

车联网中基于 MEC 的 V2X 协同缓存和资源分配

李方伟^{1,2,3}, 张海波^{1,2,3}, 王子心^{1,2,3}

(1. 重庆邮电大学通信与信息工程学院, 重庆 400065; 2. 移动通信教育部工程研究中心, 重庆 400065;
3. 移动通信技术重庆市重点实验室, 重庆 400065)

摘 要: 针对车联网的多媒体业务快速增长, 大量的数据交换为移动网络带来沉重负担的问题, 构建了一种车联网中基于移动边缘计算的 V2X 协同缓存和资源分配框架。提出了 V2X 协同缓存与资源分配机制, 实现网络内计算、缓存和通信资源的有效分配; 利用图着色模型为卸载用户分配信道; 采用拉格朗日乘子法对功率与计算资源进行分配。仿真结果表明, 在不同的系统参数下, 所提机制可以有效降低系统开销, 减少任务完成时延。

关键词: 车联网; V2X; 边缘缓存; 移动边缘计算; 资源分配

中图分类号: TN929.5

文献标识码: A

DOI: 10.11959/j.issn.1000-436x.2021007

V2X collaborative caching and resource allocation in MEC-based IoV

LI Fangwei^{1,2,3}, ZHANG Haibo^{1,2,3}, WANG Zixin^{1,2,3}

1. School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
2. Engineering Research Center of Mobile Communications of the Ministry of Education, Chongqing 400065, China
3. Chongqing Key Laboratory of Mobile Communications Technology, Chongqing 400065, China

Abstract: Aiming at the problem that with the rapidly increasing of multimedia services in IoV, a large amount of data change has brought a heavy burden on the mobile networks, a V2X collaborative caching and resource allocation framework in MEC-based IoV was constructed. A V2X cooperative caching and resource allocation mechanism was proposed to achieve the effective allocation of computing resources, caching resources, and communication resources in the network. The graph coloring model was used to allocate channels to the offloading users. Lagrange multiplier method was used to allocate power and computing resources. The simulation results show that the proposed mechanism can effectively reduce system overhead and reduce task completion delay under different system parameters.

Keywords: IoV, V2X, edge caching, mobile edge computing, resource allocation

1 引言

近年来, 随着无线通信和物联网技术的发展, 车联网 (IoV, Internet of vehicles) 已成为 5G 的重要应用场景。通过 LTE-V 或 IEEE 802.11p 技术, 车辆

可以与基础设施、行人和其他车辆进行通信。智能车辆配备的车载单元 (OBU, on board unite) 使车辆具有计算能力和存储能力, 因此支持通信的智能车辆能够作为移动缓存节点, 为终端用户带来方便。

随着车联网快速发展, 车载终端已经成为移动

收稿日期: 2020-07-07; 修回日期: 2020-09-24

通信作者: 张海波, zhanghb@cqupt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61801065, No.61601071); 长江学者和创新团队发展计划基金资助项目 (No.IRT16R72); 重庆市基础与前沿基金资助项目 (No.cstc2018jcyjAX0463)

Foundation Items: The National Natural Science Foundation of China (No.61801065, No.61601071), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT16R72), The Basic Research and Frontier Exploration Projects in Chongqing (No.cstc2018jcyjAX0463)

计算设备的重要组成部分^[1]。为了应对移动数据和云数据流量的增加带来的时延和能耗增加的问题，基于云的车联网被广泛认为是提高服务性能的新范式。在云计算网络中，通过集成通信和计算技术，应用程序可以在本地的车载终端上运行，也可以转移到远程计算云上运行^[2-5]。然而，由于回程和主干网传输的容量限制和时延波动，将云服务器放置在远离移动车辆的位置会严重降低卸载效率，并且增加传输时延。为此，业界提出了移动边缘计算(MEC, mobile edge computing)来解决这一问题，在靠近终端用户的地方执行时延敏感和上下文敏感的应用程序^[6]。MEC 将云服务推向无线网络的边缘，并在移动车载终端附近提供计算卸载服务，极大地提高了资源利用率和计算性能。此外，MEC 系统中的服务器可以实现网络内缓存功能，类似于以信息为中心的组网(ICN, information-centric networking)^[7]提供的功能，可以减少重复信息计算。

边缘缓存技术有效减少了网络中重复内容的计算与传输，是未来无线网络提高频谱利用率、缩短时延、降低网络负载的重要技术之一^[8-10]。基于全球流量特性，在短时间内，许多用户会请求一些流行的内容，这些内容占据了大部分流量^[11-12]。因此，主动缓存流行内容可以减轻回程链路的负担。在传统的蜂窝系统中，用户请求的内容必须从远离移动网络的互联网内容分发网络(CDN, content delivery network)节点获取，然后在移动核心网络上实现缓存内容^[13]，回程链路仍然受到容量限制。随着基站和低成本存储单元的发展，在大型基站和小型基站上部署缓存成为可能。同时，车联网的发展使车辆具备缓存能力，V2V(vehicle-to-vehicle)通信可以根据用户之间的社交关系，利用用户设备上的存储单元进行内容共享。

目前，已有大量学者致力于边缘缓存的研究。文献[14]设计了一个新的以信息为中心的异构网络框架，采用基于交替方向乘法器的分布式算法来求解缓存资源的分配问题。考虑到内容的流行度分布、可用的缓存大小和网络的拓扑结构，文献[15]通过设计缓存策略使系统吞吐量最大化。文献[16]提出了一种基于李雅普诺夫的在线算法来优化资源分配和内容缓存策略问题。文献[17]为了进一步降低传输时延及提高相应速率，提出了协同缓存分配和计算分流方案，MEC 服务器之间协同执行计算任务和数据缓存。在车联网中，文献[18]考虑了车

辆媒体应用的蜂窝通信网络上的主动内容传输问题，提出了一个新的理论框架来描述移动边缘网络在计算、缓存和通信资源之间的权衡，通过传输跳数、用户需求预测的规模和准确性，以及缓存副本的数量来度量计算、缓存和通信资源。文献[19]研究了高速车辆在 LTE-V2I(vehicle-to-infrastructure)网络中的部署问题，考虑 MEC 系统的回程能力、车辆速度和内容流行度分布，制定联合优化框架以最小化 MEC 系统的缓存，同时最大化平均下载百分比。然而，上述研究主要集中在 MEC 服务器侧进行内容缓存，智能车辆的闲置资源没有被有效利用。

本文利用移动边缘计算资源与周围闲置车辆资源来增强边缘缓存能力，其中智能车辆被视为与 MEC 服务器共享内容缓存任务的协作缓存代理；充分利用大量异构集成的存储资源进行内容缓存，特别是利用移动智能车辆和手持设备的资源来缓存流行内容；进一步研究车辆异构网络中基于 MEC 的任务卸载决策和资源分配问题；结合车联网的低时延要求与绿色通信的低能耗要求，将开销定义为时延与能耗的加权组合，提出了一种车联网中基于 MEC 的 V2X(vehicle-to-everything)协同缓存和资源分配机制，对网络内计算、缓存和通信资源进行有效分配，减少内容访问时延，提高资源利用率。

2 系统模型

2.1 网络模型

基于 MEC 的 V2X 协同缓存网络模型如图 1 所示。考虑一个车载云协同边缘缓存模型。在这个模型中，道路周围部署 L 个路边单元(RSU, road side unit)，表示为 $\mathcal{L} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \dots, \mathcal{M}_L\}$ ，每个 RSU 都配有一个 MEC 服务器。道路上 N 个车辆呈泊松分布，其车辆标识为 $\mathcal{N} = \{1, 2, 3, \dots, N\}$ ，由此车辆表示为 $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_N\}$ 。由于 MEC 服务器与邻近车辆均具有计算与缓存能力，因此将其统称为服务节点 $\mathcal{W} = \{w_1, w_2, w_3, \dots, w_M\}$ 。每个 RSU 覆盖范围内随机分布 n 个车辆，小区 j 的车辆集合为 $\mathcal{CV}_j = \{v_1, v_2, \dots, v_n\}$ 。车载 IEEE 802.11p OBU 具有 802.11p 网络接口和蜂窝网络接口，车辆可以通过 RSU 将任务卸载至 MEC 服务器进行计算，或者卸载至邻近车辆进行 V2V 通信。为了有效地复用频谱，V2I 模式和 V2V 模式在同一频段工作。频谱被均分为 K 个子信道，表示为 $\mathcal{K} = \{1, 2, 3, \dots, K\}$ ，每个

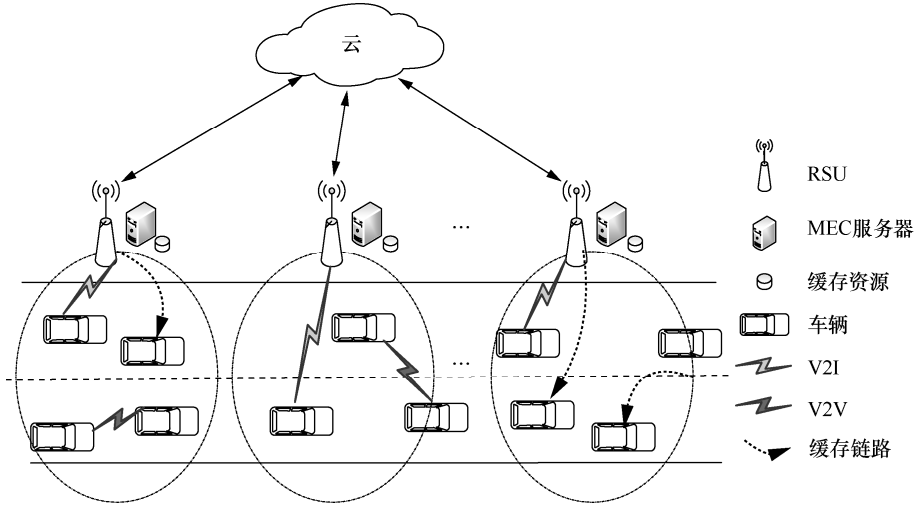


图 1 基于 MEC 的 V2X 协同缓存网络模型

子信道带宽为 B 。车辆卸载策略集合表示为 $\mathcal{A}=\{a_1, a_2, a_3, \dots, a_N\}$ ，若 $a_i=1$ ，则代表 v_i 将任务卸载到服务节点 w_m 进行计算；若 $a_i=0$ ，则代表 v_i 在本地执行计算任务。

假设在 t 时刻，缓存池存在一部分任务，当车辆有任务请求时，首先扫描周围服务节点是否存在任务缓存。如果任务缓存在服务节点上，则服务节点通知车辆任务，并在计算完成后将结果回传给车辆。通过这种方式，车辆不需要进行任务卸载，可以有效降低移动设备的能量消耗和任务卸载的时延。若服务节点上没有请求任务的缓存，则车辆进行卸载决策以及资源分配。当服务节点第一次计算完成请求任务后，考虑缓存决策。服务节点的缓存策略集合表示为 $\mathcal{G}_m=\{g_{m,1}, g_{m,2}, g_{m,3}, \dots, g_{m,n_i}\}$ ，若 $g_{m,n_i}=1$ ，则表示服务节点 w_m 将计算任务 n_i 进行缓存，以便下次请求，减少网络传输，降低计算时延。所有服务节点的缓存集合表示为 $\mathcal{AG}=\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \dots, \mathcal{G}_M\}$ 。

2.2 通信模型

在基于 MEC 的 V2X 协同缓存模型中，任务请求车辆 v_i 可以向 MEC 服务器或者临近车辆进行任务缓存请求，若服务节点不存在任务缓存，可以将其计算任务 T_i 卸载到 MEC 服务器进行 V2I 模式通信，或卸载到邻近车辆进行 V2V 模式通信，或在本地执行其计算任务。为了提高频谱利用率，V2V 模式复用 V2I 模式的上行传输信道。因为车联网环境对于时延要求苛刻，所以考虑一个车辆可以分配多个上行信道。为了更好地描述信道分配情况，本

文引入信道连接矩阵 \mathbf{C} ， \mathbf{C} 为 $N_{\text{off}} \times K$ 的二元变量矩阵，其中 N_{off} 代表卸载用户数量。信道连接矩阵 \mathbf{C} 的二元变量 $c_{i,k}$ 代表子信道 k 是否被分配给用户 i 。若 $c_{i,k}=1$ ，则表示子信道 k 被分配给用户 i 进行上行数据传输；若 $c_{i,k}=0$ ，则表示子信道 k 未被分配给用户 i 。 v_i 的上行传输速率为

$$r_{i,j} = \sum_{k \in K} B \log(1 + c_{i,k} \text{SINR}_{i,j}^k) \quad (1)$$

由于 V2X 车联网在卸载情况下的干扰环境复杂，考虑同层干扰与跨层干扰，具体而言，干扰环境是复用信道导致的同频干扰，因此任务请求车辆 v_i 在子信道上 k 的信干噪比（SINR, signal to interference plus noise ratio）表示为

$$\text{SINR}_{i,j}^k = \frac{p_i h_{i,j}^k}{\sigma^2 + I_i} \quad (2)$$

其中， p_i 为 v_i 的上行传输发送功率， $h_{i,j}^k$ 为 v_i 与服务节点 g_j 在子信道 k 上的信道增益， σ^2 为高斯白噪声功率， I_i 为 v_i 受到的干扰。

利用 I_i^D 与 I_i^C 分别表示 V2V 用户对车辆 v_i 的干扰和 V2I 用户对车辆 v_i 的干扰，则将车辆 v_i 受到的干扰 I_i 表示为 V2V 用户与 V2I 用户对车辆 v_i 的干扰之和，即 $I_i = I_i^D + I_i^C$ 。若车辆 v_i 选择的服务节点为 MEC 服务器，由于小区内采用正交频分复用（OFDM, orthogonal frequency division multiplexing）进行正交子信道分配，V2V 用户复用 V2I 用户上行信道，因此本小区内仅考虑 V2V 用户的信道复用干扰，本小区内 V2V 用户数为 $N_{D,i}$ ，则 I_i^C 和 I_i^D 可以分别表示为

$$I_i^C = \sum_{m \in \mathcal{M}} \sum_{i \in V_m, x \in V_m', m' \neq m} c_{x,k}^C p_x h_{x,j}^k \quad (3)$$

$$I_i^D = \sum_{y=1}^{N_{Dj}} c_{y,k}^D p_y h_{y,j}^k \quad (4)$$

其中, $c_{x,k}^C$ 表示子信道 k 是否被分配给 V2I 用户 x , $c_{y,k}^D$ 表示子信道 k 是否被分配给 V2V 用户 y 。 $c_{x,k}^C$ 与 $c_{y,k}^D$ 的并集则为 \mathcal{C} 。

若车辆 v_i 选择的服务节点为车辆, 即进行 V2V 模式通信, I_i^C 和 I_i^D 可以分别表示为

$$I_i^C = \sum_{x=1}^{N_C} c_{x,k}^C p_x h_{x,j}^k \quad (5)$$

$$I_i^D = \sum_{y=1, y \neq i}^{N_D} c_{y,k}^D p_y h_{y,j}^k \quad (6)$$

2.3 计算模型

假设每个任务请求车辆均有一个计算任务 $\mathcal{Z}_i = \{d_i, s_i, t_i^{\max}\}$, $i \in N$ 需要处理, 其中, d_i 为任务 \mathcal{Z}_i 的输入大小, s_i 为计算任务 \mathcal{Z}_i 所需的 CPU 周期数, t_i^{\max} 为计算任务 \mathcal{Z}_i 所能容忍的最大时延。车辆可以通过 RSU 将任务卸载至 MEC 服务器进行计算, 或者卸载至邻近车辆处理, 也可以在本地车辆执行。

1) 卸载计算

当车辆自身计算能力有限, 不足以支持任务的时延要求时, 需要将任务卸载至服务节点进行计算。任务处理过程将带来时延和能量消耗。由于回传的处理结果数据量较小, 因此本文忽略回传过程的时延和能耗^[20], 仅考虑上传时延、计算时延以及传输能耗。

定义任务请求车辆 v_i 将任务卸载至服务节点 w_j 的计算过程产生的开销为时延与能耗的加权组合, 表示为

$$u_i^{\text{off}} = \alpha t_i^{\text{off}} + \beta e_i^{\text{off}} \quad (7)$$

其中, α 与 β 分别为非负的时延与能耗的权重因子, 且满足 $\alpha + \beta \leq 1$; $t_i^{\text{off}} = \frac{d_i}{r_{i,j}} + \frac{s_i}{f_j^i}$ 为卸载时延与计算时延之和, f_j^i 为服务节点 w_j 分配给车辆 v_i 的计算资源; $e_i^{\text{off}} = p_i \frac{d_i}{r_{i,j}}$ 为传输过程的能量消耗, p_i 为上行传输功率。

2) 本地计算

假设车辆 v_i 计算能力为 F_i^1 , 不同车辆具有不同的计算能力。当车辆任务 \mathcal{Z}_i 在本地计算时, 车辆 v_i

所需要负担的开销为

$$u_i^1 = \alpha t_i^1 + \beta e_i^1 \quad (8)$$

其中, $t_i^1 = \frac{s_i}{F_i^1}$ 为计算所需时延, $e_i^1 = \varphi s_i (F_i^1)^2$ 为执行任务的能量消耗, φ 为每 CPU 周期消耗的能量的功率系数^[21]。

2.4 缓存模型

缓存模型包括 2 个部分, 即当前存在缓存 $\text{hit}_{j,i}$ 与缓存更新 $g_{j,i}$ 。 $\text{hit}_{j,i} = 1$ 表示服务节点 w_j 已缓存车辆 v_i 的内容; $\text{hit}_{j,i} = 0$ 表示服务节点 w_j 未缓存车辆 v_i 的内容, 车辆需要进行卸载或本地计算。假设在一定时间间隔内, 服务节点的缓存池中存在 X 个内容, 表示为 $\mathcal{X} = \{1, 2, \dots, X\}$ 。对于内容 $x \in \mathcal{X}$, 其流行程度服从 Zipf 分布, 内容 x 被请求的概率为

$$q_x = \frac{I(x)^{-\delta}}{\sum_{m=0}^X I(x)^{-\delta}}, \quad \forall x \in X \quad (9)$$

其中, 将缓存池中已缓存内容按照降序排列, $I(x)$ 表示内容 x 在缓存内容集合中受欢迎的等级; 参数 δ 表示流行分布倾斜, 这意味着更大的 δ 对应更高的内容复用, 即前几个流行的内容占了大部分的请求, 设置 δ 为 0.56^[22]。

缓存更新参数 $g_{j,i}$ 表示服务节点 w_j 是否缓存车辆 v_i 的内容。当服务节点 w_j 计算完成车辆 v_i 的任务 \mathcal{Z}_i 后考虑缓存更新, $g_{j,i} = 1$ 表示服务节点 w_j 将缓存任务 \mathcal{Z}_i , $g_{j,i} = 0$ 表示不缓存任务 \mathcal{Z}_i 。

不同的服务车辆拥有不同的缓存能力, 由于缓存能力有限, 因此服务节点 w_j 缓存内容总量不能超过其自身的最大缓存容量 H_j 。

2.5 问题形成

当智能车辆请求一个任务计算时, 首先检查自身缓存池是否存在内容缓存。如果内容在本地可用, 则不需要发布任务请求; 否则, 扫描周围服务节点是否存在内容缓存, 若存在, 则在服务节点计算完成后回传, 若不存在, 则需要考虑是否卸载。当任务卸载至服务节点计算完成后, 服务节点考虑缓存的更新, 之后将内容回传, 服务结束。本文目标为通过恰当的卸载与缓存决策以及通信和计算资源的分配, 使系统开销最小化。因此, 优化目标表示为

$$\begin{aligned}
\min_{\mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{F}, \mathcal{AG}} U(\mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{F}, \mathcal{AG}) &= \sum_{i=1}^N \left\{ \text{hit}_{j,i} u_i^{\text{cache}} + (1 - \text{hit}_{j,i}) g_{j,i} \left[(1 - a_i) u_i^l + a_i u_i^{\text{off}} \right] \right\} = \\
&\sum_{i=1}^N \left\{ \text{hit}_{j,i} \alpha \frac{S_i}{f_j^i} + (1 - \text{hit}_{j,i}) g_{j,i} \left\{ (1 - a_i) \left[\alpha \frac{S_i}{F_i^l} + \beta \varphi s_i (f_i^l)^2 \right] + a_i \left[\alpha \left(\frac{d_i}{r_{ij}} + \frac{S_i}{f_j^i} \right) + \beta p_i \frac{d_i}{r_{ij}} \right] \right\} \right\} \\
\text{s.t. C1: } &a_i \in \{0, 1\}, \quad \forall i \in \mathcal{N} \\
\text{C2: } &c_{i,k} \in \{0, 1\}, \quad \forall i \in \mathcal{N}, k \in \mathcal{K} \\
\text{C3: } &g_{j,i} \in \{0, 1\}, \quad \forall i \in \mathcal{N} \\
\text{C4: } &0 < p_i < p_{\max}, \quad \forall i \in \mathcal{N} \\
\text{C5: } &f_j^i > 0, \quad \forall i \in \mathcal{N} \\
\text{C6: } &\sum_{i \in \mathcal{N}} a_i f_j^i \leq F_j^{\max}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \\
\text{C7: } &(1 - a_i) t_i^{\text{local}} + a_i t_i^{\text{off}} \leq \min \left\{ t_i^{\max}, \frac{L_j}{V_u}, \frac{d_{\text{interrupt}}}{|V_u - V_v|} \right\}, \quad \forall i \in \mathcal{N} \\
\text{C8: } &\sum_{i=1}^N g_{j,i} d_i \leq H_j
\end{aligned} \tag{10}$$

其中, \mathcal{A} 为所有任务请求车辆的卸载策略集合, \mathcal{C} 为信道连接矩阵, \mathcal{P} 为卸载车辆的任务发送功率集合, \mathcal{F} 为计算资源分配策略, \mathcal{AG} 为服务节点的缓存决策。

约束条件 C1 与 C3 表示卸载决策为 0-1 决策。C2 表示信道分配矩阵为二进制变量。C4 保证了功率分配为非负值且不超过上行传输功率变化范围。C5 与 C6 表示计算资源分配不超过服务节点的最大计算能力。C7 表示时延约束, 其中 L_j 为 v_i 与 RSU 覆盖范围边界的距离, V_u 为任务请求车辆的移动速度, V_v 为服务车辆的移动速度, $d_{\text{interrupt}}$ 为最大中断距离。C8 表示服务节点的缓存内容不能超过其最大缓存容量。本文设置车辆为单向匀速行驶。由此, v_u 与 v_v 的速度差为 $|V_u - V_v|$, 其相隔距离达到最大中断距离的时间为 $\frac{d_{\text{interrupt}}}{|V_u - V_v|}$ 。此外, 由于单向行驶, 因此 v_i 驶出 RSU $_j$

覆盖范围的最大时间为 $\frac{L_j}{V_u}$ 。

3 V2X 卸载与资源分配方案

为了应对现代化车联网中移动数据服务的爆炸性增长需求, 开发高效的内容缓存和资源分配方案非常重要, 其目标是显著减少冗余数据传输, 提高内容交付效率。本文提出了一种基于 MEC 的车联网 V2X 协同缓存和资源分配机制, 根据不同

任务请求车辆对传输速率的需求, 通过图着色模型为用户分配合适信道; 针对最小化系统开销的目标, 采用拉格朗日乘子法对功率与计算资源进行分配。考虑到任务重复性, 本文利用 MEC 服务器与邻近车辆资源增强边缘缓存能力, 将缓存模型表示为背包问题, 使用动态规划法对其进行缓存决策。

3.1 基于图着色的信道分配

本文采用部分频率复用的方式进行子信道分配, 小区内 V2I 用户通过 OFDM 分配正交子信道, V2V 用户复用 V2I 用户上行信道; 相邻小区之间的用户由于频率复用而产生信道干扰。将子信道分配转化为图着色模型, 建立加权干扰图 $G = (\mathcal{V}^{\text{off}}, \varepsilon)$, 其中, \mathcal{V}^{off} 表示卸载车辆集合, $\varepsilon = \{e_{i,j}\}_{i,j \in \mathcal{V}}$ 表示用户车辆之间的干扰权重。车辆 v_i 对应的服务节点 w_x 为 MEC 服务器 ($w_x \in \mathcal{M}$) 或邻近车辆 ($w_x \in \mathcal{V}$)。基于此, 干扰权重计算式为

$$e_{i,j} = \begin{cases} 0, & i = j \text{ or } i, j \in \mathcal{CV}_x \text{ and } w_x, w_y \in \mathcal{M} \\ p_i h_{i,j}, & i \neq j \text{ and } i \in \mathcal{CV}_x, j \in \mathcal{CV}_y, x \neq y \end{cases} \tag{11}$$

其中, i 和 j 表示车辆 v_i 和 v_j 的序号标识; 第一行表示当车辆 v_i 与车辆 v_j 处于同一小区且同为 V2I 用户或者 $i = j$ 时, 相互无干扰; 第二行表示车辆 v_i 与车辆 v_j 处于不同小区时, 存在复用干扰。

引入信道连接矩阵 \mathbf{C} 为 $N^{\text{off}} \times K$ 的二元变量矩阵，其中 $N^{\text{off}} \times K$ 代表卸载用户数量。信道连接矩阵 \mathbf{C} 的二元变量 $c_{x,k}$ 代表子信道 k 是否被分配给用户 x 。建立干扰矩阵 $\mathbf{O} = [o_{n,k}]_{N^{\text{off}} \times K}$ ， $o_{n,k}$ 代表用户 n 在信道 k 上受的干扰之和。

$$o_{n,k} = \sum_{m \in \mathcal{V}^{\text{off}}} c_{mk} e_{mn} = \sum_{m \in \mathcal{V}^{\text{off}}} c_{mk} p_m h_{mn} \quad (12)$$

基于图着色的信道分配算法伪代码如算法 1 所示。

算法 1 子信道分配算法

输入 卸载任务车辆集合为 \mathcal{V}^{off} ，信道矩阵集合 \mathcal{K} ，待分配集合 $\Phi = \mathcal{V}^{\text{off}}$

输出 信道连接矩阵 \mathbf{C}

- 1) 建立加权干扰图 $G = (\mathcal{V}^{\text{off}}, \varepsilon)$ ，计算 $e_{i,j}$
- 2) 建立信道连接矩阵 $\mathbf{C} = [c_{n,k}]_{N^{\text{off}} \times K}$ ，干扰矩阵

$$\mathbf{O} = [o_{n,k}]_{N^{\text{off}} \times K}$$

- 3) 计算相互干扰 $e_{n,k}$ ，更新干扰矩阵

$$\mathbf{O} = [o_{n,k}]_{N^{\text{off}} \times K}$$

- 4) 对 $o_{n,k}$ 排序，得到 $\text{Order}_o = \text{order}(o_{n,k})$
- 5) 取 Order_o 的前 k 个进行初始信道分配，

$$\Phi = \Phi \setminus \text{Order}_o[1:k]$$

- 6) for $n = 1:k$
- 7) 计算当前传输速率 R_n
- 8) if $R_n^{\min} = \frac{d_n f_j^n}{t_n^{\max} f_j^n - s_n} \geq R_n$
- 9) $\Phi = \Phi \cup \{n\}$

10) end if

11) end for

12) while $\Phi \neq \emptyset$

13) 寻找颜色 \vec{k} 使用户 n 能获得最大速率，颜色为 k 时，用户 n 的速率为 r_n^k

$$\vec{k} = \arg \max_{k \in \mathcal{K}} \{r_n^k\}, \vec{k} \rightarrow v_n$$

$$14) \text{ if } R_n^{\min} = \frac{d_n f_j^n}{t_n^{\max} f_j^n - s_n} \leq r_n^k$$

15) $\Phi = \Phi \setminus \{n\}$

16) end if

17) 更新信道连接矩阵 \mathbf{C} ，干扰矩阵 \mathbf{O} ，未

分配用户集合 Φ

18) end while

3.2 计算资源分配

当服务节点不存在内容缓存，且车辆决定卸载

计算时，由式(10)可得以下优化问题。

$$\min_{\mathcal{P}, \mathcal{F}} U(\mathcal{P}, \mathcal{F}) = \sum_{i=1}^N \alpha \frac{d_i}{n_i \text{blb} \left(1 + \frac{p_i h_{i,j}}{\delta^2 + I_i} \right)} +$$

$$\alpha \frac{s_i}{f_j^i} + \beta p_i \frac{d_i}{n_i \text{blb} \left(1 + \frac{p_i h_{i,j}}{\delta^2 + I_i} \right)}$$

s.t. C4: $0 < p_i < p_{\max}$

C5: $f_j^i > 0, \forall i \in \mathcal{N}$

C6: $\sum_{i \in \mathcal{N}} f_j^i \leq F_j^{\max}, \forall i \in \mathcal{N}, j \in \mathcal{M}$

C7: $t_i^{\text{off}} \leq \min \left\{ t_i^{\max}, \frac{L_j}{V_u}, \frac{d_{\text{interrupt}}}{|V_u - V_v|} \right\}, \forall i \in \mathcal{N}$ (13)

对于车辆 v_i ，寻找其干扰上界为

$$\tilde{I}_i = \sum_{k=1}^K \sum_{n=1}^N c_{n,k} p_{\max} h_{n,j}^k \quad (14)$$

令 $\mathcal{G}_i = \frac{h_{i,j}}{\sigma^2 + \tilde{I}_i}$ ，可得信噪比下界 $\widetilde{\text{SINR}}_i = \mathcal{G}_i p_i$ 。

根据式(13)中的 C7，令 $T_i^{\max} = \min \left\{ t_i^{\max}, \frac{L_j}{V_u}, \frac{d_{\text{interrupt}}}{|V_u - V_v|} \right\}$ ，

且约束条件取等号可得

$$\frac{d_i}{n_i \text{blb} (1 + \mathcal{G}_i p_i)} = T_i^{\max} - \frac{s_i}{f_j^i} \quad (15)$$

令 $\xi_i = \frac{d_i}{n_i b}$ ， $\omega_i = \frac{\xi_i}{\text{lb}(1 + \mathcal{G}_i p_i)}$ ，可得

$p_i = \frac{2^{\xi_i} - 1}{\mathcal{G}_i}$ ， $f_j^i = \frac{s_i}{T_i^{\max} - \omega_i}$ 。将其代入式(13)，C4

可转化为 $\omega_i \geq \frac{\xi_i}{\text{lb}(1 + \mathcal{G}_i p_{\max})}$ ，由 C5 可得 $\omega_i < T_i^{\max}$ 。

基于以上信息，式(13)可以改写为

$$\min_{\omega} U(\omega) = \sum_{i=1}^{N^{\text{off}}} \alpha T_i^{\max} + \beta \frac{2^{\xi_i} - 1}{\mathcal{G}_i} \omega_i$$

s.t. C9: $\frac{\xi_i}{\text{lb}(1 + \mathcal{G}_i p_{\max})} \leq \omega_i \leq T_i^{\max}$

C10: $\sum_{i \in \mathcal{N}^{\text{off}}} \frac{s_i}{T_i^{\max} - \omega_i} \leq F_j^{\max}, \forall i \in \mathcal{N}, j \in \mathcal{M}$ (16)

在式(16)中对 ϖ 求二阶导可得

$$\frac{\partial^2 U(\varpi)}{\partial \varpi^2} = \frac{\beta \xi_i^2 2^{\varpi_i} \ln^2 2}{\mathcal{G}_i \varpi_i^3} > 0 \quad (17)$$

因此, 式(16)为凹函数, 可以采用 KKT (Karush-Kuhn-Tucker) 条件对其进行求解。式(16)的拉格朗日函数为

$$\begin{aligned} L(\varpi, \lambda, \mu, \rho) = & \sum_{i=1}^{N^{\text{off}}} \alpha T_i^{\text{max}} + \beta \frac{2^{\varpi_i} - 1}{\mathcal{G}_i} \varpi_i + \\ & \sum_{i=1}^{N^{\text{off}}} \lambda_i \left(\frac{\xi_i}{\log(1 + \mathcal{G}_i p_{\text{max}})} - \varpi_i \right) + \sum_{i=1}^{N^{\text{off}}} \mu_i (\varpi_i - T_i^{\text{max}}) + \\ & \rho \left(\sum_{i \in N^{\text{off}}} \frac{S_i}{T_i^{\text{max}} - \varpi_i} - F_j^{\text{max}} \right) \end{aligned} \quad (18)$$

其中, ϖ 为卸载车辆的上行传输时延向量, λ 和 μ 为拉格朗日乘子向量, λ_i 、 μ_i 、 ρ 为非负拉格朗日乘子变量, 满足 KKT 条件

$$\begin{aligned} \frac{\partial L(\varpi, \lambda, \mu, \sigma)}{\partial \varpi} = & \frac{\beta}{\mathcal{G}_i} \left(2^{\frac{\xi_i}{\varpi_i}} - 1 \right) \left(1 - \frac{\xi_i}{\varpi_i} \ln 2 \right) + \\ & \frac{\beta \ln 2 \xi_i}{\mathcal{G}_i \varpi_i} - \lambda_i + \mu_i + \rho \left(\frac{S_i}{(T_i^{\text{max}} - \varpi_i)^2} \right) \end{aligned} \quad (19)$$

$$\lambda_i \left[\frac{\xi_i}{\log(1 + \mathcal{G}_i p_{\text{max}})} - \varpi_i \right] = 0 \quad (20)$$

$$\mu_i (\varpi_i - T_i^{\text{max}}) = 0 \quad (21)$$

$$\rho \left(\sum_{i \in N^{\text{off}}} \frac{S_i}{T_i^{\text{max}} - \varpi_i} - F_j^{\text{max}} \right) = 0 \quad (22)$$

拉格朗日乘子更新规则为

$$\lambda_i(t+1) = \left[\lambda_i(t) + \gamma(t) (T_i^{\text{min}} - \varpi_i) \right]^+ \quad (23)$$

$$\mu_i(t+1) = \left[\mu_i(t) + \gamma(t) (\varpi_i - T_i^{\text{min}}) \right]^+ \quad (24)$$

$$\rho(t+1) = \left[\rho(t) + \gamma(t) \left(\sum_{i \in N^{\text{off}}} \frac{S_i}{T_i^{\text{max}} - \varpi_i} - F_j^{\text{max}} \right) \right]^+ \quad (25)$$

其中, t 表示第 t 次迭代, $\gamma(t)$ 表示迭代步长, 时间下限 $T_i^{\text{min}} = \frac{\xi_i}{\text{lb}(1 + \mathcal{G}_i p_{\text{max}})}$ 。

根据以上条件获得最优解 ϖ_i^* 后, 由 p_i^* 、 f_i^{*j} 与 ϖ_i^* 的关系可得

$$p_i^* = \frac{2^{\frac{\xi_i}{\varpi_i^*}} - 1}{\mathcal{G}_i} \quad (26)$$

$$f_i^{*j} = \frac{S_i}{T_i^{\text{max}} - \varpi_i^*} \quad (27)$$

功率与计算资源分配算法伪代码如算法 2 所示。

算法 2 功率与计算资源分配算法

输入 卸载车辆数目 N^{off} , 最大迭代次数 mt , $\gamma(0)$

输出 功率分配 \mathcal{P}^* , 计算资源分配 \mathcal{F}^*

1) for $t = 1 : mt$ do

2) $\gamma(t) = \frac{1}{100 + \gamma(t-1)}$

3) for $i = 1 : N^{\text{off}}$ do

4) 计算式(16)中的 ϖ_i

5) end for

6) 根据式(23)~式(25)更新拉格朗日乘子 $\lambda_i(t)$ 、 $\mu_i(t)$ 、 $\rho(t)$

7) end for

3.3 缓存决策

任务请求车辆 v_i 将计算任务 \mathcal{Z}_i 卸载至服务节点 w_j 执行计算, 由于服务节点缓存容量有限, 因此在计算完成以后需要选择是否将其缓存, 以备后续请求车辆使用。服务节点 w_j 的缓存容量为 H_j , 若缓存决策 $g_{j,i} = 1$, 则代表任务 \mathcal{Z}_i 被服务节点缓存, 因此服务节点需要牺牲缓存容量 d_i , 此时问题等效为使服务节点 w_j 的容量 $\{H_j - d_i\}$ 效益最大化; 若缓存决策 $g_{j,i} = 0$, 则代表任务 \mathcal{Z}_i 未被服务节点缓存, 此时问题等效为使服务节点 w_j 的容量 $\{H_j\}$ 效益最大化。

基于以上分析, 将缓存决策建模为 0-1 背包问题, 通过动态规划法^[23]对其求解, 伪代码如算法 3 所示。 $\mathbf{x}_i = (d_i, u_i^{\text{off}}, u_i^1, u_i^{\text{min}})$ 作为输入向量, 其中 $u_i^{\text{min}} = \min\{u_i^{\text{off}}, u_i^1\}$ 代表 v_i 节约的开销。对于服务节点 w_j , 其动态转移方程为

$$\begin{aligned} U'_j(i, j, H_j) = & \max \{ U'_j(i-1, j, H_j), \\ & U'_j(i-1, j, H_j - d_i) + u_i^{\text{min}} \}, \quad U'_j = -U_j \end{aligned} \quad (28)$$

其中, $U'_j(i, j, H_j)$ 代表服务节点 w_j 存在缓存容量 H_j 的情况下, 是否将用户 v_i 的内容缓存时 w_j 的开销。

算法 3 基于背包问题的缓存决策算法

输入 车辆信息 $\mathbf{x}_i = (d_i, u_i^{\text{off}}, u_i^1, u_i^{\text{min}})$, 服务节

点 w_j 状态函数 $U'_j(i, j, H_j)$ ，卸载至服务节点 w_j 的车辆数目 N_j^{off}

输出 缓存决策 \mathcal{H}

- 1) for $i = 1: N_j^{\text{off}}$ do
- 2) $U'_j(i, j, H_j) = \max\{U'_j(i-1, j, H_j), U'_j(i-1, j, H_j - d_i) + u_i^{\min}\}$
- 3) if $U'_j(i, j, H_j) \leq U'_j(i-1, j, H_j)$
- 4) $g_{j,i} = 0$
- 5) else
- 6) $g_{j,i} = 1$
- 7) end if
- 8) end for

3.4 算法总结

基于边缘缓存的 V2X 协同卸载与资源分配策略 (V2X-CCRA, V2X-collaborative caching and resource allocation) 伪代码如算法 4 所示, 原理如下。车辆首先扫描自身缓存池与周围服务节点是否存在内容缓存, 如果存在, 则进行计算或者在服务节点计算完成后回传任务结果; 如果不存在, 则加入卸载集合。针对卸载集合里的任务请求车辆, 通过基于图着色的信道分配算法得到一次信道分配矩阵, 同时根据拉格朗日 KKT 条件得到功率与计算资源分配矩阵。然后, 计算本地执行开销以及根据通信与计算资源的分配值计算卸载开销, 由于任务请求车辆均满足理性和利己主义, 目标均为尽量最小化自身计算代价, 因此, 比较本地计算与卸载至服务节点计算的执行开销, 做出卸载决策。更新卸载集合 Θ , 找到卸载集合中元素的对应卸载服务节点, 得到服务节点集合 \mathcal{Y} 。针对 \mathcal{Y} 中的每一个服务节点 w_j , 确定其服务的车辆集合 \mathcal{A}_j , 服务节点 w_j 将任务计算完成后进行缓存的更新, 考虑是否将这个任务缓存, 以备后续请求车辆使用。V2X-CCRA 研究车辆异构网络中基于 MEC 的任务卸载决策和资源分配问题, 利用 MEC 服务器与邻近车辆资源增强边缘缓存能力, 对网络内计算、缓存和通信资源进行有效分配, 减轻网络负载, 提高资源利用率。

算法 4 基于边缘缓存的 V2X 协同卸载与资源分配策略

输入 车辆数目 N , 服务节点 w_j 缓存容量 H_j , 信道集合 \mathcal{K} , 卸载集合 $\Theta = \phi$, α , β

输出 系统开销 U

- 1) for $i = 1: N$
- 2) 检查自身与服务节点缓存状态
- 3) if $\text{hit}_{j,i} = 1$
- 4) 存在内容缓存, 任务计算或者计算后回传
- 5) else
- 6) $\Theta = \Theta \cup \{i\}$
- 7) end if
- 8) end for
- 9) 通过图着色算法 (算法 1) 更新信道连接矩阵 \mathcal{C}
- 10) 通过计算功率与计算资源分配算法 (算法 2) 得到 \mathcal{P} 和 \mathcal{F}
- 11) for $i = 1: \text{length}(\Theta)$
- 12) 计算本地开销 u_i^l
- 13) 根据式(7)计算卸载开销 u_i^{off}
- 14) if $u_i^l \leq u_i^{\text{off}}$
- 15) $a_i = 0$, $\Theta = \Theta \setminus \{i\}$
- 16) else
- 17) $a_i = 1$
- 18) end if
- 19) end for
- 20) 确定 Θ 中元素 v_i 对应的服务节点 w_j 集合 \mathcal{Y}
- 21) 确定卸载到 \mathcal{Y} 中元素 w_j 的车辆集合 \mathcal{A}_j
- 22) for $j^* = 1: \text{length}(\mathcal{Y})$
- 23) for $i^* = 1: \text{length}(\mathcal{A}_j)$
- 24) $x_{i^*} = (d_{i^*}, u_{i^*}^{\text{off}}, u_{i^*}^l, u_{i^*}^{\min})$
- 25) $U'_{j^*}(i^*, j^*, H_{j^*}) = \max\{U'_{j^*}(i^*-1, j^*, H_{j^*}), U'_{j^*}(i^*-1, j^*, H_{j^*} - d_{i^*}) + u_{i^*}^{\min}\}$
- 26) if $U'_{j^*}(i^*, j^*, H_{j^*}) \leq U'_{j^*}(i^*-1, j^*, H_{j^*})$
- 27) $g_{j^*,i^*} = 0$
- 28) else
- 29) $g_{j^*,i^*} = 1$
- 30) end if
- 31) end for
- 32) end for

4 仿真分析

本文主要研究车联网中的效率优先应用, 如文件分享、地理信息收集等时延可容忍数据任务。性

能评估阶段，通过 MATLAB 平台对本文所提机制进行仿真。本节实验在 IEEE 802.11p 车辆网络场景标准和 MEC 白皮书的背景下展开，采用 3GPP 标准化中提出的信道增益模型^[24]。考虑路边有 3 个小区，每个小区都配置了 RSU 和 MEC 服务器。具体仿真参数如表 1 所示。

| 表 1 仿真参数 | |
|----------------------------------|---------------|
| 参数 | 数值 |
| 车辆最大发射功率 p_{\max} /dBm | 20 |
| 系统带宽 / MHz | 20 |
| 任务计算大小 d_i /MB | 10 ~ 20 |
| 计算任务所需 CPU 周期 s_i /周期 | 1 500 ~ 3 000 |
| 高斯白噪声功率 σ^2 /dBm | -60 |
| 权重因子设置 $\alpha = \beta$ | 0.5 |
| RSU 的覆盖直径 RL_j /m | 1 000 |
| 车辆计算能力 F_j /GHz | 1 ~ 2 |
| MEC 服务器计算能力 f_j /GHz | 10 |
| 上行传输信道数量 K /条 | 10 |
| 单小区车辆数目/辆 | 13 |
| Zipf 分布参数 δ | 0.56 |
| 车辆缓存能力 H^V /MB | 100 |
| MEC 服务器缓存能力 H^M /MB | 500 |
| 车辆移动速度/(km·h ⁻¹) | 40、60 |
| 最大中断距离 $d_{\text{interrupt}}$ /m | 345 |
| 缓存类型 | 15 |

为了评估当前提出机制的性能，将其与本地计算机制（ALCM, all local computing mechanism）、无边缓存的全卸载机制（AOCM wo.caching, all offloading computing mechanism without caching）、基于边缘缓存的全卸载计算机制（AOCM w.caching, all offloading computing mechanism with caching）、基于图着色的资源分配机制 JCOIM（joint computation offloading and interference management）^[24]、基于边缘缓存的 V2I 卸载机制 V2I-CCRA（V2I collaborative caching and resource allocation）进行比较。

图 2 显示了车辆缓存容量与总开销的关系。车辆缓存容量增加，意味着更多的内容可以被缓存到车辆上，道路上可以形成更多的车辆云。当一些车辆从云中移出或离开道路时，用户可以从云中的其他车辆或其他车辆云中获取内容，从而大大减少重复计算，降低系统开销。本文 V2X-CCRA 机制相较于基于边缘缓存的全卸载模式，优势明显。基于边缘缓存

的全卸载模式由于不考虑本地计算，任务计算全依赖服务节点，卸载过程中由于传输干扰使时延增加，同时车辆移动性原因使服务中断带来重复计算，因此系统开销较大。本文 V2X-CCRA 机制基于用户理性与利己主义，使任务请求车辆在卸载与本地计算之间进行权衡，尽量减少任务计算的代价，因此可有效降低系统开销。当车辆缓存容量逐步增长时，2 种机制的区别逐渐减小，这是由于车辆缓存容量增加，使更多的内容可以被缓存，因此需要卸载的任务减少，此时信道竞争减小，服务节点已有足够的完成计算。

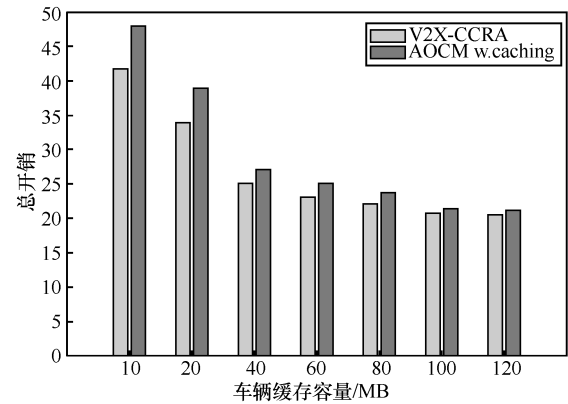


图 2 车辆缓存容量与总开销的关系

图 3 和图 4 分别为车辆数目和任务量与总开销关系。随着车辆数目与任务量的增加，总开销随之增长。传统的 ALCM 由于车辆本身计算能力有限而带来较大的时延开销。当车辆数目较小时，由于任务量较少，缓存空间与计算能力均可良好地胜任任务计算，其中由于基于边缘缓存的 AOCM w.caching 存在缓存功能，一部分任务不需要卸载计算，降低了上传拥塞，因此性能与 V2X-CCRA 相近，且优于 JCOIM 机制。然而随着任务量增加，卸载部分的任务数据增大，网络拥塞已不可避免，因此 AOCM w.caching 总开销急剧增长。此外，由于全卸载机制导致信道干扰严重且计算资源有限，因此无论是否存在缓存功能，其系统开销均随车辆数目与任务大小增加而迅速增长。JCOIM 机制应用于本文场景，由于其忽略功率分配对系统开销的影响，以及未利用周围临近车辆的闲置计算与缓存资源，因此性能不理想。本文 V2X-CCRA 机制充分利用周围闲置资源，合理分配计算与通信资源，V2X 协同卸载可以有效提高资源利用率，缓存机制可以避免重复计算以及减少网络拥塞，从而降低系统开销。

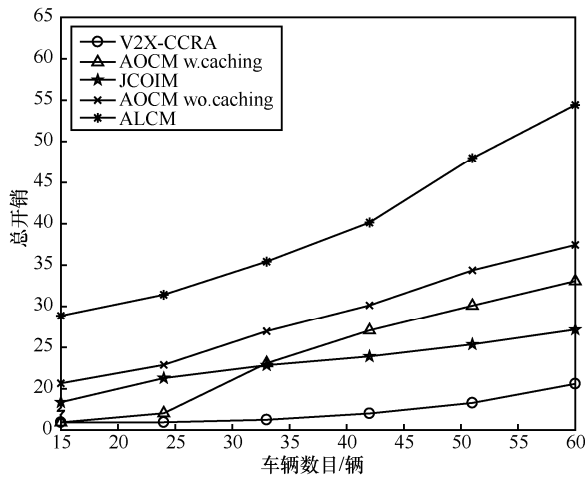


图 3 车辆数目与总开销的关系

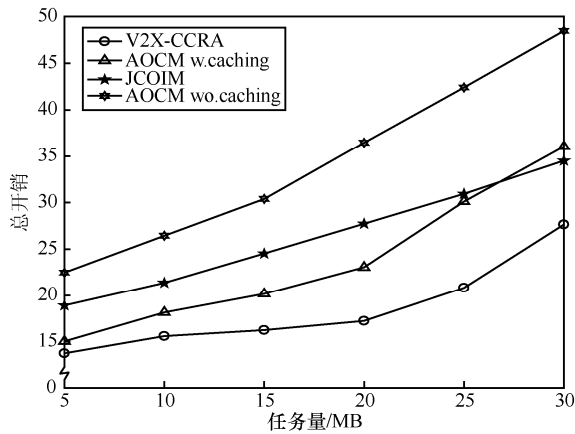


图 4 任务量与总开销的关系

图 5 为信道数量与总开销的关系。信道数量增加，干扰逐渐减小，使传输时延降低，总开销减少。全卸载机制由于受信道数量影响较大，因此随着信道数量的增加，总开销降低速度加快。无边缘缓存的 AOCM wo.caching 由于所有任务必须全部卸载，因此对信道干扰最敏感，总开销下降速度最明显，但由于服务节点计算资源有限，从而总开销一直大于 AOCM w.caching。本文基于边缘缓存的 V2X-CCRA 机制能够更加灵活地管理资源，控制卸载流量，在信道数量变换的情况下做出有效的卸载权衡，保证资源有效利用。

图 6 为缓存类型与总开销的关系。缓存类型增加，意味着更多的内容可以被缓存，减少重复计算带来的时延，因此时间消耗逐步降低。在 V2I-CCRA 中，仅 MEC 服务器具有计算与缓存功能，忽略了车辆的缓存能力，因此在服务节点可缓存类型较少时，大部分任务请求需要被卸载计算。V2X-CCRA 由于可以有效利用周围车辆的闲置资源，可以增强

边缘计算能力，因此相较于 V2I 机制总开销较小。当缓存类型逐渐增加，车辆服务器由于缓存容量限制，无法缓存太多任务。MEC 服务器的大容量缓存池可以缓存更多的内容，此时大多数任务已可以从缓存池中获得，因此 V2I 机制与 V2X 机制的差距逐渐减小。

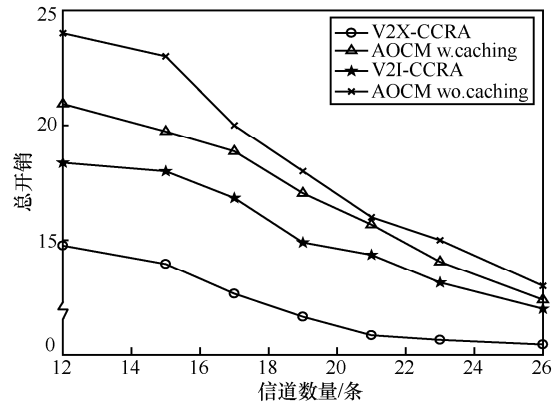


图 5 信道数量与总开销的关系

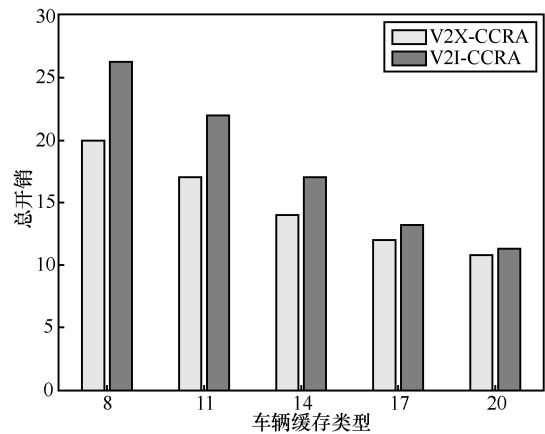


图 6 缓存类型与总开销的关系

5 结束语

本文在基于 MEC 的车联网场景中研究了基于边缘缓存的协同卸载与资源分配策略，利用 MEC 服务器与邻近车辆资源增强边缘缓存能力，提出了一种基于 MEC 的车辆网络 V2X 协同缓存和资源分配机制，对网络内计算、缓存和通信资源进行有效分配，避免重复传输，减少内容访问时延，提高资源利用率。仿真结果表明，所提机制可以有效减少任务请求车辆的计算代价以降低任务计算的系统开销，在各参数变化下均能取得较好效果。在未来工作中，将继续研究基于 MEC 的 V2X 卸载策略，同时考虑由车辆高速移动特性导致链路断开，使任

务回传失败的情况,建立基于车辆位置预测的V2X卸载框架。

参考文献:

- [1] 张海君, 张资政, 隆克平. 基于移动边缘计算的 NOMA 异构网络资源分配[J]. 通信学报, 2020, 41(4): 27-33.
ZHANG H J, ZHANG Z Z, LONG K P. Resource allocation in NOMA heterogeneous network based on MEC[J]. Journal on Communications, 2020, 41(4): 27-33.
- [2] 张海波, 荆昆仑, 刘开健. 车联网中一种基于软件定义网络与移动边缘计算的卸载策略[J]. 电子与信息学报, 2020, 42(3): 645-652.
ZHANG H B, JING K L, LIU K J. An offloading mechanism based on software defined network and mobile edge computing in vehicular networks[J]. Journal of Electronics and Information Technology, 2020, 42(3): 645-652.
- [3] ABDULRAHMAN A, YONG D, GUIYI W, et al. Collaborative security in vehicular cloud computing: a game theoretic view[J]. IEEE Network, 2018, 32(3):72-77.
- [4] SHOJAFAR M, CORDESCHI N, BACCARELLI E. Energy-efficient adaptive resource management for real-time vehicular cloud services[J]. IEEE Transactions on Cloud Computing, 2019, 7(1): 196-209.
- [5] JI L, GUO S. Energy-efficient cooperative resource allocation in wireless powered mobile edge computing[J]. IEEE Internet of Things Journal, 2019, 6(3): 4744-4754.
- [6] LIU Y, YU H, XIE S. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(11): 11158-11168.
- [7] HE Y, LIANG C, ZHANG Z, et al. Resource allocation in software-defined and information-centric vehicular networks with mobile edge computing[C]//Proceedings of the IEEE Vehicular Technology Conference. Piscataway: IEEE Press, 2017: 1-5.
- [8] FANG C, YAO H P, WANG Z W, et al. A survey of mobile information-centric networking: research issues and challenges[J]. IEEE Communications Surveys & Tutorials, 2018, 20(3): 2353-2371.
- [9] TRAN T X, POMPILI D. Adaptive bitrate video caching and processing in mobile-edge computing networks[J]. IEEE Transactions on Mobile Computing, 2019, 18(9):1965-1978.
- [10] ZHANG S, LI J, LUO H, et al. Low-latency and fresh content provision in information-centric vehicular networks[J]. arXiv Preprint, arXiv:2005.04358, 2020.
- [11] ZHANG K, LENG S, HE Y, et al. Cooperative content caching in 5G networks with mobile edge computing[J]. IEEE Wireless Communications, 2018, 25(3):80-87.
- [12] WANG S, ZHANG X, ZHANG Y, et al. A survey on mobile edge networks: convergence of computing, caching and communications[J]. IEEE Access, 2017, (5): 6757-6779.
- [13] JIANG W, FENG G, QIN S. Optimal cooperative content caching and delivery policy for heterogeneous cellular networks[J]. IEEE Transactions on Mobile Computing, 2017, 16(5):1382-1393.
- [14] ZHOU Y C, YU F R, CHEN J, et al. Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2017, 66(12): 11339-11351.
- [15] AO W C, PSOUNIS K. Fast content delivery via distributed caching and small cell cooperation[J]. IEEE Transactions on Mobile Computing, 2018, 17(5): 1048-1061.
- [16] XU J, CHEN L X, ZHOU P. Joint service caching and task offloading for mobile edge computing in dense networks[C]//Proceedings of the IEEE INFOCOM. Piscataway: IEEE Press, 2018: 207-215.
- [17] NDIKUMANA A, ULLAH S, LEANH T, et al. Collaborative cache allocation and computation offloading in mobile edge computing[C]//Proceedings of the IEEE Asia-Pacific Network Operations and Management Symposium. Piscataway: IEEE Press, 2017: 366-369.
- [18] JIAO J P, HONG X M, SHI J H. Proactive content delivery for vehicles over cellular networks: the fundamental benefits of computing and caching[J]. China Communications, 2018, 15(7): 88-97.
- [19] LUO Z H, LIWANG M H, HUANG L F, et al. Caching mechanism for mobile edge computing in V2I networks[J]. Transactions on Emerging Telecommunications Technologies, 2019: doi.org/10.1002/ett.3689.
- [20] WANG C M, LIANG C C, YU F R, et al. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2017, 16(8): 4924-4938.
- [21] ZHANG Y, ZHOU X, TENG Y, et al. Resource allocation for multi-user MEC system: machine learning approaches[C]//2018 International Conference on Computational Science and Computational Intelligence. Piscataway: IEEE Press, 2018: 794-799.
- [22] LI J, ZHANG H, JI H, et al. Joint computation offloading and service caching for MEC in multi-access networks[C]//2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications. Piscataway: IEEE Press, 2019: 1-6.
- [23] ASSI M, HARATY A R. A survey of the knapsack problem[C]//2018 International Arab Conference on Information Technology. Piscataway: IEEE Press, 2018: 1-6.
- [24] WANG C M, LIANG C C, YU F R, et al. Joint computation offloading and interference management in wireless cellular networks with mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2017, 66(8): 7432-7445.

[作者简介]



李方伟 (1960—), 男, 重庆人, 博士, 重庆邮电大学教授、博士生导师, 主要研究方向为无线资源管理。

张海波 (1979—), 男, 重庆人, 博士, 重庆邮电大学副教授、硕士生导师, 主要研究方向为车联网。

王子心 (1995—), 女, 重庆人, 重庆邮电大学硕士生, 主要研究方向为车联网、移动边缘计算。