

## 车载边缘计算中多任务部分卸载方案研究

王练<sup>\*②</sup> 闫润搏<sup>①②</sup> 徐静<sup>①②</sup>

<sup>①</sup>(重庆邮电大学计算机科学与技术学院 重庆 400065)

<sup>②</sup>(重庆邮电大学网络与信息安全重点实验室 重庆 400065)

**摘要:** 现有车载应用设备对时延有更严苛的要求, 车载边缘计算(VEC)能够充分利用网络边缘设备, 如路边单元(RSU)进行协作处理, 可有效地降低时延。现有研究多假设RSU计算资源充足, 可提供无限的服务, 但实际其计算资源会随着所需处理任务数量的增加而受限, 对时延敏感的车载应用造成限制。该文针对此问题, 提出一种车载边缘计算中多任务部分卸载方案, 该方案在充分利用RSU的计算资源条件下, 考虑邻近车辆的剩余可用计算资源, 以最小化总任务处理时延。首先在时延限制和资源约束下分配各任务在本地、RSU和邻近车辆的最优卸载决策变量比例, 其次以最小处理时延为目的在一跳通信范围内选择合适的空闲车辆作为处理部分任务的邻近车辆。仿真结果表明所提车载边缘计算中多任务部分卸载方案相较现有方案能较好地降低时延。

**关键词:** 车载边缘计算; 车载应用; 计算资源; 部分卸载

中图分类号: TN926

文献标识码: A

文章编号: 1009-5896(2023)03-1094-08

DOI: 10.11999/JEIT211620

## Research on Multi-task Partial Offloading Scheme in Vehicular Edge Computing

WANG Lian<sup>②</sup> YAN Runbo<sup>①②</sup> XU Jing<sup>①②</sup>

<sup>①</sup>(School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

<sup>②</sup>(Key Laboratory of Network and Information Security, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** Nowadays, the existing vehicular applications have more stringent requirements for delay. Vehicular Edge Computing (VEC) is able to take advantage of network edges devices, such as Road Side Unit (RSU), for collaborative processing, which can effectively reduce the latency. Most existing studies assume that RSU has the sufficient computing resources to provide the unlimited services. But in fact, its computing resources will be limited with the increase of the number of processing tasks, which will restrict the delay sensitive vehicular applications. To solve this problem, a multi-task partial offloading scheme in vehicular edge computing is proposed in this paper. To minimize the total task processing delay, the remaining available computing resources of adjacent vehicles is considered under the condition of making full use of RSU computing resources in this scheme. Firstly, under the constrains of delay and resource, the optimal offloading decision variable ratio of local, RSU and adjacent vehicle for each task are allocated. Secondly, in order to minimize processing delay, the appropriate spare vehicle is selected in one-hop range as adjacent vehicles to process part of the task. Simulation results show the scheme proposed can reduce the delay better compared with other schemes.

**Key words:** Vehicular Edge Computing(VEC); Vehicular applications; Computing resource; Partial offloading

收稿日期: 2021-12-31; 改回日期: 2022-09-23; 网络出版: 2022-09-27

\*通信作者: 王练 wanglian@cqupt.edu.cn

基金项目: 教育部产学合作协同育人项目(XT1911042), 重庆邮电大学科研启动基金(A2020-212)

Foundation Items: The Ministry of Education Industry-university Cooperative Education Project (XT1911042), Chongqing University of Posts and Telecommunications Research Start Fund (A2020-212)

## 1 引言

近年来随着车联网领域的发展,多样化的新型车载应用设备不断涌现,比如紧急预警、路径导航、出行娱乐等等,从而更好地为人们提供便利的乘车与驾驶服务<sup>[1]</sup>。这些新型的车载应用在提供服务的过程中会产生相应的任务数据,随着服务的增多,数据量相应增加,同时任务数据需要被快速处理,这要求车辆的车载单元(On Board Unit, OBU)需具备一定的计算能力。然而,仅凭目前车载单元中有限的计算能力无法满足大数据量的车载应用服务,尤其是时延敏感性应用<sup>[2,3]</sup>。因此,将移动边缘计算(Mobile Edge Computing, MEC)<sup>[4]</sup>引入到车联网中,所形成的车载边缘计算(Vehicular Edge Computing, VEC)<sup>[5]</sup>是一个有效的解决措施。车载边缘计算是指通过在路边单元(Road Side Unit, RSU)上配备MEC服务器可就近提供车辆所需数据处理计算服务,可满足新型车载应用的需求。

任务卸载作为VEC中的热点研究问题,得到了许多学术研究人员关注<sup>[6]</sup>。任务卸载是指将车辆上生成的任务数据卸载到附近配备MEC服务器的RSU上进行处理,以此来弥补车载单元计算能力的不足。考虑车辆的任务匹配,Liu等人<sup>[7]</sup>从匀速、车辆跟随和时间统计3种不同的速度模型中,推导出了任务卸载时延模型,并以优化网络总时延为目的提出了基于定价的1对1匹配算法和基于定价的1对多匹配算法。考虑车辆的移动,Yang等人<sup>[8]</sup>以最小化系统成本为目标,分别在独立MEC服务器和协同MEC服务器两种情况下提出了移动感知的任务卸载方案和基于位置的卸载方案;Zhang等人<sup>[9]</sup>研究在车载网络中任务卸载和资源分配的联合问题,通过优化计算和通信服务中产生的时延和成本,提出了高效的基于匹配的任务卸载和资源分配算法;王寒松<sup>[10]</sup>研究车辆移动可能导致的卸载中断问题,将任务切分成许多更小的任务单元,以最小化任务完成时延为目标提出了避免传输中断的自适应性任务卸载策略;Zhao等人<sup>[11]</sup>研究车辆移动可能造成的通信影响,将路边停靠车辆作为传输中继,并充分利用停靠车辆的剩余可用计算资源,以降低车辆通信系统的能耗和计算开销为目的,提出了多任务输入的车载边缘缓存卸载方案。考虑车辆的能耗,Li等人<sup>[12]</sup>研究车辆的高能耗问题,提出了3层云计算框架下的任务卸载服务自适应分配策略,以成本效益为目标提出了低复杂度的启发式资源分配算法。考虑车辆的时延,Liu等人<sup>[13]</sup>研究多个计算密集型车辆应用问题,将每个应用划分为多个具有任务依赖关系的小任务,然后小任务被卸载到RSU

的不同移动边缘服务器上处理,以最小化多个应用的平均完成时延为目的提出了一种高效的调度算法。Xiao等人<sup>[14]</sup>研究车辆密度应用问题,通过收集大规模私家车轨迹数据集来确定各个协作MEC服务器的角色,以最小化车辆应用时延为目的提出了一种基于非合作博弈的热感知协同MEC任务卸载策略。联合考虑车辆的时延和能耗,Luo等人<sup>[15]</sup>研究多车辆争夺计算和通信资源问题,通过设计博弈论模型,使单个车辆快速做出最优卸载决策,以最小化联合成本为目的,提出了一个基于自学习的分布式计算卸载算法。但以上研究工作几乎都未考虑到所处同一个RSU范围内拥有任务数据的车辆很多,同时RSU上配备服务器的计算资源有限的情况,此时使用现有研究方案会产生额外的时延。

本文考虑的场景是密集型城市道路,所以车辆之间的相对速度较慢,当车载应用的任务数据量很大时,会将其卸载到RSU的服务器上进行处理。如果1个RSU覆盖范围之内有多任务数据的车辆且RSU上配备MEC服务器的计算资源有限,就会导致任务数据不能够被及时处理,最终造成处理时延增加。针对以上问题本文提出一种以低时延为目标的多任务部分卸载方案,进一步考虑车辆雾网络(Vehicle Fog Network, VeFN)<sup>[16]</sup>,可将剩余未被及时处理的任務数据卸载到邻近车辆上,利用邻近车辆的剩余可用计算资源处理任务数据,使各个车辆充分利用本地、边缘网络和邻近车辆的计算资源,实现处理任务数据低时延的目的。

## 2 系统模型

### 2.1 网络结构模型

本文研究的场景是密集型车辆的城市道路,考虑1个单RSU多车辆的双向直行通路。如图1所示,RSU安装在道路上并部署VEC服务器,其中服务器的计算资源用 $F_{RSU}$ 表示,RSU的高度用 $h$ 表示,RSU的覆盖范围半径用 $r$ 表示。RSU覆盖范围内的车辆主要分为两种:一种是有任务数据要卸载的任务型车辆,另一种是有可用剩余计算资源并提供服务的服务型车辆,其中任务型车辆有 $n$ 辆车,记为 $V_i (i = 1, 2, \dots, n)$ ,服务型车辆有 $m$ 辆车,记为 $V_j (j = 1, 2, \dots, m)$ 。车辆均匀分布在道路上,并且在一定时间内保持匀速行驶。每个任务型车辆都含有一个任务,任务描述为 $I_i = \{D_i, C_i, t_i^{max}\} (i = 1, 2, \dots, n)$ ,其中 $D_i$ 表示车辆 $V_i$ 的任务数据大小, $C_i$ 表示处理车辆 $V_i$ 任务所需要的CPU周期数, $t_i^{max}$ 表示处理车辆 $V_i$ 任务可以容忍的最大时延阈值。每个服务型车辆都有可用剩余资源,用 $F_j (j = 1, 2, \dots, m)$ 表示。本文所用核心符号,如表1所示。

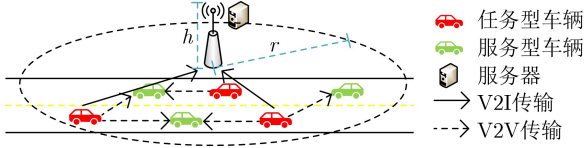


图1 网络结构模型图

表1 部分符号

符号	描述
$\alpha$	本地计算所占任务比例
$\beta$	RSU计算所占任务比例
$\gamma$	邻近车辆计算所占任务比例
$D_i$	车辆 $V_i$ 的任务数据大小
$C_i$	处理任务所需要的计算资源
$t_i^{\max}$	最大可容忍时延阈值
$t_{i,RSU}$	车辆 $V_i$ 在RSU内的最大传输时延
$t_{i,j}$	车辆 $V_i$ 在车辆 $V_j$ 覆盖范围内的可持续时延
$t_i^{\text{loc}}$	本地计算时延
$T_{i,RSU}^{\text{V2I}}$	RSU计算时延
$T_{i,j}^{\text{V2V}}$	邻近车辆计算时延
$T_i$	车辆 $V_i$ 任务处理时延
$R_i(t)$	V2I通信中信道传输速率
$d_{i,RSU}(t)$	V2I通信中随时间变化的距离变量
$t_{i,RSU}$	V2I通信中车辆 $V_i$ 在RSU内的最大传输时延
$\bar{R}_i$	V2I通信中平均传输速率
$R_{i,j}(t)$	V2V通信中信道传输速率
$d_{i,j}(t)$	V2V通信中相对距离
$t_{i,j}$	V2V通信中车辆 $V_i$ 在车辆 $V_j$ 覆盖范围内的可持续时延
$\bar{R}_{i,j}$	V2V通信中平均传输速率

## 2.2 计算模型

本方案中，任务型车辆的任务被分为3部分计算，分别是：部分任务选择在车辆本地计算，部分任务被卸载到RSU的服务器上计算，部分任务被卸载到邻近服务型车辆上计算。各部分任务的数据大小占总数据大小的决策变量分别是 $\alpha_i, \beta_i$ 和 $\gamma_i$ ，其中 $\alpha_i + \beta_i + \gamma_i = 1$ ，不同的决策变量比例会影响到总任务处理时延。

### 2.2.1 本地计算

当车辆 $V_i$ 在本地处理任务时，产生本地计算时延 $t_i^{\text{loc}}$ ，计算时延受到车辆计算能力的影响。车辆的计算能力主要来源于车载单元，不同类型的车载单元拥有不同的计算能力，用参数 $f_i^l$ 表示车辆 $V_i$ 的计算能力，因此可得到车辆 $V_i$ 的本地计算时延

$$t_i^{\text{loc}} = \frac{\alpha_i \times C_i}{f_i^l} \quad (1)$$

### 2.2.2 RSU计算

当车辆 $V_i$ 选择在RSU的服务器上处理任务时，生成的卸载时延主要分为4部分：向RSU传输数据产生的上行传输时延、任务在服务器上等待被处理的等待时延、服务器处理任务的执行时延和RSU向车辆 $V_i$ 返回结果数据的反馈时延。对于返回结果的反馈时延，由于处理完的数据远小于上行传输的数据，因此反馈时延往往忽略不计<sup>[17]</sup>，故RSU计算时延只需要计算上行传输时延、任务等待时延和执行时延。

用参数 $t_{i,RSU}^{\text{trans}}$ 表示车辆向RSU传输的上行传输时间，即

$$t_{i,RSU}^{\text{trans}} = \frac{\beta_i \times D_i}{\bar{R}_i} \quad (2)$$

车辆 $V_i$ 向RSU传输数据时，车辆 $V_i$ 必须在RSU的覆盖范围内，因此上行传输时延必须小于车辆 $V_i$ 在RSU内的最大传输时延，即

$$t_{i,RSU}^{\text{trans}} \leq t_{i,RSU} \quad (3)$$

用参数 $t_{i,RSU}^{\text{cal}}$ 表示服务器处理数据时的执行时延，即

$$t_{i,RSU}^{\text{cal}} = \frac{\beta_i \times C_i}{f_{i,RSU}} \quad (4)$$

其中，参数 $f_{i,RSU}$ 表示RSU对于车辆 $V_i$ 的计算能力。

假设RSU服务器上的计算资源用参数 $F_{RSU}$ 表示，RSU对于车辆 $V_i$ 的计算能力不得超过其计算资源，以及同一时间间隔内RSU分给每个任务型车辆的计算能力之和必须小于等于服务器上的计算资源，即

$$0 \leq f_{i,RSU} \leq F_{RSU} \quad (5)$$

$$0 \leq \sum_{i=1}^n f_{i,RSU} \leq F_{RSU} \quad (6)$$

当大量的任务型车辆需要将任务卸载到RSU上的服务器处理时，由于服务器的计算资源有限，任务数量超过服务器计算能力，一些任务需要等待，因此产生了等待时延。本方案采取先到先服务原则，根据传输时延高低分配其计算资源，传输时延低的任务优先被处理，处理完后再将其计算资源分配给未被处理的任務，因此可得到等待时延 $t_{i,RSU}^{\text{wait}}$ ，即

$$t_{i,RSU}^{\text{wait}} = t_{k,RSU}^{\text{trans}} + t_{k,RSU}^{\text{cal}} - t_{i,RSU}^{\text{trans}} \quad (7)$$

其中， $t_{k,RSU}^{\text{trans}}$ 和 $t_{k,RSU}^{\text{cal}}$ 表示车辆 $V_k$ 的传输时延和执行时延，车辆 $V_k$ 为优先被服务器处理任务的车辆。

最后，可得到RSU的计算时延 $T_{i,RSU}^{\text{V2I}}$ ，即

$$T_{i,RSU}^{\text{V2I}} = t_{i,RSU}^{\text{trans}} + t_{i,RSU}^{\text{cal}} + t_{i,RSU}^{\text{wait}} \quad (8)$$

### 2.2.3 邻近车辆计算

假设任务型车辆 $V_i$ 选择将任务卸载到服务型车辆 $V_j$ 上,产生的卸载时延由3部分组成:V2V通信的传输时延、服务型车辆处理数据的执行时延和结果反馈时延。与RSU计算同理,结果反馈时延忽略不计。

用参数 $t_{i,j}^{\text{trans}}$ 表示车辆 $V_i$ 与车辆 $V_j$ 传输数据的传输时延,即

$$t_{i,j}^{\text{trans}} = \frac{\gamma_i \times D_i}{R_{i,j}} \quad (9)$$

当车辆 $V_i$ 向 $V_j$ 传输数据时,车辆 $V_i$ 必须在 $V_j$ 的覆盖范围内才能保证数据传输,即传输时延要小于车辆 $V_i$ 在车辆 $V_j$ 覆盖范围内的可持续时间

$$t_{i,j}^{\text{trans}} \leq t_{i,j} \quad (10)$$

用参数 $t_{i,j}^{\text{cal}}$ 表示车辆 $V_j$ 处理数据的执行时延,即

$$t_{i,j}^{\text{cal}} = \frac{\gamma_i \times C_i}{f_{i,j}} \quad (11)$$

其中,参数 $f_{i,j}$ 表示车辆 $V_j$ 对于车辆 $V_i$ 的计算能力。

假设车辆 $V_j$ 剩余可用资源用参数 $F_j$ 表示,同RSU计算一样,可得

$$0 \leq f_{i,j} \leq F_j \quad (12)$$

$$0 \leq \sum_{i=1}^n f_{i,j} \leq F_j \quad (13)$$

最后,可得邻近车辆 $V_j$ 计算时延 $T_{i,j}^{\text{V2V}}$ ,即

$$T_{i,j}^{\text{V2V}} = t_{i,j}^{\text{trans}} + t_{i,j}^{\text{cal}} \quad (14)$$

### 2.3 问题公式化

由于车辆 $V_i$ 的任务被分为3部分,这3部分在同一间隔内并发处理,因此任务处理时延取决于这3部分处理的最大值。如果任务处理时延由本地计算部分决定,由于这部分并没有经过任何卸载,任务处理时延等于本地计算时延;如果任务处理时延由RSU计算时延决定,任务处理时延由传输时延、等待时延和执行时延3部分组成;如果任务处理时延由邻近车辆计算时延决定,需要由传输时延和执行时延两部分组成。综上所述,车辆 $V_i$ 任务处理时延为

$$\begin{aligned} T_i &= \max(t_i^{\text{loc}}, T_{i,\text{RSU}}^{\text{V2I}}, T_{i,j}^{\text{V2V}}) \\ &= \max\left(\frac{\alpha_i \times C_i}{f_i^{\text{l}}}, \frac{\beta_i \times D_i}{R_i} + \frac{\beta_i \times D_i}{R_i} + t_{i,\text{RSU}}^{\text{wait}}, \right. \\ &\quad \left. \frac{\gamma_i \times D_i}{R_{i,j}} + \frac{\gamma_i \times C_i}{f_{i,j}}\right) \end{aligned} \quad (15)$$

当处理 $N$ 个车辆的任务时,会生成 $N$ 个任务处理时延,因此得到总任务处理时延为

$$T = \sum_{i=1}^n T_i \quad (16)$$

本方案主要研究任务卸载方案的设计,其目的是在最大可容忍时延阈值和计算资源的约束下最小化总任务处理时延,公式化为

$$\min_{\alpha,\beta,\gamma} T = \min_{\alpha,\beta,\gamma} \sum_{i=1}^n T_i = \sum_{i=1}^n \min_{\alpha_i,\beta_i,\gamma_i} T_i \quad (17)$$

$$\text{s.t. C1: } \alpha_i + \beta_i + \gamma_i = 1, 0 \leq \alpha_i \leq 1, 0 \leq \beta_i \leq 1, 0 \leq \gamma_i \leq 1, \forall i \in \{1, 2, \dots, n\} \quad (18a)$$

$$\text{C2: } \max(t_i^{\text{loc}}, T_{i,\text{RSU}}^{\text{V2I}}, T_{i,j}^{\text{V2V}}) \leq t_i^{\text{max}}, \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\} \quad (18b)$$

$$\text{C3: } t_{i,\text{RSU}}^{\text{trans}} \leq t_{i,\text{RSU}}, \forall i \in \{1, 2, \dots, n\} \quad (18c)$$

$$\text{C4: } t_{i,j}^{\text{trans}} \leq t_{i,j}, \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\} \quad (18d)$$

$$\text{C5: } 0 \leq f_{i,\text{RSU}} \leq F_{\text{RSU}}, \forall i \in \{1, 2, \dots, n\} \quad (18e)$$

$$\text{C6: } 0 \leq \sum_{i=1}^n f_{i,\text{RSU}} \leq F_{\text{RSU}} \quad (18f)$$

$$\text{C7: } 0 \leq f_{i,j} \leq F_j, \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, m\} \quad (18g)$$

$$\text{C8: } 0 \leq \sum_{i=1}^n f_{i,j} \leq F_j, \forall j \in \{1, 2, \dots, m\} \quad (18h)$$

其中,约束C1表示 $\alpha_i, \beta_i$ 和 $\gamma_i$ 之间的关系;约束C2表示每个任务车辆处理时延不得超过最大可容忍时延阈值;约束C3,C4分别表示V2I和V2V通信的传输时延不得超过车辆在RSU和邻近车辆的停留时延;约束C5~C8表示各部分的计算能力不得超过最大可用计算资源。

## 3 多任务部分卸载方案

本方案考虑将车辆的任务分成3部分,分别在本地计算、卸载到RSU上计算和卸载到邻近车辆计算,使得处理任务的时延达到最小。首先需要合理分配好3部分任务数据的比例,其次找到合适的邻近车辆进行计算,最后根据比例计算总处理时延。

### 3.1 分配决策变量比例

对于车辆 $V_i$ ,3部分的决策变量分别用参数 $\alpha_i, \beta_i$ 和 $\gamma_i$ 表示,根据方案可得3个决策变量的分配主要是通过时延约束和速度来确定。

首先,对于 $\alpha_i$ ,本地计算时延必须小于最大可容忍时延阈值 $t_i^{\text{max}}$ ,由传输速率定义可得表达式为

$$t_i^{\text{loc}} = \frac{\alpha_i \times C_i}{f_i^{\text{l}}} \leq t_i^{\text{max}} \quad (19)$$

变换可得

$$\alpha_i \leq \frac{t_i^{\max} \times f_i^l}{C_i} \quad (20)$$

其次, 对于 $\beta_i$ , RSU的计算时延也同样要在最大可容忍时延阈值范围内完成, 同时传输时延要小于车辆 $V_i$ 在RSU下的最大传输时延 $t_{i,RSU}$ , 由 $t_{i,RSU}$ 定义公式以及式(2)—式(4)、式(7)、式(8)可得表达式为

$$t_{i,RSU}^{\text{trans}} + t_{i,RSU}^{\text{cal}} = \frac{\beta_i \times D_i}{\bar{R}_i} + \frac{\beta_i \times C_i}{f_{i,RSU}} \leq t_i^{\max} \quad (21)$$

$$t_{i,RSU}^{\text{trans}} = \frac{\beta_i \times D_i}{\bar{R}_i} \leq t_{i,RSU} \quad (22)$$

变换分别可得

$$\beta_i^1 \leq \frac{t_i^{\max}}{\left\{ \frac{D_i}{\bar{R}_i} + \frac{C_i}{f_{i,RSU}} \right\}} \quad (23)$$

$$\beta_i^2 \leq \begin{cases} \frac{(2r - l_i^0) \times \bar{R}_i}{v_i \times D_i}, & v_i > 0 \\ \frac{l_i^0 \times \bar{R}_i}{|v_i| \times D_i}, & v_i < 0 \end{cases} \quad (24)$$

最后, 对于 $\gamma_i$ , 由于是考虑将剩余未及时处理的任务卸载到邻近车辆, 因此变量 $\gamma_i$ 取决于 $\alpha_i$ 和 $\beta_i$ , 由 $\alpha_i + \beta_i + \gamma_i = 1$ 可得

$$\gamma_i^1 = 1 - \alpha_i - \beta_i \quad (25)$$

同时, 考虑到邻近车辆计算时延需要满足在最大可容忍时延阈值内, 以及V2V传输时延不得超过车辆与车辆之间的可持续时延。假设车辆 $V_j$ 作为车辆 $V_i$ 的任务卸载车辆, 因此由式(9)—式(11)、式(14)可得

$$t_{i,j}^{\text{trans}} + t_{i,j}^{\text{cal}} = \frac{\gamma_i \times D_i}{\bar{R}_{i,j}} + \frac{\gamma_i \times C_i}{f_{i,j}} \leq t_i^{\max} \quad (26)$$

$$t_{i,j}^{\text{trans}} = \frac{\gamma_i \times D_i}{\bar{R}_{i,j}} \leq t_{i,j} \quad (27)$$

变换可得

$$\gamma_i^2 \leq \frac{t_i^{\max}}{\left\{ \frac{D_i}{\bar{R}_{i,j}} + \frac{C_i}{f_{i,j}} \right\}} \quad (28)$$

$$\gamma_i^3 \leq \frac{t_{i,j} \times \bar{R}_{i,j}}{D_i} \quad (29)$$

### 3.2 选择合适邻近车辆

在V2V通信中考虑成本太高问题, 车辆的全球化信息可能无法获取, 因此车辆通信主要选择获取局部信息。车辆之间的通信受最大传输距离的限制影响, 车辆之间的距离会随着时间的推移而发生变化。因此在本方案中任务型车辆选择一跳通信范围内的空闲车辆卸载其剩余未被处理的任务数据。

在本方案中, 首先需要从服务型车辆 $V_j$ 里找到

每个任务型车辆 $V_i$ 一跳通信范围内的所有空闲车辆, 用集合 $\Gamma_{V_k}$ 表示; 其次计算每个任务型车辆选取的集合 $\Gamma_{V_k}$ 中的计算时延; 最后比较计算时延的大小, 选取计算时延最低的车辆作为要卸载的合适邻近车辆。具体过程如**算法1**所示。

### 3.3 部分任务卸载算法

在本方案中, 每个任务型车辆 $V_i$ 根据决策变量 $\alpha_i, \beta_i$ 和 $\gamma_i$ 的比例, 将其部分任务卸载到RSU服务器和符合条件的任务型车辆 $V_{i,j}$ 上, 并分别在本地、RSU、任务型车辆 $V_{i,j}$ 上处理相应的任务数据。在进行卸载与处理数据过程中会产生相应的时延, 直到车辆任务数据被处理结束, 从而得到车辆 $V_i$ 的计算时延, 如**算法2**所示, 描述了本方案卸载算法的整个过程。

对于卸载到RSU服务器上的任务被处理结束后, 如果当前车辆 $V_i$ 仍在RSU的覆盖范围之内, 数据直接传输到车辆 $V_i$ , 否则数据传送到离车辆 $V_i$ 最近的RSU服务器, 再传送到车辆 $V_i$ 。对于卸载到服务型车辆 $V_{i,j}$ 上的任务被处理结束后, 如果当前车辆 $V_i$ 和车辆 $V_{i,j}$ 之间的距离不超过车辆之间通信的最大传输距离 $r_{i,j}$ , 数据直接传送到车辆 $V_i$ , 否则数据传送到RSU服务器, 由RSU服务器传送到车辆 $V_i$ 。

算法1 选择合适邻近车辆算法

输入: 任务型车辆 $V_i$ 的相关数据参数, 服务型车辆 $V_j$ 的相关数据参数

输出: 车辆 $V_i$ 的合适邻近车辆 $V_{i,j}$

- (1) for  $V_i (i \in \{1, 2, \dots, n\})$  do
- (2) 初始化: 候选服务型车辆集合 $\Gamma_{V_k} = \emptyset$
- (3) for  $V_j (j \in \{1, 2, \dots, m\})$  do
- (4) 根据 $l_i^0$ 和 $l_j^0$ 计算 $V_i$ 与 $V_j$ 两车之间的距离:  
 $len_{i,j} = |l_i^0 - l_j^0|$
- (5) if  $len_{i,j} \leq r_{i,j}$
- (6)  $\Gamma_{V_k} = \Gamma_{V_k} + V_j$
- (7) end if
- (8) end for
- (9) for  $V_k \in \Gamma_{V_k}$  do
- (10) 初始化: 候选车辆计算时延集合 $\Phi_T = \emptyset$
- (11) 从分配决策变量比例算法获取变量 $\gamma_i^1$
- (12) 通过式(5)计算出候选车辆计算时延 $T_{i,k}$
- (13)  $\Phi_T = \Phi_T + T_{i,k}$
- (14) end for
- (15)  $V_{i,j} = \min(\Phi_T, V_k \in \Gamma_{V_k})$
- (16) return  $V_{i,j}$
- (17) end for

## 4 仿真结果

本文采用MATLAB软件进行仿真，假设仿真场景为双向直行城市道路，行驶在道路上的车辆既可以与RSU通信，又可以与车辆之间进行通信。在仿真场景中，考虑一个配备MEC服务器的RSU，

### 算法2 卸载算法

输入：任务型车辆 $V_i$ 的相关数据参数，路边单元RSU的相关数据参数，合适邻近车辆 $V_{i,j}$ ，卸载决策变量 $\alpha_i, \beta_i, \gamma_i$ ，标志位 $\text{flag}_i$

输出：总计算时延 $T$

- (1) 初始化： $T = 0$
- (2) for  $V_i (i \in \{1, 2, \dots, n\})$  do
- (3) 从分配决策变量比例算法中获取变量 $\alpha_i$
- (4) 通过式(1)计算本地计算时延 $t_i^{\text{loc}}$
- (5) 从分配决策变量比例算法中获取变量 $\beta_i$
- (6) 通过式(2)、式(4)计算RSU传输时延 $t_{i,\text{RSU}}^{\text{trans}}$ 和执行时延 $t_{i,\text{RSU}}^{\text{cal}}$
- (7) 从分配决策变量比例算法中获取变量 $\text{flag}_i$
- (8) if  $\text{flag}_i = 0$
- (9) 等待时延 $t_{i,\text{RSU}}^{\text{wait}} = 0$
- (10) 通过式(8)计算RSU计算时延 $T_{i,\text{RSU}}^{\text{V2I}}$
- (11) else if  $\text{flag}_i = 1$
- (12) 通过式(7)计算等待时延 $t_{i,\text{RSU}}^{\text{wait}}$
- (13) 通过式(8)计算RSU计算时延 $T_{i,\text{RSU}}^{\text{V2I}}$
- (14) end if
- (15) if  $T_{i,\text{RSU}}^{\text{V2I}} \leq t_{i,\text{RSU}}$  do
- (16) 处理完的结果直接发送给车辆 $V_i$
- (17) else
- (18) 处理完的结果传输到离 $V_i$ 最近的RSU服务器，然后RSU传输给 $V_i$
- (19) end if
- (20) 从分配决策变量比例算法中获取变量 $\gamma_i$
- (21) 从算法2中获取变量 $V_{i,j}$
- (22) 计算出V2V通信的平均传输速率 $\bar{R}_{i,j}$
- (23) 通过式(9)、式(11)、式(14)计算出临近车辆计算时延 $T_{i,j}^{\text{V2V}}$
- (24) 计算车辆 $V_i$ 、车辆 $V_{i,j}$ 的当前位置 $l_i^1, l_j^1$
- (25) if  $|l_i^1 - l_j^1| \leq r_{i,j}$  do
- (26) 处理完的结果直接发送给车辆
- (27) else
- (28) 处理完的数据先传输到RSU服务器，再由RSU服务器转到车辆 $V_i$
- (29)  $T_i = \max(t_i^{\text{loc}}, T_{i,\text{RSU}}^{\text{V2I}}, T_{i,j}^{\text{V2V}})$
- (30)  $T = T + T_i$
- (31) end for
- (32) return  $T$

并且任务型车辆和服务型车辆行驶在RSU附近。假设RSU的高度 $h = 20$  m，覆盖范围半径 $r = 300$  m，MEC服务器的计算资源 $F_{\text{RSU}} = 11.8$  GHz。参与计算的车辆计算能力 $f_i^l = [1, 2] \times 10^8$  cycles/s，服务型车辆剩余可用计算资源 $F_j = [1, 2.5]$  GHz，车辆之间通信的最大传输距离 $r_{i,j} = 200$  m。在道路上车辆匀速行驶，其中任务型车辆的行驶速度设定为 $v_i = 30$  km/h或者 $v_i = -30$  km/h，任务型车辆的行驶速度设定为 $v_j = [-30, 30]$  km/h，车辆在仿真时均匀分布，其车辆之间相对距离取值为 $[10, 30]$  m。本文中仿真对比方案为传统全RSU卸载方案、部分卸载方案<sup>[18]</sup>和基于排序的匹配算法(Ranking Based Matching Algorithm, RBMA)方案<sup>[9]</sup>。

如图2所示，当服务型车辆数为20，处理任务的最大可容许时延阈值为6 s时，各方案总处理时延随着任务型车辆个数增加的变化。仿真结果表明本文所提方案的总处理时延始终小于部分卸载方案。对全RSU卸载方案和本方案进行对比，当任务型车辆个数小于26时，全RSU卸载方案的总处理时延要小于本文方案，但当车辆大于26时总处理时延反而大于本文方案，这是因为在全卸载方案中当任务型车辆少的时候，卸载到RSU上的任务数据量小，配备服务器的计算资源能够及时处理数据，但随着车辆增加，配备服务器的计算资源来不及处理大量的任务数据，大量任务数据需要等待被处理而产生相应的等待时延，造成时延的快速增加；而本文方案会将处于等待的任务数据一部分卸载到邻近车辆上进行处理，能够有效地降低时延。对于RBMA方案和本文方案对比，当任务型车辆个数小于26时，RBMA方案的总处理时延要小于本文方案，但是当车辆大于26时总处理时延大于本文方案，这是由于此时RBMA方案中大量的任务数据卸载到邻近车辆，而邻近车辆车载单元的计算能力要远小于RSU的服务器的计算能力，因此造成处理时延增加；而本文方案中卸载到邻近车辆的数据处理时延需要在最大可容忍时延阈值范围之内，超过的部分任务数据卸载到

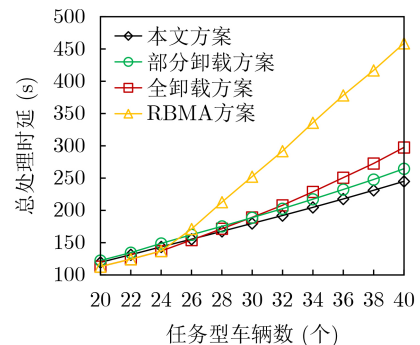


图2 不同任务型车辆数与对应的总处理时延对比

RSU等待被处理,从而减少总处理时延。因此当存在大量的任务型车辆时,本文所提方案可以明显降低任务处理时延。

如图3所示,当任务型车辆数和服务型车辆数不变,即分别为40和20时,各方案平均处理时延随着最大可容忍时延阈值的变化。仿真结果表明本文方案在时延方面明显优于部分卸载和RMBA两种方案。本文方案与部分卸载方案中平均处理时延差距逐步减小,这是由于随着最大可容忍时延阈值的增加,车辆在本地车载单元上处理的的数据逐步增加,卸载到RSU服务器和邻近车辆的任务量变小,那么任务处理时延大部分取决于本地计算时延,故两者方案差距逐步递减。对于RMBA方案中时延递减情况,主要是因为随着最大可容忍时延阈值的增加,有的车辆的任务数据在本地车载单元即可处理完成,卸载到RSU和邻近车辆的任务数据量降低,从而任务数据的总处理时延降低,相应的平均处理时延也逐渐降低。

如图4所示,在其他指标固定的情况下,任务的总处理时延随着RSU上配备服务器的计算资源增加的变化。仿真结果表明随着计算资源的增加,所有方案中的总处理时延皆在下降,这是由于随着计算资源的增加,RSU的计算能力提升,相应卸载到RSU上处理的任务量增加,因此使得任务的总处理时延降低。在所有方案中RMBA方案因为其更多地

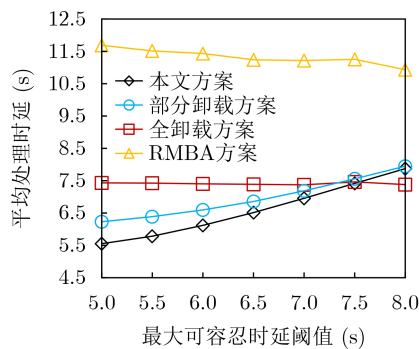


图3 不同最大可容忍时延阈值对应的车辆平均处理时延对比

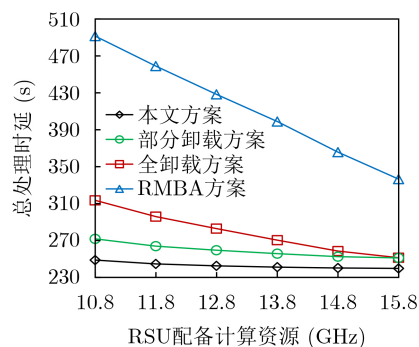


图4 不同RSU配备计算资源下的总处理时延对比

依赖于RSU上的计算资源数,随着计算资源增加,原卸载到邻近车辆的任务有部分卸载到RSU上,所以其总处理时延下降的幅度比较大。另外3种方案不仅仅依赖于RSU配备的计算资源,因此时延下降得较为缓慢。总体而言,本方案的性能仍优于另外3种方案。

## 5 结束语

本文在城市交通场景下设计车联网下的单RSU多车辆系统模型,并以此提出一种车载边缘计算中多任务部分卸载方案,解决了现有RSU上计算资源受限的问题。该方案为保障多车辆用户的低时延、高可靠的需求,设计了一种基于车载边缘计算的部分任务卸载算法。该算法通过最大可容忍时延阈值和设备计算资源的约束来获取任务决策变量的最佳卸载比例,从而优化RSU端的等待时延,并依据邻近车辆中最小计算时延指标来选取合适的空闲车辆。仿真结果表明,在多车辆任务卸载情况下,邻近车辆的计算资源减轻了RSU端的负担,使任务等待时延降低。相同条件下,本文方案相较于其他方案降低7%~17%的总时延,在任务处理时延上更优,证明了该方案的有效性。

## 参考文献

- [1] FENG Cheng and JING Weipeng. Deadline-constrained data aggregation scheduling in urban vehicular networks[C]. 19th Annual IEEE International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 2017: 1-5. doi: [10.1109/HealthCom.2017.8210827](https://doi.org/10.1109/HealthCom.2017.8210827).
- [2] MAO Yuyi, ZHANG Jun, and LETAIEF K B. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems[C]. IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, USA, 2017: 1-6. doi: [10.1109/WCNC.2017.7925615](https://doi.org/10.1109/WCNC.2017.7925615).
- [3] LIU Yongnan, GUAN Xin, PENG Yu, et al. Blockchain-based task offloading for edge computing on low-quality data via distributed learning in the internet of energy[J]. *IEEE Journal on Selected Areas in Communications*, 2022, 40(2): 657-676. doi: [10.1109/JSAC.2021.3118417](https://doi.org/10.1109/JSAC.2021.3118417).
- [4] DAI Yueyue, XU Du, MAHARJAN S, et al. Joint computation offloading and user association in multi-task mobile edge computing[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(12): 12313-12325. doi: [10.1109/TVT.2018.2876804](https://doi.org/10.1109/TVT.2018.2876804).
- [5] HUANG C M, CHIANG M S, DAO D T, et al. V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC)

- architecture[J]. *IEEE Access*, 2018, 6: 17741–17755. doi: [10.1109/ACCESS.2018.2820679](https://doi.org/10.1109/ACCESS.2018.2820679).
- [6] HOU Xueshi, LI Yong, CHEN Min, *et al.* Vehicular fog computing: A viewpoint of vehicles as the infrastructures[J]. *IEEE Transactions on Vehicular Technology*, 2016, 65(6): 3860–3873. doi: [10.1109/TVT.2016.2532863](https://doi.org/10.1109/TVT.2016.2532863).
- [7] LIU Pengju, LI Junluo, and SUN Zhongwei. Matching-based task offloading for vehicular edge computing[J]. *IEEE Access*, 2019, 7: 27628–27640. doi: [10.1109/ACCESS.2019.2896000](https://doi.org/10.1109/ACCESS.2019.2896000).
- [8] YANG Chao, LIU Yi, CHEN Xin, *et al.* Efficient mobility-aware task offloading for vehicular edge computing networks[J]. *IEEE Access*, 2019, 7: 26652–26664. doi: [10.1109/ACCESS.2019.2900530](https://doi.org/10.1109/ACCESS.2019.2900530).
- [9] ZHANG Yifan, QIN Xiaoqi, and SONG Xianxin. Mobility-aware cooperative task offloading and resource allocation in vehicular edge computing[C]. *IEEE Wireless Communications and Networking Conference Workshops*, Seoul, Korea (South), 2020: 1–6. doi: [10.1109/WCNCW48565.2020.9124825](https://doi.org/10.1109/WCNCW48565.2020.9124825).
- [10] 王寒松. 车联网中基于MEC的计算任务卸载策略研究[D]. [硕士学位论文], 北京邮电大学, 2019.  
WANG Hansong. Research of computing offloading scheme for MEC-enabled vehicular networks[D]. [Master dissertation], Beijing University of Posts and Telecommunications, 2019.
- [11] ZHAO Junhui, DONG Peiyang, MA Xiaoting, *et al.* Mobile-aware and relay-assisted partial offloading scheme based on parked vehicles in B5G vehicular networks[J]. *Physical Communication*, 2020, 42: 101163. doi: [10.1016/j.phycom.2020.101163](https://doi.org/10.1016/j.phycom.2020.101163).
- [12] LI Xin, DANG Yifan, AAZAM M, *et al.* Energy-efficient computation offloading in vehicular edge cloud computing[J]. *IEEE Access*, 2020, 8: 37632–37644. doi: [10.1109/ACCESS.2020.2975310](https://doi.org/10.1109/ACCESS.2020.2975310).
- [13] LIU Yujiong, WANG Shangguang, ZHAO Qinglin, *et al.* Dependency-aware task scheduling in vehicular edge computing[J]. *IEEE Internet of Things Journal*, 2020, 7(6): 4961–4971. doi: [10.1109/JIOT.2020.2972041](https://doi.org/10.1109/JIOT.2020.2972041).
- [14] XIAO Zhu, DAI Xingxia, JIANG Hongbo, *et al.* Vehicular task offloading via heat-aware MEC cooperation using game-theoretic method[J]. *IEEE Internet of Things Journal*, 2020, 7(3): 2038–2052. doi: [10.1109/JIOT.2019.2960631](https://doi.org/10.1109/JIOT.2019.2960631).
- [15] LUO Qiyuan, LI Changle, LUAN T H, *et al.* Self-learning based computation offloading for internet of vehicles: Model and algorithm[J]. *IEEE Transactions on Wireless Communications*, 2021, 20(9): 5913–5925. doi: [10.1109/TWC.2021.3071248](https://doi.org/10.1109/TWC.2021.3071248).
- [16] LI Haotian, LI Xujie, ZHANG Mingyue, *et al.* Multicast-oriented task offloading for vehicle edge computing[J]. *IEEE Access*, 2020, 8: 187373–187383. doi: [10.1109/ACCESS.2020.3030943](https://doi.org/10.1109/ACCESS.2020.3030943).
- [17] WANG Yanting, SHENG Min, WANG Xijun, *et al.* Mobile-edge computing: Partial computation offloading using dynamic voltage scaling[J]. *IEEE Transactions on Communications*, 2016, 64(10): 4268–4282. doi: [10.1109/TCOMM.2016.2599530](https://doi.org/10.1109/TCOMM.2016.2599530).
- [18] REN Jinke, YU Guanding, CAI Yunlong, *et al.* Partial offloading for latency minimization in mobile-edge computing[C]. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017: 1–6. doi: [10.1109/GLOCOM.2017.8254550](https://doi.org/10.1109/GLOCOM.2017.8254550).

王 练：女，博士，教授，研究方向为无线可靠传输。

闫润搏：女，硕士生，研究方向为车联网计算卸载。

徐 静：女，硕士生，研究方向为无线可靠传输。

责任编辑：余 蓉