

基于稀疏网络编码的移动边缘缓存网络稳健性维持方法研究

殷俊¹, 夏欣然¹, 张登银¹, 柯亚琪², 杨余旺²

(1. 南京邮电大学物联网学院, 江苏 南京 210003; 2. 南京理工大学计算机科学与工程学院, 江苏 南京 210094)

摘要: 稀疏网络编码通过对若干数据块进行线性组合后再存储可提高移动边缘缓存 (MEC, mobile edge caching) 网络的缓存空间利用率, 降低内容调度的开销和复杂度。但 MEC 网络采用普通用户设备作为缓存节点存储这些编码内容, 因此用户设备的离开/失效会削弱该方法的健壮性。针对缓存节点失效后的编码内容恢复问题, 首先, 提出了一种基于查找表的编码内容管理方法, 可有效地组织缓存节点的稀疏编码内容; 其次, 建立了失效内容恢复的数学问题模型, 并证明了该问题属于 NP 类问题; 最后, 鉴于问题求解复杂性, 给出了一种启发式的失效节点内容恢复算法以维持 MEC 网络的健壮性, 先寻找可行的备用缓存节点集合, 再恢复失效内容。仿真测试进一步验证了该算法在恢复时延和恢复能耗上的良好性能。

关键词: 移动边缘缓存网络; 网络编码; 稳健性; 恢复算法

中图分类号: TN929.5

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2025.00387

Research on robustness maintenance for sparse network coding based mobile edge caching network

YIN Jun¹, XIA Xinran¹, ZHANG Dengyin¹, KE Yaqi², YANG Yuwang²

1. School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

2. School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Abstract: Sparse network coding, achieved by linearly combining several data blocks before storage, can enhance the cache space utilization of mobile edge caching (MEC) networks, reduce content scheduling overhead and complexity. However, as MEC networks utilize ordinary user devices as cache nodes to store this encoded content, the departure or failure of these user devices undermines the robustness of this approach in the network. In order to address the recovery of encoded content following cache nodes' failures, firstly, a lookup table-based encoding content management method was proposed. This method effectively organized the sparse encoding content within cache nodes. Secondly, a mathematical problem model for content recovery from failures was established, which was demonstrated that this problem belonged to the NP class. Finally, considering the complexity of problem-solving, a heuristic algorithm for recovery of failed nodes was presented to maintain the robustness of the MEC network. This algorithm initially sought a feasible set of alternative cache nodes, and then proceeded to recover the failed content. Simulation results further validate the effectiveness and good performance of this algorithm in terms of recovery latency and energy consumption.

Key words: MEC network, network coding, robustness, recovery algorithm

收稿日期: 2023-12-05; 修回日期: 2024-03-11

通信作者: 张登银, zhangdy@njupt.edu.cn

基金项目: 国防基础科研计划项目 (No. JCKY2023***007); 南邮自然科学基金资助项目 (No. NY221095, No. NY223106); 江苏省研究生实践创新计划项目 (No. SJCX24_0335)

Foundation Items: The National Defense Basic Research Foundation of China (No. JCKY2023***007), Nanjing University of Posts and Telecommunications Science Foundation (No. NY221095, No. NY223106), Jiangsu Province Graduate Practice Innovation Program (No. SJCX24_0335)

0 引言

随着智能手机和移动设备的广泛普及，人们对高速、低时延通信服务的需求也在不断增加。然而，在传统的移动网络架构中，用户请求的数据需要通过核心网络传输，导致网络回程（backhaul）链路的带宽压力增大，尤其在高峰时段，这会严重影响网络性能和用户体验^[1]。为了缓解这一问题，研究者提出了移动边缘缓存（MEC, mobile edge caching）网络。

MEC网络通过将热门内容提前存储在离用户更近的边缘服务器或设备上，减轻了核心网络的负载，降低了传输时延，提高了数据访问速度。同时，结合设备对设备（D2D, device-to-device）通信技术，MEC网络允许用户之间直接进行内容分享，可更有效地利用用户设备的存储空间^[2-3]。

然而，由于边缘节点和用户设备的存储空间有限，且用户设备会频繁地移动位置，MEC网络面临的主要问题是高效地管理存储空间，以最大化存储空间的利用^[4-5]。为了解决这一问题，研究者们^[6-10]引入了网络编码技术，对待缓存内容进行编码，然后再分布式地存储到用户设备上。当内容请求用户需要获取某个内容（文件）时，只需要从这些用户设备上获取若干编码块即可解码出原文件。由于避免了数据块价值各异的问题，网络编码方法极大地提高了内容分发和调度效率，增强了系统的鲁棒性。

尽管在MEC网络中引入网络编码技术具有上述优势，但以随机线性网络编码（RLNC, random linear network coding）为代表的网络编码、解码依赖矩阵乘法和求逆运算，运算量较大^[10-13]，如何设计更优的码字构造方法，降低节点运算成本是一个待解问题。

更重要的是，近年来研究人员^[14-15]发现，随着携有缓存内容的节点离开，基于网络编码的分布式存储系统通常只能在某个用户请求文件失效时才意识到系统内存储的目标文件不可用，这将难以保证系统的服务质量（QoS, quality of service）。

本文面向稀疏随机线性网络编码（SRLNC, sparse random linear network coding）^[16-17]的MEC网络，提出了一种健壮性维持方法。本文讨论SRLNC是由于其相较于经典的RLNC，SRLNC参与编码

的数据块量少，具有编码效率低、解码时延小等优势^[18-20]，因此更适用于MEC网络的应用场景。

本文的具体贡献如下：

- 1) 提出了一种基于查找表的稀疏网络编码块管理方法，可高效地组织用户设备的编码内容；
- 2) 针对MEC网络缓存节点离开后的失效内容修复问题，建立了最优恢复问题的数学模型，并证明了该问题属于NP类问题；
- 3) 给出了一种启发式的节点失效恢复算法，采用两阶段思路，首先寻找备用节点集合，再进行编码内容生成，仿真结果亦验证了该算法的有效性。

1 系统模型

典型移动边缘缓存网络系统架构如图1所示^[4, 12, 21-26]，由一个云基站、若干个边缘节点和随机分布在边缘节点周围的用户设备构成。在此3层网络架构中，云基站、边缘节点和用户设备可以共同进行内容缓存和传递。根据MEC部署目的的不同，边缘节点可以由配置了边缘（存储）服务器的小基站充当^[21]，亦可以是加载了部署区域内的Wi-Fi接入点（AP, access point）^[4, 21-22]。

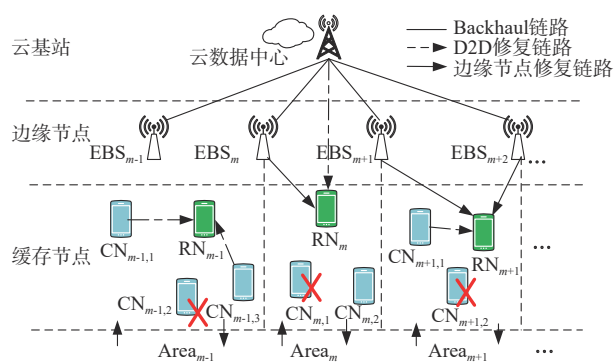


图1 典型移动边缘缓存网络系统架构

用户设备是处于系统内的智能手机、平板电脑等，可以用于数据缓存，并且近邻范围内的这些用户设备可以使用Wi-Fi或者蓝牙等D2D链路相互通信。为了方便阐述，缓存内容的用户设备又称为缓存节点，而某个节点由于失效离开系统后，用以替代该节点的用户设备被称为后备节点/恢复节点。

MEC网络中的用户设备内容存储是以块（block）为单位进行的^[12, 15]，若块大小为 B ，则一个具有 F 字节的内容文件首先经过分片操作，被分成 n 个相等大小的源数据块（ $n=F/B$ ）。若 $\text{mod}(F, n) \neq 0$,

则说明不能等分，需在最后一个块末尾补 NULL 直至齐平。

尽管每个用户设备可能同时落在多个边缘节点的覆盖区域内，但只能与一个边缘节点相连。而每个边缘节点管理所辖区域内用户设备的内容发布、订阅等任务，具体的实现机制见文献[4, 12]，在此不作赘述。

1.1 编码内容缓存

在基于网络编码的 MEC 网络中，用户设备的内容存储的是编码块而不是这些源数据块。如果某个源文件 \mathbf{X} 分片操作后包含源数据块 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ ，那么任意一个编码块 \mathbf{b} 由 n 个源数据块在某有限域上的线性组合生成。

$$\mathbf{b} = \mathbf{c} \cdot \mathbf{X} = (c_1, \dots, c_n) \cdot (\mathbf{x}_1, \dots, \mathbf{x}_n)^T \quad (1)$$

其中，向量 \mathbf{c} 称为编码块 \mathbf{b} 的编码向量，其元素是从该有限域中随机生成的；当 $\|\mathbf{c}\|_0 = 2$ 时，编码块 \mathbf{b} 是一个二稀疏随机线性编码块，而此时编码向量 \mathbf{c} 的非零元个数为 2；不失一般性，若记非零元为 c_i 和 c_j ，则可记 \mathbf{b} 为 \mathbf{b}_{i,j,c_i,c_j} ；在非零元 c_i 和 c_j 的非零取值不影响价值的情况下，一些研究^[27]也直接简称为 $\mathbf{b}_{i,j}$ ，即

$$\mathbf{b}_{i,j} = c_i \mathbf{x}_i + c_j \mathbf{x}_j \quad (2)$$

1.2 内容请求

当用户设备连接到某个 MEC 网络的边缘节点时，该边缘节点将根据当前热门内容和其他用户发布的与该用户订阅的文件类型相关的新内容进行推送。当用户确认某个文件请求后，会首先通过 D2D 向周边邻居设备请求下载尽可能多的该文件编码块，若周边邻居不能满足下载全部编码块，将转由边缘节点传输剩余块。在此过程中，边缘节点根据分布式哈希表 (DHT, distributed Hash table) 等数据结构可进行目标内容的缓存节点定位与匹配^[4, 12, 26]，并将结果推送给相应的节点。用户请求到与源内容相等数量的编码块 (n 个)，即可以以很高的概率解码出源文件^[28]。

1.3 内容损失与修复

当用户设备由于设备关机或离开某一边缘节点的覆盖区域时，该用户设备中缓存的数据（编码块）对于该子区域的 MEC 网络来说已经损失。损失的编码块需要及时重建并缓存到具有可用存储容量的后备用户设备上，才能保持 MEC 网络的健壮性。因此，需要解决当一个区域内，某个内容缓存

节点离开 MEC 网络后，如何以最小的通信时延恢复丢失编码块的问题。注意这里讨论的编码块恢复是指等价修复，即仅要求修复后的编码块对应的编码系数向量非零元与原编码块的编码系数向量非零元位置一致，而不要求非零元值也一致。

尽管云基站包含了所有内容，但请求云基站进行恢复的时延、通信开销较大，故本文考虑一个尽可能少请求云基站的丢失数据恢复机制，从而充分利用用户设备间可进行 D2D 通信的条件，进行数据恢复，仅当周围缓存节点无可用数据时再考虑向云基站请求数据。因此损失编码块的修复需考虑如下两个问题。

1) 全部由缓存节点恢复、部分由缓存节点恢复部分由云基站恢复、全部由云基站恢复 3 种情况下，哪种方法恢复时延最小？

2) 若全部或部分由缓存节点恢复，需选择哪些缓存节点进行恢复是最优的？

由于编码内容的恢复过程与内容的缓存管理密切相关，因此，在介绍所提的恢复方法前，本文首先给出必要的编码内容在内容缓存节点的管理方法。

2 编码内容缓存管理

缓存管理的首要任务是表征用户设备、缓存节点所缓存的编码块，方便编码内容分享和缓存节点失效时编码内容的修复。本文提出缓存内容查找表 (CCT, cache content lookup table) 来记录节点缓存中编码块的信息。

定义 1 缓存内容查找表

内容查找表是一个二维表，每一行对应一个编码块的编码向量；用户设备/缓存节点在时刻 t 关于文件 f 的缓存内容查找表 CCT _{t} 的第 i 行元素含义为

$$\text{CCT}_t^i(j) = \begin{cases} c_i \text{ 或 } c_j, & \text{第 } i \text{ 个编码块是 } \mathbf{b}_{i,j} \text{ 或 } \mathbf{b}_{j,i} \\ 0, & \text{其他} \end{cases} \quad (3)$$

在编码内容损失后的恢复过程中需要频繁生成某一编码块，通过缓存内容查找表可以很容易判断在 t 时刻某用户设备是否能够产生某编码块 $\mathbf{b}_{m,n}$ ，即查该用户设备对应文件的每一行，使存在 $k \in \{1, \dots, n\}$ ，满足 $\text{CCT}_t^k(m) \neq 0$ 且 $\text{CCT}_t^k(n) \neq 0$ 。此外，鉴于 CCT 是稀疏二维表，因此实践中可以采用三元组等方式节省存储开销。

3 健壮性维持方法

3.1 缓存节点失效与修复问题模型

假设所考虑的 MEC 网络包含 N 个缓存节点 (记为 $\mathbf{CN} = \{\mathbf{CN}_1, \mathbf{CN}_2, \dots, \mathbf{CN}_N\}$) 和 U 个边缘节点 (记为 $\mathbf{EBS} = \{\mathbf{EBS}_1, \mathbf{EBS}_2, \dots, \mathbf{EBS}_U\}$)；每个缓存节点最多可存储 C 个编码块，且不会存重复的编码块。由于后备缓存节点及其恢复过程都是工作在电池驱动的智能终端上，因此恢复机制的评价从两个方面进行分析评估：恢复能耗和恢复时延。为了量化分析恢复能耗，引入自由空间信道模型^[29]，一个节点向距离 d 的节点发送 l bit 数据的能耗可表示为

$$E_T = \begin{cases} lE_c + l\varepsilon_{fs}d^2, & d < d_0 \\ lE_c + l\varepsilon_{mp}d^4, & d \geq d_0 \end{cases} \quad (4)$$

接收数据能耗为

$$E_R = lE_c, \quad (5)$$

其中， E_c 是发射电路每发送 1 bit 信息的能量消耗， ε_{fs} 是自由空间损耗系数， ε_{mp} 是多径衰弱损耗系数， d_0 是自由空间传输模式和多径衰弱传输模式的切换阈值，可通过式(6)得到，

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \quad (6)$$

不失一般性，假设缓存节点 \mathbf{CN}_i ($i \in N$) 失效，即其缓存的 c 个编码块 (记为 $\mathbf{c}_i = \{c_1, c_2, \dots, c_c\}$) 全部丢失。下面将从恢复任意编码块 c_i ($c_i \in \mathbf{c}_i$) 的恢复时延和恢复能耗来进行具体分析。

1) 周边缓存节点不可生成编码块

后备节点的周边缓存节点上都不具备恢复任意一个编码块 c_i 的条件，如在图 1 的区域 Area_m ，后备节点 \mathbf{RN}_m 只能转向管辖该区域的边缘节点 \mathbf{EBS}_m 获取；边缘节点继而从云基站请求相关内容后返回给后备节点 \mathbf{RN}_m 。考虑到云基站恢复内容时不仅需要通信时延，还需要考虑计算、转码、资源调度等时间开销 (记为 w)，则所需的通信时延可表示为

$$D_{c_i, \mathbf{EBS}} = \min(d_{i, \mathbf{EBS}} + w) \quad (7)$$

$$E_{c_i, \mathbf{EBS}} = f(d_{i, \mathbf{EBS}}) \quad (8)$$

其中， $d_{i, \mathbf{EBS}}$ 表示后备节点 i 和边缘节点 \mathbf{EBS} 之间的通信距离， $D_{c_i, \mathbf{EBS}}$ 表示全部由云基站恢复数据块 c_i 的最小通信时延， $E_{c_i, \mathbf{EBS}}$ 表示全部由云基站恢复数据块 c_i 的最小恢复能耗。

2) 周边缓存节点可恢复部分编码块

如在图 1 的区域 Area_{m+1} 内的后备节点 \mathbf{RN}_{m+1} ，此

时周边缓存节点，可以恢复出部分丢失的编码块，因此对于恢复任意一个编码块 c_i 分为以下 3 种情况。

a) 若匹配各周边缓存节点 \mathbf{CN}_j 中的 CCT 表，发现存在与 c_i 原始数据对相同的编码块，即可直接恢复出数据块 c_i ，此时的时延可记为 $d_{i,j}$ 。其中， $d_{i,j}$ 表示存储节点 i 和 j 之间的通信距离。

b) 发现周边缓存节点 \mathbf{CN}_j 中存在的编码块经一次再编码后可生成与 c_i 等价的编码块，此时恢复时延可表示为 $d_{i,j} + o_{c_i}$ 。其中， o_{c_i} 是再编码产生一个编码块的运算时延。

c) 发现周边缓存节点 \mathbf{CN}_j 和 \mathbf{CN}_k 存在的编码块经一次再编码后可生成 c_i 的编码块，此时恢复数据块 c_i 一个编码块的时延可表示为 $d_{i,j} + d_{i,k} + o_{c_i}$ 。

所以，利用缓存节点恢复一个编码数据块，再向边缘节点请求一个数据块从而恢复数据块 c_i 的最优通信时延可表示为

$$D_{c_i, j, \mathbf{EBS}} = \min\{d_{i,j} + d_{i, \mathbf{EBS}} + w, d_{i,j} + d_{i, \mathbf{EBS}} + w + o_{c_i}, d_{i,j} + d_{i,k} + d_{i, \mathbf{EBS}} + w + o_{c_i}\} \quad (9)$$

$$E_{c_i, j, \mathbf{EBS}} = f(d_{i, \mathbf{EBS}}, d_{i,j}, d_{i,k}), \quad (10)$$

其中， $D_{c_i, j, \mathbf{EBS}}$ 表示缓存节点和云基站共同恢复数据块 c_i 的最优通信时延， $E_{c_i, j, \mathbf{EBS}}$ 表示缓存节点和云基站共同恢复数据块 c_i 的最优恢复能耗。

3) 周边缓存节点可生成全部编码块信息

后备节点的周边缓存节点有全部数据，如图 1 的区域 Area_{m+2} ，此时周边缓存节点具备恢复丢失数据块的所有条件，且可能不止一个缓存节点中含有可用数据，此时需考虑如何选取最优恢复节点的问题。本文用集合 R 表示可以再编码生成 c_i 编码块的节点对， $R = \{(\mathbf{CN}_j, \mathbf{CN}_k), \dots, (\mathbf{CN}_m, \mathbf{CN}_n)\}$ ，因此周边缓存节点具有全部数据时又分以下 3 种情况。

a) 缓存节点 j 和 k 分别有一个与 c_i 原始数据相同的编码块，可以直接恢复出两个 c_i 的编码块，此时恢复数据块 c_i 的时延可表示为 $d_{i,j} + d_{i,k} + 2o_{c_i}$ 。

b) 缓存节点 j 有一个与 c_i 原始数据相同的编码块，用户设备对 \mathbf{CN}_m 和 \mathbf{CN}_n 所含数据可再编码生成一个 c_i 的编码块，此时恢复数据块 c_i 的时延可表示为 $d_{i,j} + d_{i,m} + d_{i,n} + 2o_{c_i}$ 。

c) 用户设备对 \mathbf{CN}_j 、 \mathbf{CN}_k 和 \mathbf{CN}_m 、 \mathbf{CN}_n 可分别再编码生成两个 c_i 的编码块，此时恢复数据块 c_i 的时延可表示为 $d_{i,j} + d_{i,k} + d_{i,m} + d_{i,n} + 2o_{c_i}$ 。

因此, 完全利用用户设备中现有数据恢复丢失数据块 c_i 的最优通信时延可表示为

$$D_{c_i,j,k} = \min \{d_{i,j} + d_{i,k} + 2o_{c_i}, d_{i,j} + d_{i,m} + d_{i,n} + 2o_{c_i}, d_{i,j} + d_{i,k} + d_{i,m} + d_{i,n} + 2o_{c_i}\} \quad (11)$$

$$E_{c_i,j,k} = f(d_{i,j}, d_{i,k}, d_{i,m}, d_{i,n}) \quad (12)$$

其中, $D_{c_i,j,k}$ 表示全部由云基站恢复数据块 c_i 的最优通信时延, $E_{c_i,j,EBS}$ 表示缓存节点和云基站共同恢复数据块 c_i 的最优恢复能耗。

综上, 对于丢失的数据块 c_i , 恢复它的最优通信时延可表示为

$$D_{c_i} = \min \{D_{c_i,EBS}, D_{c_i,j,EBS}, D_{c_i,j,k}\} \quad (13)$$

最优恢复能耗为

$$E_{c_i} = \min \{E_{c_i,EBS}, E_{c_i,j,EBS}, E_{c_i,j,k}\} \quad (14)$$

则恢复失效节点 i 丢失的所有数据块 c_i 的最优恢复时延 D_{Cost} 和能耗 E_{Cost} 为

$$D_{Cost} = \sum_{c_i=1}^c D_{c_i}, c < C \quad (15)$$

$$E_{Cost} = \sum_{c_i=1}^c E_{c_i}, c < C \quad (16)$$

3.2 最优缓存节点失效修复问题性质

定义2 最小顶点覆盖问题

设一个无向图 $G = (V, E)$, 其中, V 表示顶点集合, 共有 n 个顶点, E 表示边集合, 共有 m 条边。目标是找到顶点尽可能少的顶点子集, 使这些顶点能够覆盖图中的每条边, 这里的覆盖是指一条边的两个顶点至少有一个在顶点子集中。

引理1 最小顶点覆盖问题是一个NP-Complete问题。

证明 见文献[30]第4节, 第135~136页。

定理1 移动边缘缓存网络中可提供数据的后备缓存节点的选择问题是一个NP问题。

证明 不失一般性, 假设某缓存网络中有 n 个可提供数据的缓存节点, 可提供数据的缓存节点组合有 m 个。若仅考虑特例, 即丢失数据恢复时仅考虑全部由缓存节点恢复, 最优缓存节点的选取问题和移动传感器部署问题是相同的, 所以后备缓存节点的选择问题是该问题的特化, 而由引理1可知该问题是NP难问题, 因此后备缓存节点的选择问题是NP类问题。

3.3 两阶段启发式缓存节点失效的内容恢复算法

鉴于NP问题的复杂性, 本文提出了一种启发

式的恢复算法, 通过快速恢复丢失编码块内容来保持系统的健壮性。算法思路是: 首先, 对失效节点的周边缓存节点的数据进行查找获取周边缓存节点集合 R ; 其次, 根据集合 R 状态选择合适的恢复数据的方式; 最后, 得到恢复失效节点全部数据的最小恢复时延。MEC网络缓存节点失效的启发式恢复算法如算法1所示。

算法1 MEC网络缓存节点失效的启发式恢复算法

输入 缓存节点CN, 失效的缓存节点 CN_i 及其缓存的编码文件块 c

输出 最小恢复时延 D_{Cost} 和恢复每个编码块 c_i 的设备号restore

步骤1 获取周边的备用缓存节点集合 R

初始化; //初始化 N 个用户设备缓存 c 个编码块数据, 以及用户设备之间、用户和边缘节点间的通信距离, 并用无穷大表示不可通信

缓存节点 CN_i 失效; //在 N 个用户设备中随机选择一个设备, 清空其缓存内容模拟节点失效

for $i=1:1:c$ //与失效节点每个编码块对比

if $CCT'_n(i) \neq 0$ 且 $CCT'_m(i) \neq 0$

$R \leftarrow [R; n, m]$; //当前两个缓存节点内的数据可再编码生成 c_i , 记录在集合 R 中

end if

end for

步骤2 数据恢复

for $i=1:1:c$ //对失效节点每个编码块恢复

$D_{c_i,EBS} \leftarrow \min(d_{i,EBS} + w)$;

$E_{c_i,EBS} \leftarrow f(d_{i,EBS})$; //全部由云基站恢复时最小通信时延和能耗

restore($i, :$) $\leftarrow [0, 0]$; //记: 录恢复数据块 c_i 的缓存节点号, 这里用0表示云基站

$D_{c_i,j,EBS} \leftarrow \min\{d_{i,j} + d_{i,EBS} + w, d_{i,j} + d_{i,EBS} + w + o_{c_i}, d_{i,j} + d_{i,k} + d_{i,EBS} + w + o_{c_i}\}$;

$E_{c_i,j,EBS} \leftarrow f(d_{i,EBS}, d_{i,j}, d_{i,k})$; //周边节点与云基站共同恢复

restore($i, :$) $\leftarrow [j, 0]$; //缓存节点 j 与云基站共同恢复数据

$D_{c_i,j,k} \leftarrow \min\{d_{i,j} + d_{i,k} + 2o_{c_i}, d_{i,j} + d_{i,m} + d_{i,n} + 2o_{c_i}, d_{i,j} + d_{i,k} + d_{i,m} + d_{i,n} + 2o_{c_i}\}$;

```

 $E_{c_i,j,k} \leftarrow f(d_{i,j}, d_{i,k}, d_{i,m}, d_{i,n}); //全部由周$ 
    边节点恢复
    restore( $i, : \leftarrow [j, k];$ 
 $D_{c_i} \leftarrow \min\{D_{c_i,EBS}, D_{c_i,j,EBS}, D_{c_i,j,k}\}; //恢复失效$ 
    节点丢失的文件块 $c_i$ 的最小通信时延
 $E_{c_i} \leftarrow \min\{E_{c_i,EBS}, E_{c_i,j,EBS}, E_{c_i,j,k}\}; //恢复失效$ 
    节点丢失的文件块 $c_i$ 的最小通信能耗
end for
 $D_{Cost} \leftarrow \sum_{c_i=1}^c D_{c_i};$ 
 $E_{Cost} \leftarrow \sum_{c_i=1}^c E_{c_i};$ 
return result  $\leftarrow (D_{Cost}, E_{Cost}, \text{restore})$ 
    
```

由于在获取周边缓存节点集合 R 时，最坏情况下除失效节点外，需遍历剩余 $N-1$ 个缓存节点，与失效节点 i 丢失的 c 个数据进行对比，故获取集合 R 的时间复杂度为 $O(cN^2)$ ，而在数据恢复阶段，需要对失效节点 c 个编码块进行处理，故时间复杂度为 $O(c)$ ，因此整个启发式恢复算法的时间复杂度为 $O(cN^2)$ 。

4 仿真测试

本节仿真测试了所提的MEC网络系统中节点失效和恢复的过程。实验中，20个缓存节点和5个边缘节点随机部署在面积为 600 m^2 的MEC网络通信范围内，每个边缘节点管辖一定用户设备，MEC区域内节点加入或离开系统的行为服从泊松分布^[31]。内容文件 X 被分割成40等份，初始阶段每个缓存节点随机存储一定数量的编码块，数量不超过缓存节点的最大存储容量。仿真参数设置见表1。

表1 仿真参数设置

参数	取值
MEC网络通信面积/ m^2	600
缓存节点数量 N	20
边缘节点数量 U	5
文件块大小/ MB	[1, 10]
网络带宽/ $(\text{Mbit} \cdot \text{s}^{-1})$	100
文件分块数量 K	40
缓存节点最大缓存容量 $C/\text{块}$	[1, 10]
缓存节点间不可达率	[0, 1]
边缘节点通信距离/ m	[1, 50]
缓存节点间通信距离/ m	[1, 100]
发射电路消耗能量 $E_e/(\text{nJ} \cdot \text{bit}^{-1})$	50
自由空间损耗系数 ϵ_k	10

在仿真中，对比测试了两种具有代表性的方法：1) 云基站直接恢复：不考虑缓存节点间协作通信恢复丢失数据，直接全部由云基站提供数据完成恢复；2) 尽力恢复^[32]：每个与后备节点可通信的缓存节点设备都提供一个编码块，再编码后形成新的编码内容，完成恢复。

仿真从缓存节点最大缓存容量、云基站传输数据时延、缓存节点间的不可达率和文件块大小4个关键参数对恢复数据的时延进行评估。本文所评测的恢复时延主要由云基站传输时延、缓存节点间传输时延和数据恢复时延组成。

缓存节点最大缓存容量对恢复时延的影响如图2所示，缓存节点最大缓存容量对恢复能耗的影响如图3所示。从仿真结果可知，本文方法的恢复时延基本保持在 $1\sim 25 \text{ ms}$ ，恢复能耗基本保持在 $6 \mu\text{J}$ 之下，相较于另两个传统恢复方式性能较优。当缓存节点最大缓存容量较大时，所有后备用户设备内可用数据较多，本文方法对云基站数据的依赖相对较小，故恢复时延和能量开销低；而另外两种方法由

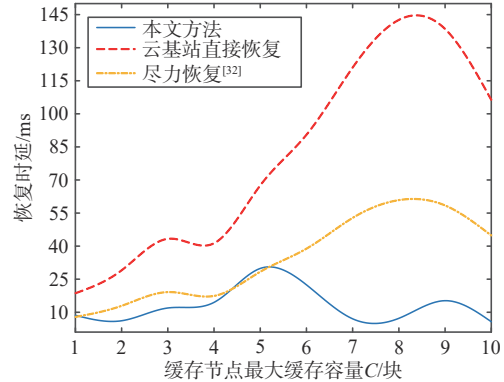


图2 缓存节点最大缓存容量对恢复时延的影响

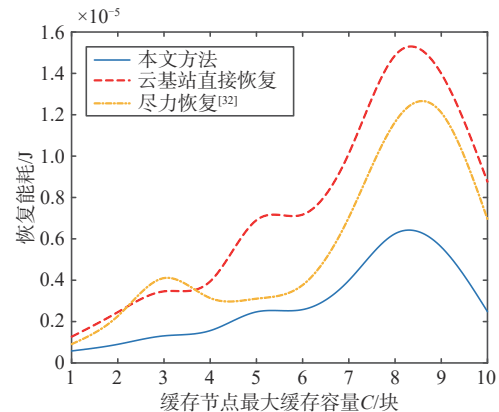


图3 缓存节点最大缓存容量对恢复能耗的影响

于缓存节点最大缓存容量增加，相应丢失的数据也增多，恢复开销增加。

缓存节点间的不可达率对恢复时延的影响如图4所示，缓存节点间的不可达率对恢复能耗的影响如图5所示。当缓存节点间不可通速率较小时，本文为最优恢复方案。根据本文方法，当缓存节点间的不可达率较高时，导致利用后备用户设备协同通信恢复数据的方法受限，要恢复丢失数据需要更多地依赖云基站，所以恢复开销较尽力恢复方法高，且缓存节点间的不可达率较高时，由于本文方案也依赖云基站恢复数据，所以此时时延、能耗曲线与全部由云基站直接恢复曲线走势基本一致，尽力恢复机制下则由于不可通信设备过多导致无法成功恢复数据，本文直接记恢复时延和能耗为0。实验结果产生波动的主要原因是每轮测试时，边缘节点、用户设备这些节点的拓扑结构都发生了改变，因此产生了一定波动。

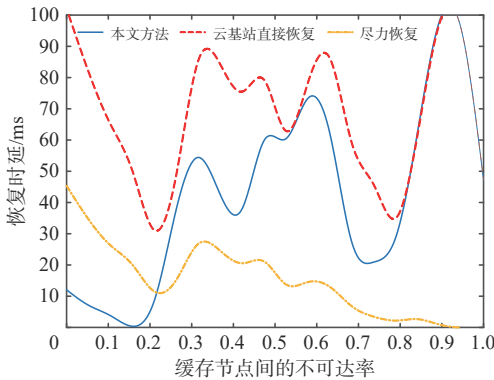


图4 缓存节点间的不可达率对恢复时延的影响

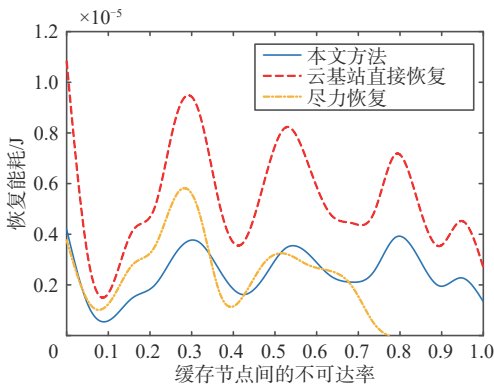


图5 缓存节点间的不可达率对恢复能耗的影响

文件块大小对恢复时延的影响如图6所示，文件块大小对恢复能耗的影响如图7所示。由仿真结果可知，本文所提方法恢复开销最优。当文件块较

大时，本文方法更多通过缓存节点协助恢复数据，恢复开销较小；尽力恢复方法则由于文件块较大，每个缓存节点提供数据时时延较大导致最终总体恢复开销大。

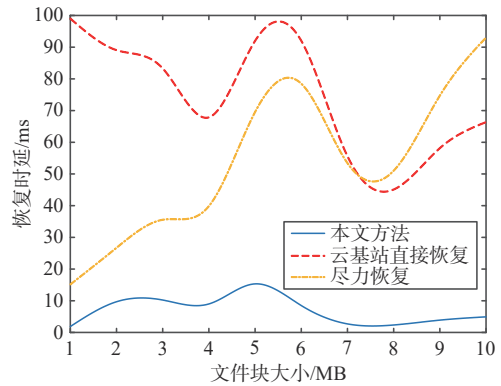


图6 文件块大小对恢复时延的影响

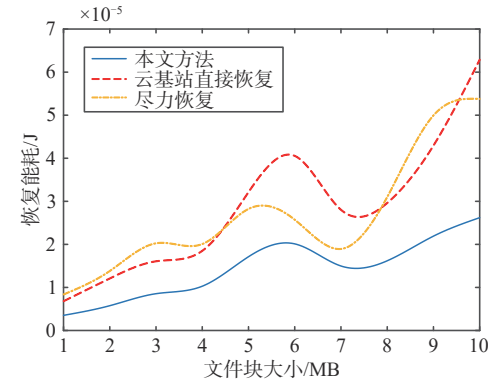


图7 文件块大小对恢复能耗的影响

3种算法下不同文件块大小时缓存节点最大缓存容量对恢复时延的影响如图8所示。通过仿真结果可知，当文件块为5 MB和10 MB时，本文方法恢复时延总体最优。在本文方法中，当文件块过小且缓存节点最大缓存容量也较小时，恢复时延主要

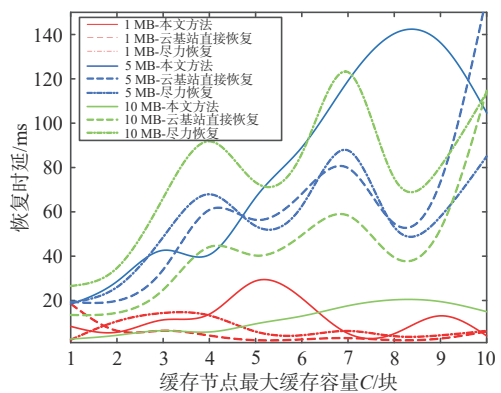


图8 3种算法下不同文件块大小时缓存节点最大缓存容量对恢复时延的影响

受云基站传输时延影响，恢复时延略高，但总体来看本文恢复方法仍然优于传统恢复方法。

5 结束语

基于网络编码的MEC网络，由于缓存节点是普通用户设备，它们的离开、掉电或失效会造成网络健壮性的损失。本文提出了一种失效节点的编码内容恢复方法，首先，通过构建一种基于查找表的编码内容管理方法，从而高效地组织缓存节点的稀疏编码内容；然后，针对节点离开后的恢复问题，建立了最优恢复问题数学模型，并证明了该问题属于NP类问题。考虑到NP类问题求解随用户设备规模的增加而迅速增加，本文给出了一种启发式的缓存节点失效内容恢复算法，先选取最优后备节点，再选取编码内容；最后，仿真结果验证了所提方法在恢复时延和能耗方面的优势。下一步工作拟将该算法在Thyme^[21]等MEC网络应用系统中展开进一步部署测试。

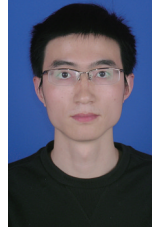
参考文献：

- [1] HASSEBO A. The road to 6G, vision, drivers, trends, and challenges[C]//Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). Piscataway: IEEE Press, 2022: 1112-1116.
- [2] ZENG H R, LI X, BI S Z, et al. Delay-sensitive task offloading with D2D service-sharing in mobile edge computing networks[J]. IEEE Wireless Communications Letters, 2022, 11(3): 607-611.
- [3] 张厚浩, 李晗琳, 高林. 移动边缘计算中的分层资源部署与共享策略[J]. 物联网学报, 2021, 5(1): 11-18.
ZHANG H H, LI H L, GAO L. Hierarchical resource deployment and sharing strategy in mobile edge computing[J]. Chinese Journal on Internet of Things, 2021, 5(1): 11-18.
- [4] YIN J, ZHAN M Q, ZHU M, et al. Salango: a simplified load balancing scheme for edge nodes in mobile edge caching networks[C]//Proceedings of the GLOBECOM 2023 - 2023 IEEE Global Communications Conference. Piscataway: IEEE Press, 2023: 1681-1686.
- [5] YAO J J, HAN T, ANSARI N. On mobile edge caching[J]. IEEE Communications Surveys & Tutorials, 2019, 21(3): 2525-2553.
- [6] ZEWAIL A A, YENER A. Coded caching for combination networks with cache-aided relays[C]//Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT). Piscataway: IEEE Press, 2017: 2433-2437.
- [7] HE W S, SU Y H, XU X T, et al. Cooperative content caching for mobile edge computing with network coding[J]. IEEE Access, 2019, 7: 67695-67707.
- [8] FU Y R, YU Q, WONG A K Y, et al. Exploiting coding and recommendation to improve cache efficiency of reliability-aware wireless edge caching networks[J]. IEEE Transactions on Wireless Communications, 2021, 20(11): 7243-7256.
- [9] DE OLIVEIRA ALVES L H, REBELATTO J L, SOUZA R D, et al. Network-coded cooperative LoRa network with D2D communication[J]. IEEE Internet of Things Journal, 2022, 9(7): 4997-5008.
- [10] ZAKERINASAB M R, WANG M A. Practical network coding for the update problem in cloud storage systems[J]. IEEE Transactions on Network and Service Management, 2017, 14(2): 386-400.
- [11] DAI M J, WANG X, WANG H, et al. Bandwidth overhead-free data reconstruction scheme for distributed storage code with low decoding complexity[J]. IEEE Access, 2017, 5: 6824-6832.
- [12] WANG Y, GU S S, ZHAO L, et al. Repairable fountain coded storage systems for multi-tier mobile edge caching networks[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2310-2322.
- [13] WOOLSEY N, WANG X Y, CHEN R R, et al. FLCD: a flexible low complexity design of coded distributed computing[J]. IEEE Transactions on Cloud Computing, 2023, 11(1): 470-483.
- [14] ZIMMERMANN S, SCHWENCK P, CABRERA J A, et al. Grade to the edge: how many unreliable nodes does it take to break a content delivery network? [C]//Proceedings of the 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). Piscataway: IEEE Press, 2022: 331-336.
- [15] SAGOR M, STOLERU R, HAROON A, et al. R-drive: resilient data storage and sharing for mobile edge clouds[C]//Proceedings of the 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS). Piscataway: IEEE Press, 2022: 171-179.
- [16] LI X L, MOW W H, TSANG F L. Singularity probability analysis for sparse random linear network coding[C]//Proceedings of the 2011 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2011: 1-5.
- [17] YANG S H, YEUNG R W. Batched sparse codes[J]. IEEE Transactions on Information Theory, 2014, 60(9): 5322-5346.
- [18] SEHAT H, PAHLEVANI P. An analytical model for the partial intercept probability in sparse linear network coding[J]. IEEE Communications Letters, 2020, 24(4): 725-728.
- [19] CHEN W L, LU F, DONG Y. Improved expression for rank distribution of sparse random linear network coding[J]. IEEE Communications Letters, 2021, 25(5): 1472-1476.
- [20] YIN J, WEN J X, LING X, et al. Moncode: a multiscale network coding software implementation library[C]//Proceedings of the 2021 13th International Conference on Wireless Communications and Signal Processing (WCSP). Piscataway: IEEE Press, 2021: 1-5.
- [21] SILVA J A, CERQUEIRA F, PAULINO H, et al. It's about Thyme: on the design and implementation of a time-aware reactive storage system for pervasive edge computing environments[J]. Future Generation Computer Systems, 2021, 118: 14-36.

- [22] SILVA J A, VIEIRA P, PAULINO H. Data storage and sharing for mobile devices in multi-region edge networks[C]//Proceedings of the 2020 IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). Piscataway: IEEE Press, 2020: 40-49.
- [23] 蔡艳, 吴凡, 陆音, 等. D2D协作的边缘缓存系统能耗分析[J]. 南京邮电大学学报(自然科学版), 2022, 42(1): 13-22.
CAI Y, WU F, LU Y, et al. Energy consumption analysis on D2D cooperative edge caching system[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2022, 42(1): 13-22.
- [24] 岳雯雁, 赵夙, 朱琦. D2D网络中SVC视频文件的缓存策略[J]. 南京邮电大学学报(自然科学版), 2022, 42(2): 54-65.
YUE W Y, ZHAO S, ZHU Q. SVC video caching strategy in D2D network[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2022, 42(2): 54-65.
- [25] CHEN Q, WANG W, YU F R, et al. Content caching oriented popularity prediction: a weighted clustering approach[J]. IEEE Transactions on Wireless Communications, 2021, 20(1): 623-636.
- [26] YIN J, ZHAN M Q, ZHANG Z W, et al. Research on the content sharing system for mobile edge caching networks: a hierarchical architecture[C]//Proceedings of the 2022 15th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). Piscataway: IEEE Press, 2022: 1-6.
- [27] ORTOLF C, SCHINDELHAUER C, VATER A. Paircoding: improving file sharing using sparse network codes[C]//Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services. Piscataway: IEEE Press, 2009: 49-57.
- [28] CAI Q C, LO K T. Two blocks are enough: on the feasibility of using network coding to ameliorate the content availability of BitTorrent swarms[J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(8): 1682-1694.
- [29] 朱开磊, 孙爱晶. 基于布谷鸟优化K均值的WSN分簇路由算法[J]. 物联网学报, 2022, 6(1): 73-81.
ZHU K L, SUN A J. WSN clustering routing algorithm based on cuckoo search algorithm optimized K-means[J]. Chinese Journal on Internet of Things, 2022, 6(1): 73-81.
- [30] FOWLER R J, PATERSON M S, TANIMOTO S L. Optimal packing and covering in the plane are NP-complete[J]. Information Processing Letters, 1981, 12(3): 133-137.
- [31] SUN S, LIU M, JIAO Z Z, et al. User-centric content sharing via cache-enabled device-to-device communication[J]. Journal of Network and Computer Applications, 2018, 115: 103-115.
- [32] FITZEK F H P, TOTH T, SZABADOS A, et al. Implementation

and performance evaluation of distributed cloud storage solutions using random linear network coding[C]//Proceedings of the 2014 IEEE International Conference on Communications Workshops (ICC). Piscataway: IEEE Press, 2014: 249-254.

[作者简介]



殷俊(1989-), 男, 南京邮电大学物联网学院副教授, 主要研究方向为网络编码理论及其在物联网的应用。



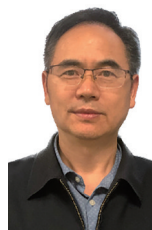
夏欣然(2000-), 女, 南京邮电大学物联网学院硕士生, 主要研究方向为网络编码。



张登银(1964-), 男, 博士, 南京邮电大学物联网学院研究员, 主要研究方向为现代通信网络、信号与信息处理技术等。



柯亚琪(1992-), 女, 南京理工大学计算机科学与工程学院博士生, 主要研究方向为物联网。



杨余旺(1966-), 男, 南京理工大学计算机科学与工程学院教授, 主要研究方向为下一代网络、编码理论与应用、物联网等。