

## 基于网络编码的蜂窝车联网边缘内容缓存方法

夏欣然<sup>1</sup>, 陈烨<sup>1</sup>, 殷俊<sup>1</sup>, 窦如林<sup>2</sup>, 张登银<sup>1</sup>

(1. 南京邮电大学物联网学院, 江苏 南京 210003; 2. 金陵科技学院软件工程学院, 江苏 南京 211169)

**摘要:** 边缘节点内容缓存收益是车联网边缘缓存系统的核心性能指标。基于车载用户的移动特性, 对边缘节点的缓存部署方法进行数学建模, 同时引入网络编码技术对比分析随机缓存方法和基于网络编码的随机缓存方法的性能差异, 提出了一种基于网络编码的车联网边缘内容缓存算法, 给出了算法缓存收益理论界限。实验结果表明, 所提方法优于经典算法, 对瞬时车速等不确定性因素具有更好的鲁棒性, 且路边单元处的缓存内容命中率提升逾 20%。

**关键词:** 蜂窝车联网; 边缘缓存; 移动感知; 网络编码

**中图分类号:** TN929.5

**文献标志码:** A

**doi:** 10.11959/j.issn.2096-3750.2026.00409

## Content caching for roadside units in cellular-vehicle to everything based on network coding

Xia Xinran<sup>1</sup>, Chen Ye<sup>1</sup>, Yin Jun<sup>1</sup>, Dou Rulin<sup>2</sup>, Zhang Dengyin<sup>1</sup>

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

2. School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China

**Abstract:** The cache revenue of edge nodes is a key performance metric for the cellular-vehicle caching system. Based on the mobility characteristics of vehicular users, the caching deployment method at edge nodes was mathematically modeled. Network coding technology was introduced to comparatively analyze the performance differences between random caching and network coding-based random caching methods. A network coding-based edge content caching algorithm for vehicular networks was proposed, and a theoretical limits of the algorithm's caching benefits were analyzed. Experimental results showed that the proposed method outperforms the classic algorithms, exhibits greater robustness against uncertain factors such as instantaneous vehicle speed, and increases a cache hit rate improvement of over 20% at roadside units.

**Key words:** cellular-vehicle to everything, edge caching, mobility aware, network coding

### 0 引言

随着移动通信和物联网 (IoT, Internet of things) 技术的持续发展, 蜂窝车联网 (C-V2X, cellular-vehicle to everything) 凭借其低时延、高带宽和高

可靠性等特点<sup>[1]</sup>, 已成为未来车辆自动驾驶、智慧交通系统等车路协同应用的关键支撑技术<sup>[2-3]</sup>。

C-V2X架构包括云平台、边缘服务器 (ES, edge server)、边缘节点 (EN, edge node)、车载单元 (OU, onboard unit)、用户智能设备等关键组件, 可

收稿日期: 2024-03-26; 修回日期: 2025-07-11

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); 国家自然科学基金项目 (No. 62471241); 南京邮电大学自然科学基金项目 (No. NY221095, No. NY223106); 江苏省研究生实践创新计划 (No. SJCX24\_0335, No. SJCX25\_0365)

**Foundation Items:** The National Defense Basic Research Foundation of China (No. JCKY2023\*\*\*007), The National Natural Science Foundation of China (No. 62471241), NJUPT Science Foundation (No. NY221095, No. NY223106), Jiangsu Province Graduate Practical Innovation Program (No. SJCX24\_0335, No. SJCX25\_0365)

实现乘客的娱乐、工作、社交等活动在移动出行中的延伸。其中，边缘节点既可以是部署在道路侧或者交叉路口的专用路边单元 (RSU, road side unit)，亦可由小基站、Wi-Fi 接入点等通用网络设备承担<sup>[4]</sup>。为满足日益增长的乘客娱乐体验需求，内容提供商逐渐将内容从云端转移到边缘节点，使服务尽可能地贴近车辆用户，从而减少内容访问时延，降低核心网络负载<sup>[4-7]</sup>，并借助边缘计算等方法进一步提高网络效率和系统容错能力<sup>[8-9]</sup>。然而，这种转变也带来了一个关键问题，即与云中心存储方式相比，边缘节点的存储空间有限。因此，需要制定高效的内容缓存策略，以最大化边缘节点内容缓存收益<sup>[10-13]</sup>。

由于车联网边缘节点的缓存策略通常针对的是流媒体等具有时延约束的应用场合，需要将内容按顺序缓存到若干边缘节点上，并按车辆移动顺序进行传递，传统内容缓存方法（如 LRU<sup>[14]</sup>、LFU<sup>[15]</sup>等）往往难以直接适用<sup>[4]</sup>。鉴于此，现有研究主要分成两类：一类是基于车辆移动轨迹预测的缓存策略<sup>[10,16-17]</sup>，通过预测车辆未来的位置，提前在边缘节点缓存可能被这些车辆请求的内容。另一类是根据内容的受欢迎度<sup>[18-19]</sup>或者目标用户群体的兴趣偏好<sup>[20-22]</sup>来决策边缘节点的缓存内容。然而，这两类方法均需要对每辆车的行驶轨迹或车内用户的兴趣、社交行为进行统计，不可避免地引发了对用户隐私侵犯的担忧。此外，它们对车辆移动过程的顺序缓存需求考虑不足。

区别于上述两类方法，文献[4]提出了一种车辆移动感知概率 (MAP, mobility-aware probabilistic) 缓存策略。MAP 缓存策略以最大化缓存命中率为目标，不再追踪具体车辆的行驶轨迹，转而统计分析每个边缘节点的车辆停留时间分布，据此形成在这些边缘节点上目标内容的部署策略。尽管 MAP 缓存策略实现了对车辆用户目标内容的预测性缓存，但受到 RSU 缓存空间大小的限制，无法缓存所有目标文件，仍然有很大一部分内容需要从回程链路获取。

本文提出一种基于 MAP 策略的改进方法，通过引入网络编码技术，可更高效地利用 RSU 的缓存内容。本文的主要贡献总结如下。

(1) 建立了可感知车辆移动的 C-V2X 边缘节点内容缓存问题分析模型，能够分析特定边缘节点

所请求内容的发布概率。

(2) 提出了一种基于网络编码的边缘节点内容缓存算法。该算法可感知车辆移动，将内容分成固定大小的块，并进行网络编码后部署在用户将途经的边缘节点上，利用编码后信息的混合优势，获得更高缓存收益。

(3) 设计了基于蒙特卡洛模拟的仿真实验系统，且实验结果表明，相较于多种经典缓存策略，本文所提算法具有更高的命中概率和更低的回程流量开销。

## 1 系统模型

基于城市环境的 C-V2X 内容边缘缓存系统架构如图 1 所示。边缘节点由部署在路灯或道路交叉路口的各个 RSU 构成。每个 RSU 均配备有存储器，用于缓存目标内容，并向路过的车辆用户提供服务。假设每个待缓存内容文件由  $K$  个数据块组成，各数据块由其序列号  $k$  进行唯一标识，其中  $1 \leq k \leq K$ 。

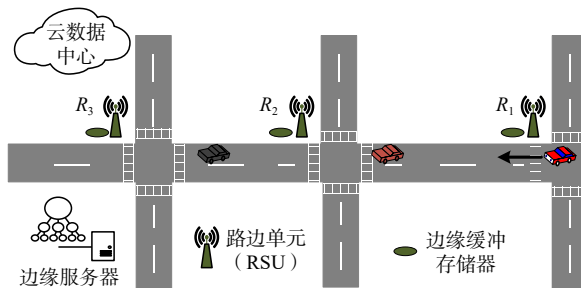


图1 C-V2X 内容边缘缓存系统架构

在图 1 所示的场景中，道路沿线的若干 RSU 由一台 C-V2X ES 统一管控。由于该 ES 与基于车联网的交通流量感知系统相连接，因此可以假设 ES 能够知悉或预测车辆在接下来的几分钟内将要经过的 RSU 序列，并了解每个边缘节点处车辆停留时间的分布情况（如通过统计历史交通信息来获取）。同时，ES 还知悉 RSU 缓存中的可用空间与经过用户发出的内容请求。基于这些信息，ES 可以确定每个边缘节点应该缓存哪些内容的哪些数据块，并相应地从云中心下载对应数据块并部署至指定的边缘节点上。

当车辆用户进入某一个 RSU 的覆盖范围时，其应用程序会发起一批新的内容块请求，指明它正在接收的内容项及期望获取的数据块号  $k$ 。RSU 将检查本地是否缓存了该块号  $k$ 。若未缓存，则需要通过回程链路从云数据中心下载，这一操作显然伴

随着较高的带宽消耗和传输时延。在数据块的传输过程中，RSU会依次发送下一个请求块，如果本地缓存中没有存储该块，则再次借助回程链路获取。

因此，C-V2X内容边缘缓存系统的目标是建立一种高效的缓存策略，将内容块合理地存储在RSU中，以确保车辆能够以高吞吐量、低时延的方式获取内容，并且使请求的数据块有很高的概率能直接从RSU本地缓存中获取，而不是从回程链路获取。

## 2 基于网络编码的车联网边缘节点内容缓存算法

类似P2P (peer-to-peer) 内容分发网络，C-V2X边缘节点内容缓存同样以内容块/数据块为单位进行<sup>[4,23-25]</sup>。在ES的管理下，待缓存文件将首先被分割成固定大小的数据块。若文件大小不能被等分，则最后一个块通过补零操作对齐至规定尺寸。

假设某个源文件 $Y$ 经过分割操作后包含 $n$ 个内容块/数据块，即 $Y = \{y_1, y_2, \dots, y_n\}$ 。不同于MAP等非编码缓存算法，在基于网络编码<sup>[26-30]</sup>的边缘节点缓存策略中，各RSU用户设备中存储的是编码块，而不是这些源数据块。编码过程可记为 $E = C \cdot Y$ ，其中 $C$ 表示编码矩阵， $E = (e_1, \dots, e_m)^T$ 则是编码运算后产生的 $m (m \geq n)$ 个编码块集合，集合中的任意一个编码块 $e_i$ 是由 $n$ 个源数据块在一个选定的有限域（如 $GF(2^8)$ ）上通过线性组合生成，可表示为：

$$e_i = c_i \cdot Y = (c_{i,1}, \dots, c_{i,n}) \cdot (y_1, \dots, y_n)^T, i \in [1, m] \quad (1)$$

其中， $c_i$ 表示编码块 $e_i$ 的编码向量，其元素是从运算有限域中随机生成的。

显然，从编码块解码出源内容块就是求解线性方程组 $E = C \cdot Y$ 的过程。根据线性代数理论，当用户车辆获取到 $n$ 个线性无关的编码块（对应 $n$ 个线性无关的编码向量）时，即可以通过Gauss或Gauss-Jordan等解码算法求解并恢复源文件内容<sup>[31]</sup>。

为说明基于网络编码的概率缓存算法优势，以图1所示架构为基础，对比某车辆分别采用经典非网络编码方法和基于网络编码方法时，车联网RSU的内容缓存过程。

假设该车辆在接下来的几分钟内将依次经过3个RSU节点 $\{R_1, R_2, R_3\}$ ，这些RSU节点为待缓存内容的边缘节点。设待缓存文件 $f$ 包含3个内容块： $f = \{f_1, f_2, f_3\}$ ，而且每个RSU可以为这个车辆用户提供1个内容块大小的缓存空间。

在MAP的内容缓存算法<sup>[4]</sup>中，根据RSU的次序，将源文件的3个数据块依次部署在对应RSU上。基于MAP的RSU内容缓存算法的部署方案如图2所示。

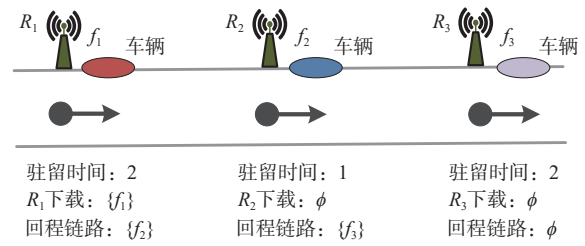


图2 基于MAP的RSU内容缓存算法的部署方案

由于车辆在每个RSU覆盖范围的逗留时间受当前车流、路况及驾驶习惯等因素影响，因此其在各RSU覆盖区域内能够下载内容的数量有所不同。假设车辆在 $R_1$ 、 $R_2$ 、 $R_3$ 的覆盖范围分别驻留2、1、2个时间单位，且每个时间单位可下载一个内容块。

当车辆处于 $R_1$ 覆盖范围时，首先可以从 $R_1$ 的存储空间处下载到 $f_1$ 内容块。由于车辆在 $R_1$ 处会驻留2个时间单位，第二个时间单位车辆用户将从回程链路下载 $f_2$ 。当车辆用户到达 $R_2$ 处时，需求的内容块为 $f_3$ ，但 $R_2$ 缓存的是 $f_2$ ，因此依然需要从回程链路下载 $f_3$ 。因此，在MAP算法下，车辆用户仅能够从RSU直接获得1个内容块，缓存命中率为1/3。

在基于网络编码的RSU内容缓存算法中，首先需要对源文件的内容块进行编码处理。经编码处理后得到的编码块集合为 $E = \{e_1, e_2, e_3, \dots, e_m\}$ ，且 $E = A \cdot f$ ，其中 $A$ 为编码系数矩阵。由于编码运算工作在有限域上进行，因此编码后的内容块大小和源文件内容块是相同的。编码完成后，将各个编码块分别部署在对应的RSU处。不同于MAP算法，经过网络编码后的编码块是等价的，因此可以任意部署在不同的RSU处，其部署方案如图3所示。

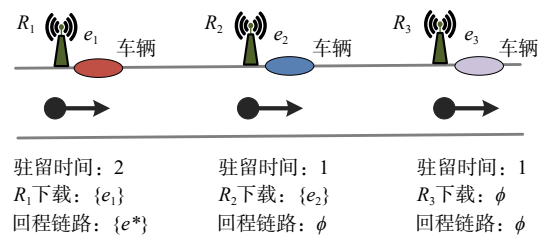


图3 基于网络编码的RSU内容缓存算法的部署方案

假设车辆在 $R_1$ 、 $R_2$ 、 $R_3$ 覆盖范围内的停留时间依然是2、1、2个时间单位。当车辆处于 $R_1$ 覆盖范

围时，可以首先从 $R_1$ 的存储空间处下载编码块 $e_1$ 。由于车辆在 $R_1$ 处会驻留2个时间单位，因此第二个时间单位车辆用户将从回程链路下载一个编码块 $e_2$ 。 $e_2$ 可以是部署过程剩余的编码块（如 $e_4$ ），也可以是ES根据源数据块动态编码生成的块。不失一般性，假设 $e_2=e_4$ 。当车辆用户进入 $R_2$ 范围，下载 $R_2$ 中缓存的编码块 $e_2$ 。至此，用户车辆已经获得3个编码块，满足解码条件，可通过线性解码恢复原始文件。因此，采用基于网络编码的RSU内容缓存算法，车辆用户能够从RSU直接获得2个内容块，缓存命中率为2/3，比MAP算法高1倍。

需要指出的是，上文中RSU协作缓存内容的示例是假定ES不能支配RSU分配给该用户缓存内容的空间大小，即RSU空间受限模式。文献[22]还提出了另一种应用场景——允许ES动态调配各RSU用于缓存该内容的空间大小，即RSU空间自由支配模式。由于这两种模式均以最大化边缘节点缓存命中率为目标，在下文缓存收益的分析中，将不再对其展开具体区分。

### 3 算法分析

记 $R$ 为将服务于目标车辆的RSU集合，并假设集合元素已按照车辆用户的访问次序进行了排序，即 $R=\{R_1, R_2, R_3, \dots, R_m\}$ 。记待缓存的源文件包含内容块集合 $D=\{d_1, d_2, d_3, \dots, d_m\}$ 。令 $X_i$ 表示该车辆用户在 $R_i$ 覆盖区内逗留时所接收到的最后一个内容块编号的随机变量，并定义 $X_0=0$ ，则车辆用户在 $R_i$ 覆盖区逗留时下载的内容块集合可表示为 $(X_{i-1}, X_i]$ 。因此，对于第 $k$ 个内容块，其能够在车辆用户处于 $R_i$ 覆盖区内时被下载的概率为：

$$p_i(k) = P(k \in (X_{i-1}, X_i]) = P(X_{i-1} < k \leq X_i) \quad (2)$$

若记随机变量 $S_i$ 为该车辆用户在 $R_i$ 覆盖区逗留时下载的总内容块数，则 $S_i$ 的概率密度函数主要取决于以下两个因素：（1）车辆在 $R_i$ 覆盖范围内的逗留时长；（2）车辆在 $R_i$ 覆盖范围内获得的实际数据传输吞吐量，而吞吐量又取决于RSU缓存的内容是否命中和无线数据传输速率。根据 $S_i$ 和 $X_i$ 的定义， $X_i$ 可以表示为：

$$X_i = X_{i-1} + S_i = X_{i-2} + S_{i-1} + S_i = \dots = \sum_{j=1}^i S_j \quad (3)$$

其中， $j$ 是求和下标，从1到 $i$ ，表示RSU的序号。

**定理 1:** 一个车辆用户将要经过的RSU集合

$R=\{R_1, R_2, R_3, \dots, R_m\}$ ，处于 $R_i$ 的覆盖范围内时，则该车辆用户下载第 $k$ 个内容块的概率 $p_i(k)$ 为：

$$p_i(k) = \sum_{j=0}^{k-1} P(S_i \geq k-j | X_{i-1} = j) P(X_{i-1} = j) \quad (4)$$

$$p_i(k) = \sum_{j=0}^{k-1} P(X_{i-1} = j | S_i \geq k-j) P(S_i \geq k-j) \quad (5)$$

同时，在 $R_i$ 的覆盖范围内，该车辆用户平均下载的文件块数为：

$$E(S_i) = \sum_{k=1}^K p_i(k) \quad (6)$$

**证明:** 由全概率公式知 $p_i(k) = \sum_{j=0}^{k-1} P(S_i \geq k-j, X_{i-1} = j)$ ，因此根据贝叶斯公式可分别推导出式（4）和式（5）；由离散型随机变量的期望定义亦可得到式（6）。定理得证。

需要指出的是，定理1成立的前提条件并不要求假设车辆用户处于不同RSU覆盖范围时下载的内容块数相互关联。如果进一步放松条件，假设车辆用户逗留在不同RSU覆盖范围的时间是独立的，即车辆用户在不同RSU的下载内容块数是独立的，则 $p_i(k) = \sum_{j=0}^{k-1} P(S_i \geq k-j) P(X_{i-1} = j)$ ；若进一步假设 $S_i$ 服从独立同分布，则可以得到文献[4]的推论1。

**推论 1<sup>[4]</sup>:** 给定任一车辆用户将要经过的RSU集合 $R=\{R_1, R_2, R_3, \dots, R_m\}$ ，若该车辆用户处于不同 $R_i$ 覆盖范围时所下载内容块数 $S_i$ 独立且同分布，则该车辆用户将要在其处于 $R_i$ 的覆盖范围时下载第 $k$ 个内容块的概率 $p_i(k)$ 为：

$$p_i(k) = f_s * p_{i-1}(k) \quad (7)$$

其中， $f_s$ 是 $S$ 的概率密度函数，\*表示卷积运算。

**证明:** 具体过程参见文献[4]。

对于具有 $m$ 个RSU的某一蜂窝车联网区域，假设某车辆用户欲访问的兴趣文件包含 $n$ 个内容块（即 $|R|=m, |D|=|E|=n$ 且 $m \leq n$ ），记一个内容缓存算法将该兴趣文件部署在各RSU后的部署结果为 $I=\{\{R_1, D_1\}, \{R_2, D_2\}, \dots, \{R_m, D_m\}\}$ ，其中 $D_i$ 表示缓存在 $R_i$ 的内容块集合，且对于 $\forall i, j$ 有 $| \cup D_i|=m, D_i \cap D_j = \emptyset$ 。由于基于网络编码方法的RSU上部署的是编码块而不是原始内容块，为示区分，记基于网络编码方法的内容部署结果为 $I_{NC}=\{\{R_1, E_1\}, \{R_2, E_2\}, \dots, \{R_m, E_m\}\}$ ，其中 $E_i$ 表示缓存在 $R_i$ 的编码块集合，且对于 $\forall i, j$ 有 $| \cup E_i|=m, E_i \cap E_j = \emptyset$ 。

基于上述分析，本文给出基于网络编码的边缘

内容缓存算法的简要伪代码，如算法1所示。该算法分为三个阶段：第一阶段为编码部署阶段，首先将源文件分块并进行网络编码，以生成编码块集合，然后根据定理1估算各RSU处的平均下载块数，并依次为每个RSU分配相应数量的编码块进行存储；第二阶段为请求响应阶段，车辆用户依次从途经的RSU下载已缓存的编码块；第三阶段为解码恢复阶段，车辆用户收集到足够数量的线性无关编码块后，通过高斯消元等方法解码恢复原始文件。

**算法1** 基于网络编码的车联网边缘内容缓存算法

**输入：**待缓存内容文件  $Y$  和 RSU 集合  $R = \{R_1, R_2, R_3, \dots, R_m\}$

**输出：**部署结果  $I_{NC} = \{\{R_1, E_1\}, \{R_2, E_2\}, \dots, \{R_m, E_m\}\}$

**阶段1：**基于网络编码的RSU内容部署。

对源文件  $Y = \{y_1, y_2, \dots, y_n\}$  进行分块

生成编码向量矩阵  $C$

生成编码块集合  $E$ ，即  $E = C \cdot Y$

**For**  $i=1$  to  $m$  do

据定理1估算车辆用户处于  $R_i$  时的平均下载块数  $E(S_i)$

从  $E$  中选择  $E(S_i)$  个编码块（记为  $E_i$ ）并存储至  $R_i$

$E(S_i) = E - E_i$

**End for**

**阶段2：**RSU响应车辆用户内容请求。

**For**  $i=1$  to  $m$  do

**If** ( $X_i < n$ )

车辆用户从  $R_i$  处下载缓存内容

**Else**

break;

**End if**

**End for**

**阶段3：**车辆用户根据 Gauss 消元等算法，从下载的编码块中解码出源文件  $Y$  的内容。

## 4 仿真实验

### 4.1 实验设置

为验证所提出的基于网络编码的车联网边缘内容缓存算法的性能，采用蒙特卡洛方法模拟测试车辆用户通过3个RSU处获取内容服务的场景。与文

献[10-11]一致，待测试的兴趣文件库包含  $r=10^3$  个视频文件，且文件库中文件的流行度服从参数为  $\gamma=0.8$  的 Zipf 分布，即文件库中流行度排名为  $i$  的文件被用户请求的概率  $f_{r_i}$  可以表示为：

$$f_{r_i} = \frac{1}{i^\gamma \sum_{j=1}^r \frac{1}{j^\gamma}} \quad (8)$$

作为实验对照，选用 MAP 随机缓存算法<sup>[4]</sup>和 POP (popularity-based caching) 算法<sup>[4]</sup>作为对比方法。POP 算法是一种经典的内容缓存算法，其根据数据块的流行度由高至低按顺序缓存至 RSU。具体仿真参数设置见表1。

表1 仿真参数位置

参数	取值
车辆(用户)数/个	$10^4$
视频内容库文件数 $r$ /个	$10^3$
视频内容库文件流行度参数 $\gamma$ /个	{0.8, 1.0, 1.2}
兴趣内容数据块数 $n$ /个	{30, 60, 90}
数据块大小/kB	{1, 2, 4}
RSU处下载数据块数的泊松分布	{10, 20, 30}
网络编码有限域	GF(2 <sup>8</sup> )
网络编码运算方法	二维查表法
MAP可重叠缓存策略阈值 $\tau$	0.7

在实验指标选取方面，缓存命中率用于衡量目标车辆用户从RSU缓存空间直接获取的内容块数占总文件内容块数的比例，是评估边缘节点缓存收益的关键指标。因此，本实验重点比较不同算法的缓存命中率。此外，边缘服务器将分别采用以下两种部署策略作为各种算法的执行依据。

(1) 策略1：RSU缓存空间受限策略。不同RSU缓存的内容各异，且每个RSU缓存空间可至多缓存用户  $i$  的兴趣文件  $\lambda_i$  个数据块。

(2) 策略2：RSU缓存空间自由策略。不同RSU缓存的内容可部分重叠，但缓存策略需保证兴趣文件的每个数据块具有最小访问概率（即阈值  $\tau$ ）。

### 4.2 实验结果

若某个车辆用户感兴趣的内容包含  $n=30$  个数据块，且该车辆用户经过每个RSU处下载的内容块数量独立且同分布。例如，服从均值  $\lambda$  的泊松分布，即

$$P(S_i = k) = \frac{e^{-\lambda} \cdot \lambda^k}{k!} \quad (9)$$

3个RSU对每个数据块  $k$  的下载概率  $\varphi(k)$  如图4

所示。从图4的灰色线段可以观察到，在RSU  $R_1$ 处，随着数据块序号的增加， $\varphi(k)$ 迅速下降。这是因为随着车辆用户的移动，用户可能从其他RSU下载数据块的概率迅速增加，尤其是对于序号较大的数据块。由于设定的下载块数服从均值 $\lambda=10$ 的泊松分布，当 $k>20$ 时， $\varphi(k)$ 将趋于0。对于 $R_2$ ， $\varphi(k)$ 呈帽形分布。初始时 $\varphi(k)$ 取值极小，是因为若要从 $R_2$ 下载这些块，车辆用户通过 $R_1$ 覆盖范围内的速度可能非常快，导致车辆用户来不及下载任何数据块，但这一事件发生的概率较低。在 $R_2$ 覆盖范围， $\varphi(k)$ 的最大值在第15块附近，这是因为通过 $R_1$ 时平均下载10个块，因此在 $R_2$ 覆盖范围将访问区间 $[10, 20]$ ，且要下载的数据块恰好在这个区间中间（即15）。在 $R_3$ 处的数据块下载概率仍然呈帽形分布，但具有更宽的帽檐，这是因为随着车辆用户经过 $R_1$ 和 $R_2$ 后，已下载数据块的不确定性会累积增大。

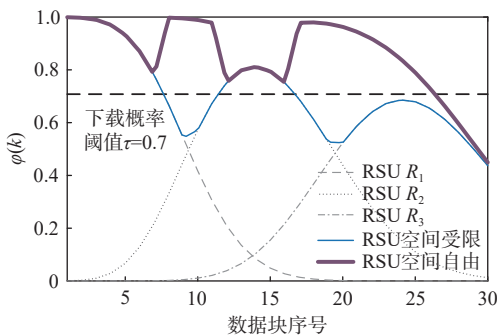


图4 3个RSU对每个数据块k的下载概率

与图4设定相同，即车辆用户感兴趣的内容包含 $n=30$ 个数据块，且该车辆用户经过每个RSU处下载的内容块数量独立且同分布，服从均值 $\lambda = 10$ 的泊松分布。3种算法的缓存命中率对比如图5所示。其中，网络编码的有限域运算基于Moncode库<sup>[32]</sup>提供的二维查表法实现。为便于与MAP方法和POP方法在同一尺度上进行比较，网络编码方法在每个RSU处缓存内容的编码块所占空间大小均

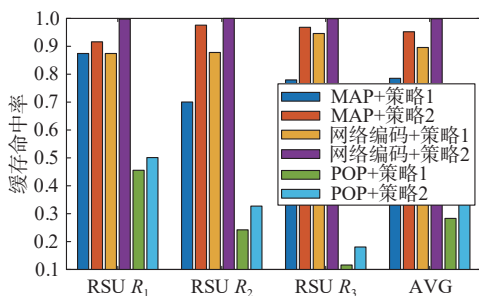


图5 3种算法的缓存命中率对比

与对比方法保持一致。

图5表明，2种经典算法的平均缓存命中率均低于网络编码算法。具体而言，在3个RSU处，MAP算法的平均缓存命中率分别为0.875、0.701和0.780；POP算法的平均缓存命中率分别为0.465、0.251和0.124；网络编码算法的平均缓存命中率分别为0.874、0.878和0.946，显著高于MAP算法。此外，随着RSU序号的增加，2种对比算法的缓存命中率逐渐降低。这是因为MAP算法缓存的是原始数据块，随着车辆移动，后续RSU缓存的数据块可能已被车辆通过回程链路在之前RSU处下载，从而失去效用；POP算法由于根据数据块内容流行度进行缓存，随着车辆的移动，后续RSU缓存的数据块流行度逐渐降低，被车辆用户请求的概率下降，故命中率降低。相比之下，网络编码算法却能够随着RSU序号的增加而维持较高的缓存命中率，这是因为编码块融合了原数据块的信息，后续RSU处缓存的编码块依然有同等价值。此外，实验结果还表明，在缓存受限策略下，3种算法的性能均明显差于缓存自由策略，这说明分配给各RSU的缓存空间大小也是影响算法性能的重要因素。

RSU处平均下载数据块数对算法性能的影响如图6所示。从图6可以看出，随着车辆平均下载块数发生变化，尤其是到达临界值即 $\lambda = 10$ 时，不同算法的缓存命中率表现也呈现出不同的趋势：MAP算法的缓存命中率持续增加；POP算法的缓存命中率总体较低且逐渐下滑；网络编码算法也出现了一定程度的下滑，但仍表现出了较好的稳定性。由于车辆用户从RSU处下载的块数既受到用户设备和RSU间的通信速率影响，还受到车辆通过各RSU覆盖范围的速度影响，因此这个实验事实上反映了网络编码算法性能在车辆移动速度、下载速率波动等环境下仍具有较好的鲁棒性。

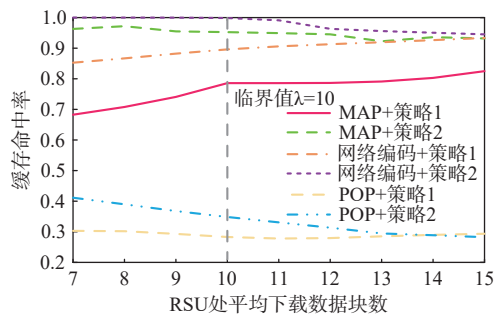


图6 RSU处平均下载数据块数对算法性能的影响

数据块访问率阈值对算法性能的影响如图7所示。随着策略2的阈值从0.55增加到0.95，MAP算法的缓存命中率迅速增加，但一直低于网络编码算法的命中率。网络编码算法的最大波动仅为9.5%，远小于MAP算法的20.6%，说明本文所提算法对最低访问阈值的设置不敏感，具有更强的适应性。

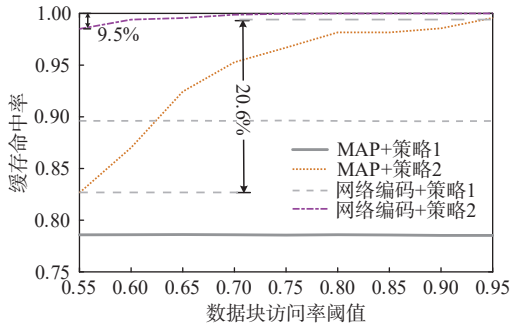


图7 数据块访问率阈值对算法性能的影响

## 5 结束语

针对蜂窝车联网边缘节点的内容缓存优化问题，本文提出了一种基于网络编码的内容缓存算法。该算法通过感知车辆的移动轨迹，对拟缓存的内容块进行网络编码，并按需实施存储，从而提高缓存命中率。实验结果表明，该算法对网络下载率、车辆移动速率等外界因素的变化具有较好的鲁棒性。

鉴于本文所提缓存算法是一个通用方案，其设计不依赖特定的内容、用户行为或兴趣分布，未来工作可以从以下两个方向进一步探索：其一，面向流式内容获取场景的编码内容部署策略。当前方法主要是将兴趣内容文件分块后进行部署，而随着视频流等流传输应用的逐渐普及，如何在该环境下进行更高效的编码内容部署与调度，是一个可拓展的研究方向。其二，用户设备的普遍异构性。实际场景中，用户设备在屏幕尺寸、计算能力等方面存在显著差异，如何在兼顾用户设备多样性的同时，提供差异化服务，是另一个值得进一步探索的方向。

## 参考文献：

- [1] Chen C, Yao G R, Liu L, et al. A cooperative vehicle-infrastructure system for road hazards detection with edge intelligence[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(5): 5186-5198.
- [2] Gupta S K, Khan J Y, Ngo D T. A 5G-based vehicular network architecture to enhance road safety applications[C]//*Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. Piscataway: IEEE Press, 2021: 1-7.
- [3] 郭延永, 刘佩, 袁泉, 等. 网联自动驾驶车辆道路交通安全研究综述[J]. *交通运输工程学报*, 2023, 23(5): 19-38. Guo Y Y, Liu P, Yuan Q, et al. Review on research of road traffic safety of connected and automated vehicles[J]. *Journal of Traffic and Transportation Engineering*, 2023, 23(5): 19-38.
- [4] Mahmood A, Casetti C, Chiasserini C F, et al. Mobility-aware edge caching for connected cars[C]//*Proceedings of the 2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. Piscataway: IEEE Press, 2016: 1-8.
- [5] Hou L, Lei L, Zheng K, et al. A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4512-4520.
- [6] Wang C W, Wang Z Y, Xu L X, et al. Collaborative caching in vehicular edge network assisted by cell-free massive MIMO[J]. *Chinese Journal of Electronics*, 2023, 32(6): 1218-1229.
- [7] Qiao G H, Leng S P, Maharjan S, et al. Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks[J]. *IEEE Internet of Things Journal*, 2020, 7(1): 247-257.
- [8] 朱思峰, 王钰, 陈昊, 等. 车联网边缘计算场景下基于改进型NSGA-II算法的边缘服务器部署决策[J]. *物联网学报*, 2024, 8(1): 84-97. Zhu S F, Wang Y, Chen H, et al. Edge server deployment decision based on improved NSGA-II in the Internet of vehicles edge computing scenario[J]. *Chinese Journal on Internet of Things*, 2024, 8(1): 84-97.
- [9] Feng L, Li W J, Lin Y X, et al. Joint computation offloading and URLLC resource allocation for collaborative MEC assisted cellular-V2X networks[J]. *IEEE Access*, 2020, 8: 24914-24926.
- [10] Zhao L, Li H X, Lin N, et al. Intelligent content caching strategy in autonomous driving toward 6G[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(7): 9786-9796.
- [11] Huang X N, Zhao G D, Chen Z. Segment-based random caching in device-to-device(D2D) caching networks[C]//*Proceedings of the 2015 International Symposium on Wireless Communication Systems (ISWCS)*. Piscataway: IEEE Press, 2015: 731-735.
- [12] 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.
- [13] Yao J J, Han T, Ansari N. On mobile edge caching[J]. *IEEE Communications Surveys & Tutorials*, 2019, 21(3): 2525-2553.
- [14] Aggarwal C, Wolf J L, Yu P S. Caching on the world wide web[J]. *IEEE Transactions on Knowledge and Data Engineering*, 1999, 11(1): 94-107.
- [15] Hasslinger G, Ntougias K, Hasslinger F, et al. Comparing web cache implementations for fast O(1) updates based on LRU, LFU and score gated strategies[C]//*Proceedings of the 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design*

- of Communication Links and Networks (CAMAD). Piscataway: IEEE Press, 2018: 1-7.
- [16] Serbetci B, Goseling J. On optimal geographical caching in heterogeneous cellular networks[C]//Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway: IEEE Press, 2017: 1-6.
- [17] Yang C, Liu Y, Chen X, et al. Efficient mobility-aware task offloading for vehicular edge computing networks[J]. IEEE Access, 2019, 7: 26652-26664.
- [18] Chen Q, Wang W, Zhang Z Y. Clustered popularity prediction for content caching[C]//Proceedings of the ICC 2019 – 2019 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2019: 1-6.
- [19] 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.
- [20] Sun Z F, Nakhai M R. Distributed learning-based cache replacement in collaborative edge networks[J]. IEEE Communications Letters, 2021, 25(8): 2669-2672.
- [21] Wu D P, Li J F, He P, et al. Social-aware graph-based collaborative caching in edge-user networks[J]. IEEE Transactions on Vehicular Technology, 2023, 72(6): 7926-7941.
- [22] Chiang Y, Hsu C H, Wei H Y. Collaborative social-aware and QoE-driven video caching and adaptation in edge network[J]. IEEE Transactions on Multimedia, 2021, 23: 4311-4325.
- [23] Lin J, Huang S Q, Zhang H L, et al. A novel coalition game based distributed cooperative content caching in mobile edge networks[J]. Vehicular Communications, 2023, 44: 100689.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] Zakerinasab M R, Wang M A. Practical network coding for the

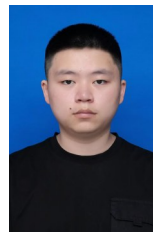
update problem in cloud storage systems[J]. IEEE Transactions on Network and Service Management, 2017, 14(2): 386-400.

- [31] 殷俊, 沙雪琪, 王磊, 等. 基于网络编码的协作恢复机制线性可解性研究[J]. 通信学报, 2021, 42(5): 216-229.
- Yin J, Sha X Q, Wang L, et al. Research on linear solvability of network coding based cooperative recovery scheme[J]. Journal on Communications, 2021, 42(5): 216-229.
- [32] 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.

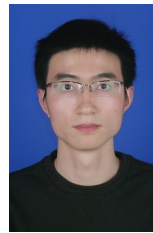
### [作者简介]



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应用技术。



陈焜(2001–), 男, 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应用技术。



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应用技术。



窦如林(1979–), 男, 金陵科技学院软件工程学院正高级实验师, 主要研究方向为计算机网络、大数据。



张登银(1964–), 男, 博士, 南京邮电大学研究员, 主要研究方向为现代通信网络、信号与信息处理技术。