doi:10.11959/j.issn.1000-436x.2015275

“双码”架构下的云存储多节点修复协作编码

谢显中[1]，黄倩[1,2]，王柳苏[1]

(1. 重庆邮电大学 宽带接入网络研究所，重庆 400065；2. 重庆邮电大学 移通学院 计算机科学系，重庆 401520)

摘 要：针对云存储中现有多节点失效修复模型的不足，给出了一种可以对多个系统节点或冗余节点同时修复的多节点协作的精确修复码，证明了其存在性，并且将此修复码与具有健康节点协作的 MDS 双码架构模型相结合，以达到对多节点修复的同时，降低修复带宽、修复链路数和单个中间节点需要处理的数据量。通过数值仿真结果表明，本模型与修复方案在以上 3 个方面具有较大改进，尤其削弱了修复时中间节点的负荷，且随着云存储中节点数量的增多，本方案的优势更加明显。

关键词：云存储；多节点协作精确修复码；协作修复；双极大距离可分码模型

中图分类号：TP302 **文献标识码**：A

Collaboration coding to multi-node repair program under the twin-MDS codes framework in cloud storage systems

XIE Xian-zhong[1], HUANG Qian[1, 2], WANG Liu-su[1]

(1. Institute of Broadband Access Technologies, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 2. Department of Computer Science, College Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing 401520, China)

Abstract: A multi-node exact repair code scheme, which can repair multiple system nodes or redundant nodes simultaneously, was shown and proved to against the disadvantages of the existing multi-node repair model in cloud storage. The multi-node exact repair code was combined with a twin-MDS codes framework with health cooperative nodes. In this way, repair bandwidth, the number of repair links and the amount of data to be treated in an intermediate node were reduced, while multi-node were repaired. Finally, numerical simulation results show that this scheme has greater improvements. In particular, it reduces the load in an intermediate node. And the advantages was more obvious with the more storage nodes in cloud storage.

Key words: cloud storage; multi-node exact repair code; collaboration repair; twin-MDS codes model

1 引言

随着大数据的迅猛发展，如何在云存储/分布式存储的可靠性与带宽消耗之间找到一个平衡点至关重要。2007 年，Dimakis 等[1]第一次将网络编码思想应用于云存储/分布式存储，并提出了基于网络编码思想的再生码方案。再生码（regenerating codes）是指在云存储（cloud storage）系统中，将网络编码（network codes）技术和失效节点修复技术相结合的编码方案[2,3]。利用再生码技术可以在提高云存储可靠性的同时，减少修复需要的网络流量和带宽，进一步降低存储和修复代价。因此，各类再生码和失效节点修复方案受到广泛重视。Suh 等[4]提出了一种满足最优存储容量-修复带宽折衷曲线的极大距离可分（MDS, maximum distance separable）码，并通过对偶变换的形式将该修复方案同时应用于系统节点和冗余节点的修复。但文献[4]的精确修复 MDS（E-MDS, exact MDS）码只能解决单节点

收稿日期：2015-10-29

基金项目：国家自然科学基金资助项目（61271259，0872037）；重庆市自然科学基金资助项目（CTSC2011jjA40006，CSTC2010BB2415）；重庆市教委科学技术研究基金资助项目（KJ120501，KJ110530）

Foundation Items: The National Natural Science Foundation of China (61271259, 0872037); The Natural Science Foundation of Chongqing (CTSC2011jjA40006, CSTC2010BB2415); The Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJ120501, KJ110530)

失效的问题，而未涉及多节点失效的情况。

对于多节点失效修复问题，多节点协作修复是有效的解决方案[5~12]。文献[5]给出了联合再生节点方案，但需要更多的传输信道，这使修复过程更加繁琐，导致修复的不稳定。文献[6,7]所阐述的多节点灵活再生方案主要是指在修复过程中，中间节点下载所需数据的灵活性，即可同时选择从原健康节点和其他再生节点下载数据并通过干扰对齐来修复失效节点，虽然有效地降低了修复带宽，但仍有修复时间不同步、修复过程复杂等问题。文献[8]在多节点联合修复时把干扰对齐独立地运用其中，提高了修复效率，但在精确修复时这个方案只在 k=2 时有效，为了使其有更大的适应性，还有很多问题需要解决。文献[9,10]分别将其扩展到其他 k 值组合时的最小带宽再生码（MBRC, minimum-bandwidth regenerating codes）和最小存储再生码（MSRC, minimum-storage regenerating codes），但在 MBRC 和 MSRC 无法取得折中平衡。Shum 等[11]通过改变部分中间节点选取的帮助节点，提出了一种多节点联合修复（MCR, multi-node cooperative regenerating）码。MCR 码虽然可以减少修复带宽，但也使修复过程更加繁琐，需要更多的传输信道，导致修复的不稳定。具有健康节点协作的多节点修复模型[12]在进行多节点修复时，帮助节点把用来修复失效节点的数据直接传输到从健康节点中选取的中间节点处。然后在此中间节点上进行相关计算和处理，在保证最低修复带宽的前提下，同步节点修复过程，减少修复链路数目，简化修复过程，减少对网络资源的浪费和依赖，以此增加系统可靠性，从而达到安全高效修复节点的目的。但由于此模型将主要的计算和处理工作放在中间节点上，所以中间节点的存储容量和运算负荷较大，系统稳定性差。

为解决上述多节点协作修复问题，本文首先根据文献[13]的 MDS 双码(twin-MDS codes)架构模型结合具有健康节点协作的多节点修复模型，给出了一种具有健康节点协作的 MDS 双码架构模型；这不但能解决具有健康节点协作的多节点修复方案[12]中的存储容量和运算负荷较大的问题，而且具有数据传输链路少、修复带宽小、多节点同步修复的优势。进一步，本文对文献[4]中的 E-MDS 码进行扩展，给出了一种适用于多个系统节点和冗余节点同时修复的多节点协作的精确修复（MER, multi-node exact repair）编码方案，并证明了其存在性。最后，通过数值仿真对比表明，本文的模型与方案在修复带宽、数据传输链路具有较大改进，且随着云存储中节点数量的增多优势更加明显。

2 MDS 双码架构下健康节点协作的多节点修复模型

2.1 健康节点协作的多节点修复模型简介

具有健康节点协作的多节点修复模型的基本思想是在修复 r 个节点失效时，d 个健康节点互相协作，并从 d 个健康节点中选择一个健康节点作为节点 m 代替再生节点，失效节点的修复过程就直接在节点 m 上进行，以此省掉修复过程中的中间节点环节。具有健康节点协作的多节点修复过程如图 1 所示。

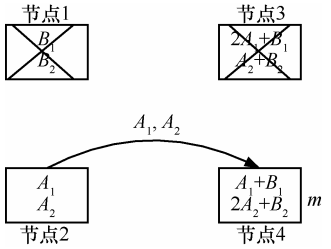

图 1 具有健康节点协作的多节点修复过程

设原始用户文件可分为 4 个大小相同的数据块（A_1，A_2，B_1，B_2），并存储在 2 个系统节点（节点 1、节点 2）中，然后运用具有 MDS 性质的再生码对原始文件进行网络编码，再生成与系统节点同等大小的 2 个冗余节点（节点 3、节点 4）。假设节点 1 和节点 3 失效，选择节点 4 为节点 m，节点 2 直接将其存储的数据块 A_1 和 A_2 传输给节点 m，然后节点 m 将接收到的数据再与其自身存储的数据进行运算，则可以同时得到 2 个失效节点的全部数据，并分别输出从而修复出失效节点 1 和失效节点 3。

在具有健康节点协作的多节点修复模型中，其他健康节点传送自己存储的数据到节点 m，则需要传送 d−1 次大小为 β 的数据，然后再通过线性运算并输出 r 个大小为 α 的数据，因此，成功修复所有失效节点所需的总修复带宽为 $\gamma=(d-1)\beta+r\alpha$。因此，可以看出图 1 中的数据传输量为 6 个数据块，传输次数（传输线路）为 3，并且在一个修复过程就达到了同时修复 2 个失效节点的目的。

因此在云存储环境中，当存储节点相隔距离非常大的情况下，具有健康节点协作的多节点修复模

型能使整个修复过程更加安全、简便。该模型在修复失效节点时下载数据是同步的，且修复步骤简便、安全易实施，有效地减少了所需的数据传输链路数，保证了高效率的修复节点，减少了对资源的浪费。虽然具有健康节点协作的多节点修复模型的优势明显，但仍然有中间节点 m 存储容量、运算负荷较大的问题。因此，下一节将引入 MDS 双码架构模型及其编码过程。

2.2　MDS 双码架构简介

在 MDS 双码架构下，云存储系统中的存储节点（n 个）被分成类型 1（n_1 个）和类型 2（n_2 个），如图 2 所示。

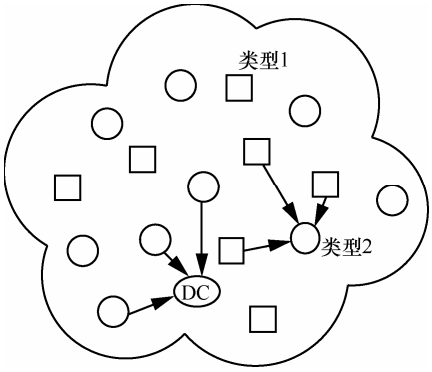

图 2　MDS 双码架构下节点修复和源文件重建的数据下载方式

如果想从类型 1 的节点下载数据，则该数据可以运用线性码 C_1 进行编码解码操作；而想从类型 2 的节点下载数据，则数据先通过简单的移位操作改变顺序，然后运用线性码 C_2 进行编码解码操作，并且这 2 种线性码（C_1、C_2）可以相同。另外，当一种类型的存储节点失效时，需要从另一种类型的健康存储节点子集下载数据以修复这个失效节点；而如果想要重建出整个原始用户文件，则需要从同一种类型节点子集下载数据。因为 MDS 双码架构下的两种线性码都满足 MDS 特性，因此任意选择 k 个同类型节点就可以恢复出整个原始用户文件。

假设原始用户文件为在 F_q 的有限域里的 M 个信息符号，每个存储节点存储 k 个信息字符，对于类型 i=0，1，令 C_i 为区间$[n_i, k]$中在有限域 F_q 中任意的线性码（具有 MDS 特性），生成矩阵为 $\boldsymbol{G}_i$，把矩阵 $\boldsymbol{G}_i$ 的第 l 列表示为 $g_{(i,l)}(1\leqslant l\leqslant n_i)$。

首先，用户数据被分为 k^2 个数据片段，然后对于每一个数据片段进行编码。把这 k^2 个数据符号排列在 $k\times k$ 阶信息矩阵 $\boldsymbol{A}_1$ 中，且 $\boldsymbol{A}_2=\boldsymbol{A}_1{}^t$，其中，上标 t 表示矩阵的置换。通过编码 C_i 和信息矩阵 $\boldsymbol{A}_i$ 来获取这些信息数据，存储在类型 i 的每一个节点中的 k 个信息符号对应到 $k\times n_i$ 阶矩阵 $\boldsymbol{A}_i\boldsymbol{G}_i$ 中的每一列，类型 i 的节点 $l(1\leqslant l\leqslant n_i)$存储的数据字符在对应在矩阵的第 l 列，表示为 $\boldsymbol{A}_i g_{(i,l)}(1\leqslant l\leqslant n_i)$。易知，每个类型 i 的节点 l 与矩阵 $\boldsymbol{G}_i$ 中列 $g_{(i,l)}$一一对应，因此称 $g_{(i,l)}$为所对应存储节点的编码向量。如图 3 所示。

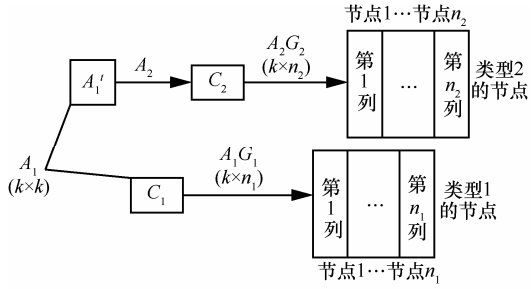

图 3　MDS 双码架构下 2 种类型码的编码过程

为更加具体地描述 MDS 双码架构下的节点修复过程，下面给出一个简单的修复过程示例。假设类型 1 的节点 1 和节点 4 同时失效，除了类型 2 的节点 2、3、4 把所需数据传输给修复 $g_{(1,1)}$的中间节点外，类型 2 的节点 1、2、3 也要将数据传输到修复 $g_{(2,4)}$的中间节点处，最终修复出这 2 个失效节点的数据，如图 4 所示。

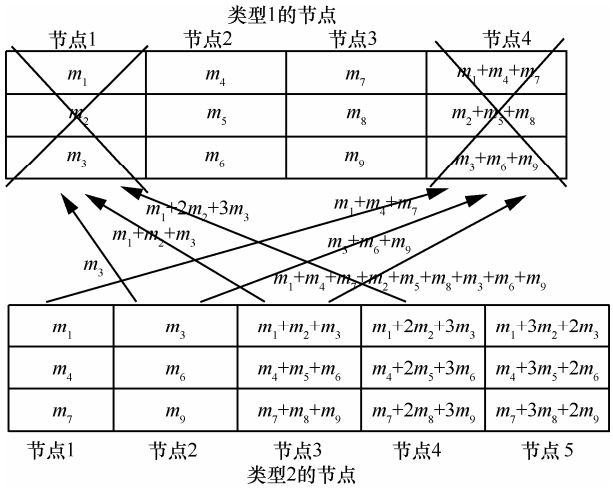

图 4　MDS 双码架构的多节点修复

2.3　具有健康节点协作的 MDS 双码架构模型

为了进一步提高对失效节点的修复效率、系统可靠性，同时降低成本，可将 MER 码运用于 MDS 双码架构模型中，对多节点失效进行修复。利用 MDS 双码架构模型与具有健康节点协作的多节点修复模型相结合，不仅达到更优的修复效果，而且

还可以克服文献[12,13]中存在的诸多问题。

具有健康节点协作的 MDS 双码架构模型如图 5 所示。将存储节点根据双码结构进行布局，用户原始文件被存放在 2 种类型的节点中，并分别用 C_1 码和 C_2 码为类型 1 和类型 2 的存储节点编码。然后在两部分各选择一个健康节点作为修复对方失效节点的中间节点 m_1 和 m_2。假设类型 2 的 r_2 个节点损坏，则属于类型 1 的用来修复这些节点的健康节点的个数为 d_1，传输到 m_1 的链路数为 d_1-1；同样的，若是类型 1 的 r_1 个节点损坏，则传输到 m_2 的链路数为 d_2-1，然后在中间节点进行线性运算，得到失效节点中的数据，输出再生节点。

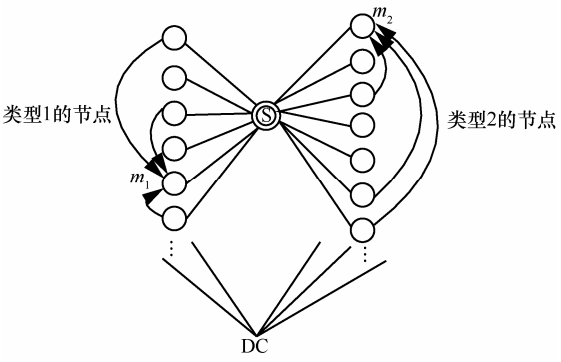

图 5 具有健康节点协作的 MDS 双码架构模型

假设云存储系统中有总共有 r 个节点失效，其中，属于 C_1 码和 C_2 码这 2 种类型的失效节点个数分别为 r_1 和 r_2。本文分别从 d_1 和 d_2 个节点下载数据来修复所有失效节点，系统管理员从 2 个类型的健康节点中分别选择 2 个节点作为中间节点 m_1 和 m_2，其他健康节点分别传送数据到与自身同类型的接收端 m_1 或者 m_2，则需要传送 d_i-1 次大小为 β_i 的数据，再通过线性运算并输出 r_i 个 α 大小的数据。因此，修复全部失效节点所需的修复带宽为 $\gamma=\gamma_1+\gamma_2$，其中，$\gamma_i=(d_i-1)\beta_i+r_i\alpha$。

从上面的分析可以看出，与简单的具有健康节点协作的多节点修复模型相比，只增加了少量的存储节点，但中间节点的数据存储量少了一半，明显地减小了中间节点 m 的存储及运算负担。同时，在同样的修复条件下，当 2 种类型的节点中都存在多节点失效的情况时，在传输链路方面，传输链路数减少了 1 条。因为多了 1 个中间节点就少了 1 次健康节点传输数据的过程，在这个方案中 $d=d_1-1+d_2-1=d_1+d_2-2$，而具有健康节点协作的多节点修复模型中 $d=d_1+d_2-1$。在修复带宽方面，每条链路传输的数据量为 α 没变，但传输链路减少了一条，也就是减少了 β 大小的修复带宽，这里 $\beta=\alpha$。

综上所述，整个系统的可靠性、灵活性都大大增加，既保留、甚至优化了具有健康节点协作的多节点修复方案的数据传输链路少、修复带宽小、多节点同步修复的优势；也解决了这个方案中间节点存储、运算负担大的问题，在大量存储节点的网络中，此优势会更加明显。在下一节中，本文将给出一种运用于具有健康节点协作的 MDS 双码架构模型中的，能适用于多个系统节点或冗余节点的 MER 码，并证明其存在性。

3 MDS 双码架构下具有健康节点协作的 MER 修复方案

Suh 等[4]给出了一种 $(n, k, d)=(2k, k, 2k-1)$ 的 E-MDS 码。对于该 E-MDS 码，可以计算 $M=k\beta(d-k+1)$，$\beta=1$，则有 $\alpha=\frac{M}{k}=k$。这里进一步将 E-MDS 码再进行简要地概括，并为了强调 E-MDS 码的对偶性，将使用与文献[4]不同的符号表述。

设 k 阶非奇异矩阵 $\boldsymbol{X}=[x_{ij}]$，$\boldsymbol{Y}=[y_{ij}]$，$\boldsymbol{\Psi}=[\psi_{ij}]$，$\boldsymbol{\Phi}=\boldsymbol{\Psi}^{-1}=[\phi_{ij}]$ 满足

$$\boldsymbol{X}=\boldsymbol{Y}\boldsymbol{\Psi}\ ,\ \boldsymbol{Y}=\boldsymbol{X}\boldsymbol{\Phi} \tag{1}$$

其中，矩阵 $\boldsymbol{\Psi}$、$\boldsymbol{\Phi}$ 为初等变换矩阵，矩阵 $\boldsymbol{X}$ 的列向量为 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_k$，矩阵 $\boldsymbol{Y}$ 的列向量为 $\boldsymbol{y}_1,\cdots,\boldsymbol{y}_k$。设 $K=\{1,\cdots,k\}$，则对于 $\forall i\in K$，式(1)可以表示为

$$\begin{cases}\boldsymbol{x}_i=\psi_{1i}\boldsymbol{y}_1+\psi_{2i}\boldsymbol{y}_2+\cdots+\psi_{ki}\boldsymbol{y}_k\\ \boldsymbol{y}_i=\varphi_{1i}\boldsymbol{x}_1+\varphi_{2i}\boldsymbol{x}_2+\cdots+\varphi_{ki}\boldsymbol{x}_k\end{cases} \tag{2}$$

设矩阵 $\tilde{\boldsymbol{X}}=(\boldsymbol{X}^{\mathrm{T}})^{-1}$，$\tilde{\boldsymbol{Y}}=(\boldsymbol{Y}^{\mathrm{T}})^{-1}$。其中，对矩阵 $\boldsymbol{X}$（或 $\boldsymbol{Y}$）的列向量进行转置和求逆操作后得到矩阵 $\tilde{\boldsymbol{X}}$（或 $\tilde{\boldsymbol{Y}}$）的列向量 $\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_k$（或 $\tilde{\boldsymbol{y}}_1,\cdots,\tilde{\boldsymbol{y}}_k$）。设 $k\times k$ 阶矩阵 $\boldsymbol{\Theta}=[\boldsymbol{\theta}_1\cdots\boldsymbol{\theta}_k]$，$\boldsymbol{D}=[\boldsymbol{\delta}_1\cdots\boldsymbol{\delta}_k]$。其中，$\boldsymbol{\theta}_i$ 表示节点 i 中存储的长度为 k 的列向量，$\boldsymbol{\delta}_i$ 表示节点 $k+i$ 中存储长度为 k 的列向量。

文献[4]中给出了 E-MDS 编码方案的两种构造方法：一种方法是将节点 1 到节点 k 设为系统节点，节点 $k+1$ 到节点 $2k$ 设为冗余节点；另一种方法是将节点 $k+1$ 到节点 $2k$ 设为系统节点，节点 1 到节点 k 设为冗余节点。首先对于第一种构造方法，系统节点 i 中存储的是未编码的原始数据分块，而冗余节点 $k+i$ 中存储的是对系统节点中的数据分块进行线性变换后的编码数据块，其中，$\forall i\in K$。因此，

k 个冗余节点中的数据可以表示为

$$\boldsymbol{D}=\omega\tilde{\boldsymbol{Y}}\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{X}+\sigma\boldsymbol{\Theta}\boldsymbol{\Psi} \tag{3}$$

其中，编码系数ω和σ是有限域 **GF**(q)上定义的参数。设矩阵$\boldsymbol{\Theta}\boldsymbol{\Psi}$的第 j 列向量为$\boldsymbol{\lambda}_j=\sum_{\varepsilon=1}^{k}\psi_{\varepsilon j}\boldsymbol{\theta}_\varepsilon$，则式（3）用列向量形式可表示为

$$\boldsymbol{\delta}_j=\left(\omega\sum_{i=1}^{k}\tilde{\boldsymbol{y}}_i\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\theta}_i\right)+\sigma\boldsymbol{\lambda}_j,\ \forall j\in K \tag{4}$$

对于第二种构造方法，k 个冗余节点中的数据可以表示为

$$\boldsymbol{\Theta}=\omega'\tilde{\boldsymbol{X}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{Y}+\sigma'\boldsymbol{D}\boldsymbol{\Phi} \tag{5}$$

其中，编码系数ω'和σ'是有限域 **GF**(q)上定义的参数。设矩阵 $\boldsymbol{D}\boldsymbol{\Phi}$的第 j 列向量为$\boldsymbol{\lambda}_j'=\sum_{\varepsilon=1}^{k}\varphi_{\varepsilon j}\boldsymbol{\delta}_\varepsilon$，则式（5）用列向量形式可表示为

$$\boldsymbol{\theta}_j=\left(\omega'\sum_{i=1}^{k}\tilde{\boldsymbol{x}}_i\boldsymbol{y}_j^{\mathrm{T}}\boldsymbol{\delta}_i\right)+\sigma'\boldsymbol{\lambda}_j',\ \forall j\in K \tag{6}$$

由式（3）和式（5）可以推出，如果编码系数ω、σ、ω'和σ'满足

$$\omega\omega'+\sigma\sigma'=1,\ \sigma\omega'+\omega\sigma'=0 \tag{7}$$

则 E-MDS 码的构造具有严格的对偶性。即若有函数 $F(\boldsymbol{\Theta})=\omega\tilde{\boldsymbol{Y}}\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{X}+\sigma\boldsymbol{\Theta}\boldsymbol{\Psi}$和函数 $G(\boldsymbol{D})=\omega'\tilde{\boldsymbol{X}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{Y}+\sigma'\boldsymbol{D}\boldsymbol{\Phi}$，则 $G(F(\boldsymbol{\Theta}))=\boldsymbol{\Theta}$和 $F(G(\boldsymbol{D}))=\boldsymbol{D}$ 成立。

但是，该 E-MDS 码并不适合多节点修复。为了给具有健康节点协作的 MDS 双码架构模型构造一种适合的再生码，这里对 E-MDS 码进行扩展，从而得到一种适用于多个系统节点或冗余节点的多节点修复网络编码方案——多节点精确修复（MER, multi-node exact repair）码。该修复码不仅保持了 E-MDS 码的最小修复带宽，而且可以适用于多节点同时失效的精确修复情况。

定义 1　(柯西矩阵)[14]　若 $m\times n$ 阶矩阵 $\boldsymbol{\Psi}=\left[\dfrac{1}{a_i-b_j}\right]$，其变量 a_i 和 b_j 是有限域 **GF**(q)上满足 $a_i-b_j\neq0$，$a_i\neq a_j$，$b_i\neq b_j$ 的参数，其中，$1\leqslant i\leqslant m$，$1\leqslant j\leqslant n$。则称矩阵 $\boldsymbol{C}$ 为柯西（Cauchy）矩阵。

定理 1　若 E-MDS 码的参数矩阵 $\boldsymbol{Y}$、$\boldsymbol{\Psi}$和编码系数ω、σ、ω'和σ'满足。

1) 矩阵 $\boldsymbol{Y}$ 是有限域 **GF**(q)上 $k\times k$ 阶的非奇异矩阵。

2) 矩阵$\boldsymbol{\Psi}$是有限域 **GF**(q)上 $k\times k$ 阶的柯西矩阵。

3) 系数ω、σ、ω'和σ'是满足式（7）的非零变量。

4) 对于$\forall i$，$j\in K$ 有$\psi_{ij}\phi_{ij}\neq1$。

则存在能同时修复 r 个失效节点的，并满足再生码割集边界值的 MER 码。

证明　选取 E-MDS 码的第一种构造方法。即对于$\forall i\in K$，矩阵$\boldsymbol{\Theta}$的列向量对应于系统节点 i 的数据，矩阵 $\boldsymbol{D}$ 的列向量对应于冗余节点 $k+i$ 的数据。因为 MER 码至多允许 r 个节点失效，所以 MER 码的帮助节点数量 $d=n-r$。

在节点修复过程中，若系统节点 i 失效，则帮助节点将各自存储的向量与列向量 $\boldsymbol{y}_i$ 做内积操作，然后传给再生节点 i'；若冗余节点 $k+i$ 失效，则帮助节点将各自存储的向量与列向量 $\boldsymbol{x}_i$ 做内积操作，然后传给再生节点$(k+i)'$。

首先证明当 $r\geqslant2$ 时，且 r 个失效节点全为系统节点或 r 个失效节点全为冗余节点的情况。由于矩阵$\boldsymbol{\Theta}$和矩阵 $\boldsymbol{D}$ 的对称性，所以这 2 种情况可以等效处理。

设 r 个节点失效（节点 $k+1$ 到节点 $k+r$），在修复过程中，再生节点$(k+i)'$将收到帮助节点发送的数据 $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_1,\cdots,\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_k$ 和 $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_j=\omega\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\lambda}_i+\sigma\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\lambda}_j$，其中，$\forall i\in\{1,\cdots,r\}$，$\forall j\in\{r+1,r+2,\cdots,k\}$。从式（4）可知，再生节点$(k+i)'$需要修复的数据为

$$\boldsymbol{\delta}_i=\left(\omega\sum_{\varepsilon=1}^{k}\tilde{\boldsymbol{y}}_\varepsilon\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_\varepsilon\right)+\sigma\boldsymbol{\lambda}_i \tag{8}$$

其中，式（8）的前半部分由接收到的数据$(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_1,\cdots,\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_k)$组成；后半部分，当$r+1\leqslant j\leqslant k$时，再生节点$(k+i)'$根据收到的数据 $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\delta}_j=\omega\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\lambda}_i+\sigma\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\lambda}_j$ 可计算得到

$$\begin{cases}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\lambda}_i=\displaystyle\sum_{\varepsilon=1}^{k}\psi_{\varepsilon i}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_\varepsilon\\ \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\lambda}_i=\dfrac{1}{\omega}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\delta}_j-\dfrac{\sigma}{\omega}\left(\displaystyle\sum_{\varepsilon=1}^{k}\psi_{\varepsilon j}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\theta}_\varepsilon\right)\end{cases} \tag{9}$$

当 $1\leqslant j\leqslant r$ 时，再生节点$(k+i)'$将从其余 $r-1$ 个再生节点中请求数据 $\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\lambda}_i$，其中，$i\neq j$。最后根据 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_k$ 的相互独立性解出向量$\boldsymbol{\lambda}_i$。

其次，当 $r=2$ 时，且 2 个失效节点中既有系统节点也有冗余节点。设节点 u 和节点 $k+v$ 失效，其中，$\forall u$，$v\in K$。在修复失效节点 u 和 $k+v$ 过程中，再生节点 u'将收到来自帮助节点的数据

$$\begin{cases}\boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\theta}_i\\ \boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\delta}_j=\omega\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\theta}_u+\sigma\boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\lambda}_j\end{cases} \tag{10}$$

再生节点$(k+v)'$将收到来自帮助节点的数据

$$\begin{cases}\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\theta}_i \\ \boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\delta}_j=\omega\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\lambda}_v+\sigma\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\lambda}_j\end{cases} \tag{11}$$

其中，$\forall i\in K\backslash\{u\}$，$\forall j\in K\backslash\{v\}$。

因为正交关系$\sum_\varepsilon\psi_{i\varepsilon}\varphi_{\varepsilon j}$服从克罗内克$\delta_{ij}$函数（当$i=j$时，$\sum_\varepsilon\psi_{i\varepsilon}\varphi_{\varepsilon j}=1$；当$i\neq j$时，$\sum_\varepsilon\psi_{i\varepsilon}\varphi_{\varepsilon j}=0$。），所以节点$(k+v)'$中接收到的数据式（11）的线性组合$\sum_{j=1,j\neq v}^{k}\varphi_{ju}\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\delta}_j+(\omega+\sigma)\sum_{i=1,i\neq u}^{k}\psi_{iv}\varphi_{vu}\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\theta}_i$可简化为由$\boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\lambda}_v$和$\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\theta}_u$的线性组合形式

$$\omega\boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\lambda}_v+\left(\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}\right)\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\theta}_u \tag{12}$$

然后节点$(k+v)'$将式（12）发送给节点u'。因为此时节点u'已收到数据式（10），所以可以将式（12）中的干扰数据$\omega\psi_{iv}\boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\theta}_i$消除。因此，根据式（10）可计算得到

$$\begin{cases}\left(\omega\psi_{uv}\boldsymbol{y}_u^{\mathrm{T}}+\left(\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}\right)\boldsymbol{x}_v^{\mathrm{T}}\right)\boldsymbol{\theta}_u \\ \left(\omega\boldsymbol{x}_j^{\mathrm{T}}+\sigma\psi_{uj}\boldsymbol{y}_u^{\mathrm{T}}\right)\boldsymbol{\theta}_u\end{cases} \tag{13}$$

此时若$k\times k$阶矩阵

$$\boldsymbol{A}=\begin{bmatrix}\left(\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}\right)\boldsymbol{x}_v^{\mathrm{T}}+\omega\psi_{uv}\boldsymbol{y}_u^{\mathrm{T}} \\ \omega\boldsymbol{x}_1^{\mathrm{T}}+\sigma\psi_{u1}\boldsymbol{y}_u^{\mathrm{T}} \\ \vdots \\ \omega\boldsymbol{x}_{v-1}^{\mathrm{T}}+\sigma\psi_{u,v-1}\boldsymbol{y}_u^{\mathrm{T}} \\ \omega\boldsymbol{x}_{v+1}^{\mathrm{T}}+\sigma\psi_{u,v+1}\boldsymbol{y}_u^{\mathrm{T}} \\ \vdots \\ \omega\boldsymbol{x}_k^{\mathrm{T}}+\sigma\psi_{uk}\boldsymbol{y}_u^{\mathrm{T}}\end{bmatrix} \tag{14}$$

是非奇异矩阵，则向量$\boldsymbol{\theta}_u$可被恢复在节点u'处。因此，证明MER码的存在性转换为证明矩阵$\boldsymbol{A}$可逆。

下面分别从$\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}$等于零和不等于零这2种情况进行证明。

当$\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}=0$时，式（14）可简化为矩阵

$$\boldsymbol{B}=\begin{bmatrix}\omega\psi_{uv}\boldsymbol{y}_u^{\mathrm{T}} \\ \omega\boldsymbol{X}_{K\backslash\{v\}}^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix}\omega\psi_{uv}\sum_{\varepsilon=1}^{k}\varphi_{\varepsilon u}\boldsymbol{x}_\varepsilon^{\mathrm{T}} \\ \omega\boldsymbol{X}_{K\backslash\{v\}}^{\mathrm{T}}\end{bmatrix} \tag{15}$$

其中，矩阵$\boldsymbol{X}_{K\backslash\{v\}}=[\boldsymbol{x}_1\cdots\boldsymbol{x}_{v-1}\boldsymbol{x}_{v+1}\cdots\boldsymbol{x}_k]$。显然式（15）中的矩阵$\boldsymbol{B}$可以进一步简化为一个非奇异矩阵。

当$\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}\neq0$时，根据式（2）可得$\boldsymbol{y}_u=\sum_{\varepsilon=1}^{k}\varphi_{\varepsilon u}\boldsymbol{x}_\varepsilon$。因此，式（14）可分解为

$$\begin{aligned}\boldsymbol{C}&=\boldsymbol{C}_1\boldsymbol{C}_2\\&=\begin{bmatrix}\sigma-\sigma\psi_{uv}\varphi_{vu} & \omega\psi_{uv}\boldsymbol{\Phi}_{K\backslash\{v\},u}^{\mathrm{T}} \\ \sigma\varphi_{vu}\boldsymbol{\Psi}_{u,K\backslash\{v\}} & \omega\boldsymbol{I}+\sigma\boldsymbol{\Psi}_{u,K\backslash\{v\}}\boldsymbol{\Phi}_{K\backslash\{v\},u}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_v^{\mathrm{T}} \\ \boldsymbol{X}_{K\backslash\{v\}}^{\mathrm{T}}\end{bmatrix}\end{aligned} \tag{16}$$

其中，$\boldsymbol{\Psi}_{u,K\backslash\{v\}}=[\psi_{u1}\cdots\psi_{u,v-1}\ \psi_{u,v+1}\cdots\psi_{uk}]^{\mathrm{T}}$，$\boldsymbol{\Phi}_{K\backslash\{v\},u}=[\varphi_{1u}\cdots\varphi_{v-1,u}\ \varphi_{v+1,u}\cdots\varphi_{ku}]^{\mathrm{T}}$，矩阵$\boldsymbol{I}$是$(k-1)\times(k-1)$阶的单位矩阵。所以证明矩阵$\boldsymbol{A}$的非奇异性也转换为证明式（16）中矩阵$\boldsymbol{C}$的前半部分$\boldsymbol{C}_1$的非奇异性。对$\boldsymbol{C}_1$进行分解可得

$$\begin{aligned}\boldsymbol{C}_1&=\boldsymbol{\Lambda}+\boldsymbol{g}\boldsymbol{f}^{\mathrm{T}}\\&=\begin{bmatrix}\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu} & 0 \\ 0 & \omega\boldsymbol{I}\end{bmatrix}+\\&\quad\begin{bmatrix}\omega\psi_{uv} \\ \sigma\boldsymbol{\Psi}_{u,K\backslash\{v\}}\end{bmatrix}\begin{bmatrix}\varphi_{vu} & \boldsymbol{\Phi}_{K\backslash\{v\},u}^{\mathrm{T}}\end{bmatrix}\end{aligned} \tag{17}$$

因为$\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}$和$\omega$是非零元素，所以对角矩阵$\boldsymbol{\Lambda}$是可逆的。

因此根据 Sherman-Morrison 公式可以推出。若$1+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{g}\neq0$，则矩阵$\boldsymbol{C}_1$是非奇异的。因为$\sum_{\varepsilon=1}^{k}\psi_{i\varepsilon}\varphi_{\varepsilon j}$服从克罗内克$\delta_{ij}$函数，所以有$\sum_{\varepsilon=1}^{k}\psi_{u\varepsilon}\varphi_{\varepsilon u}\equiv1$，从而可推出

$$1+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{g}=\frac{\sigma(\sigma+\omega)\left(1-\psi_{uv}\varphi_{vu}\right)^2}{\sigma-(\sigma+\omega)\psi_{uv}\varphi_{vu}} \tag{18}$$

将式（7）中的2个等式平方后相减，可得到$(\omega^2-\sigma^2)(\omega'^2-\sigma'^2)=1$。因此，结合定理1中的约束条件，有$\omega^2\neq\sigma^2$，$\sigma\neq0$，$\psi_{uv}\varphi_{vu}\neq1$。所以$1+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\boldsymbol{g}\neq0$。至此，证明得到式（14）中矩阵$\boldsymbol{A}$的非奇异性，MER码的存在性，以及节点$u'$可以修复出向量$\boldsymbol{\theta}_u$。

因为MER码继承了E-MDS码的对偶性，所以节点$(k+v)'$可以在接收到节点u'发送来的数据$\omega'\boldsymbol{x}_v^{\mathrm{T}}\boldsymbol{\lambda}_u'+(\sigma'-(\sigma'+\omega')\varphi_{vu}\psi_{uv})\boldsymbol{y}_u^{\mathrm{T}}\boldsymbol{\delta}_v$后，将节点$k+v$中丢失的数据恢复出来。

这样完成定理1的证明。

根据以上的证明过程可以看出，MER码作为E-MDS码的扩展，不但保持了E-MDS码在节点修复过程中满足再生码的割集边界的性质，而且可以实现对r个系统节点（冗余节点），或2个节点（单个系统节点和单个冗余节点）的精确修复。

4 架构模型的数值仿真分析对比

在多节点修复问题中，前面本文已经理论上分

析了具有健康节点协作的 MDS 双码架构模型与文献[12,13]中的架构模型比较，本文提出的架构模型在修复带宽、数据传输链路、中间节点存储量运算量、系统可靠灵活性等方面都有一定的改进。

接下来，本文进行数值仿真对比。把几种多节点修复模型（原始修复[1]、依次修复[15]、联合修复[11]、健康节点协作修复[12]）与本文修复方案进行对比。为方便，当 $k=d, r=n-k$，且 $n<2k$ 时，把相关参数用集合的形式表示为$(n, k, d)(\alpha, B)$。

4.1　修复带宽比较

在多节点失效的环境下，MDS 双码架构模型的节点的修复过程中用来修复失效节点的每一个健康节点的信息字符都是相互独立的。因此，每个健康节点只需识别中间节点的编码向量。并且在此架构中，整个修复过程是以一种分布式方式完成的。这种方式可以使整个修复系统更容易实现，并进一步减少节点之间的修复带宽消耗。用 $\overline{\gamma}$ 表示平均每个失效节点所消耗修复带宽的大小，各修复模型的平均修复带宽分析结果如表 1 所示。可以明显地看出，本文修复模型每个节点平均的修复带宽最优。并且随着云存储中节点数量的增多，优势更加明显。

表 1　　**修复带宽比较**

$(n, k, d)(\alpha, B)$	原始修复	依次修复	联合修复	健康节点协作修复	本文修复模型
(4,2,2) (2,4)	$\overline{\gamma}$=4	$\overline{\gamma}$=3	$\overline{\gamma}$=3	$\overline{\gamma}$=3	$\overline{\gamma}$=3
(7,4,4)(21,84)	$\overline{\gamma}$=84	$\overline{\gamma}$=51.3	$\overline{\gamma}$=42	$\overline{\gamma}$=42	$\overline{\gamma}$=35
(14,8,8)(80,640)	$\overline{\gamma}$=640	$\overline{\gamma}$=261.3	$\overline{\gamma}$=173.3	$\overline{\gamma}$=173.3	$\overline{\gamma}$=160

4.2　数据传输链路数比较

在多节点修复过程中，在网络上传输数据所用传输链路数 f 越小，系统可靠性越大。各修复模型所用的传输链路数结果如表 2 所示。可以明显地看出，本方案不仅减少了修复带宽，也有效地减少了修复传输链路数目，简化修复过程，增加可靠性，减少对网络资源的浪费，达到了安全高效地修复节点的目的。

表 2　　**数据传输链路数比较**

$(n, k, d)(\alpha, B)$	联合修复	健康节点协作修复	本文修复模型
(4,2,2)(2,4)	f=6	f=3	f=3
(7,4,4)(21,84)	f=18	f=6	f=5
(14,8,8)(80,640)	f=78	f=13	f=12

具有健康节点协作的多节点修复方案虽然传输链路数已经大大减少，但是本方案在同等参数下比它的传输链路数少 1，在减少中间节点存储运算负担的同时，减少了数据传输链路数，从而减少链路传输数据失败的几率，使系统更加可靠。

4.3　中间节点上的数据量比较

由于在 MDS 双码架构下健康节点协作的多节点修复方案中，2 种编码均会选择一个健康节点作为修复对方编码中失效节点的中间节点 m_1 和 m_2，并分别将修复所需的数据传输给相应的中间节点（m_1 或 m_2）。因此相比健康节点协作修复方案[12]，分散了当修复多个失效节点时，单个中间节点上的数据量。

各修复模型中，单个中间节点上的最大数据量如表 3 所示。虽然中间节点的总数据量并没有明显减少，但本文的修复模型通过增加中间节点个数，使得单个中间节点的处理数据量减少，从而分散了中间节点的处理压力。

表 3　　**单个中间节点上的数据量比较**

$(n, k, d)(\alpha, B)$	健康节点协作修复	本文修复模型
(4,2,2)(2,4)	4	2
(7,4,4)(21,84)	21	11
(14,8,8)(80,640)	86.7	44

除了以上 3 个方面的优势外，MDS 双码架构模型的另一个主要优势在与它适用于任意的线性纠删码，而 MER 编码方案即是一种适用于多个系统节点和冗余节点的多节点修复线性网络编码方案。因此只要在 E-MDS 码的基础上使各参数满足定理 1 中的约束条件，存在一种 MER 码可适用于具有健康节点协作的 MDS 双码架构模型，使系统性能更优。

5　结束语

本文首先根据 MDS 双码架构模型结合具有健康节点协作的多节点修复模型，给出了一种具有健康节点协作的 MDS 双码架构模型。该模型在具有健康节点协作的多节点修复方案的基础上，解决了中间节点存储、运算负担大的问题，尤其在海量存储节点的网络中，此优势会更加明显。其次，本文通过约束参数条件，将适用于单节点修复的 E-MDS 码扩展成了适用于多个系统节点或冗余节点同时修复的 MER 码，并证明了其存在性。并在理论意义上将

MER 码与具有健康节点协作的 MDS 双码架构模型相结合，以达到对多节点修复的同时，降低修复带宽、修复链路数和单个中间节点需要处理的数据量。最后本文将具有健康节点协作的 MDS 双码架构模型与现有的几种架构模型进行数值仿真对比。结果表明，在进行多节点修复时，本文给出的架构模型减少了修复带宽和数据传输链路数，降低了中间节点的数据处理量，进一步提高系统可靠性。对于下一步工作，将从 MER 码的具体构造和双码架构下的 MER 码在现实网络中的实现进行研究。

参考文献：

[1] DIMAKIS A G, GODFREY P B, WAINWRIGHT M J, *et al*. Network coding for distributed storage systems[A]. IEEE International Conference on Computer Communications (INFOCOM)[C]. 2007. 2000-2008.

[2] DIMAKIS A G, RAMCHANDRAN K, WU Y, *et al*. A survey on network codes for distributed storage[J]. Proceedings of the IEEE, 2011, 99(3): 476-489.

[3] RASHMI K V, SHAH N B, KUMAR P V. Optimal exact regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction[J]. IEEE Transactions on Information Theory, 2011, 57(8): 5227-5239.

[4] SUH C, RAMCHANDRAN K. Exact-repair MDS code construction using interference alignment[J]. IEEE Transactions on Information Theory, 2011, 57(3): 1425-1442.

[5] HU Y, XU Y, WANG X, *et al*. Cooperative recovery of distributed storage systems from multiple losses with network coding[J]. IEEE Journal on Selected Areas in Communications, 2010, 28(2): 268-275.

[6] WANG X, XU Y, HU Y, *et al*. MFR: multi-loss flexible recovery in distributed storage systems[A]. IEEE International Conference on Communications (ICC)[C]. 2010. 1-5.

[7] 谢显中, 黄倩, 王柳苏, 等. 一种云存储中基于干扰对齐的多节点精确修复方法[J]. 电子学报，2014, 42 (10): 1873-1881.
XIE X Z, HUANG Q, WANG L S, *et al*. A multi-node exact repair method in cloud storage based on interference alignment[J], Acta Electronica Sinica, 2014, 42 (10): 1873-1881.

[8] LE S N. Exact scalar minimum storage coordinated regenerating codes[A]. IEEE International Symposium on Information Theory Proceedings (ISIT)[C]. 2012. 1197-1201.

[9] WANG A, ZHANG Z. Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage[A]. IEEE International Conference on Computer Communications (INFOCOM)[C]. 2013. 400-404.

[10] LI J, LI B. Cooperative repair with minimum-storage regenerating codes for distributed storage[A]. IEEE International Conference on Computer Communications (INFOCOM)[C]. 2014. 316-324.

[11] SHUM K W, HU Y. Cooperative regenerating codes[J]. IEEE Transactions on Information Theory, 2013, 59(11): 7229-7258.

[12] 谢显中, 王柳苏, 黄倩, 等. 具有健康节点协作的高效多节点修复方案[J]. 北京邮电大学学报. 2014, 37(1):52-56.
XIE X Z, WANG L S, HUANG Q, *et al*. Efficient multi-node regenerating program with healthy nodes collaboration in distributed storage systems[J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37(1):52-56.

[13] RASHMI K V, SHAH N B, KUMAR P V. Enabling node repair in any erasure code for distributed storage[A]. IEEE International Symposium on Information Theory(ISIT)[C]. Saint Petersburg, Russia, 2011. 1235-1239.

[14] LI X, ZHENG Q, QIAN H, *et al*. Toward optimizing cauchy matrix for cauchy reed-solomon code[J]. IEEE Communications Letters, 2009, 13(8): 603-605.

[15] WU Y, DIMAKIS A G. Reducing repair traffic for erasure coding-based storage via interference alignment[A]. IEEE International Symposium on Information Theory Proceedings (ISIT)[C]. Seoul, 2009. 2276-2280.

作者简介：

谢显中（1966-），男，四川通江人，博士，重庆邮电大学教授，主要研究方向为无线和移动通信技术。

黄倩（1988-），女，重庆人，硕士，重庆邮电大学移通学院讲师，主要研究方向为云存储、个人通信。

王柳苏（1990-），女，重庆人，重庆邮电大学硕士生，主要研究方向为云存储、个人通信。