i symbols from a storage node i).

In another more particular example, data collector **120** can download all symbols required to perform data reconstruction from a set of storage nodes **110**. More particularly, for example, data collector **120** can download α symbols from each of K storage nodes **110**, wherein K is not greater than S. In such an example, a total number of $K\alpha$ symbols can be downloaded from K storage nodes **110**.

In some embodiments, data collector **120** can be connected to each storage node **110** through one or more communication channels **130** that can include a command channel **132**, a data channel **134**, a feedback channel **136**, etc. Any suitable information can be transmitted through communication channels **130** to facilitate data reconstructions and/or node regeneration. For example, in some embodiments, data collector **120** can communicate particular information with each storage node **110** through one or more communication channels. In a more particular example, a storage node **110** can transmit an encoding matrix associated with the storage node to data collector **120** through command channel **132**. In another more particular example, a storage node **110** can transmit one or more data symbols to data collector **120** through data channel **134** (e.g., an orthogonal frequency-division multiple access channel or OFDMA channel). In yet another more particular example, data collector **120** can transmit information about the number and/or the identities of the symbols to be downloaded from one or more storage nodes **110** through feedback channel **136**.

In some embodiments, system **100** can regenerate the data stored in a failed storage node. For example, in some embodiments in which a storage node **110** that stores a symbols fails or leaves system **100**, system **100** can regenerate the α symbols and create a new storage node. In a more particular example, the α symbols can be regenerated by downloading a suitable number of symbols from one or more of the surviving storage nodes. In some embodiments, for example, the new storage node can download β symbols from each of a set of d surviving storage nodes. In such an example, a total number of $\gamma=d\beta$ symbols can be downloaded from the surviving storage nodes for node regeneration. In some embodiments, upon the creation of the new storage node, data collector **120** can reconstruct the data stored in system **100** by downloading $K\alpha$ symbols from K storage nodes **110** as described above.

In some embodiments, the amount of symbols downloaded for data reconstruction (e.g., $K\alpha$) and the amount of symbols downloaded for node regeneration (e.g. $\gamma=d\beta$) can be described using an optimal tradeoff curve. For example, the optimal tradeoff curve may have a minimum-storage regenerating (MSR) point corresponding to coding schemes with the best efficiency for data reconstruction (e.g., $K\alpha=M$). As another example, the optimal tradeoff curve may have a minimum-bandwidth regenerating (MBR) point corresponding to coding schemes with the best efficiency for node regeneration (e.g., $d\beta=\alpha$).

Turning to FIG. **2**, a generalized schematic diagram of an example of a data collector and a storage node of FIG. **1** that can be used in accordance with some embodiments of the disclosed subject matter is shown.

As illustrated, storage node **110** can include one or more antennas **212**, a transceiver **214**, a hardware processor **216**, a storage device **218**, and/or any other suitable components. In some embodiments, transceiver **214** can transmit data symbols, linear combination coefficients, and/or other suitable information to data collector **120** through antennas **212**. Transceiver **214** can also receive feedback signals containing information about one or more symbols to be downloaded from storage node **110** through antennas **212**. In some embodiments, transceiver **214** can pass the feedback signals to hardware processor **216**. Hardware processor **216** can then process the feedback signals and identify the symbols to be transmitted to data collector. In some embodiments, storage node **110** can store suitable data in storage device **218**. For example, storage device **218** can store a data symbols, a set of linear combination coefficient (e.g., an encoding matrix) associated with the data symbols, and/or other suitable data

As shown, data collector **120** can include one or more antennas **222**, a transceiver **224**, a hardware processor **226**, a storage device **228**, and/or any other suitable components. In some embodiments, transceiver **224** can receive suitable data and/or commands transmitted from one or more storage nodes **110** through one or more antennas **222**. The data and/or commands can then be stored in storage device **228** and/or passed to hardware processor **226**. Hardware processor **226** can perform channel estimation, wireless resource allocation, and/or other suitable functions based on the received data, commands, and/or other suitable information. In some embodiments, hardware processor **226** can generate one or more feedback signals containing information about the results of the channel estimation and/or wireless resource allocation. In some embodiments, the feedback signals can be transmitted to one or more storage nodes **110** through transceiver **224** and antenna(s) **222**.

In some embodiments, storage node **110** and data collector **120** can be implemented in any suitable devices. For example, they can be implemented in mobile computers, mobile telephones, mobile access cards, wireless routers, wireless access points and/or any other suitable wireless devices.

In some embodiments, each of transceivers **214** and **224** can include both a receiver and a transmitter in some embodiments. In some embodiments, each transceiver can include one or more multi-input multi-output (MIMO) transceivers where each includes multiple antennas (e.g., such as two transmit antennas and four receive antennas (some of which may also be transmit antennas)).

In some embodiments, each of hardware processors **216** and **226** can include any suitable hardware processor, such as a microprocessor, a micro-controller, digital signal processor, dedicated logic, and/or any other suitable circuitry.

In some embodiments, each of storage devices **218** and **228** can include any suitable circuitry that is capable of storing data symbols, linear combination coefficients, feedback signals, computer readable instructions, etc. For example, each of storage devices **218** and **228** can include a hard drive, a solid state storage device, a removable storage device, etc.

It should be noted that storage node **110** and data collector **120** can include any other suitable components. For example, in some embodiments, each of storage node **110** and data collector **120** can include a modulator, a demodulator, etc.

Turning to FIG. **3**, a flow chart of an example **300** of a process for partial downloading for wireless distributed storage networks in accordance with some embodiments of the disclosed subject matter is shown. In some embodiments, process **300** can be implemented in a data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**).

As illustrated, process **300** can start by receiving a set of linear combination coefficients and pilot symbols from at least one storage node at **302**. The set of linear combination coefficients can contain any suitable information about one or more symbols stored in the storage node. For example, the combination coefficients can include one or more encoding matrices as defined in equation (4). In a more particular example, a storage node can transmit an encoding matrix $H^{(i)}$

that is associated with the storage node through a command channel. Additionally, the storage node can transmit one or more pilot symbols to the data collector through a data channel.

Next, at **304**, process **300** can perform a channel estimation on one or more communication channels. For example, the data collector can estimate one or more channel gains of a communication channel that connects a particular storage node to the data collector. In a more particular example, channel gains $\{g_j^{(i)}\}_{j\in\mathcal{N}}$ can be estimated for the storage node, where $g_j^{(i)}$ denotes the complex gain of channel j from storage node i ($i\in S$) to the data collector. In some embodiments, the channel gains can be estimated based on any suitable models. For example, the channel gains between the data collector and a particular storage node can be modeled by a complex Gaussian random variable $\mathcal{N}_{\mathcal{C}}(0, d^{-2})$, where d represents the distance between the data collector and the storage node.

At **306**, process **300** can perform a wireless resource allocation based on the results of the channel estimation. For example, the data collector can estimate the number of symbols that can be transmitted from a particular storage node to the data collector over a particular communication channel based on the estimated channels gains. In a more particular example, the data collector can be connected to S storage nodes via N orthogonal wireless channels $\mathcal{N}=\{1, 2, \ldots, N\}$. In some embodiments, each of the wireless channels can have a suitable bandwidth (e.g., a bandwidth W) and a suitable duration (e.g., a duration T). In such an example, the number of symbols that can be transmitted from storage node i ($i\in S$) to the data collector over channel j ($j\in\mathcal{N}$) can be estimated as follows:

$$c\left(|g_j^{(i)}|^2 P_j\right) = \frac{WT}{B} log_2\left(1 + k\frac{|g_j^{(i)}|P_j}{\sigma^2}\right), \quad (6)$$

where P_j denotes the transmission power of channel j; σ^2 is the power of background noise; W denotes the bandwidth of channel j; T denotes the duration of channel j; and $k<1$ accounts for the rate loss due to the practical modulation and coding, compared with the ideal case of Gaussian signaling and infinite-length code. In some embodiments, the transition power of a communication channel can be a function of the amount of information transmitted through the communication channel.

In some embodiments, data can be transmitted over a channel in unit of a symbol. For example, one or multiple symbols can be transmitted over the channel. In such an example, wireless resources (e.g., power, bandwidth, etc.) can be allocated to minimize the total transmission power of N channels while achieving successful data reconstruction.

At **308**, process **300** can transmit the results of the wireless resource allocation to one or more of the storage nodes. In some embodiments, the results of the wireless resource allocation can be transmitted in any suitable manner. For example, the data collector can generate a feedback signal containing information about the number of the symbols and/or the identities of the symbols to be downloaded from a particular storage node. The data collector can then transmit the feedback signal to the particular storage node through a suitable communication channel (e.g., a feedback channel that connects the data collector to the particular storage node).

At **310**, process **300** can receive a set of symbols transmitted from at least one storage node. For example, a storage node can transmit a set of symbols to the data collector based on the feedback signal transmitted from the data collector. In a more particular example, in response to receiving the feedback signal, the storage node can identify the symbols chosen by the data collector by wireless resource allocation and transmit the identified symbols to the data collector through a suitable communication channel (e.g., a data channel that connects the data collector to the storage node).

As described herein, a portion of the symbols stored in each storage node can be downloaded. For example, a data collector can use a partial downloading scheme that downloads a portion of the symbols from any suitable storage node. In a more particular example, for storage nodes and a data collector linked by wireless channels, the data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**) can download symbols in a roughly even manner from the storage nodes such that the total number of downloaded symbols is equal to the number of data symbols to be reconstructed.

It should be noted that the reconstructability of the original data can be considered when performing such a partial downloading scheme. For example, for $i\in S$, let μ_i be the number of symbols downloaded from storage node i. Since $\mu_i\leq\alpha$, it can be assumed that the downloaded symbols are linear combinations of the symbols in node i given by $s^T H^{(i)} A^{(i)}$, where $A^{(i)}$ is an $\alpha\times\mu_i$ matrix. The data s can then be reconstructed from the downloaded symbols $s^T[H^{(i)}A^{(i)}]_{i\in S}$ if:

$$\text{rank}([H^{(i)}A^{(i)}]_{i\in S})=M \quad (7)$$

For each $i\in S$, the matrix $A^{(i)}$ can be a full column rank, since otherwise at least one downloaded symbol can be expressed as a linear combination of other symbols downloaded from the same storage node, which means this symbol is redundant and should be removed to reduce the downloading bandwidth.

However, it should be noted that the search for the linear combination matrices $\{A^{(i)}\}_{i\in S}$ that satisfy the above-mentioned equation (7) may be computationally prohibitive. While it may be simpler to directly download the stored symbols from each storage node without performing such a linear combination, it can be determined whether there is a loss in optimality. That is, for some symbols $\{\mu_i\}_{i\in S}$, it can determined whether the above-mentioned equation (7) can be satisfied by performing linear combination but cannot be satisfied by simply downloading the stored symbols without linear combination. It has been determined that, in terms of the number of symbols downloaded from the storage nodes, downloading the symbols stored in the storage nodes directly and downloading their linear combinations are equivalent in some embodiments. This can be represented as follows:

> If there exists $\alpha\times\mu_i$ matrices $A^{(i)}$ for $i\in S$ such that the above-mentioned equation (7) is satisfied, then there exist $M\times\mu_i$ submatrices $\overline{H}^{(i)}\subseteq H^{(i)}$, $i\in S$, such that the matrix $\overline{H}^{S}\subseteq[\overline{H}^{(i)}]_{i\in S}$ is of rank M.

Accordingly, given $\mu=[\mu_1, \mu_2, \ldots, \mu_S]$, $\mu_i\in\{0, 1, \ldots, \alpha\}$ for $1\leq i\leq S$, the data can be μ-reconstructable if it can be reconstructed via downloading μ_i symbols from storage node i for $i\in S$, which is equivalent to that there exist $M\times\mu_i$ submatrices $\overline{H}^{(i)}\subseteq H^{(i)}$, $i\in S$, such that the matrix $\overline{H}^{S}\subseteq[\overline{H}^{(i)}]_{i\in S}$ is of rank M.

In some embodiments, a portion of the symbols in a storage node for data reconstruction can be downloaded at the minimum-storage regenerating (MSR) point or at the minimum-bandwidth regenerating (MBR) point. This can be performed, for example, by the data collector.

Generally speaking, to analyze the μ-reconstructability at either the MSR or the MBR point, one or more sufficient conditions on $\{\mu_i\}_{i\in S}$ such that data can be reconstructed (e.g., by the data collector **120** illustrated in FIGS. **1** and **2**) via downloading μ_i symbols from storage node i, $i\in S$ can be

determined. In response, a partial downloading scheme given a set $\{\mu_i\}_{i\in S}$ that satisfied this condition can be provided.

In some embodiments, the data can be reconstructed by downloading a portion of the symbol from any suitable storage nodes at the MSR point. In this example, the data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**) can use a partial downloading scheme such that the number of downloaded symbols should be no less than the total number of data symbols to be reconstructed. This can be represented by:

$$\Sigma_{i\in S}\mu_i \geq M \tag{8}$$

It should be noted that the μ-reconstructability can be considered for all coding schemes satisfying the constraint of the MSR point that Kα=M in some embodiments. In some embodiments, since, for any size-K subset $\mathcal{R}\subseteq$S, the data s can be reconstructed from the M downloaded symbols $s^T[H^{(i)}]_{i\in S}$, the square matrix $[H^{(i)}]_{i\in\mathcal{R}}$ is of rank M. Thus, in such embodiments, for any subset $\mathcal{R}\subseteq$S with $|\mathcal{R}|\leq$K, the columns of matrix $[H^{(i)}]_{i\in\mathcal{R}}$ are linearly independent.

Accordingly, $\Sigma_{i\in S}\mu_i\geq$M, which states that the number of downloaded symbols should be no less than the total number of data symbols to be reconstructed, can also be a sufficient condition for μ-reconstructability for the MSR point.

In some embodiments, it can be determined or tested whether a condition is sufficient for μ-reconstructability for the MSR point. FIG. **4** shows an illustrative example of a process for determining whether a condition is sufficient for μ-reconstructability for the MSR point in accordance with some embodiments of the disclosed subject matter. It should be noted that, in process **400** of FIG. **4** and any other process or method described herein, some steps can be added, some steps can be omitted, the order of the steps can be re-arranged, and/or some steps can be performed simultaneously.

Generally speaking, to determine whether $\Sigma_{i\in S}\mu_i\geq$M is a sufficient condition for μ-reconstructability for the MSR point, process **400** selects μ_i symbols $s^T\overline{H}^{(i)}$ from storage node i for any $\{\mu_i\}_{i\in S}$ satisfying $\Sigma_{i\in S}\mu_i$=M, where $\overline{H}^{(i)}$ is an M×μ_i submatrix of $H^{(i)}$, i∈S such that $\overline{H}^S=[H^{(i)}]_{i\in S}$ is of rank M. Process **400** continues to select symbols that are linearly independent of the selected symbols until M symbols have been selected.

Process **400** begins at **410** by selecting a storage node i. At **420**, process **400** determines whether the selected storage node is a feasible storage node. In some embodiments, a storage node i can be considered feasible if the number of downloaded symbols from it is smaller than μ_i.

Upon determining that the storage node is a feasible storage node at **420**, process **400** can determine whether all of the symbols in the feasible storage node are linearly independent of the selected symbols at **430**. Upon determining that the symbols in the feasible storage node are linearly independent of the selected symbols, a symbol from the feasible storage node can be selected and written as a linear combination of the selected symbols at **440**. Based on the linear combination, a selected symbol can be replaced with another symbol from the same storage node such that the symbol from that feasible storage node can be selected to further increase the rank of $\overline{H}^S$ at **450**.

Process **400** can determine whether M symbols have been selected at **460**. If M symbols have not been selected, process **400** can return to **410** and can continue to select symbols that are linearly independent of the selected symbols from one or more storage nodes. Upon selecting M symbols, process **400** can turn to process **500** of FIG. **5**.

As shown in FIG. **5**, process **500** begins at **510** by initializing $\overline{H}^{(i)}$ as null matrices for i∈S. It should be noted that, in the selection process, let λ_i be the number of columns of $\overline{H}^{(i)}$ (the number of symbols already selected from storage node i). It should also be noted that $V=\{i\in S:\lambda_i<\mu_i\}$ is the set of storage nodes that does not satisfy the downloading requirement, $V^0=\{i:\lambda_i=\mu_i\}$, and $\tilde{H}_{(i)}=H^{(i)}\backslash\overline{H}^{(i)}$ for i∈S.

At **520**, process **500** can extract a column of $\tilde{H}^{(i)}\triangleq H^{(i)}\backslash\overline{H}^{(i)}$ and add the extracted column to $\overline{H}^{(i)}$ for some $i\in V\triangleq\{i\in S:\lambda_i<\mu_i\}$. This extraction and addition can continue until $\overline{H}^S=[H^{(i)}]_{i\in S}$ is of rank M.

When it is determined that rank $(\overline{H}^S)<$M at **530**, process **500** can check each column of $\tilde{H}^V=[\tilde{H}^{(i)}]_{i\in V}$ to determine whether it is linearly independent of the columns of $\overline{H}^S$ at **540**. Upon determining that the column is linearly independent, process **500** can add the column to $\tilde{H}^V$ and, thus, increase the rank by one at **550**.

Otherwise, process **500** can turn to process **600** of FIG. **6**.

Process **600** begins by randomly selecting some $i_0\in V$ and a column

$$h_{j_0}^{(i_0)}$$

of $\tilde{H}^{(i_0)}$. This can, for example, be expressed as a linear combination of the columns in $\overline{H}^S$. For example:

$$h_{j_0}^{(i_0)}=\sum_{i\in W}\sum_{j\in\mathcal{J}}\gamma_j^{(i)}h_j^{(i)},\qquad \gamma_j^{(i)}\neq 0 \tag{9}$$

where $\mathcal{W}$ is the set of storage nodes, and $\mathcal{J}_i\subseteq\{1,2,\ldots,\alpha\}$, for each $i\in\mathcal{W}$, is the set of column indices of $\overline{H}^{(i)}$ that participate in the linear combination representation of $h_{j_0}^{(i_0)}$ shown above in (9). It should be noted that $\mathcal{W}_0=\mathcal{W}\backslash\{i_0\}$ is the set of storage nodes other than i_0 that participates in the above-mentioned linear combination of (9).

At **620**, process **600** determines that a column

$$h_{j_1}^{(i_1)}$$

of $[\tilde{H}^{(i)}]_{i\in\mathcal{W}_0}$ exists that is linearly independent from the columns of $\overline{H}^S$. Again, it should be noted that a property of the MSR point is that the columns of $H^{\mathcal{R}}$ are linearly independent for any $|\mathcal{R}|\leq$K and rank $(H^{\mathcal{R}})$=M. Accordingly, span$(H^{\mathcal{R}})$=Q=GF(q)$^{\mathcal{M}}$ for any $|\mathcal{R}|\geq$K. Since the above-mentioned linear combination (9) involves the columns from matrices $\overline{H}^{(i)}$ for $i\in\mathcal{W}\cup\{i_0\}$, $|\mathcal{W}\cup\{i_0\}|\geq$K+1 and $|\mathcal{W}_0|=|\mathcal{W}\backslash\{i_0\}|\geq$K, and thus:

$$Q=\text{span}(H^{W_0})\subseteq\text{span}([\overline{H}^{S\backslash W_0}|H^{W_0}])=\text{span}([\overline{H}^S|\tilde{H}^{W_0}])$$

Accordingly, rank$([\overline{H}^S|\tilde{H}^{W_0}])$=M. It should be noted that, by assuming that rank $(\overline{H}^S)<$M, it follows that there exists a column

$$h_{j_1}^{(i_1)}$$

of $\tilde{H}^{W_0}$ that is linearly independent from the columns of $\overline{H}^S$.

It should be noted that, for the linear combination shown above in (9) in some embodiments, replacing a column

$$h_{j_1}^{(i_1)}$$

for a $j \in \mathcal{J}_1$ with

$$h_{j_0}^{(i_0)}$$

does not change span ($\overline{H}^S$), but provides space for

$$h_{j_1}^{(i_1)},$$

which increases the rank of $\overline{H}^S$ by one. Accordingly, process **600** removes

$$h_j^{(i_1)}$$

from $\overline{H}^S$ and then adds

$$h_{j_0}^{(i_0)} \text{ and } h_{j_1}^{(i_1)}$$

to $\overline{H}^S$, thereby increasing the rank of $\overline{H}^S$ by one.

In some embodiments, a symbol selection process for a partial downloading scheme at the MSR point can be provided. For example, FIG. **7** shows an illustrative example of a recursive symbol selection process for a partial downloading scheme at the MSR point in accordance with some embodiments of the disclosed subject matter. It should be noted that Λ can be used to represent the number of columns of $\overline{H}^S$ and $\overline{G}^S$ can be used to represent the Gaussian elimination representation of $\overline{H}^S$ via column transformation ($\overline{G}^S=\overline{H}^S T$).

At **710**, process **700** can determine whether h is linearly independent of the columns of $\overline{H}^S$. At **720**, upon determining that h is linearly independent of the columns of $\overline{H}^S$, process **700** can add H to $\overline{H}^S$ and, accordingly, update $\overline{G}^S$.

Alternatively, upon determining that h is not linearly independent of the columns of $\overline{H}^S$, process **700** can randomly select $i_0 \in V$ and a column

$$h_{j_0}^{(i_0)}$$

of $\tilde{H}^{(i_0)}$, which can be expressed as a linear combination of the vectors in $\overline{G}^S$. For example:

$$h_{j_0}^{(i_0)} = \overline{G}^S t_0.$$

Allowing $\overline{H}^S=\overline{G}^S T_0$ to represent the Gaussian elimination procedure for $\overline{H}^S$,

$$h_{j_0}^{(i_0)}$$

can be represented as:

$$h_{j_0}^{(i_0)} = \overline{H}^S T_0^{-1} t_0$$

It should be noted that this is an explicit representation of the equation

$$h_{j_0}^{(i_0)} = \sum_{i \in W} \sum_{j \in \mathcal{J}} \gamma_j^{(i)} h_j^{(i)},$$

$$\gamma_j^{(i)} \neq 0.$$

Referring back to FIG. **7**, process **700** can search for a column

$$h_{j_1}^{(i_1)}$$

this is linearly independent of $\overline{H}^S$ based on the Gaussian eliminated matrix $\overline{G}^S$. It should be noted that, since replacing

$$h_j^{(i_1)} \text{ with } h_{j_0}^{(i_0)}$$

does not change the spanned space and thus the Gaussian eliminated matrix $\overline{G}^S$, process **700** only needs to update the Gaussian eliminated matrix $\overline{G}^S$ for adding

$$h_{j_1}^{(i_1)}$$

to $\overline{H}^S$.

In some embodiments, the data can be reconstructed by downloading a portion of the symbols from any suitable storage nodes at the MBR point. For example, the data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**) can use a partial downloading scheme that downloads a portion of the symbols from any suitable storage nodes at the MBR point.

A data block of:

$$M = \frac{K(K+1)}{2} + K(d-K)$$

symbols can be stored among S storage nodes, where each node stores $\alpha=d$ symbols. The M symbols can be represented using the following d×d symmetric matrix:

$$B = \begin{bmatrix} B^{(1)} & B^{(2)} \\ (B^{(2)})^T & 0 \end{bmatrix}$$

where $B^{(1)}$ is a K×K symmetric matrix storing

$$K\frac{(K+1)}{2}$$

symbols and $B^{(2)}$ is a K×(d−K) matrix storing K(d−K) symbols. For encoding, the matrix B is premultiplied by an S×d Vandermonde matrix given by Ψ, and each node i∈S stores the d symbols corresponding to the i^{th} row of $\psi_i B$, where ψ_i is the i^{th} row of Ψ.

In some embodiments, a partial downloading scheme for reconstructing data at the MBR point based on the above coding scheme is provided. As shown in FIG. **8**, since the matrix $B^{(1)}$ is symmetric, the data symbols in $B^{(2)}$ and in the upper triangular part of $B^{(1)}$ can be decoded. As also shown in FIG. **8**, the data symbol can be divided into d columns $\{b_j\}_{1\le j\le d}$, where b_j for $1\le j\le K$ are in the upper triangular part of $B^{(1)}$, and b_j for $K+1\le j\le d$ are the columns of $B^{(2)}$. The data can be reconstructed in the backward order of $b_d, b_{d-1}, \ldots, b_1$. It should be noted that the number of symbols in b_j, $1\le j\le d$, can be given by $\min\{j, K\} \triangleq \theta_j$. Let $(\psi_i B)_j$ be the j^{th} symbol in storage node i which is the product of ψ_i and the j^{th} column of B.

As shown in the example flow diagram of FIG. **9**, a partial downloading scheme that performs a backward reconstruction process **900** is provided in accordance with some embodiments of the disclosed subject matter.

At **910**, process **900** can begin by reconstructing $B^{(2)}$ of matrix B. Referring back to FIG. **8**, to reconstruct b_j, the data collector can download the symbols $(\psi_i B)_j$ for i belonging to a subset $\mathcal{R} \subseteq S$, here the size $|\mathcal{R}_j| \ge K = \theta_j$. Since Ψ is a Vandermonde matrix, b_j can be reconstructed with K downloaded symbols.

At **920**, process **900** can then reconstruct $B^{(1)}$ of matrix B. This can be done, for example, by reconstructing b_j for $1\le j\le K$ in the order of $b_k, b_{k-1}, \ldots, b_1$. Referring back to FIG. **8**, when reconstructing b_j since b_l for $j<l\le K$ have been reconstructed and $B^{(1)}$ is symmetric, the part of $B^{(1)}$ shown in area **802** is known from the previous reconstruction. Then, as $B^{(2)}$ has been reconstructed in **910** and thus is known, reconstructing b_j amounts to downloading the symbols $(\psi_i B)_j$ for $i \in \mathcal{R}_j \subseteq S$, where the size $|\mathcal{R}| \ge j = \theta_j$. As mentioned above, since Ψ is a Vandermonde matrix, b_j can be reconstructed with j downloaded symbols.

Based on the above partial downloading scheme, let $\eta_j^{(i)}=1$ if the data collector downloads $(\psi_i B)_j$ to reconstruct b_j. Otherwise, let $\eta_j^{(i)}=0$. Accordingly, the minimum requirement for data construction can be represented as:

$$\sum_{i=1}^{S} \eta_j^{(i)} = \theta_j,\ 1 \le j \le d$$

The data is μ-reconstructable if there exists $\eta_j^{(i)} \in \{0,1\}$ for $1\le i\le S$ and $1\le j\le d$ such that:

$$\mu_i = \sum_{i=1}^{d} \eta_j^{(i)},\ i \in S$$

and the above-equations are satisfied.

In some embodiments, a sufficient condition in terms of $\{\mu_i\}_{i\in S}$ is provided for the two above-mentioned equations to hold.

One condition for the two above-mentioned equations is that, for any subset $A \subseteq S$:

$$\sum_{i\in A} \mu_i = \sum_{i\in A}\sum_{i=1}^{d} \eta_j^{(i)} = \sum_{i=1}^{d}\sum_{i\in A} \eta_j^{(i)} \le \sum_{j=1}^{d} \min\{\theta_j, |A|\}$$

Since $\Sigma_{i\in A}\eta_j^{(i)} \le |A|$ and $\Sigma_{i\in A}\eta_j^{(i)} \le \Sigma_{i\in S}\eta_j^{(i)} = \theta_j$ for all $1\le j\le d$. Denoting the sorted $\{\mu_i\}_{i\in S}$ in decreasing order as $\mu^{(1)} \ge \mu^{(2)} \ge \ldots \ge \mu^{(S)}$ provides:

$$\sum_{i=1}^{l} \mu^{(i)} \le \sum_{j=1}^{d} \min\{\theta_j, l\}$$

for $1 \le l \le d$, and

$$\sum_{i=1}^{S} \mu^{(i)} = \sum_{j=1}^{d} \theta_j.$$

Since $\theta_j = \min\{j, K\}$, the above-mentioned equations can be represented as:

$$\sum_{i=1}^{l} \mu^{(i)} \le dl - \frac{l(l-1)}{2}$$

for $1 \le l \le d$, and

$$\sum_{i=1}^{S} \mu^{(i)} = M.$$

In some embodiments, the sufficiency of the condition can be tested via an illustrative partial downloading scheme shown in FIG. **10**. More particularly, given $\{\mu_i\}_{i\in S}$ satisfying the two above-mentioned equations.

$$\{\eta_{j_p}^{(i_k)}\}_{i\in S, 1\le j\le d}$$

can be determined.

Turning to FIG. **10**, process **1000** can begin by initializing at **1010**. Such an initialization can include ranking $\{\mu_i\}_{i\in S}$ in decreasing order $\mu_{i_1} \ge \mu_{i_2} \ge \ldots \ge \mu_{i_S}$ and letting $\theta_j = \min\{j, K\}$, $j=1, \ldots, K$.

At **1020**, process **1000** can then determine

$$\{\eta_{j_p}^{(i_k)}\}_{1\le j\le d}$$

for k=1, 2, . . . S. At **1022**, process **1000** can include ranking $\{\theta_j\}_{1\le j\le d}$ in decreasing order $\theta_{j_1} \ge \theta_{j_2} \ge \ldots \ge \theta_{j_d}$. At **1024**, process **1000** can then let

$$\eta_{j_p}^{(i_k)} = 1$$

for $1\le p\le \mu_{i_k}$ and letting

$$\eta_{j_p}^{(i_k)} = 0$$

for $\mu_{i_k}+1 \le p \le d$. At **1026**, process **1000** can subtract

$$\{\eta_{j_p}^{(i_k)}\}_{1 \le j \le d}$$

from θ_{j_p} and update $\theta_{j_p}=\theta_{j_p}-1$ for $1 \le p \le \mu_{i_k}$.

It should be noted that the above symbol selection scheme of FIG. **10** can be used to obtain $\eta_j^{(i)}$, $i \in S$, $1 \le j \le d$ to reconstruct the data.

As described above, in some embodiments, the data collector can determine the number of symbols to be downloaded from each channel and storage node based on suitable channel and power allocation schemes.

In accordance with some embodiments, the number of symbols to be downloaded over channel j can be given by:

$$X_j = c(P_j \Sigma_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2) \tag{10}$$

where: $i \in S$;

$j \in \mathcal{N}$;

$$c(\cdot) \text{is } c\left(P_j \sum_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2\right) = \frac{WT}{B} log_2\left(1 + \mathcal{H} \frac{P_j \sum_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2}{\sigma^2}\right)$$

as described above in connection with equation (6);

$\beta_j^{(i)}=1$ if the data collector (e.g., data collector **120** in FIGS. **1** and **2**) downloads symbols from storage node i using channel j and $\beta_i^{(i)}=0$ otherwise; and

P_j be the transmission power for channel j.

Based on equation (10), the number of symbols μ_i to be downloaded from a storage node i can then be calculated as:

$$\mu_i = \Sigma_{j=1}^{N} \beta_j^{(i)} X_j, i \in S \tag{11}$$

In some embodiments, it may be desirable to minimize the total power used across all N channels during data reconstruction. A minimum total power used during such a reconstruction can be represented in some embodiments as:

$$min_{\{\beta_j^{(i)}, P_j\}_{i \in S, j \in N}} \sum_{j \in N} P_j$$

such that: the data is μ-reconstructable as described above;

$X_j = c(P_j \Sigma_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2)$;

$\mu_i = \Sigma_{j=1}^{N} \beta_j^{(i)} X_j$, $i \in S$;

$X_j \in \{0, 1, 2, \ldots, \alpha\}$ for $j \in \mathcal{N}$; and

$\Sigma_{i \in S} \beta_j^{(i)} \le 1$, $j \in \mathcal{N}$ $\beta_j^{(i)} \in \{0,1\}$.

As reflected by $\Sigma_{i \in S} \beta_j^{(i)} \le 1$, $j \in \mathcal{N}$; $\beta_j^{(i)} \in \{0,1\}$, in some embodiments, each channel, if used, may be restricted to being used only to transmit symbols from one node.

In accordance with some embodiments, based on channel estimation results, the data collector can run a channel and power allocation process to determine the number of symbols to be downloaded from each storage node. Any suitable channel and power allocation process can be used in some embodiments.

According to equation (10), letting $p(\cdot)$ be the inverse function $c(\cdot)$ of the capacity function in equation (6), the transmission power of a channel j can be represented as:

$$P_j = \frac{p(X_j)}{\sum_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2} \tag{12}$$

Based on equation (12), the minimum total power used during a reconstruction, represented above by

$$min_{\{\beta_j^{(i)}, P_j\}_{i \in S, j \in N}} \sum_{j \in N} P_j,$$

can alternatively be represented as

$$min_{\{\beta_j^{(i)}, X_j\}_{i \in S, j \in N}} \sum_{j \in N} \frac{p(X_j)}{\sum_{i \in S} \beta_j^{(i)} |g_j^{(i)}|^2} \tag{13}$$

such that:

$$\sum_{j=1}^{N} X_j = M;\ X_j \in \mathbb{Z}^+ \cup \{0, 1\};$$

$$\sum_{i \in S} \beta_j^{(i)} \le 1,\ j \in \mathcal{N};\ \beta_j^{(i)} \in \{0, 1\}$$

In some embodiments, each channel $j \in N$ can then be assigned to a node i as follows:

$\beta_j^{(i_j)}=1$, for $i_j = \arg\max |g_j^{(i)}|$; and $\beta_j^{(i)}=0$, otherwise.

Letting

$$p_j(X_j) = \frac{1}{\left|g_j^{(i_j)}\right|^2} p(X_j),$$

the power allocation problem then becomes:

$$min_{\{X_i\}_{i \in N}} \sum_{j=1}^{N} p_j(X_j) \tag{14}$$

such that:

$$\sum_{j=1}^{N} X_j = M;\ X_j \in \mathbb{Z}^+ \cup \{0\}.$$

In some embodiments, a greedy algorithm can be used to find this minimum power. For example, in some embodiments, the minimum power can be found as follows:

1) Initialize $X_j=0$ for $j \in \mathcal{N}$;
2) While $\Sigma_{j \in N} X_j < M$, do the following:
   a) For $j \in \mathcal{N}$, let $\Delta P_j = p_j(X_j+1) - p_j(X_j)$ be the power increment for channel j; and
   b) Find the channel $j_0 = \arg\min_{\mathcal{N}} \Delta P_j$ with the minimum power increment, and update $X_{j_0} \leftarrow X_{j_0}+1$; and
3) Output X_j, for $j \in \mathcal{N}$.

Turning to FIG. **11**A, an example of a process **1100** for finding a minimum power is illustrated. In some embodiments, process **1100** can be implemented in a data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**).

As shown, after process **1100** begins, the process can initialize the number of symbols for each channel to zero at **1104**. Next, at **1106**, process **1100** can, for each channel,

calculate the increase in power required for transmitting another symbol on that channel. This calculation can be performed in any suitable manner. For example, as described above, this calculation can be performed using the following equation: $\Delta P_j = p_j(X_j+1) - p_j(X_j)$. Process **1100** can then select the channel with the minimum increase in power at **1108**. Next, at **1110**, process **1100** can allocate a symbol to be transmitted on the selected channel. At **1112**, process **1100** can determine whether all symbols have been allocated. If not, process **1100** can loop back to **1106**. Otherwise, at **1114**, process **1100** can provide the total numbers of symbols to be transmitted on each channel to a suitable process that provides this data to the storage nodes via a feedback channel.

As described above, based on the solution $\{X_j\}_{j\in\mathcal{N}}$ obtained by a greedy algorithm, the number of symbols assigned to each storage node can be given by $\mu_i = \sum_{j\in\mathcal{N}} \beta_j^{(i)} X_j$ for $i\in S$, in some embodiments.

In some embodiments, if $\{\mu_i\}_{i\in S}$ violates a constraint that $\mu_i \le \alpha$, $i\in S$ and the data collector is operating at the MSR point, an adjustment can be performed by identifying each storage node i with $\mu_i > \alpha$, and reassigning symbols to be downloaded on one of the storage nodes channels to another storage node to decrease μ_i, until $\mu_i < \alpha$ for all $i\in S$. This adjustment can be performed in any suitable manner in some embodiments. For example, in some embodiments, the adjustment can be performed as follows:

1) While $\mu_i > \alpha$ for some $i\in S$, do the following:
   a) Find a storage node i with $\mu_i > \alpha$ and the set of assigned channels denoted as $\mathcal{N}_i = \{j: \beta_j^{(i)} = 1\}$;
   b) Find the storage node $i'\in S\backslash\{i\} \triangleq S'$ and the channel $j\in\mathcal{N}_i$ that minimizes the power increment of reassigning the X_j symbols in channel j to node i' such that $\mu_i + X_j \le \alpha$, i.e.,

$$(i_0, j_0) = argmin_{(i',j)\in S'\times N_i : \mu_{i'} + X_j \le \alpha} p_j^{(i')}(X_j) - p_j^{(i)}(X_j) \quad (15)$$

2) Reassign the X_{j_0} symbols in channel j_0 to storage node i_0, by letting $\beta_{j_0}^{(i_0)} = 1$ and $\beta_{j_0}^{(i)} = 0$.

Turning to FIG. **11**B, an example of a process **1101** for performing an adjustment when the data collector is operating at the MSR point is illustrated. In some embodiments, process **1101** can be implemented in a data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**). As shown, process **1101** may be a continuation of process **1100** of FIG. **11**A. In some embodiments, and in some instances, such as when the data collector is not operating at the MSR point, process **1101** can be skipped or omitted.

After process **1101** begins, the process can set an index i equal to one at **1114**. Next, at **1116**, the process can determine if the number of symbols to be downloaded from node i is greater than the number of symbols stored at that node. If so, process **1101** can then calculate the power increase to each other node that would result from reassigning the symbols on a channel assigned to that node to that channel on another node at **1118**. This calculation can be made in any suitable manner. For example, in some embodiments, the calculation can be made using: $p_j^{(i')}(X_j) - p_j^{(i)}(X_j)$. Next, at **1120**, process **1101** can select as node i the node that (1) will have the minimum power increase if the symbols to be transmitted on that channel are reassigned to that node to and (2) that will not have more symbols to be transmitted from that node that are present in that node after reassignment. At **1122**, process **1101** can the reassign the symbols to the channel on the new node i.

After performing **1122** or determining at **1116** that the number of symbols to be downloaded from node i is not greater than the number of symbols stored at that node, process **1101** can branch to **1124** at which i can be incremented. Then, at **1126**, process **1101** can determine whether i is greater than M, the number of symbols to be downloaded. If not, process **1101** can loop back to **1116**. Otherwise, process **1101** can complete.

In some embodiments, if the data collector is operating at the MBR point, the following constraint can be applied against $\{\mu_i\}_{i\in S}$:

$$\sum_{i=1}^{l} \mu^{(i)} \le dl - \frac{l(l-1)}{2}, \text{ for } 1 \le l \le d;$$

and

$$\sum_{i=1}^{S} \mu^{(i)} = M$$

As described above, this constraint can require that $\mu^{(1)} \le d$, $\mu^{(1)} + \mu^{(2)} \le 2d-1$, $\mu^{(1)} + \mu^{(2)} + \mu^{(3)} \le 3d-3$, etc.

In some embodiments, if $\{\mu_i\}_{i\in S}$ violates this constraint and the data collector is operating at the MBR point, an adjustment can be performed as follows:

1) Select as i the storage node with the maximum μ_i;
2) Select as i' the storage node with the minimum μ_i;
3) Reassign one symbol to be downloaded from storage node i so that the symbol will be downloaded from storage node i';
4) Recalculate $\{\mu_i\}_{i\in S}$;
5) Determine if the constraint is still violated; and
6) If so, loop back to 1), otherwise end.

Turning to FIG. **11**C, an example of a process **1102** for performing an adjustment when the data collector is operating at the MBR point is illustrated. In some embodiments, process **1102** can be implemented in a data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**). As shown, process **1102** may be a continuation of process **1101** of FIG. **11**B. In some embodiments, and in some instances, such as when the data collector is not operating at the MBR point, process **1102** can be skipped or omitted.

After process **1102** begins, the process can select as i the storage node with the maximum μ_i at **1128**. Next, at **1130**, process **1102** can select as i' the storage node with the minimum μ_i. Then, at **1132**, the process can reassign one symbol to be downloaded from storage node i so that the symbol will be downloaded from storage node i'. At **1134**, process **1102** can recalculate $\{\mu_i\}_{i\in S}$ as described above. Then, at **1136**, the process can determine if the constraint is still violated. If so, the process can loop back to **1128**. Otherwise, the process can end.

As described above, in some embodiments, the data collector can regenerate data of a failed storage node by downloading d data symbols from other storage nodes. These downloaded data symbols can then be loaded onto a new storage node.

In some embodiments, when the data collector is operating at the MSR point, this can be accomplished by downloading one symbol from each of any $\alpha+K-1$ storage nodes. More particularly, for example, in some embodiments, when the data collector is operating at the MSR point and $\alpha \ge K-1$, this can be accomplished by choosing any $d=\alpha+K-1$ storage nodes and downloading one symbol from each of the d nodes.

In some embodiments, when the data collector is operating at the MBR point, this can be accomplished by downloading one symbol from each of any α storage nodes. More particularly, for example, in some embodiments, when the data collector is operating at the MBR point, this can be accomplished by selecting any $d=\alpha$ nodes and downloading one symbol from each of them.

Then, the wireless resource allocation for both the MSR and MBR points can be similarly formulated as described above, i.e., to minimize the power consumption of downloading d symbols from d storage nodes.

In some embodiments, the data collector can attempt to minimize the total power used to download these symbols subject to the constraints of: $\Sigma_{i\in S}\mu_i=d$; and $\mu_i\in\{0,1\}$ for $i\in S$.

This attempt to minimize the total power can be performed in any suitable manner. For example, in some embodiments, a greedy algorithm can be used to find this minimum power. For example, in some embodiments, the minimum power can be found as follows:

1) Initialize $X_j=0$ for $j\in\mathcal{N}$;

2) While $\sum_{j\in\mathcal{N}}X_j<d$, do the following:

a) For $j\in\mathcal{N}$ where X_j equals 0, calculate the power for transmitting one symbol $p_j(1)$ on channel j; and

b) Find the channel $j_0=\arg\min_{j\in\mathcal{N}}p_j(1)$ with the minimum power, and select that channel j as to be used to download a symbol: $X_{j_0}\leftarrow 1$; and

3) Output X_j for $j\in\mathcal{N}$.

Turning to FIG. **12**, an example of a process **1200** for finding a minimum power is illustrated. In some embodiments, process **1200** can be implemented in a data collector (e.g., data collector **120** as illustrated in FIGS. **1** and **2**).

As shown, after process **1200** begins, the process can initialize the number of symbols for each channel to zero at **1204**. Next, at **1206**, process **1200** can, for each unused channel, calculate the power required for transmitting a symbol on that channel. This calculation can be performed in any suitable manner. Process **1200** can then select the channel with the minimum power at **1208**. Next, at **1210**, process **1100** can allocate a symbol to be transmitted on the selected channel. At **1212**, process **1100** can determine whether all symbols have been allocated. If not, process **1200** can loop back to **1206**. Otherwise, at **1214**, process **1200** can provide the total numbers of symbols to be transmitted on each channel to any suitable process that needs this data. In some embodiments, in the event that the allocated symbols $\{\mu_i\}_{i\in S}$ violates a constraint that $\mu_i\leq 1$, $i\in S$, a reassignment method similar to that described above in connection with FIG. **11**B can be used to reassign one or more of the allocated symbols for regeneration.

Accordingly, methods, systems, and media for partial downloading in wireless distributed networks are provided.

It should be noted that processes of FIGS. **3**, **5-7**, **9**, **10**, **11**A, **11**B, **11**C, and **12** can be performed concurrently in some embodiments. It should also be noted that the above steps of the flow diagrams of FIGS. **3**, **5-7**, **9**, **10**, **11**A, **11**B, **11**C, and **12** may be executed or performed in any order or sequence not limited to the order and sequence shown and described in the figures. Furthermore, it should be noted, some of the above steps of the flow diagrams of FIGS. **3**, **5-7**, **9**, **10**, **11**A, **11**B, **11**C, and **12** may be executed or performed substantially simultaneously where appropriate or in parallel to reduce latency and processing times. And still furthermore, it should be noted, some of the above steps of the flow diagrams of FIGS. **3**, **5-7**, **9**, **10**, **11**A, **11**B, **11**C, and **12** may be omitted.

In some embodiments, any suitable computer readable media can be used for storing instructions for performing the processes described herein. For example, in some embodiments, computer readable media can be transitory or non-transitory. For example, non-transitory computer readable media can include media such as magnetic media (such as hard disks, floppy disks, etc.), optical media (such as compact discs, digital video discs, Blu-ray discs, etc.), semiconductor media (such as flash memory, electrically programmable read only memory (EPROM), electrically erasable programmable read only memory (EEPROM), etc.), any suitable media that is not fleeting or devoid of any semblance of permanence during transmission, and/or any suitable tangible media. As another example, transitory computer readable media can include signals on networks, in wires, conductors, optical fibers, circuits, any suitable media that is fleeting and devoid of any semblance of permanence during transmission, and/or any suitable intangible media.

Although the invention has been described and illustrated in the foregoing illustrative embodiments, it is understood that the present disclosure has been made only by way of example, and that numerous changes in the details of embodiment of the invention can be made without departing from the spirit and scope of the invention, which is only limited by the claims which follow. Features of the disclosed embodiments can be combined and rearranged in various ways.

What is claimed is:

1. A method for selecting numbers of symbols to be transmitted on a plurality of channels, comprising:

receiving, from each of a plurality of storage nodes, information about one or more symbols stored in the storage node, wherein each of the one or more symbols represents a portion of a data block and wherein each of the plurality of storage nodes is connected via a channel of the plurality of channels;

for each of the plurality of channels, calculating using a hardware processor at least an increase in transmission power of that channel if it transmits a first symbol that represents a first portion of the data block from a corresponding storage node of the plurality of storage nodes;

selecting, using the hardware processor, a first channel of the plurality of channels to transmit the first symbol and a second channel of the plurality of channels to transmit a second symbol that represents a second portion of the data block based at least in part on the calculation indicating that the first channel will have the smallest increase in transmission power from transmitting the first symbol and based at least in part on the received information about one or more symbols stored in a first storage node, wherein the first storage node of the plurality of storage nodes is connected via the first channel and a second storage node of the plurality of storage nodes is connected via the second channel; and

indicating that the first symbol is to be transmitted by the first channel and that the second symbol is to be transmitted by the second channel using the hardware processor.

2. The method of claim **1**, further comprising determining a reduction in the number of symbols to be transmitted on a channel of the plurality of channels.

3. The method of claim **2**, further comprising:

determining that the number of symbols to be transmitted from a third storage node of the plurality of storage nodes exceeds the number of symbols stored at the third storage node, wherein the third storage node is connected via a third channel of the plurality of channels;

calculating a power increase that would occur by reassigning a symbol from the third channel to each of a subset of the plurality of channels wherein each storage node of a subset of the plurality of storage nodes is connected via a corresponding channel of the subset of the plurality of channels; and

selecting a fourth storage node of the subset of the plurality of storage nodes to which the symbol is to be reassigned based on the calculating.

4. The method of claim **3**, further comprising transmitting symbols at a minimum-storage regenerating (MSR) point.

5. The method of claim **2**, further comprising:

selecting the first storage node as having a maximum number of symbols to be transmitted;

selecting the second storage node as having a minimum number of symbols to be transmitted; and

reassigning a symbol as to be transmitted from the second storage node instead of the first storage node.

6. The method of claim **5**, further comprising transmitting symbols at a minimum-bandwidth regenerating (MBR) point.

7. A system for selecting numbers of symbols to be transmitted on a plurality of channels, comprising:

at least one hardware processor; and

memory containing computer-executable instructions that, when executed by the hardware processor, cause the hardware processor to:

receive, from each of a plurality of storage nodes, information about one or more symbols stored in the storage node, wherein each of the one or more symbols represents a portion of a data block and wherein each of the plurality of storage nodes is connected via a channel of the plurality of channels;

for each of the plurality of channels, calculate at least an increase in transmission power of that channel if it transmits a first symbol that represents a first portion of the data block from a corresponding storage node of the plurality of storage nodes;

select a first channel of the plurality of channels to transmit the first symbol and a second channel of the plurality of channels to transmit a second symbol that represents a second portion of the data block based at least in part on the calculation indicating that the first channel will have the smallest increase in transmission power from transmitting the first symbol and based at least in part on the received information about one or more symbols stored in a first storage node, wherein the first storage node of the plurality of storage nodes is connected via the first channel and a second storage node of the plurality of storage nodes is connected via the second channel; and

indicate that the first symbol is to be transmitted by the first channel and that the second symbol is to be transmitted by the second channel.

8. The system of claim **7**, wherein the instructions further cause the at least one hardware processor to determine a reduction in the number of symbols to be transmitted on a channel of the plurality of channels.

9. The system of claim **8**, wherein the instructions further cause the at least one hardware processor to:

determine that the number of symbols to be transmitted from a third storage node of the plurality of storage nodes exceeds the number of symbols stored at the third storage node, wherein the third storage node is connected via a third channel of the plurality of channels;

calculate a power increase that would occur by reassigning a symbol from the third channel to each of a subset of the plurality of channels wherein each storage node of a subset of the plurality of storage nodes is connected via a corresponding channel of the subset of the plurality of channels; and

select a fourth storage node of the subset of the plurality of storage nodes to which the symbol is to be reassigned based on the calculating.

10. The system of claim **9**, wherein the instructions further cause the at least one hardware processor to cause symbols to be transmitted a minimum-storage regenerating (MSR) point.

11. The system of claim **8**, wherein the instructions further cause the at least one hardware processor to:

select the first storage node as having a maximum number of symbols to be transmitted;

select the second storage node as having a minimum number of symbols to be transmitted; and

reassign a symbol as to be transmitted from the second storage node instead of the first storage node.

12. The system of claim **11**, wherein the instructions further cause the at least one hardware processor to cause symbols to be transmitted at a minimum-bandwidth regenerating (MBR) point.

13. A non-transitory computer-readable medium containing computer-executable instructions that, when executed by a processor, cause the processor to perform a method for selecting numbers of symbols to be transmitted on a plurality of channels, the method comprising:

receiving, from each of a plurality of storage nodes, information about one or more symbols stored in the storage node, wherein each of the one or more symbols represents a portion of a data block and wherein each of the plurality of storage nodes is connected via a channel of the plurality of channels;

for each of the plurality of channels, calculating at least an increase in transmission power of that channel if it transmits a first symbol that represents a first portion of the data block from a corresponding storage node of the plurality of storage nodes;

selecting a first channel of the plurality of channels to transmit the first symbol and a second channel of the plurality of channels to transmit a second symbol that represents a second portion of the data block based at least in part on the calculation indicating that the first channel will have the smallest increase in transmission power from transmitting the first symbol and based at least in part on the received information about one or more symbols stored in a first storage node, wherein the first storage node of the plurality of storage nodes is connected via the first channel and a second storage node of the plurality of storage nodes is connected via the second channel; and

indicating that the first symbol is to be transmitted by the first channel and that the second symbol is to be transmitted by the second channel.

14. The non-transitory computer-readable medium of claim **13**, wherein the method further comprises determining a reduction in the number of symbols to be transmitted on a channel of the plurality of channels.

15. The non-transitory computer-readable medium of claim **14**, wherein the method further comprises:

determining that the number of symbols to be transmitted from a third storage node of the plurality of storage nodes exceeds the number of symbols stored at the third storage node, wherein the third storage node is connected via a third channel of the plurality of channels;

calculating a power increase that would occur by reassigning a symbol from the third channel to each of a subset of

the plurality of channels, wherein each storage node of a subset of the plurality of storage nodes is connected via a corresponding channel of the subset of the plurality of channels; and

selecting a fourth storage node of the subset of the plurality of storage nodes to which the symbol is to be reassigned based on the calculating.

16. The non-transitory computer-readable medium of claim **15**, wherein the method further comprises transmitting symbols at a minimum-storage regenerating (MSR) point.

17. The non-transitory computer-readable medium of claim **14**, wherein the method further comprises:

selecting the first storage node as having a maximum number of symbols to be transmitted;

selecting the second storage node as having a minimum number of symbols to be transmitted; and

reassigning a symbol as to be transmitted from the second storage node instead of the first storage node.

18. The non-transitory computer-readable medium of claim **17**, wherein the method further comprises transmitting symbols at a minimum-bandwidth regenerating (MBR) point.

19. A method for selecting numbers of symbols to be transmitted on a plurality of channels, comprising:

receiving, using a hardware processor of a data collection device, from each of a plurality of storage nodes connected to a data collection device via a channel of the plurality of channels, information about one or more symbols stored in that storage node, wherein each of the one or more symbols represents part of a data block;

calculating, using the hardware processor, for each of the plurality of channels, an increase in transmission power of that channel if it transmits a first symbol from a corresponding storage node of the plurality of storage nodes;

selecting, using the hardware processor, a first channel of the plurality of channels based at least in part on the calculation indicating that the first channel will have the smallest increase in power from transmitting the first symbol and based at least in part on the information about one or more symbols stored in a first storage node of the plurality of storage nodes;

in response to selecting the first channel for transmitting the first symbol, calculating, using the hardware processor, for each of the plurality of channels, an increase in transmission power of that channel if it transmits a second symbol from a corresponding storage node of the plurality of storage nodes, wherein the calculation of the increase in transmission power of the first channel if it transmits the second symbol is based on the calculation of the increase in transmission power of the first channel if it transmits the first symbol;

selecting, using the hardware processor, a second channel of the plurality of channels based on the calculation indicating that the second channel will have the smallest increase in power from transmitting the second symbol and based at least in part on the information about one or more symbols stored in a second storage node of the plurality of storage nodes; and

indicating, using the hardware processor, that the first symbol is to be transmitted by the first storage node using the first channel and that the second symbol is to be transmitted by the second storage node using the second channel.

* * * * *