

分类号	<u>TN929.5</u>	密级	<u>公开</u>
UDC	<u>621.39</u>	学位论文编号	<u>D-10617-30852-(2021)-01006</u>

重庆邮电大学硕士学位论文

中文题目	<u>基于 MEC 的车联网任务卸载与迁移</u> <u>机制研究</u>
英文题目	<u>Research on MEC-based Task</u> <u>Offloading and Migration Mechanism</u> <u>in Internet of Vehicles</u>
学 号	<u>S180131004</u>
姓 名	<u>张益峰</u>
学位类别	<u>工程硕士</u>
学科专业	<u>电子与通信工程</u>
指导教师	<u>张海波 副教授</u>
完成日期	<u>2021 年 3 月 25 日</u>

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含他人已经发表或撰写过的研究成果，也不包含为获得重庆邮电大学或其他单位的学位或证书而使用过的材料。与我一同工作的人员对本文研究做出的贡献均已在论文中作了明确的说明并致以谢意。

作者签名：张益峰

日期：2021 年 6 月 4 日

学位论文版权使用授权书

本人完全了解重庆邮电大学有权保留、使用学位论文纸质版和电子版的规定，即学校有权向国家有关部门或机构送交论文，允许论文被查阅和借阅等。本人授权重庆邮电大学可以公布本学位论文的全部或部分内容，可编入有关数据库或信息系统进行检索、分析或评价，可以采用影印、缩印、扫描或拷贝等复制手段保存、汇编本学位论文。

(注：保密的学位论文在解密后适用本授权书。)

作者签名：张益峰

日期：2021 年 6 月 4 日

导师签名：张益峰

日期：2021 年 6 月 4 日

摘要

随着车联网的发展,各种车辆应用需要更低的任务处理时延。车辆本身计算能力有限,为此在车联网中引入移动边缘计算(Mobile Edge Computing, MEC)技术。云服务被转移到无线接入网络的边缘,用户可以将计算密集型任务卸载到网络边缘,缓解本地的计算压力。另外,为了满足新兴应用的需要,车联网体系结构还必须具备灵活性、可编程性、可靠性和可伸缩性,因此将其与软件定义网络(Software Defined Network, SDN)架构结合。然而在基于 MEC 的软件定义车联网系统中,依然存在着任务处理时延较高的问题。因此,本文着重研究了用户任务卸载与迁移时延优化问题。主要贡献如下:

1. 研究了基于 MEC 的软件定义车联网任务迁移时延优化问题。由于车辆用户的移动,在任务卸载过程中,未完成的计算任务需要从原始路边单元迁移到目标路边单元,由此产生虚拟机(Virtual Machine, VM)迁移问题。将任务迁移时延优化问题建模为一个混合整数非线性问题,利用两阶段算法,将原始优化问题化解为两个子优化问题。利用 SDN 的深度可编程性来动态地重新配置 VM 迁移路由,通过 Dijkstra 路由算法找到本次任务迁移的最佳路径,然后利用 Q-learning 算法来优化本次任务的迁移时延,从而减少车辆用户移动时的总任务时延,改善用户服务体验。仿真结果表明,本方案可以有效降低任务迁移时延。

2. 研究了基于 NOMA-MEC 的车联网任务卸载、缓存和迁移时延优化问题。在 MEC 技术与非正交多址访问(Non-orthogonal Multiple Access, NOMA)技术相结合的车联网系统中,针对用户处理计算密集型和时延敏感型任务时面临的高时延问题,提出了一种基于博弈论和 Q-learning 的任务卸载、缓存与迁移优化策略。首先,对基于 NOMA-MEC 的车联网任务卸载时延、缓存时延与迁移时延进行建模;其次,采用合作博弈模型获得最优用户分组以实现卸载时延优化;最后,通过 Q-learning 算法优化用户分组中的缓存迁移联合时延。仿真结果表明,本方案可以有效提高卸载效率,降低任务处理时延。

关键词: 移动边缘计算, 车联网, 任务卸载, 任务迁移

Abstract

With the development of the Internet of Vehicles, various vehicle applications require lower task processing latency. The computing capability of vehicles is limited, so Mobile Edge Computing (MEC) technology is introduced into the Internet of Vehicles. Cloud services are transferred to the edge of the wireless access network, and users can offload computing-intensive tasks to the edge of the network in order to relieve the pressure of local computing. In addition, to satisfy the needs of emerging applications, the vehicular network architecture must provide more services with flexibility, programmability, reliability, and scalability. Thus, MEC is combined with the Software Defined Network (SDN) architecture to improve the system performance. However, in the MEC-based software-defined vehicular network, there still exists the problem of high task processing delay. Therefore, this thesis discusses the user task offloading and migration delay optimization problem. The main contributions are presented as follows:

1. The delay optimization problem of task migration of software-defined vehicular network based on MEC is studied. Due to the movement of vehicles, the undone computing tasks need to be migrated from the original roadside unit to the target roadside unit, leading to the migration problem of virtual machines. Model the task migration delay optimization problem as a mixed integer nonlinear problem, and the original optimization problem is divided into two sub-optimization problems through two-stage algorithm. Utilize the deep programmability of SDN to reconfigure virtual machine migration routes dynamically. The Dijkstra routing algorithm is employed to find the best path for this task migration. Then, the Q-learning algorithm is adopted to optimize the migration delay of the current task, which consequently reduces the total task delay when the vehicle users are moving and enhances the user service experience. The simulation results demonstrate that this scheme can effectively lower the user's service delay.

2. The optimization of task offloading, caching and migration delay of the Internet of Vehicles based on NOMA-MEC is studied. In the Internet of Vehicles that combining MEC and NOMA technology, addressing at the high latency problems faced by users when processing computationally intensive and latency-sensitive tasks, an optimization strategy of task offloading, migration and cache based on game theory and Q learning is proposed. First, the offloading delay, migration delay and cache delay of the Internet of Vehicles task

are constructed based on NOMA-MEC. Then, the cooperative game model is applied to obtain the optimal user grouping to optimize the offloading delay. Finally, in order to avoid local optima, the Q learning algorithm is involved to optimize the joint delay of the migration cache in the user group. The simulation results indicate that the proposed scheme can effectively improve the offloading efficiency and reduce the task processing delay.

Keywords: mobile edge computing, vehicular network, task offloading, task migration

目录

图录	VII
表录	IX
注释表	X
第 1 章 绪论	1
1.1 研究背景及意义	1
1.1.1 车联网研究背景及意义	1
1.1.2 MEC 技术研究背景及意义	2
1.2 国内外研究现状	3
1.2.1 车联网研究现状	3
1.2.2 任务卸载研究现状	4
1.2.3 任务迁移研究现状	5
1.3 本文主要工作	6
1.4 本文组织结构	7
第 2 章 车联网相关技术概述	9
2.1 车联网概述	9
2.2 MEC 技术及其应用	11
2.2.1 MEC 技术概述	11
2.2.2 计算卸载	12
2.2.3 边缘缓存	13
2.2.4 任务迁移	14
2.3 软件定义车联网技术概述	15
2.3.1 软件定义车联网相关概念	15
2.3.2 软件定义车联网发展方向	17
2.4 NOMA 技术概述	17
2.4.1 NOMA 技术简介	17
2.4.2 NOMA 与 MEC 的结合	18

2.4.3 MEC-NOMA 系统关键问题	18
2.5 本章小结	19
第 3 章 基于 MEC 的软件定义车联网任务迁移策略	20
3.1 引言	20
3.2 系统模型	21
3.2.1 系统分析	21
3.2.2 任务模型	22
3.2.3 任务卸载、计算与迁移模型	23
3.2.4 问题建模	24
3.3 任务迁移优化算法	25
3.3.1 基于 Dijkstra 的最短迁移路径选择算法	26
3.3.2 基于 Q-learning 的任务迁移优化算法	27
3.3.3 复杂度分析	31
3.4 仿真分析	31
3.5 本章小结	34
第 4 章 基于 NOMA-MEC 的车联网任务卸载、缓存与迁移策略	35
4.1 引言	35
4.2 系统模型	36
4.2.1 网络模型	36
4.2.2 传输模型与卸载模型	37
4.2.3 迁移模型与缓存模型	38
4.2.4 问题建模	39
4.3 车联网任务卸载、缓存与迁移优化算法	40
4.3.1 基于 NOMA 的合作博弈卸载决策算法	41
4.3.2 基于 Q-learning 的任务缓存与迁移优化算法	43
4.3.3 时间复杂度	46
4.4 仿真分析	46
4.5 本章小结	50
第 5 章 总结与展望	51

5.1 全文工作总结	51
5.2 研究工作展望	51
参考文献	53
致谢	59
攻读硕士学位期间从事的科研工作及取得的成果	60

图录

图 1.1 基于 MEC 的车联网模型图	3
图 1.2 论文组织架构	7
图 2.1 车联网的五种通信模式示意图	9
图 2.2 缓存协同处理示意图	13
图 2.3 任务迁移示意图	15
图 2.4 SDN 基本架构图	15
图 2.5 软件定义车联网模型图	16
图 3.1 任务迁移优化策略流程框图	21
图 3.2 系统模型图	21
图 3.3 任务迁移优化算法流程框图	25
图 3.4 基于 Dijkstra 的最短迁移路径选择算法流程图	26
图 3.5 基站分布数学模型图	26
图 3.6 Q-learning 状态过程转化图	29
图 3.7 基于 Q-learning 的任务迁移优化算法流程图	30
图 3.8 任务不迁移与任务迁移时延对比图	32
图 3.9 不同车辆数下平均时延对比图	32
图 3.10 迁移时间对比图	33
图 3.11 迁移百分比对比图	33
图 3.12 迁移失败百分比对比图	34
图 4.1 车联网任务卸载、缓存与迁移策略流程框图	36
图 4.2 系统模型图	36
图 4.3 基于 NOMA-MEC 的车联网任务缓存与迁移策略流程框图	40
图 4.4 基于合作博弈算法的卸载决策流程图	43
图 4.5 迭代次数与平均时延对比图	47
图 4.6 任务数据大小与平均时延对比图	48
图 4.7 数据大小与平均时延关系图	48
图 4.8 缓存内容总数与用户平均时延关系图	49

图 4.9 任务数据大小与卸载时延对比图	49
图 4.10 车辆用户数量与总体时延对比图	50

表录

表 3.1 基于 Dijkstra 的最短迁移路径选择算法..... 27

表 3.2 基于 Q-learning 的任务迁移优化算法 30

表 3.3 仿真参数表 31

表 4.1 基于合作博弈算法的卸载决策 42

表 4.2 基于 Q-learning 的任务缓存与迁移优化算法 45

表 4.3 仿真参数表 46

注释表

AU	Application Unit, 应用单元
BS	Base Station, 基站
CDMA	Code Division Multiple Access, 扩频码分多址
CSI	Channel State Information, 全信道状态信息
CPU	Central Processing Unit, 中央处理器
C-RAN	Centralized Radio Access Network, 集中式无线接入网
D2D	Device to Device, 设备对设备通信
DSRC	Dedicated Short Range Communication, 专用短距离通信技术
FCC	Federal Communications Commission, 美国联邦通信委员会
GA	Greedy Algorithm, 贪婪算法
GPS	Global Positioning System, 全球定位系统
ICN	Information Centric Networking, 信息中心网络
ITS	Intelligent Transport Systems, 智能交通系统
IoV	Internet of Vehicles, 车辆互联网
MCC	Mobile Cloud Computing, 移动云计算
MEC	Mobile Edge Computing, 移动边缘计算
NOMA	Non-orthogonal Multiple-Access, 非正交多址访问
NFV	Network Functions Virtualization, 网络功能虚拟化
OBU	On Board Units, 车载单元
OMA	Orthogonal Multiple-Access, 正交多址访问
OFDMA	Orthogonal Frequency Division Multiple Access, 正交频分多址
PLS	Physical layer Security, 物理层安全性
QoS	Quality of Service, 服务质量
RAN	Radio Access Network, 无线接入网
RSU	Roadside Units, 路边单元
R2V	Roadside to Vehicle, 路边智能设备对车辆通信

SDN	Software Defined Network, 软件定义网络
SFC	Service Function Chains, 服务功能链
SIC	Successive Interference Cancellation, 连续干扰消除
V2V	Vehicle to Vehicle, 车辆对车辆通信
V2I	Vehicle to Infrastructure, 车辆对基础设施通信
V2P	Vehicle to Pedestrian, 车辆对行人通信
V2X	Vehicle to Everything, 车对万物通信
VM	Virtual Machine, 虚拟机
VANET	Vehicular Ad hoc Networks, 车辆自组网
V2R	Vehicle to Roadside Unit, 车对路边单元通信
V2S	Vehicle to Sensors, 车对传感器通信
WAT	Wireless Access Technology, 无线接入技术
WHO	World Health Organization, 世界卫生组织

第 1 章 绪论

1.1 研究背景及意义

1.1.1 车联网研究背景及意义

在信息时代,机动车辆从简单的机械设备发展成具有不同属性的智能传感器主体,技术的进步增强了车辆安全性和用户驾驶体验。但是,在交通运输系统中,仍然存在着一些亟待解决的问题。例如,城市车辆流量的急速增加^[1],导致车辆拥堵与碰撞等交通问题大幅增加,进而影响社会经济。良好的车辆通信能力,在提高交通安全等方面具有巨大作用^[2]。因此,汽车公司、科研组织和相关部门都在致力于实现车到万物通信(Vehicle to Everything, V2X)技术, V2X 包括车辆对车辆通信(Vehicle to Vehicle, V2V)、车辆对基础设施通信(Vehicle to Infrastructure, V2I)和车辆对行人通信(Vehicle to Pedestrian, V2P)^[3]。

支持 V2X 通信的技术通常有两种:专用短距离通信技术(Dedicated Short Range Communication, DSRC)和蜂窝网络技术。

DSRC 技术可实现 DSRC 设备之间的短程信息交换,通常应用于智能交通系统(Intelligent Transport Systems, ITS)。常用的 DSRC 设备有车载单元(On Board Units, OBU)、路边单元(Roadside Units, RSU)和行人携带的手持设备等。为了促进 DSRC 技术的发展,美国联邦通信委员会(Federal Communications Commission, FCC)为基于 DSRC 的应用分配专用的无线电频谱波段。然而,在车辆密度较高的场景下,DSRC 支持的 V2X 通信性能较差。此外,预计未来车载互联网数据流量会迅速增加,分配的 DSRC 无线电频谱将无法满足应用需求。由上述介绍可知,DSRC 技术存在一定的局限性。基于此,将具有高容量和高覆盖范围的蜂窝网络技术应用于 V2X 通信,凭借其广泛部署的基础设施,在未来可以最大程度上适用于 V2X 体系架构。但是,蜂窝网络会导致 V2X 通信时延较长,这会对应用程序造成影响。此外,蜂窝网络容量能否满足其蜂窝用户不断增加的数据流量也尚待研究。因此,研究团体一直致力于研究 DSRC 技术与蜂窝技术相结合的 V2X 通信架构,以实现 V2X 通信的可靠性和有效性,进而减少交通拥堵与碰撞问题,保障车辆行驶安全与用户驾驶体验。

1.1.2 MEC 技术研究背景及意义

思科白皮书报告显示, 2017 年至 2022 年, 全球数据流量将以 26% 的复合年均增长率增长, 到 2022 年将达到每月 122 EB^[4]。此外, 应用服务产生的流量将以更高的增长率增长。然而, 大多数设备(如智能汽车)的处理能力有限, 不能满足许多应用程序的计算需求。这些应用程序会生成、处理和存储大量数据, 并依赖大量计算资源, 解决该问题的一个可行技术是移动边缘计算(Mobile Edge Computing, MEC)技术。

在 MEC 之前, 已经有一些类似的计算概念, 例如, 移动云计算(Mobile Cloud Computing, MCC)、朵云和雾计算。MCC 结合了云计算、移动计算和无线通信网络等技术, 将计算功能和数据存储功能从移动设备转移到云中, 从而使服务提供商能够提供更复杂的应用服务^[5]。但是, MCC 的低伸缩性、高延迟以及带宽有限是其无法回避的缺点。朵云是 Satyanarayanan 等人在 2009 年提出的边缘计算概念^[6], 指的是资源丰富的受信任计算机或位于网络边缘位置并与互联网保持良好连接的计算机集群。朵云将云计算转移到网络边缘, 当用户进行移动性管理时, 若用户资源受限, 则朵云可以为其提供服务。但是, 用户无法长距离访问朵云, 也无法同时使用 WiFi 和蜂窝连接^[7]。雾计算将云计算从核心网扩展到网络边缘, 减少了中央云的数据量^[8]。其在许多应用中扮演着重要角色^[9], 如智能城市、联网车辆、智能电网、无线传感器、执行器网络以及智能建筑等应用。但是, 雾节点难以实现自我管理, 并且需要中央云的支持。朵云和雾计算的相似之处在于都没有集成到移动网络体系结构中, 因此其通常由私有企业部署, 很难为私人用户保证服务质量(Quality of Service, QoS)^[10, 11]。

2014 年底, 欧洲电信标准协会 MEC 行业规范小组提出 MEC 概念。作为集中式无线接入网(Centralized Radio Access Network, C-RAN)架构补充, MEC 可以在移动用户附近的无线接入网络(Radio Access Network, RAN)内为其提供云计算功能^[12]。为了使 5G、WiFi 和固定连接等异构接入技术可以与 MEC 更好地结合, MEC 行业规范小组在 2017 年正式将 MEC 的名称更改为多边缘访问计算^[13]。在此范围扩展之后, 网络运营商可以将 MEC 服务器部署在 RAN 的任意位置, 例如基站(Base Station,

BS)、光网络单元和 WiFi 接入点等位置。这不仅可以更好地促进通信功能，还可以更好地为用户提供计算、缓存和控制服务。

典型的基于 MEC 的车联网架构由支持 V2X 的车辆、RSU 和 BS 组成，如图 1.1 所示。RSU 为车辆提供协助和连接服务，BS 作为城市蜂窝系统基础设施的一部分，除了服务车辆外，还可服务于其他用户。服务器具有 V2X 应用程序所需功能，如中央处理器 (Central Processing Unit, CPU)、内存、存储等。这些服务器部署在 RSU、BS、运营商核心网络以及互联网上。车辆和各种 RSU 装备传感设备，如雷达、摄像头、全球定位系统(Global Positioning System, GPS)等，可以收集关于当前道路状况信息并发送到 RSU。为了控制车辆信息（速度、调整、超车、变道等），RSU 上的 V2X 应用程序可以将收集到的信息作为任务输入参数生成任务。MEC 服务器部署在 RSU 上，其计算和存储能力远大于车载单元，当车辆用户将任务卸载至 MEC 服务器时，可以降低任务时延，提升用户体验。

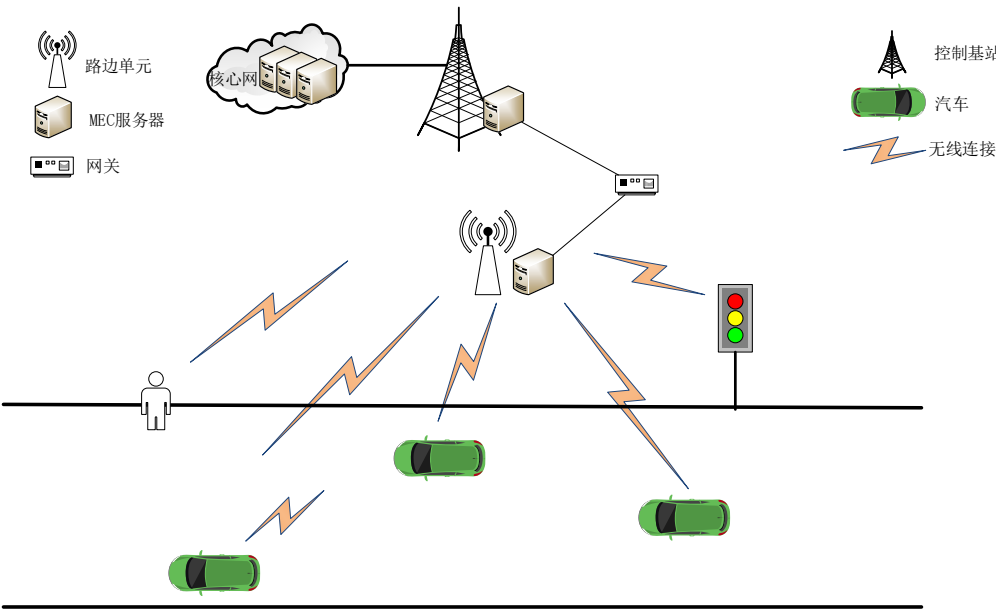


图 1.1 基于 MEC 的车联网模型图

1.2 国内外研究现状

1.2.1 车联网研究现状

车辆自组网(Vehicular Ad Hoc Networks, VANET)支持车辆通信技术，允许部署 V2V 服务^[14]。V2V 通信通常使用基于 IEEE 802.11p 的 DSRC 技术在车辆之间交换

消息^[15]。然而, 基于 IEEE 802.11p 的分布式信道可能会产生包碰撞, 因此其可靠性较低^[16]。蜂窝车联网通信技术可以提高传播道路安全消息的可靠性, 但是随着道路交通流量的增加, DSRC 和蜂窝车联网的性能都会急剧下降^[17]。所以车联网与软件定义网络(Software Defined Network, SDN)等新兴技术的结合是大势所趋。

SDN 和 MEC 是实现新一代车载网络的关键技术^[18]。SDN 将控制平面和数据平面分开, 在通用硬件上控制平面软件。SDN 控制器保留网络全局视图, 并可以传达在应用层实现的指导策略。此外, MEC 允许 MEC 服务器与不同的网络元素并置, 例如宏基站、小型基站、路由器、交换机和光学单元等^[19]。SDN 控制器在边缘计算处的分布式拓扑结构减轻了全局的计算压力^[20]。因此, 基于 SDN-MEC 的车联网体系结构可以降低网络延迟以及提高网络资源利用率^[21-23]。

基于 SDN-MEC 的车载网络研究引起了汽车厂商、移动运营商和学术界的关注。文献[18]提出一种基于 SDN 的车载自组织网络架构, 可以控制无线资源的分配, 提高传输效率。文献[24]提出一种 MEC 支持的软件定义车载网络, 在此架构中 V2X 通信速率稳定, 提高了系统的稳定性。文献[25]提出一种 5G-V2X 架构, 该架构使用 Side Link 替代 DSRC 进行 V2V 通信, 可以更加有效地传播车载数据。文献[26]提出一种基于三种协作通信的协作代码传播方案, 包括路边设备对车辆通信(Roadside to Vehicle, R2V)、V2V 通信和 R2V-V2V 组合通信, 降低了整体计算成本。

然而, 上述工作重点是优化基于 SDN-MEC 的车载网络中的资源分配方案以及降低计算成本。事实上, 随着车辆流量急速地增加, V2X 的数据传播方案的延迟性也是一个不可忽视的问题^[27]。因此, 本文工作的主要目标是降低任务延迟。

1.2.2 任务卸载研究现状

近年来, ITS 发展迅速, 该系统可提高驾驶员的体验质量以及减少事故数量。V2X 应用程序在 ITS 中起到关键作用, 大部分的程序需要在非常短的时间内执行数十亿次计算操作。因此, 为了同时满足延迟和计算要求, 欧洲电信标准协会引入了 MEC 概念。根据该概念, 计算服务被移动到无线电接入网边缘, 自身计算能力有限的车辆可以将任务卸载至 MEC 服务器处进行计算。目前的车联网架构限制了任务卸载决策的灵活性, 为了解决该问题, 研究人员基于 SDN 架构提出任务卸载方案, SDN 控制器可以有效地管理网络并且提供任务卸载决策。文献[28]提出一种基于

SDN 的 MEC 框架, 该框架从无线网络中收集网络信息, 例如拓扑结构、带宽和信号强度等信息, 提高了整体计算卸载效率。文献[29]将任务卸载和资源分配作为联合优化问题, 最大程度地提高了总体服务效率。文献[30]利用集中式 SDN 控制器管理通信车辆之间的端到端连接, 提出一种 V2V 数据卸载方案, 提高了卸载效率。文献[31]提出了一种基于 SDN 的 MEC 框架, 可提高网络扩展性。文献[32]提出一种基于 SDN 的任务卸载方案, 可以最大程度减少任务失败率。

边缘计算中的任务卸载也是研究热点。文献[33]提出一种基于雾计算的节能卸载决策, 以确保实时任务的 QoS 并最小化设备的能耗。文献[34]分析了任务转移到雾节点的位置, 最大程度地减少了移动设备的功耗。文献[35]提出基于优先级调度的计算卸载算法, 在 MEC 中引入了优先级卸载模型, 提高了卸载效率。文献[36]提出一种延迟感知视频边缘分析系统, 提高了 MEC 计算任务的卸载效率。文献[37]提出了一种联合计算卸载方法, 该方法根据可用的计算资源, 可以将任务卸载到基站, 并且启用无人飞行器充当移动雾节点进行计算, 减少了阻塞概率, 并最大程度地减少了用户设备和无人机的能耗。文献[38]在 MEC 中提出了一种状态行动奖励算法, 降低了系统成本。文献[39]提出了一种近端网络解决方案, 该解决方案通过使用基于深度 Q-learning 的代码分载方法在 MEC 中进行计算分载, 提高了在大量卸载请求状况下的卸载效率。

上述工作主要研究任务卸载过程中的设备能耗问题和服务效率问题, 但是若卸载时间过长, 可能会对某些时延敏感型应用造成不利的影响。因此, 本文对卸载时间进行进一步优化。

1.2.3 任务迁移研究现状

在车联网中, 任务迁移通过虚拟机(Virtual Machine, VM)迁移技术实现。目前关于 VM 迁移的研究考虑了资源优化问题。文献[40]根据动态网络流量数据对 VM 进行群集。将频繁通信的 VM 放到同一位置, 或尽可能近的位置, 以此来减少位置之间的流量。文献[41]提出基于粒子群优化的 VM 分配方法, 优化了应用程序的执行效率和数据中心节点之间的数据传输效率。文献[42, 43]提出 VM 放置和迁移优化策略, 可选择出位于计算节点和存储节点之间的最佳节点, 提高数据传输效率。文献[44]提出基于程序密集型的 VM 迁移, 对目标文件仍驻留在同一存储节点中的 VM

重新分配,来最大程度提高文件访问效率。文献[45]研究了 VM 的流量感知功能,在计算节点上分配高流量的 VM 来降低总通信成本。然而,这些方法的局限性在于没有考虑 CPU 属性、CPU 完成应用程序速度以及队列等待时间,这可能导致应用程序性能下降。

关于 VM 迁移的安全性研究也是重点。云计算服务的动态性质,使攻击者能够增加攻击面,使漏洞管理复杂化,事件响应时间呈指数增长^[46]。文献[47]提出一种虚拟防火墙系统,旨在保护 VM 免受虚拟网络的侵害。文献[48]为 VM 建立共享内存通道,可以降低入侵概率。文献[49]认为通过监视缓存使用情况,攻击者可以确定目标 VM。一方面,通过监视缓存和行为分析可以避免跨 VM 的信息泄漏^[50]。于是,相关文献旨在减少与缓存监视相关的漏洞。文献[51]认为数据中心虚拟化和中间件共享可以有效地减少缓存监视漏洞,但是该方案共享资源利用效率较低。另一方面,采用 VM 监控策略,可以隔离监控系统的攻击^[52]。因此,文献[53]提出虚拟化的入侵容忍系统方案,在具有安全漏洞的 VM 监控环境下,提高了 VM 请求的成功概率。

上述工作旨在研究 VM 迁移的资源分配问题以及安全性问题。但是在车辆密集等场景下,若迁移时间过长,会造成数据堵塞、传输效率低下等问题。因此,本文对 VM 迁移时间优化进行研究。

1.3 本文主要工作

本文研究工作来自于重庆市基础研究与前沿探索项目。在该项目的科研指导方针下,主要对基于 MEC 的车联网中任务卸载与任务迁移问题进行探究。详细工作如下:

1. 构建软件定义车联网络场景,研究用户在车辆移动期间任务迁移问题。由于车辆用户移动,当任务从原始路边单元迁移到目标路边单元时,会产生 VM 迁移问题。将任务时延优化问题建模为一个混合整数非线性问题。首先通过 Dijkstra 算法在城市基站分布模型中找到最优迁移路径,之后通过 Q-learning 算法求得最优路径上的任务迁移时延。

2. 构建基于 NOMA-MEC 的车联网环境,对多个车辆用户的任务卸载、缓存与迁移时延进行优化。具体来讲,将任务时延分为两部分,任务卸载时延与任务缓存迁移联合时延。通过合作博弈算法对车辆用户进行分区,以达到卸载时延的优化。

利用 Q-learning 算法对状态空间和动作空间进行建模，获得优化后的缓存迁移联合时延。

1.4 本文组织结构

各章节之间的联系如下图 1.2 所示。



图 1.2 论文组织架构

本文主要研究车联网中基于 MEC 的任务卸载与迁移方案,全文结构安排如下:

第 1 章: 绪论, 首先阐述课题研究的背景意义, 其次对车联网研究现状与任务卸载、迁移研究现状进行总结, 最后简要说明本文的主要工作内容和组织结构。

第 2 章: 车联网相关技术概述, 该章节详细介绍了车联网相关技术与原理, 并对 MEC 技术与车联网相关技术的结合进行多维展开, 探讨其工作的未来发展趋势。

第 3 章: 提出一种车联网任务迁移优化方案, 在该方案下, 首先建立数学模型, 提出一个两阶段算法。第一阶段通过 Dijkstra 算法在城市基站分布模型中找到最优迁移路径, 第二阶段通过 Q-learning 算法优化第一阶段求得最优路径上的任务迁移时间, 从而优化任务时延。

第 4 章：提出一种基于 NOMA-MEC 车联网任务卸载、缓存与迁移优化方案。将任务时延分为两部分，卸载时延与缓存迁移联合时延。通过车辆用户的合作博弈建模，对车辆用户进行分区，以优化卸载时延。利用 Q-learning 算法对缓存迁移联合时延进行优化，获得最优的时延奖励。

第 5 章：总结与展望，对全文工作进行了总结，反思目前工作的不足，并对未来的研究方向和内容进行了展望。

第 2 章 车联网相关技术概述

2.1 车联网概述

无线通信领域的技术进步，促进了智能运输系统中车辆通信技术的发展。智能车辆中配有支持无线接入技术(Wireless Access Technology, WAT)的设备，例如显示屏、传感器、GPS 接收器、应用单元(Application Unit, AU)、CPU 等。这些设备可以实现 V2V 和车对路边单元通信(Vehicle to Roadside, V2R)，V2V 和 V2R 统称为 VANET。

不同于 VANET，将互联网和其它异构网络相结合的车辆网络称为车辆互联网(Internet of Vehicles, IoV)。如图 2.1 所示，IoV 的异构网络体系结构包括五种类型的车辆通信模式，包括 V2V、V2R、V2I、V2P 和车对传感器通信(Vehicle to Sensors, V2S)。IoV 通信体系结构不仅包含车辆和 RSU，还包括其它通信设备。IoV 的异构车辆网络框架可以更好地监督和指导车辆，并且为多媒体应用程序提供了可靠的通信平台。

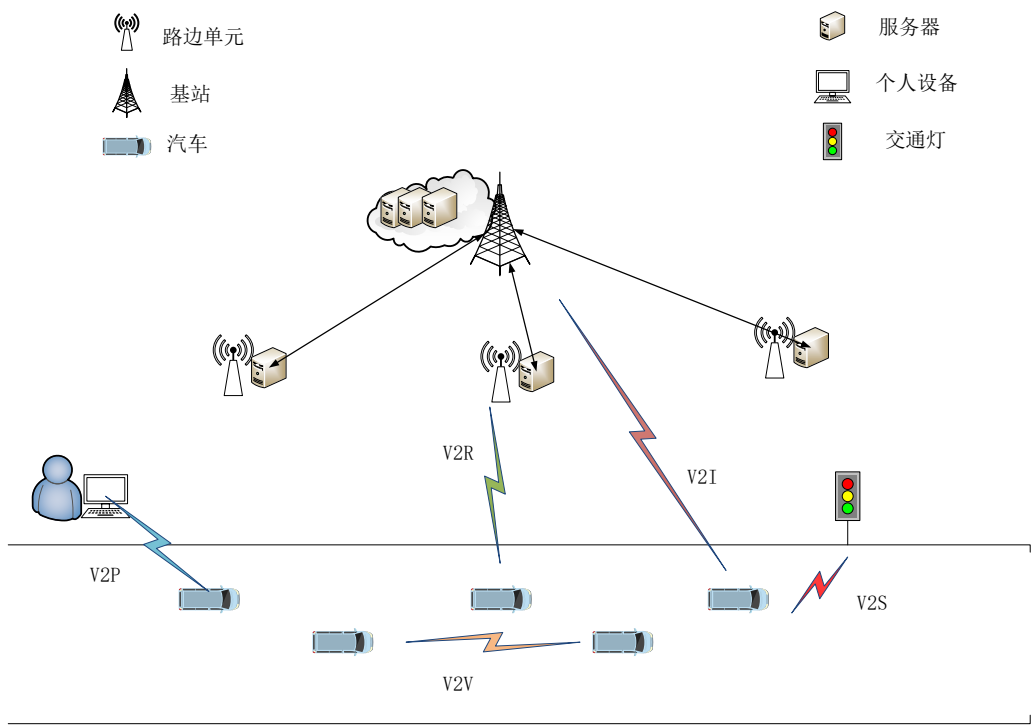


图 2.1 车联网的五种通信模式示意图

IoV 架构包括环境感知层、网络层、计算层和应用层。环境感知层通过车内的传感器从车辆周围环境中收集数据,包括车辆位置和道路状况等数据信息。网络层负责管理车辆之间的短距离通信。计算层负责对收集到的数据进行处理、储存和分析。应用层可以提供开放性服务和封闭性服务。开放性服务是指互联网提供商提供的在线应用服务,包括实时交通服务以及在线视频服务等。封闭性服务是指特定的车联网应用服务,包括控制面板指令服务和交通指令服务等。

IoV 设计动机分为三个方面。首先,VANET 在走向商业化的道路上遇到了技术难题。其次,随着机动车数量的增加,交通安全问题日益凸显。最后也是最重要的一环,巨大的市场机遇与政策导向是 IoV 走向应用的强大推力。

1. VANET 中的商业化技术难题

(1) VANET 是自组织网络体系结构,如果车辆与自组织网络断开连接,即便车辆还在道路上,也无法与其它网络进行协作。所以,在目前的 VANET 框架下,网络连接无法得到保证,而商业应用程序需要可靠的网络连接。因此,商业应用程序无法适应 VANET 架构。

(2) 基于大数据的计算决策在目前的 VANET 框架下很难实现,这是由于计算资源和存储资源有限,以及车辆与云服务连接的不稳定性所造成的。

(3) VANET 是基于交通环境局部知识的体系架构,使得 ITS 应用程序中服务的准确性明显较低,无法为用户带来良好的驾驶体验。

由此可知,VANET 体系无法与商业化程序兼容。这使得研究团体开始对 IoV 体系进行设计与开发。

2. 交通安全问题

与 VANET 相比, IoV 将为车辆通信提供更可靠的框架,有效减少交通伤亡。

世界卫生组织(World Health Organization, WHO)报告称,每年因道路交通事故而导致的死亡总数为 125 万,每天的死亡人数接近 3424 人,道路交通事故是十大致死原因之一。报告表明,迫切需要更可靠的车辆通信技术来减少交通伤亡。

3. 市场机遇

IoV 为汽车行业、IT 设备制造业、软件行业和 Internet 服务提供商等一系列行业提供了巨大的市场机会。在物联网的发展中,汽车行业是发展最快的行业之一。

到 2025 年，联网汽车的年销量将达到 8100 万辆，80% 的新车将采用某种形式的联网驱动技术，IoV 每年产生的潜在经济价值估计为 210-740 亿美元。

由上述可知，IoV 的设计与开发在社会经济与交通安全等方面具有十分重要的意义。为此，本文对 IoV 体系下的任务卸载与迁移机制展开研究。

2.2 MEC 技术及其应用

2.2.1 MEC 技术概述

1. MEC 的出现

在移动终端设备的应用程序中，数据流量开始迅速增加。移动终端设备计算和存储资源有限，并且受其电池寿命的限制，促使网络运营商修改其现有基础架构。为此引入 MCC，将云计算功能扩展到移动终端设备，目的是集中管理核心网络中的计算和存储资源。但是，由于与设备的距离较远以及回程带宽的限制，MCC 无法满足新兴移动应用程序的低延迟要求。为解决这一问题，计算和存储资源应放置在尽可能靠近移动终端设备的位置。这种在移动终端设备附近部署云服务器的技术被称为 MEC。与 MCC 相比，MEC 将分布式计算和缓存资源放置在与移动终端设备较近的物理位置，例如基站和路边单元等位置。因此，在一定程度上解决了通信延迟较高的问题，也降低了回程数据流量，延长了移动设备的电池寿命。

2. 实施 MEC 的关键技术

为了更好地实现 MEC 框架，并提高其可操作性，可以结合使用多种集成技术，主要是 SDN，网络功能虚拟化(Network Functions Virtualization, NFV)和信息中心网络(Information Centric Networking, ICN)等技术，如下所述。

(1) SDN

SDN 将管理网络的控制平面与传输数据的数据平面分开，形成了可编程且具有应用感知能力的智能网络。SDN 有助于解决经典的网络问题，例如路由和 IP 地址转换等问题，也可以解决新型问题，例如用户移动性管理以及物联网系统安全性等问题。SDN 可灵活控制 MEC 之间的网络流量，进而可以将计算资源和缓存资源无缝集成。

(2) NFV

NFV 利用虚拟化技术实现网络功能设计、部署和管理,并且独立于底层物理网络设备。NFV 功能包括传统功能与新型功能。其中,传统功能包括防火墙、深度数据包检查、演进数据包核心元素等。新型功能包括网络编码、数据聚合服务等。NFV 按顺序组合了单个虚拟网络功能,因此可以在服务功能链(Service Function Chains, SFC)中将复杂功能进行模块化处理。

(3) ICN

因特网最初是用来实现主机对主机通信,如今主要用于内容分发。ICN 旨在缩小传统互联网和当前应用程序(例如高清视频点播流、3D 游戏)之间的差距,以及解决增强现实和虚拟现实带来的流量爆炸问题。为了提高内容分发效率,ICN 重新设计网络体系结构,减轻了 MEC 服务器上的带宽压力并提高了数据传输效率。

2.2.2 计算卸载

计算卸载的主要研究目标是 최소화设备等待时间、能量消耗以及增强卸载过程中的安全性。

大多数研究根据应用场景考虑了可分割和不可分割的计算任务。不可分割任务或可分割任务的各个子任务可以在本地执行(如果移动设备有足够的计算资源,并且可以容忍本地执行的延迟),也可以通过设备对设备通信(Device to Device, D2D)或中继辅助卸载到相邻的移动设备或者 MEC 服务器进行计算。

将任务分配到其它终端设备或 MEC 服务器的策略称为卸载决策。由于终端设备的移动性,以及无线信道和终端资源的随机性,卸载决策通常是基于终端设备的计算需求和 MEC 服务器的资源约束等方面来制定。此外,制定卸载决策还需要综合考虑无线网络动态和设备异构性等方面。

任务卸载通常以分布式方式或集中式方式(在基站处)进行。集中式方式适用于小型网络,但对于大型网络而言,为了适应网络动态,需要从整个网络域中频繁收集数据并进行后续集中处理,纯粹的集中方法变得效率低下。在大规模网络中,这种集中式处理会导致较长的信令延迟,较大的控制信令开销和较高的计算复杂度。未来的研究需要探索组合决策方法,即将一定范围的决策托管给局部节点,而全局决策仍然在中央控制器上完成,进而提高任务计算卸载效率。

2.2.3 边缘缓存

移动视频和社交网络已经占据了无线网络流量的很大比例，数据密集型的网页以及社交网络应用的请求，可以通过 MEC 缓存服务器、本地缓存和附近其他用户的缓存协同处理，如图 2.2 所示。当用户在基站范围内的不同时间段请求内容时，来自用户缓存的内容可以减少基站的内容重复传输。

MEC 缓存研究的主要目标是用户能耗最小化，吞吐量最大化以及降低成本。目前 MEC 缓存方法的主要策略如下：1. 将缓存内容放置在其它终端设备并且通过 D2D 通信链路进行交付；2. 将缓存内容放置在 MEC 服务器并从该服务器进行交付；3. 将缓存内容放置在中心云进行交付。

为了实现缓存性能目标，通常使用一些数学工具（例如博弈论、随机几何和启发式优化）来解决优化问题。当玩家（基站和用户）相互竞争以最大化其利益时，通常使用博弈论。在某些现有约束（例如有限的缓存大小，缓存状态和设备的剩余电池寿命）下一般采用优化算法来最大化性能指标。随机几何方法一般用于具有随机拓扑的网络结构。

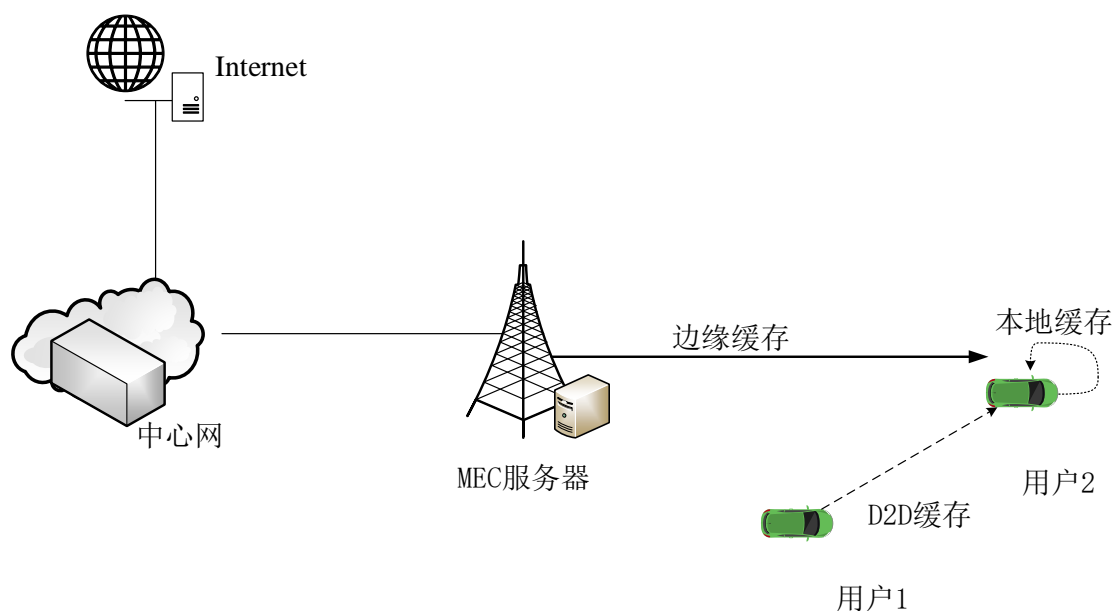


图 2.2 缓存协同处理示意图

MEC 缓存策略通常采用分布式控制或集中式控制。现有的研究大多数考虑了集中式控制，集中式控制通常在基站处实现。因为基站会跟踪所需的信息，例如用户

位置、偏好、请求以及内容流行度等。未来的工作应该深入研究在具有多基站的分布式网络系统下的 MEC 缓存机制，以突破现有研究的局限性。

2.2.4 任务迁移

当移动车辆卸载任务时会面临任务迁移问题。资源虚拟化技术是任务迁移的关键技术，可在共享硬件平台上执行任务。VM 是资源虚拟化技术的重要部分，通常依靠软件实现，可以在其中安装和运行操作系统或程序。在云计算中，应用程序和服务托管在 VM 上，并为其分配了专用资源，以匹配应用程序需求。

任务迁移，也就是 VM 迁移，是将 VM 从其当前物理机转移到新物理机的过程。VM 迁移可以在 VM 正在处理任务时完成，称为实时或联机 VM 迁移。脱机 VM 迁移是指在迁移之前暂停 VM 任务，然后进行迁移。

在车辆云中，VM 迁移的主要功能之一是确保不会因为网络断开或车辆故障而丢失正在处理的任务。当车辆断开网络连接时，正在处理的任务可能会丢失。通过 VM 迁移，可以在与车辆云断开连接之前，将正在执行任务的 VM 迁移到新的位置，从而保证任务可以正常运行。

由于节点的高度移动性和间歇性连接，车辆云中会发生大量的 VM 迁移。如果没有有效的 VM 迁移方案，丢失的任务必须由另一个节点重新启动。这将导致任务完成时间较高，性能下降和资源利用效率降低。由此研究团体一直致力于探索有效的 VM 迁移方案。

任务迁移如图 2.3 所示。当 RSU-1 通讯范围内的车辆 A 要使用云服务时，服务器会定制 VM-A 为车辆服务。当车辆行驶到 RSU-2 时，有两种方法可以继续使用云服务。一种方法是通过 VM 迁移技术，将 VM-A 迁移至车辆附近的 RSU-2，从而使车辆可以继续使用云服务。虽然 VM 迁移会引入迁移成本（与 VM 的内存大小相对应），但也会降低流量数据的传输成本。另一种方法是将 VM-A 保留在 RSU-1 中，车辆 A 与 RSU-1 通信以获取 VM-A 上的服务。显然，后一种方法的迁移成本为零，但由于通信距离（或跳数）更长，可能会带来更高的执行成本。因此，有必要研究 VM 迁移方案来提高任务完成效率，降低任务迁移成本，从而让用户获得更好地服务体验。

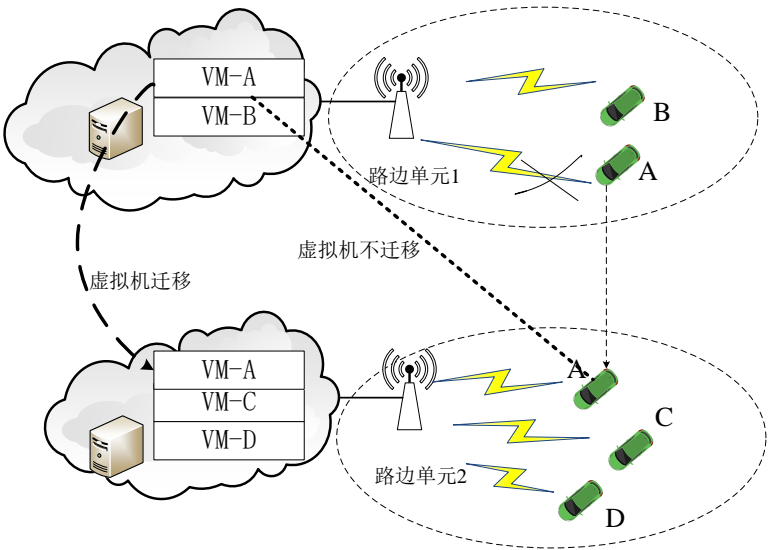


图 2.3 任务迁移示意图

2.3 软件定义车联网技术概述

2.3.1 软件定义车联网相关概念

SDN 在发送数据到网络设备之前，在控制器中创建规则。控制器可以管理网络行为，提高了传统网络的拓展性和伸缩性。图 2.4 为 SDN 基本架构图。

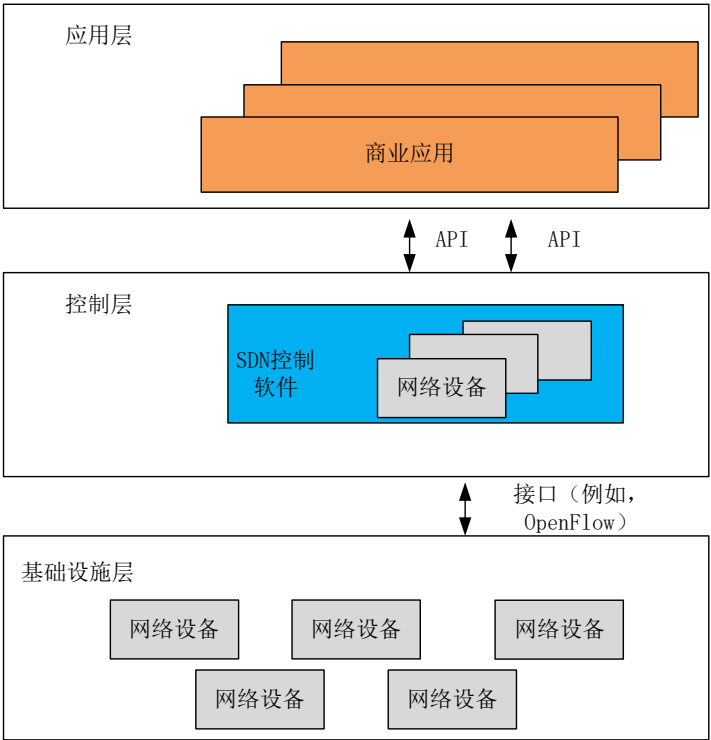


图 2.4 SDN 基本架构图

软件定义车联网模型图如图 2.5 所示，相关概念介绍如下：

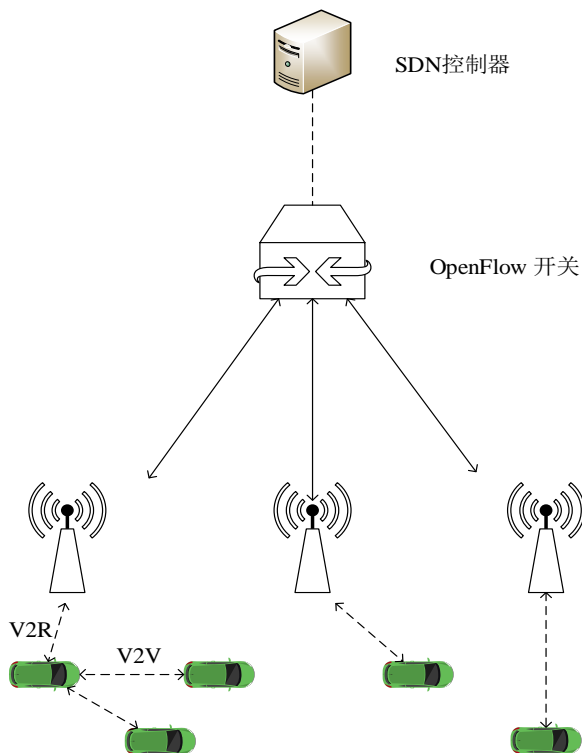


图 2.5 软件定义车联网模型图

1. 控制平面

控制平面用于管理数据平面中的车辆和路边单元。它可以存储 SDN 交换机的状态，并依靠 GPS 收集车辆位置和速度等数据。SDN 控制器可以分析数据来制定转发决策，例如在转发信息包到达目的地时制定路由决策。

2. 数据平面

相互连通的网络组件构建成数据平面。数据平面组件由包含 SDN/OpenFlow (OF) 交换机的车辆和路边单元组成。车辆数据包含方向、位置和速度等信息，被放置在路边单元中的 OF 流表中。

3. OF 协议

OF 协议一般用于 SDN 数据平面和控制平面之间的通信接口。OF 协议包括三个部分，即 OF 开关，协议和控制器。

4. OpenRadio

OpenRadio 通常建立在一个多核硬件平台上。核心组件是一个软件抽象层，通过声明性接口对基带和 MAC 层进行编程。操作员通过定义规则进行相关操作。规

则一般是有关数据包参数的逻辑词,例如报头字段、接收信号强度、信道频率和其它可能被编程的字段。操作包括诸如数据编码/解码和信道流量调度等行为。

5. OpenRoads

OpenRoads 为研究人员提供了两种网络控制手段,即使用 OF 协议控制数据路径和使用管理设备接口控制设备配置,进一步增强了 OpenRoads 的网络管理。

2.3.2 软件定义车联网发展方向

软件定义车联网的发展方向是通过结合新的元素来增强性能,满足车联网的应用要求。

结合地理信息服务可以帮助 SDN 控制器进行路径预测,尤其是在高度动态的情况下,提高了安全服务的数据分发效率。结合云计算的存储和处理能力可以满足信息娱乐应用程序的需求。结合雾计算可以解决高延迟和缺乏移动性支持等问题。由于 SDN 的优势,软件定义车联网集中式体系结构已经实现了车载网络的 QoS 改进。但是在大型网络中,控制器可能放置在离车辆较远的地方,从而导致任务延迟较高。同样,在高度密集的情况下,单个控制器通常会过载,并且无法响应大量请求。因此,目前的研究工作开始探索控制平面的分布式发展,以克服使用单个控制器的局限性。

2.4 NOMA 技术概述

2.4.1 NOMA 技术简介

NOMA 思想的关键是将叠加编码技术应用于基站,将多用户检测和连续干扰消除(Successive Interference Cancellation, SIC)等干扰消除技术应用于用户。NOMA 可以让多个用户共享相同的时频资源,与传统的正交多址访问(Orthogonal Multiple Access, OMA)相比,频谱效率更高。NOMA 有两个主要类别:功率域 NOMA 和代码域 NOMA。功率域 NOMA 利用用户之间的信道增益差异,在功率域中多路复用户户,而代码域 NOMA 使用用户特定的序列来共享整个可用无线电资源。基于代码域的访问策略的典型示例有低密度扩频码分多址(Code Division Multiple Access,

CDMA)和基于低密度扩展的正交频分多址(Orthogonal Frequency Division Multiple Access, OFDMA)。

2.4.2 NOMA 与 MEC 的结合

MEC 和 NOMA 的结合可以显著提高用户满意度和网络性能。NOMA 在提高频谱效率和减少传输等待时间等方面具有许多优势。MEC 不仅可以为用户带来便利,而且可以改善总体网络性能。

MEC 和 NOMA 的结合可以加强 5G 网络业务效率。一方面, NOMA 有望大幅增加用户数量。另一方面, MEC 中的边缘计算可在 RAN 边缘附近为终端用户提供计算资源。因此 MEC 能够将计算资源从集中式分布到网络边缘,并为大量用户提供服务,具有支持大规模连接和分布式计算的潜力。

MEC 和 NOMA 的结合可以实现低延迟传输。由于 5G 网络不会依赖单一技术,因此须从多种角度优化网络。MEC 将云计算功能移至网络边缘,在该边缘生成和处理数据。与云计算相比,MEC 在边缘运行的服务能够更好地满足用户的低延迟要求。类似地, NOMA 中灵活调度和免授权访问也可为 5G 网络用户降低传输延迟。

2.4.3 MEC-NOMA 系统关键问题

1. 资源分配优化问题

在 MEC-NOMA 网络中,可以同时优化通信和计算资源分配,以增强系统性能。将 NOMA 上行传输应用于 MEC 系统时,多个用户同时将任务分配到 MEC 服务器。在考虑系统总能耗的情况下,通过控制每个用户的卸载计算负载和传输功率,设计出最小化系统总延迟的最优策略和次优策略是一个值得研究的课题。

2. 安全通信问题

安全和保护隐私的通信也是研究重点,尤其是将 NOMA 应用于 MEC 系统时。例如,两个用户通过 NOMA 技术同时将任务卸载到 MEC 服务器。当执行 SIC 时,一个用户可以解码另一用户的消息。在此期间,窃听者或攻击者可能会解码移动用户的消息。因此,提高基于 NOMA-MEC 系统的物理层安全性(Physical Layer Security, PLS)是值得研究的关键问题。

2.5 本章小结

本章的第一小节对车联网技术进行了详细的介绍。第二小节主要介绍 MEC 技术的发展以及其网络场景，分别为计算卸载、任务迁移以及边缘缓存。本章的第三小节对软件定义车联网技术进行概述，并说明其未来的发展方向。本章的第四小节主要介绍 NOMA 技术，以及 NOMA 与 MEC 相结合的优势和关键问题。

第3章 基于 MEC 的软件定义车联网任务迁移策略

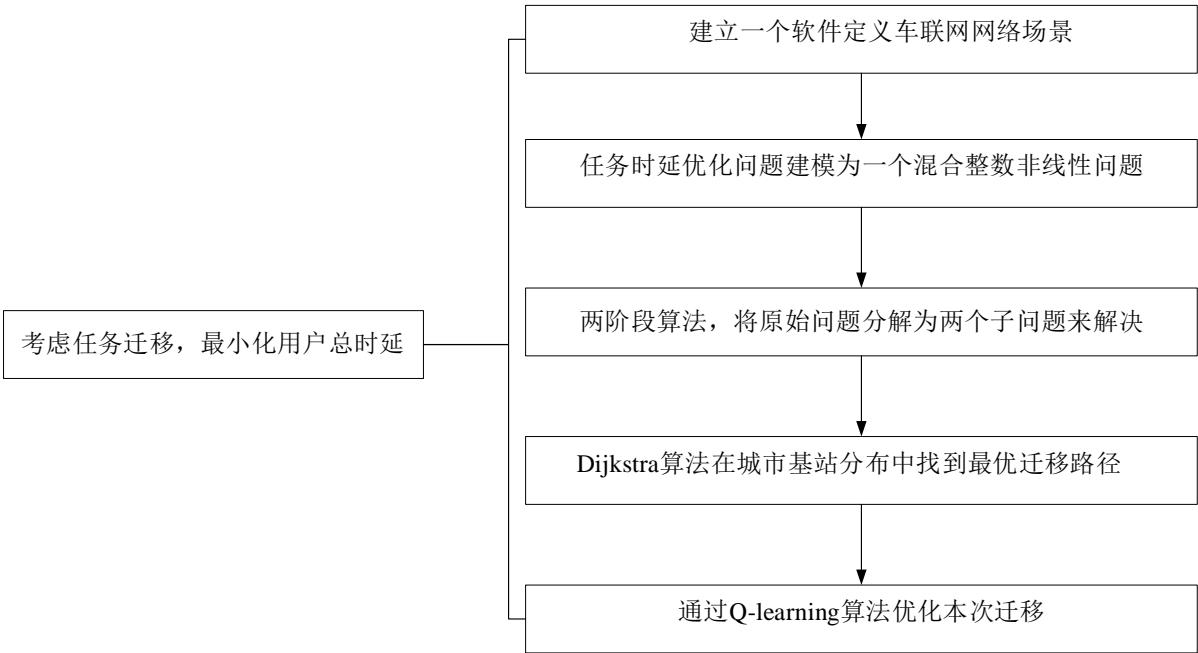
3.1 引言

近年来,汽车行业硬件和软件技术发展迅速,各种先进的通信技术使驾驶更安全、更舒适。由于汽车运行着大量的计算密集型应用程序,使得具有有限计算能力的汽车终端难以满足日新月异的新业务需求,所以研究者提出利用 MEC 技术为附近的移动车辆终端提供计算服务。另外,在新一代车载网络中,传统的体系架构难以应对复杂车辆网场景下的全局网络管理和资源调度等问题。而 SDN 技术可以加快数据调度、提高车辆网络的资源利用率和增强服务管理。因此,在基于 MEC 的车联网中引入 SDN 形成软件定义车联网体系架构。在基于 SDN-MEC 的车联网系统下,当车辆用户处于动态时,会面临车辆任务迁移等一系列移动性管理问题。通过 VM 迁移技术,可以有效解决该问题。所以本章把 SDN、VM 迁移和车联网结合,利用 SDN 提高车联网的可扩展性,可以更好地管理 VM 迁移路由。并且 SDN 控制器可实时监视和收集网络状态,具备全局视图。因此三者的结合可以提供一个完整的结构框架,可管理车联网的资源分配和 VM 迁移问题。

本章提出基于 MEC 的软件定义车联网任务迁移策略,本章工作归结于以下几点:

1. 构建一个软件定义车联网网络场景。基站配有 MEC 服务器,用户在移动过程中将产生计算任务。SDN 控制器具有全局视图,可以集中调度车辆与更新基站信息。
2. 研究用户在车辆移动期间的任务迁移问题。车辆的任务分为三部分执行,即本地执行、原始路边单元执行和目标路边单元执行。由于车辆用户的移动,任务需从原始路边单元迁移到目标路边单元,由此产生 VM 迁移问题,优化目标是最小化迁移过程中的时延。
3. 将任务时延优化问题建模为一个混合整数非线性问题。提出一个两阶段算法求解。第一阶段提出基于 Dijkstra 的最短迁移路径选择算法,在城市基站分布模型中找到最优迁移路径以减少资源浪费以及时间消耗;第二阶段提出基于 Q-learning 的任务迁移优化算法,优化任务迁移时延。

图 3.1 为任务迁移优化策略流程框图。



3.2 系统模型

3.2.1 系统分析

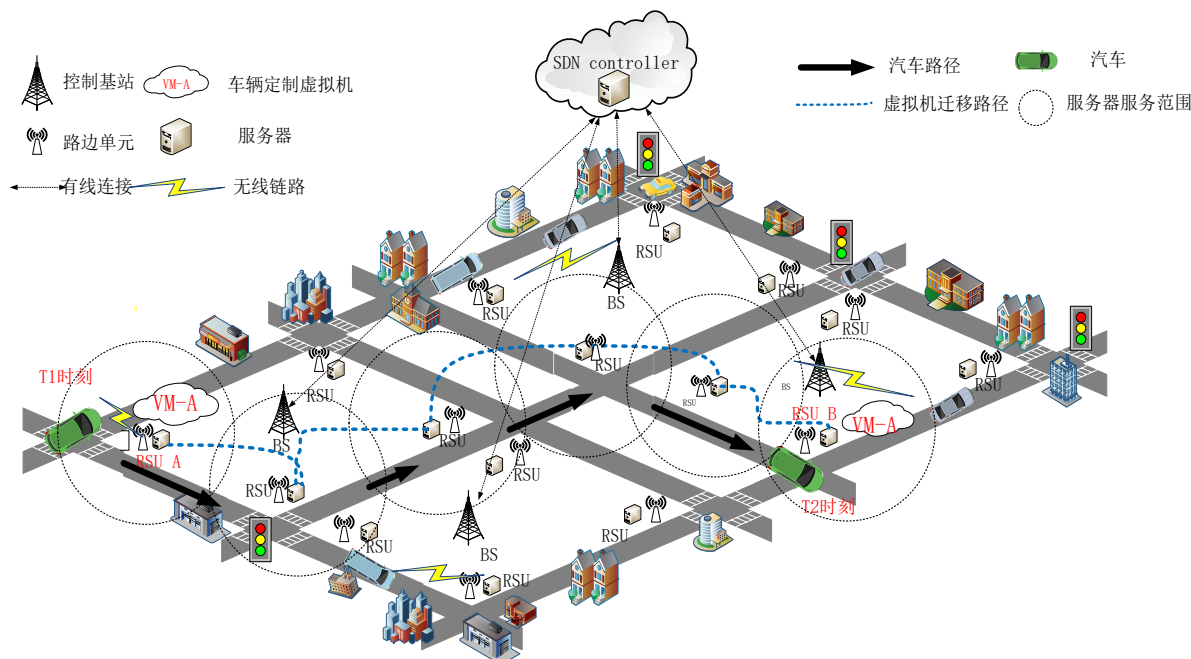


图 3.2 是基于 MEC 的软件定义车联网系统框架。其中 RSU、BS、车辆和其它无线设备之间的数据流代表数据平面，车辆可以通过蜂窝网络接口定期向控制器更新其状态，包括位置和运动信息等内容。SDN 控制器代表控制平面，基于给定的服务，执行带宽分配、路由协议等决策。基于接收到的控制信息，诸如 RSU 和 BS 之类的设备将进行对应的操作。SDN 的应用平面由车辆的服务请求所构成。

软件定义车联网服务迁移场景如图 3.2 所示，有 M 个路边单元，每个路边单元部署一个 MEC 服务器，将其表示为 $Z = \{z_1, z_2, \dots, z_m, \dots, z_M\}$ ，其中 $1 \leq m \leq M$ ， z_m 表示第 m 个服务器。车辆可以通过蜂窝网络向路边单元和基站卸载任务。有 A_n 个任务车辆，表示为 $V = \{v_1, v_2, \dots, v_{A_n}\}$ 。车辆用户采用 OFDMA 与基站相连，也就是说，在一个基站覆盖范围内的用户信道之间是正交的。

如图 3.2 所示，车辆 V_i 在 RSU A 的通信范围内请求服务时，RSU A 会对车辆 V_i 进行任务 VM 定制，产生 VM-A。由于车辆的移动性，会存在任务还没有完成而该车辆驶出了 RSU A 的服务范围的可能，当车辆驶出 RSU A 的范围到达 RSU B 时，有两种方案：一种是 VM-A 迁移到 RSU B；另一种是 VM-A 仍然留在 RSU A，车辆与 RSU A 进行通信获得服务。显然后者通信距离较远，导致时间延迟较高。因此本章重点针对前者研究 VM 迁移策略来减少用户任务迁移时延。

3.2.2 任务模型

每个计算任务可以被描述成 $\Gamma_i(D_i, C_i)$ ， D_i 表示计算任务 Γ_i 的输入数据大小， C_i 表示完成任务 Γ_i 所需的 CPU 周期数。在本章中，任务被分为三部分执行，车辆本地执行，原始路边单元执行以及目的路边单元执行。之所以将任务限制在两个计算节点进行执行，是因为如果在多个计算节点计算，会增加通信链路端口的负载，当任务车辆增多时，可能会造成堵塞，导致传输效率降低。因此将任务限制在开始节点和目的节点进行计算，这样可以改善整体系统效率。任务数据及 CPU 周期数分配情况如下所示：

$$D_i = D_{il} + D_{io} + D_{id} = \varepsilon D_i + \phi D_i + \varphi D_i \quad (3.1)$$

$$C_i = C_{il} + C_{io} + C_{id} = \alpha C_i + \beta C_i + \chi C_i \quad (3.2)$$

其中, D_{il} 、 D_{io} 、 D_{id} 分别为车辆本地、原始路边单元以及目的路边单元的任务数据大小, C_{il} 、 C_{io} 、 C_{id} 分别表示为本地部分任务、原始路边单元以及目的路边单元的任务所需要的 CPU 周期数。 ε 、 ϕ 、 φ 、 α 、 β 、 χ 为分配比例系数。

3.2.3 任务卸载、计算与迁移模型

当车辆有任务卸载时, 首先将任务卸载至起始基站的服务器, 假设车辆在离开起始基站的服务范围时, 计算任务没有完成, 则任务将从起始服务器迁移至目的基站的服务器, 最后目的基站将结果回传给车辆。

车辆 V_i 向基站 z_m 发送任务的上行传输速率 $R_{i,n,m}^T$ 为:

$$R_{i,n,m}^T = w \log_2 \left(1 + \frac{p_{i,n,m} h_{i,n,m}}{\sigma^2 + I} \right) \quad (3.3)$$

$$h_{i,n,m} = 10^{-(128.1 + 37.6 \log_{10} d_{i,n,m})/10} \quad (3.4)$$

其中, w 是信道带宽, $p_{i,n,m}$ 、 $h_{i,n,m}$ 分别是车辆 V_i 在信道 n 向基站 z_m 上传任务的传输功率和信道增益^[54]。 σ^2 是汽车终端的高斯白噪声功率, I 是信道的干扰。 $d_{i,n,m}$ 是车辆 V_i 上传任务时与起始服务器 z_m 的距离。

当任务回收时, 由于任务计算结果的数据量远小于输入数据量的大小, 回收时延很小, 因此一般不考虑回收时延^[55]。

车辆将部分数据上行传输给路边单元的时间为:

$$t^T = \frac{D_{io} + D_{id}}{R_{i,n,m}^T} \quad (3.5)$$

车辆本地计算时间为:

$$t_l = \frac{C_{il}}{F_i} \quad (3.6)$$

其中, F_i 表示车辆自身的计算能力。

边缘计算时间由原始 MEC 服务器以及目的 MEC 服务器两部分构成, 如下所示:

$$t_{io} = \frac{C_{io}}{F_{io}} \quad (3.7)$$

$$t_{id} = \frac{C_{id}}{F_{id}} \quad (3.8)$$

其中, F_{io} 和 F_{id} 为原始 MEC 服务器以及目的 MEC 服务器分配给该任务的计算资源。

任务迁移即 VM 迁移, 目前普遍采取的是基于内存预拷贝的在线迁移机制。由文献[56]可知, VM 迁移时间分为两部分, 迁移时间和宕机时间。在服务器 Z_m 中, 数学模型为:

迁移时间:

$$T_{mig} = \frac{D_{id}}{L_m} \times \frac{1 - \lambda_m^{N_m+1}}{1 - \lambda_m} \quad (3.9)$$

其中, $\lambda_m = \frac{R_m}{L_m}$ 为迁移模型参数, R_m 为迁移期间的脏页率, L_m 为 VM 所处数据中心的网络带宽, N_m 为迭代次数。

宕机时间:

$$T_{down} = \frac{D_{id}}{L_m} \times \lambda_m^{N_m} + t_{res} \quad (3.10)$$

其中, t_{res} 为 VM 在目的物理机的重启时间。

假设任务 $\Gamma_i(D_i, C_i)$ 迁移路径上经过的服务器数为 e , 其中 $1 \leq e \leq M$ 。总迁移时间消耗为:

$$T_M = T_{mig} + T_{down} = \sum_{m=1}^e \left(\frac{\varphi D_i}{L_m} \times \frac{1 - \lambda_m^{N_m+1}}{1 - \lambda_m} + \frac{\varphi D_i}{L_m} \times \lambda_m^{N_m} + t_{res} \right) \quad (3.11)$$

3.2.4 问题建模

将总体任务时延优化问题建模为一个混合整数非线性问题, 目标问题由以下几部分构成: 车辆上传时延, 任务计算时延, 任务迁移时延。相比于任务上传过程及任务计算过程, 任务迁移过程更为复杂, 所以本章工作重点在于任务迁移时延的优化。计算任务总时延优化目标函数如下所示:

$$\min T(\lambda_m, N_m) = \min \sum_{i=1}^{A_n} \left[\frac{\phi D_i + \varphi D_i}{R_{i,n,m}^T} + \frac{\alpha C_i}{F_i} + \frac{\beta C_i}{F_{io}} + \frac{\chi C_i}{F_{id}} + \sum_{m=1}^e \left(\frac{\varphi D_i}{L_m} \times \frac{1 - \lambda_m^{N_m+1}}{1 - \lambda_m} + \frac{\varphi D_i}{L_m} \times \lambda_m^{N_m} + t_{res} \right) \right]$$

C1: $\phi, \varphi \in (0, 1)$
 $\alpha + \beta + \chi = 1$
 $\alpha, \beta, \chi \in (0, 1)$

C2: $0 < F_{io}, F_{id} < F_B$

C3: $\varphi D_i < A_{LINK}$

C4: $0 < \lambda_m < 1$

C5: $0 < N_m < N_{MAX}$

(3.12)

其中，C1 是任务模型中输入数据和完成任务所需 CPU 周期数的权重因子的限制条件。C2 表示 MEC 服务器分配给该任务的计算资源要小于 MEC 服务器总的计算资源 F_B 。C3 表明在 VM 迁移时传输的数据量要小于链路的容量 A_{LINK} 。C4 表示 VM 迁移时间模型参数 λ_m 的取值范围。C5 表明迭代次数的范围，其中 N_{MAX} 表示最大的迭代次数。

3.3 任务迁移优化算法

本章对公式(3.12)中的任务迁移时间部分进行优化，即优化目标函数如下：

$$\min T_M(\lambda_m, N_m) = \min \sum_{i=1}^{A_n} \sum_{m=1}^e \left(\frac{\varphi D_i}{L_m} \times \frac{1 - \lambda_m^{N_m+1}}{1 - \lambda_m} + \frac{\varphi D_i}{L_m} \times \lambda_m^{N_m} + t_{res} \right) \quad (3.13)$$

优化算法分为两个阶段。第一个阶段是从原始路边单元到目的路边单元的多条迁移路径中选择一条最优的路径进行迁移，第二个阶段是优化第一阶段求得最优路径上的任务迁移时间。算法流程框图如图 3.3 所示。

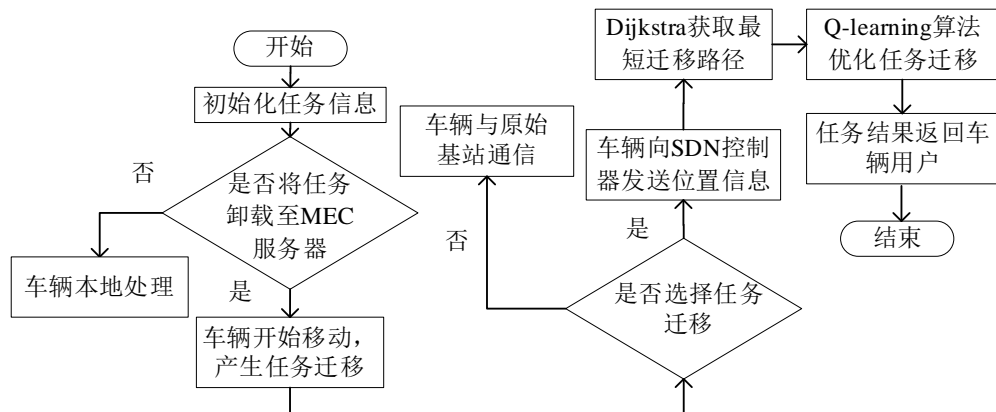


图 3.3 任务迁移优化算法流程框图

3.3.1 基于 Dijkstra 的最短迁移路径选择算法

图 3.4 是基于 Dijkstra 的最短迁移路径选择算法流程图。

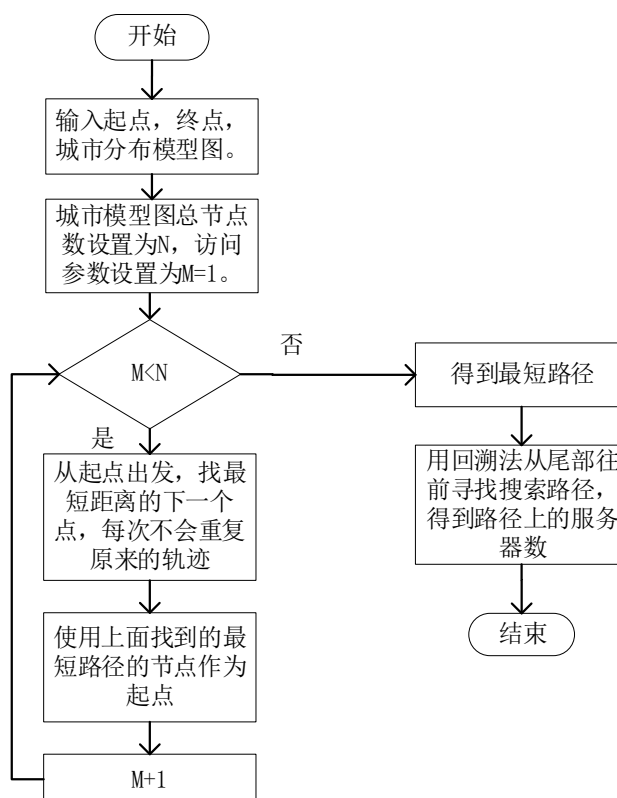


图 3.4 基于 Dijkstra 的最短迁移路径选择算法流程图

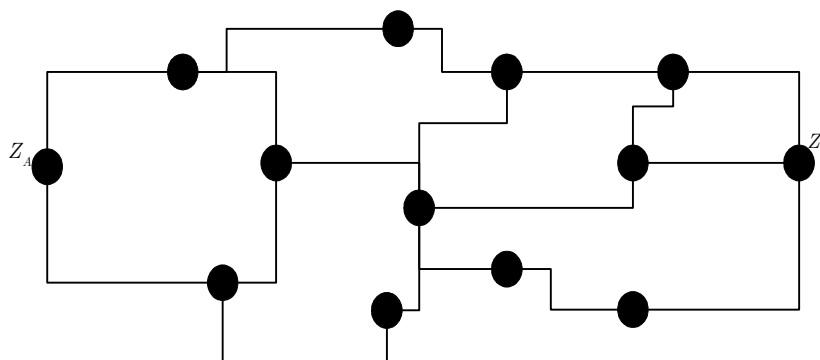


图 3.5 基站分布数学模型图

假设基站分布数学模型如图 3.5 所示, 该图的数学模型为 $G(Z, X, K)$, Z 表示服务节点, X 为服务节点之间的通信链路, K 为链路的距离权重。

在基于 MEC 的软件定义车联网系统框架下, 车辆可以通过蜂窝网络接口定期向 SDN 控制器更新其状态, 包括位置和运动信息等内容。SDN 控制器位于控制平面, 通过车辆的 GPS 系统可以获知车辆行驶轨迹, 得到车辆起点信息与终点信息。

假设车辆起点位置附近的基站 MEC 服务器为 Z_A ，车辆可以将任务卸载到 Z_A 。当车辆行驶到终点位置附近时，SDN 控制器可以为其提供附近的基站信息，将终点处的基站 MEC 服务器记为 Z_B 。这一过程中，如果车辆终点信息发生更改，SDN 控制器会通过实时的车辆轨迹为车辆选择新的终点基站。由此可知，车辆任务所在的 VM 需要从 Z_A 到 Z_B 进行 VM 迁移。然后， Z_B 把结果回传给车辆。为了降低任务迁移时延，需要在基站分布数学模型图中找出从 Z_A 到 Z_B 的最短路径以最小化任务迁移时延。Dijkstra 算法是寻找最短路径的一种搜索算法，通过该算法，可以找到从 Z_A 到 Z_B 的最短路径，以及在这条路径上 VM 迁移所经过的 MEC 服务器数，即目标函数 (3.13) 中的 e 。表 3.1 是基于 Dijkstra 的最短迁移路径选择算法伪代码。

表 3.1 基于 Dijkstra 的最短迁移路径选择算法

算法 3.1: 基于 Dijkstra 的最短迁移路径选择算法

- 1: 输入: 城市分布模型图为 $G(Z, X, K)$ ，起点路边单元 Z_A 的位置，目的路边单元 Z_B 的位置。
 - 2: 输出: 最短路径 I ，路径经过的 MEC 服务器个数 e 。
 - 3: 记录城市分布模型的服务器总数 n
 - 4: 生成初始节点
 - 5: 从起点出发，找最短距离的下一个点，每次不重复原轨迹
 - 6: 使用上面找到的最短路径的节点作为起点
 - 7: 如果通过此节点，则从起点到每个节点的路径长度会更短，然后更新并记录前一个节点
 - 8: 用回溯法从尾部往前寻找搜索路径，得到 I 以及路径上的服务器数 e
-

3.3.2 基于 Q-learning 的任务迁移优化算法

Q-learning 是一种无模型的强化学习算法，基本思想是训练由状态、动作、奖励和下一个状态组成的元组。在 Q-learning 算法中，在状态和动作之间构造一个 Q 表以存储 Q 值，这些 Q 值是在特定时间采取措施的奖励，再加上下一步的最大期望值。然后，根据 Q 值选择可获得最大奖励的动作以获得尽可能多的回报^[57]。

Q 值的公式如下：

$$Q(\xi, a) \leftarrow (1 - \alpha)Q(\xi, a) + \alpha[r(\xi, a) + \gamma \max_{a'} Q(\xi', a')] \quad (3.14)$$

其中, ξ 表示状态集, 且 $\xi, \xi' \in \xi$ 。 A 表示动作集, 且 $a, a' \in A$ 。 α 是学习率, $0 < \alpha < 1$ 。 $r(s, a)$ 表示在状态 s 下执行动作 a 将得到的回报, γ 表示折扣因子, 决定时间的远近对回报的影响程度^[58]。 $Q(\xi, a)$, $Q(\xi', a')$ 表示在相应状态动作下的价值回报。

本章先证明优化目标函数(3.13)为非凸优化问题, 为此利用 Q-learning 算法求解目标函数(3.13), 通过建立状态集和动作集, 将目标函数设置为算法可获得的奖励, 达到求解的目的。详细情况如下:

引理: 给定约束条件下, 优化目标函数(3.13)是一个非凸优化问题。

证明: 多元函数求偏导问题实际是一元函数的微分法问题。即证明

$$f(\lambda_m, N_m) = \frac{1 - \lambda_m^{N_m+1}}{1 - \lambda_m} + \lambda_m^{N_m} \quad (3.15)$$

为非凸问题。对公式(3.15)求二阶偏导得:

$$\frac{\partial^2 f}{\partial \lambda_m^2} = \frac{2\lambda_m^{(N_m-1)N_m(N_m+1)} - \lambda_m^{N_m(N_m-2)(N_m-1)}}{\lambda_m - 1} - \frac{2(\lambda_m^{N_m} - 2\lambda_m^{N_m+1} + 1)}{(\lambda_m - 1)^3} + \frac{2\lambda_m^{N_m(N_m-1)} - 2\lambda_m^{N_m(N_m+1)}}{(\lambda_m - 1)^2} \quad (3.16)$$

$$\frac{\partial^2 f}{\partial \lambda_m \partial N_m} = 2\lambda_m^{N_m} - \lambda_m^{N_m-1} - \lambda_m^{(N_m-1)N_m \ln(\lambda_m)} + \frac{2\lambda_m^{N_m(N_m+1) \ln(\lambda_m)}}{\lambda_m - 1} - \frac{2\lambda_m^{(N_m+1) \ln(\lambda_m)} - \lambda_m^{N_m \ln(\lambda_m)}}{(\lambda_m - 1)^2} \quad (3.17)$$

$$\frac{\partial^2 f}{\partial N_m \partial \lambda_m} = 2\lambda_m^{N_m} - \lambda_m^{N_m-1} - \lambda_m^{(N_m-1)N_m \ln(\lambda_m)} + \frac{2\lambda_m^{N_m(N_m+1) \ln(\lambda_m)}}{\lambda_m - 1} - \frac{2\lambda_m^{(N_m+1) \ln(\lambda_m)} - \lambda_m^{N_m \ln(\lambda_m)}}{(\lambda_m - 1)^2} \quad (3.18)$$

$$\frac{\partial^2 f}{\partial N_m^2} = -\frac{\lambda_m^{N_m \ln \lambda_m^2} - 2\lambda_m^{(N_m+1) \ln \lambda_m^2}}{\lambda_m - 1} \quad (3.19)$$

其 Hessian 矩阵表达式为:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial \lambda_m^2} & \frac{\partial^2 f}{\partial \lambda_m \partial N_m} \\ \frac{\partial^2 f}{\partial N_m \partial \lambda_m} & \frac{\partial^2 f}{\partial N_m^2} \end{bmatrix} \quad (3.20)$$

在 $0 < \lambda_m < 1$, $0 < N_m < N_{\text{MAX}}$ 且 N_m 为整数的条件下, $\frac{\partial^2 f}{\partial N_m^2} < 0$, $\frac{\partial^2 f}{\partial \lambda_m^2} > 0$,

$\frac{\partial^2 f}{\partial \lambda_m \partial N_m} = \frac{\partial^2 f}{\partial N_m \partial \lambda_m}$ 。所以余子式为:

$$|\mathbf{H}| = \frac{\partial^2 f}{\partial N_m^2} \times \frac{\partial^2 f}{\partial \lambda_m^2} - \frac{\partial^2 f}{\partial \lambda_m \partial N_m} \times \frac{\partial^2 f}{\partial N_m \partial \lambda_m} < 0 \quad (3.21)$$

其余子式的值小于 0，所以式(3.15)为非凸优化问题。即式(3.13)亦为非凸优化问题，因此可以采用 Q-learning 算法来求解此问题。

首先建立状态集 $\xi(R_m, L_m)$ ，动作集 $A(\lambda_m, N_m)$ 。其中， λ_m 为迁移模型参数， R_m 为迁移期间的脏页率， L_m 为 VM 所处数据中心的网络带宽， N_m 为迁移期间的迭代次数。动作集参数的变化会导致状态集随之发生变化，Q-learning 算法在多个服务器中不断迭代后，可达到收敛，使得奖励最大化。奖励设置为：

$$r = \frac{1}{T_M} \quad (3.22)$$

Q-learning 算法可实现奖励的最大化，即 T_M 的最小化，进而达到优化 T_M 的目的。

由上节可以得知迁移的路径以及经过的 MEC 服务器数 e 。由此假设有 e 个状态， $\xi = \{\xi_1, \xi_2, \dots, \xi_e\}$ ，进而可以通过 Q-learning 算法实现在途经 e 个服务器后的整个迁移过程的优化。每通过一次动作集 A 的选择，状态会随之转换，进而得到奖励 r 。如图 3.6 所示。

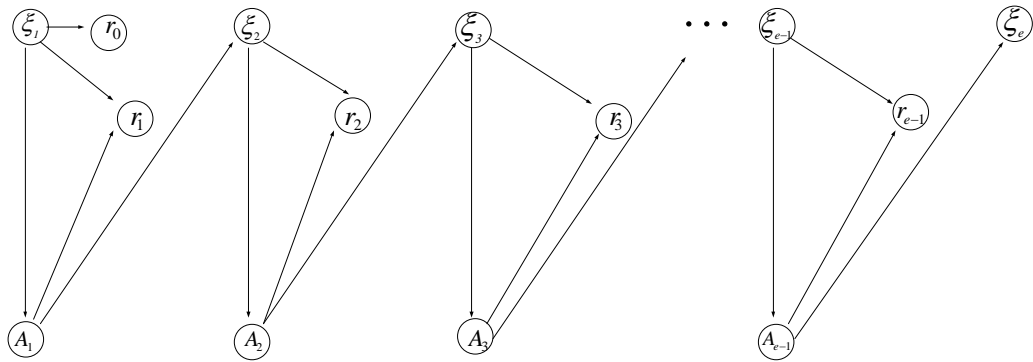


图 3.6 Q-learning 状态过程转化图

经过多次学习后， e 个服务器的累加奖励为：

$$T_M = \frac{1}{r_{0\max}} + \frac{1}{r_{1\max}} + \frac{1}{r_{2\max}} + \frac{1}{r_{3\max}} + \dots + \frac{1}{r_{e-1\max}} \quad (3.23)$$

其中， $\frac{1}{r_{0\max}}$ 为初始状态的奖励，即初始奖励。 $\frac{1}{r_{1\max}}$ 、 $\frac{1}{r_{2\max}}$ 等为状态转化时，在途经的服务器上产生的奖励，也就是该服务器上可得到的最小迁移时延。总体迁移时延由 e 个服务器最小迁移时延累加而得，进而实现总体迁移时延的最小化。

表 3.2 为基于 Q-learning 的任务迁移优化算法伪代码，图 3.7 是基于 Q-learning 的任务迁移优化算法流程图。

表 3.2 基于 Q-learning 的任务迁移优化算法

算法 3.2: 基于 Q-learning 的任务迁移优化算法

- 1: 输入: 状态集 ξ , 动作集 A
- 2: 输出: 累计奖励 T_M
- 3: 初始化 Q 矩阵
- 4: **for** 每一次探索
- 5: 随机选择一个初始状态 ξ
- 6: 通过贪婪策略选择动作
- 7: 利用选择的动作进入下一个状态 ξ'
- 8: $Q(\xi, a) \leftarrow (1 - \alpha)Q(\xi, a) + \alpha[r(\xi, a) + \gamma \max_{a'} Q(\xi', a')]$
- 9: $\xi \rightarrow \xi'$
- 10: **Until** 到达最终状态
- 11: **end**
- 12: 多次探索后, Q 矩阵趋于收敛。
- 13: 通过 Q 矩阵得到每个状态的奖励 r , 奖励累加后得到最终奖励 T_M 。

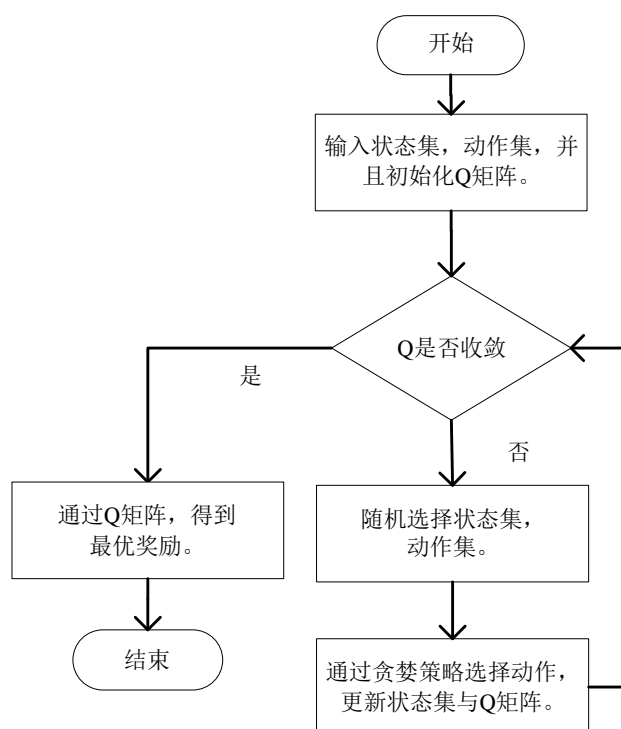


图 3.7 基于 Q-learning 的任务迁移优化算法流程图

3.3.3 复杂度分析

本章通过一个二阶段算法来优化目标时延函数，第一个阶段通过路由算法得到任务迁移的最佳路径，然后通过第二阶段的强化学习算法优化该路径上的迁移时延。第一阶段 Dijkstra 算法时间复杂度为 $O(n^2)$ ，第二阶段 Q-learning 算法时间复杂度为 $O(n)$ 。所以整个算法时间复杂度为 $O(n^2 + n) = O(n^2)$ 。

3.4 仿真分析

本节基于 MATLAB 平台对所提出的软件定义车联网服务迁移策略进行验证。根据 MEC 白皮书和 IEEE 802.11p 标准，相关参数如表 3.3 所示。

表 3.3 仿真参数表

参数	值
输入数据的大小/ MB	0-10
噪声功率/ dBm	-114
车辆用户的 CPU 周期频率/ GHz	0.5-1
传输带宽/ MHz	20
最小脏页率/ Mbps	5
车辆最大传输功率/ dbm	20
MEC 服务器 CPU 周期频率/ GHz	6
RSU 覆盖半径/ m	250
最小 λ	0.2
任务所需 CPU 周期数/ GHz	0.1-1
最大迭代次数	30

图 3.8 是不迁移与迁移车辆用户任务计算总时延对比图。图中可以看出，迁移后时延明显降低，这是因为当车辆驾驶出原始路边单元 MEC 服务器的服务范围后，若依然采取与原 MEC 服务器进行通信传输，由于车辆与其对应的 VM 之间通信距离较远，跳数增多，所以花费时间较长，也会造成通信资源的浪费。如果采用 VM 迁移的方式，VM 迁移发生在 MEC 服务器之间，最后在目标服务器完成与车辆的通信，这样的方式花费时间较少，也节省了通信资源。图 3.9 是在不同车辆数目下平均时延的对比图。显然，车辆数越大，计算任务越多，平均时延越高。

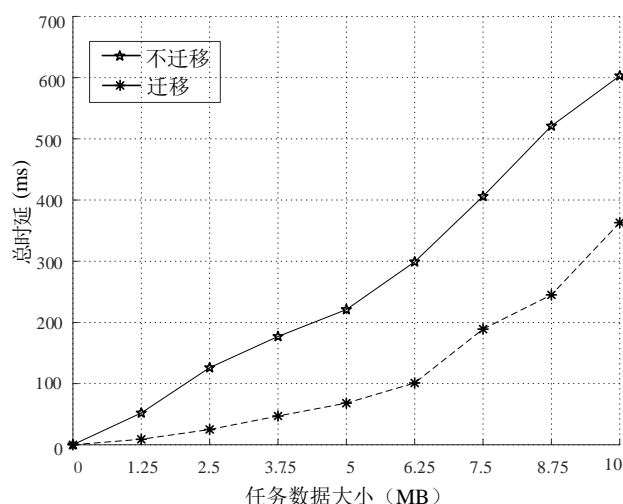


图 3.8 任务不迁移与任务迁移时延对比图

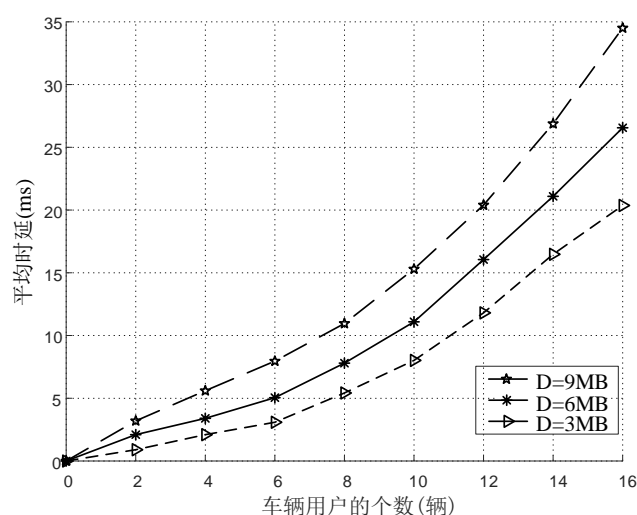


图 3.9 不同车辆数下平均时延对比图

本章与马尔科夫链算法^[59]、节能迁移算法^[60]和负载均衡算法^[61]进行比较，马尔科夫链算法优势在于可以在用户 QoS 和迁移成本之间实现良好的折中，节能迁移算法优化了迁移过程中的能量消耗，负载均衡算法利用自动部署机制减少了能耗和 VM 迁移数量。但是它们都忽略了迁移时间的影响。

图 3.10 描述了本章方案与马尔科夫链算法、节能迁移算法和负载均衡算法等方法在不同迁移数据下迁移时间对比图。可以看到，与其它方法相比，本章的方法总体迁移时间消耗较小。这是因为 Q-learning 是一个动态不断学习的过程，可以选择出最优的迁移参数进行迁移，再加上利用 Dijkstra 算法选择出了任务迁移的最短路径，与其它方法相比，迁移时延更短，可以进行最优的迁移。

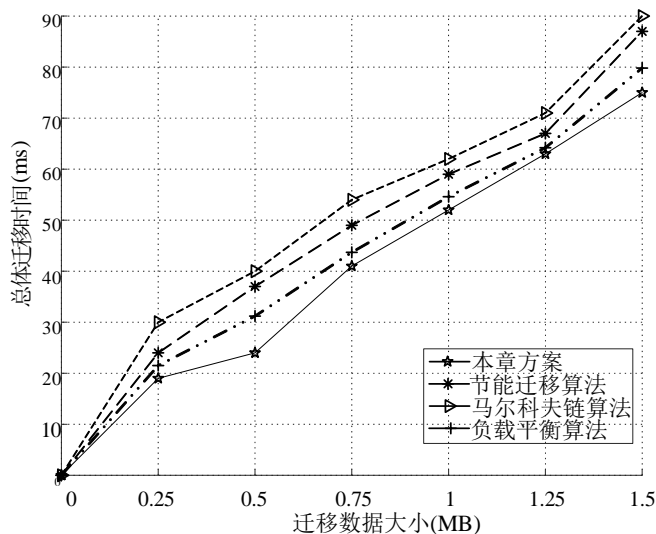


图 3.10 迁移时间对比图

图 3.11 显示了由本章方案与节能迁移算法、马尔科夫链算法和负载均衡算法等策略执行的 VM 迁移百分比对比图。可以观察到，与其它方法相比，本章算法的 VM 迁移数量更少。节能迁移算法、马尔科夫链算法和负载均衡算法没有考虑到时间因素，导致一些不必要的迁移。

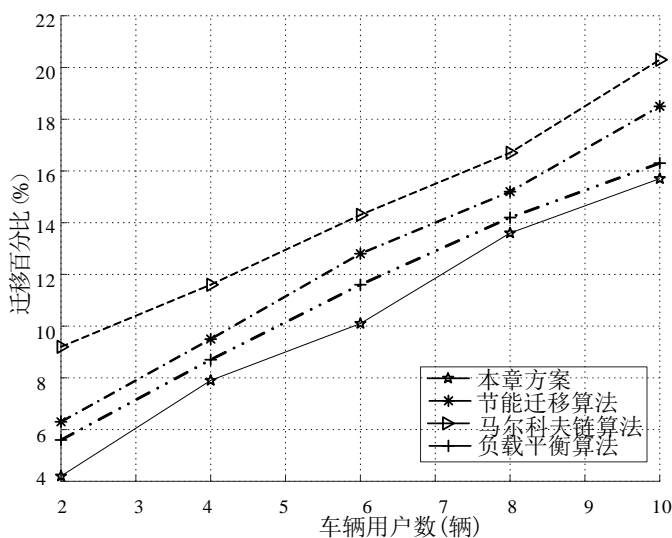


图 3.11 迁移百分比对比图

图 3.12 显示了本章的迁移算法与节能迁移算法、马尔科夫链算法和负载均衡算法等策略的迁移失败百分比对比。本章的算法迁移失败百分比比较小，这是因为它与其它方法相比，减少了任务的迁移次数，从而避免了一些不必要的迁移。本章通过路径选择算法，在已知原始路边单元 MEC 服务器和目的路边单元 MEC 服务器的位置后，找到了最短的迁移路径，进而降低了迁移失败百分比。VM 迁移的时间也因此得到减少，同时也减少了服务延迟，优化了用户体验。

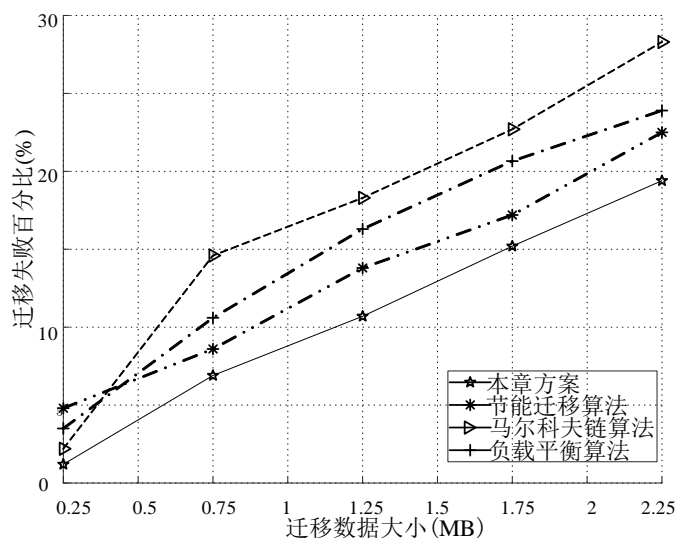


图 3.12 迁移失败百分比对比图

3.5 本章小结

该章节研究了基于 MEC 的软件定义车联网任务迁移问题。通过 Dijkstra 路由算法找到本次任务迁移的最佳路径，并通过 Q-learning 算法来优化最佳路径上的任务迁移时延。仿真结果表明本章的算法优于其它方案，任务延迟显著降低。未来的工作将继续开展车联网中关于任务迁移的工作，优化关于用户体验的其它因素。

第4章 基于 NOMA-MEC 的车联网任务卸载、缓存与迁移策略

4.1 引言

随着车联网的快速发展,移动数据流量将呈现爆炸式增长。在用户密集的环境中,大量的数据传输会导致网络拥塞,传统的车联网系统难以为用户提供高质量服务。为此在车联网中引入 MEC 技术,用户可以将计算密集型任务卸载到网络边缘,缓解计算压力,进而减少能耗和任务延迟。然而在基于 MEC 的车联网架构下依然存在多用户卸载时延较高的问题。为了提高多用户场景下的资源利用率和优化卸载时延,NOMA 技术被应用于车联网。

在基于 NOMA-MEC 的车联网用户任务卸载过程中,当多个用户产生相同的任务卸载请求时,没有必要对此进行重复计算,由此引入任务缓存机制。此外,当车辆移动时会产生任务迁移问题。VM 迁移技术可以促进负载均衡、方便故障管理和服务器维护。它允许任务以 VM 的形式在共享的硬件平台上执行,进而解决任务迁移问题。

虽然关于任务缓存与任务迁移的研究工作较多,但是共同考虑任务卸载、缓存与迁移的工作较少。由于车辆的移动性以及车联网中请求内容的不断增长,会出现多个用户同时进行任务卸载、缓存与迁移的场景,这很容易造成数据拥堵,进而导致任务处理时延过高。综上所述,本章主要的工作为:

1. 构建一个多小区多用户的软件定义车联网网络场景。SDN 控制器可以集中调度车辆与基站信息,每个基站配置 MEC 服务器和缓存容器。
2. 研究用户在车辆移动期间任务协作卸载时延以及任务缓存迁移联合时延优化的问题,将任务时延优化问题建模为一个混合整数非线性问题。
3. 为了解决上述问题,提出一个两阶段算法。首先通过合作博弈建模,对车辆用户进行分组处理,优化任务卸载时延。然后对用户分组后上传至 MEC 服务器的任务进行动作集和状态集建模,利用 Q-learning 算法优化任务缓存迁移联合时延。

图 4.1 为车联网任务卸载、缓存与迁移策略流程框图。

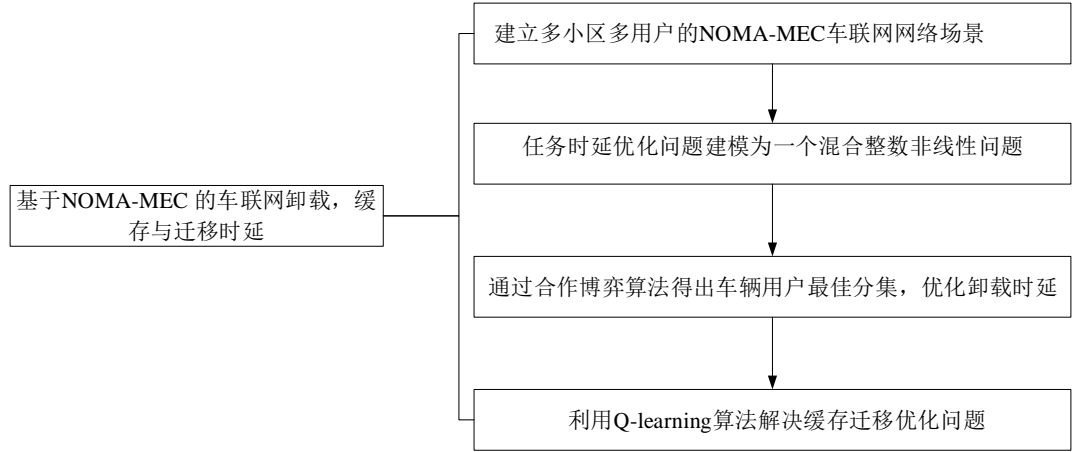


图 4.1 车联网任务卸载、缓存与迁移策略流程框图

4.2 系统模型

4.2.1 网络模型

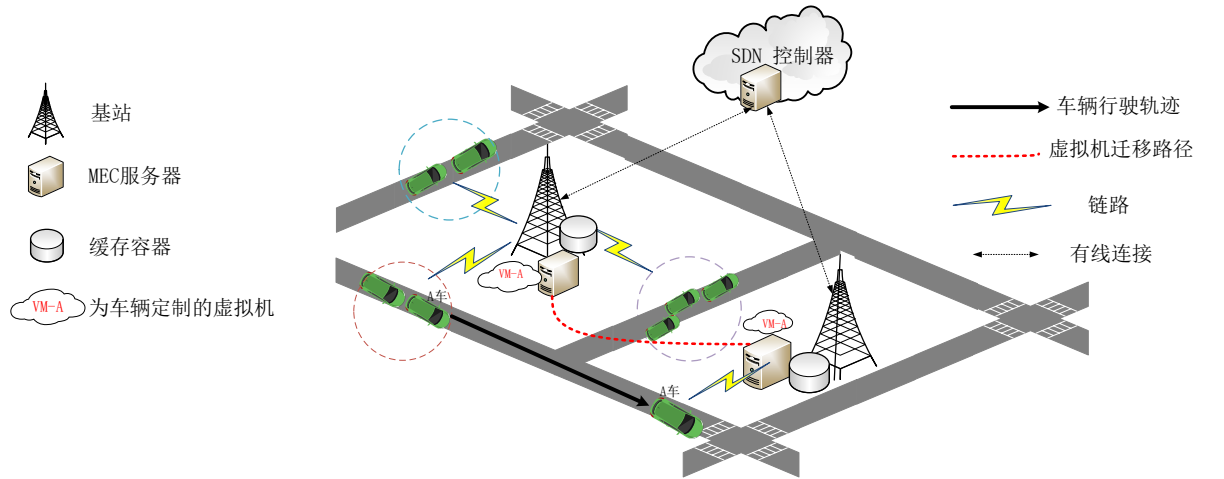


图 4.2 系统模型图

系统模型如图 4.2 所示。有 M 个基站，每个基站部署一个 MEC 服务器，MEC 服务器可以提供缓存或迁移服务，表示为 $Z = \{z_1, z_2, \dots, z_m, \dots, z_M\}$ ，其中 $1 \leq m \leq M$ ， z_m 表示第 m 个服务器（基站）。附近有 A 个车辆用户，表示为 $V = \{v_1, v_2, \dots, v_A\}$ ，且 $A \geq M$ 。车辆的任务模型为 $\Gamma_i(D_i, C_i)$ 。其中 D_i 表示计算任务 Γ_i 的输入数据大小， C_i 表示完成任务 Γ_i 所需的 CPU 周期数。

在基于 NOMA 技术的车联网中，多个车辆用户向同一个基站 z_m 卸载任务时，可以共享同一个子载波。MEC 服务器接收到的用户信号不仅包含该用户的信号，还包含使用同一子载波的其他用户的干扰信号。假设有 N 个子载波，把车辆用户分成

N 组, 每组用户共享同一子载波。也就是向同一个服务器卸载任务的多个用户分成一个组, 车辆用户分组集为 $S = \{S_1, S_2, \dots, S_n, \dots, S_N\}$, 其中 S_n 表示使用第 n 个子载波的车辆用户分组, 该分组中车辆用户数为 A_n , 并且满足 $\sum_{n=1}^N A_n = A$ 。信道增益为 $h_{n,z_m} = \{h_{1,n,z_m}, h_{2,n,z_m}, \dots, h_{A_n,n,z_m}\}$, 其中 h_{a,n,z_m} 表示车辆用户 $a (a \in S_n)$ 在基站 z_m 卸载任务时, 在子载波 n 上的信道增益。

4.2.2 传输模型与卸载模型

当多个车辆用户向同一个基站 z_m 卸载任务时, 假设基站具有 N 个子载波的全信道状态信息(Channel State Information, CSI)。在用户分组 S_n 中, 根据信道增益对用户进行排序, 即 $|h_{1,n,z_m}|^2 \leq \dots \leq |h_{A_n,n,z_m}|^2$ 。

服务器应用 SIC 来解码每个用户的信号, 在第一阶段, 解码信道增益最高的用户数据, 将来自所有其它用户的信号视为干扰。接收机对该用户的数据解码后, 就可以重构出其信号, 并从接收到的总信号中减去该信号。下一阶段解码增益次高的用户数据, 重复此步骤直到最后一个用户, 即最后一个用户只有高斯噪声干扰。因此, 当用户 a 将信号上传给基站 z_m 的服务器后, 用户的信干噪比描述为:

$$SINR_{a,n,z_m} = \frac{p_a |h_{a,n,z_m}|^2 d_{a,z_m}^{-\mu}}{\sum_{i=1}^{a-1} p_i |h_{i,n,z_m}|^2 d_{i,z_m}^{-\mu} + \sigma^2} \quad (4.1)$$

其中, p_a 为用户 a 向基站 z_m 上传任务时的发射功率, p_i 为其它用户的发射功率。 σ^2 是汽车终端的高斯白噪声功率。 $d_{a,z_m}^{-\mu}$ 表示路径损耗, 其中 d_{a,z_m} 表示车辆用户 a 与基站 z_m 中 MEC 服务器的距离, $\mu > 2$ 为路径损耗指数。

用户 a 向基站 z_m 卸载任务的上行传输速率为 $R_{a,z_m} = w \log_2(1 + SINR_{a,n,z_m})$, w 表示用户的信道传输带宽。

因此, 在用户 a 向基站 z_m 卸载任务的上行传输时间为:

$$T^L = \frac{D_i}{R_{a,z_m}} \quad (4.2)$$

车辆本地计算任务 Γ_i 时, 车辆本身计算能力为 F_i , 且 $F_i > 0$ 。本地计算时延为:

$$t^l = \frac{C_i}{F_i} \quad (4.3)$$

MEC 服务器计算任务 Γ_i 时, MEC 服务器分配给该任务的计算资源为 F_M 。服务器计算时延为:

$$t^M = \frac{C_i}{F_M} \quad (4.4)$$

4.2.3 迁移模型与缓存模型

与第三章有所不同, 本章考虑任务缓存迁移联合优化, 假设服务器只发生一次任务迁移。主要采用基于预复制迁移的 VM 迁移模型^[62]。

VM 迁移时间分为两个部分。第一部分为内存迁移时间:

$$T_{mig} = \frac{D_i}{L} \times \frac{1 - \lambda^{E+1}}{1 - \lambda} \quad (4.5)$$

其中, $\lambda = \frac{R}{L}$, R 为迁移期间的脏页率, L 为 VM 所处数据中心的网络带宽。 E 为数据拷贝的迭代次数。

第二部分为迁移过程中的停机拷贝时间:

$$T_{down} = \frac{D_i}{L} \times \lambda^E + t_{res} \quad (4.6)$$

其中, t_{res} 为 VM 在目的物理机的重启时间。

当车辆用户向 MEC 服务器请求任务内容时, 若请求的内容在 MEC 服务器已缓存, 则可以减少回程延迟和回程带宽。基于文献[63]可知, 任务请求内容流行程度满足 Zipf 分布。对于任务 Γ_i , 其缓存命中率为:

$$P(\Gamma_i) = \frac{I(\Gamma_i)^{-\varsigma}}{\sum_{i=1}^{N_f} I(\Gamma_i)^{-\varsigma}} \quad (4.7)$$

其中, $I(\Gamma_i)$ 表示任务 Γ_i 受欢迎的等级, N_f 表示内容总数, ς 为 Zipf 分布参数, 一般设置为 0.56^[64]。

用户所请求缓存任务的回程速率为:

$$R_c = \varpi P(\Gamma_i) \quad (4.8)$$

其中, ϖ 为缓存容器中数据平均传输速率。

当任务内容的大小为 D_i , 则其缓存回传时延为:

$$t^b = \frac{D_i}{R_c} \quad (4.9)$$

4.2.4 问题建模

本章的目的是优化用户的任务卸载时延、任务迁移时延与任务缓存时延。当多个用户同时进行任务卸载、缓存与迁移时, 用户首先会检查自身能力是否可支持任务计算, 如若不能, 会通过 NOMA 系统将任务上传给附近的 MEC 服务器。然后, MEC 服务器将会检查缓存池, 若任务计算结果存在, MEC 服务器会将任务结果回传给用户。这一过程中, 由于车辆用户的移动性, MEC 服务器同时也会进行任务迁移。本章考虑任务卸载、任务缓存与任务迁移时延的联合优化, 假设任务只进行一次迁移。整个任务计算时延包括任务卸载时延、任务缓存时延与任务迁移时延, 具体如下:

$$\begin{aligned} \min U(p_a, \lambda, E, \varpi) = \\ \min \sum_{i=1}^{A_h} \left(\frac{D_i}{w \log_2(1 + \text{SINR}_{a,n,z_m})} + \frac{C_i}{F_M} + \frac{D_i}{L} \times \frac{1 - \lambda^{E+1}}{1 - \lambda} + \frac{D_i}{L} \times \lambda^E + t_{res} + \frac{D_i}{\varpi P(\Gamma_i)} \right) \\ \begin{aligned} C1: & 0 < F_M < F_B \\ C2: & 0 \leq p_a \leq P_{MAX} \\ C3: & D_i < A_{LINK} \\ C4: & 0 < \lambda < 1 \\ C5: & 0 < E < E_{MAX} \\ C6: & N_f \leq C_{MEC} \\ C7: & 50 < \varpi < 100 \end{aligned} \end{aligned} \quad (4.10)$$

其中, C1 表示 MEC 服务器分配给该任务的计算资源要小于 MEC 服务器总计算资源 F_B 。C2 表明发射功率 p_a 在允许的最大范围内且非负。C3 表明在 VM 迁移时传输的数据量要小于链路的容量 A_{LINK} 。C4 和 C5 表明了 VM 迁移时间模型参数 λ 和迭代次数 E 的取值范围。C6 表明缓存内容总和 N_f 不宜超过服务器的存储范围 C_{MEC} 。C7 表明数据平均传输速率 ϖ 的取值范围。

4.3 车联网任务卸载、缓存与迁移优化算法

在基于 NOMA-MEC 的车联网系统中，当用户面对计算密集型和时延敏感性任务时，任务处理时延尤为重要，是用户 QoS 的重要保障。任务处理时延主要为任务卸载时延与任务缓存迁移联合时延两个部分，本章将优化算法分为两个阶段。第一阶段，提出基于 NOMA 的合作博弈卸载决策算法，通过子载波之间的协作来提供用户分组方案，从而优化卸载时延；第二阶段，在用户分组后，当任务传至服务器，使用 Q-learning 算法优化任务缓存迁移联合时延。基于 NOMA-MEC 的车联网任务缓存与迁移策略流程如图 4.3 所示。

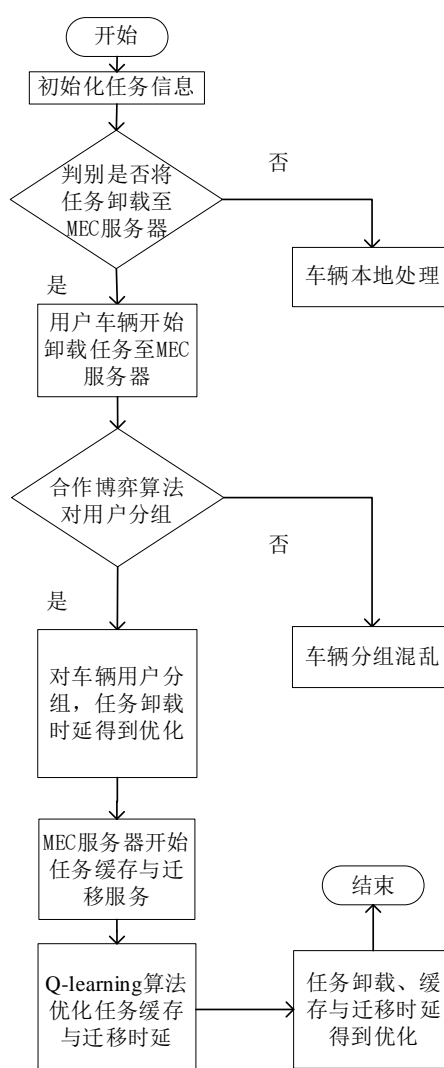


图 4.3 基于 NOMA-MEC 的车联网任务缓存与迁移策略流程框图

4.3.1 基于 NOMA 的合作博弈卸载决策算法

当多个用户同时向基站 z_m 卸载任务时，将用户进行高效分组，即为每个子载波合理地分配用户。合理分组使得 NOMA 系统整体卸载效率提高，实现多用户卸载时延优化。本章采用合作博弈算法，建立博弈论模型 (V, S, ψ) 。 V 是车辆用户集合， $S = \{S_1, S_2, \dots, S_n, \dots, S_g, \dots, S_N\}$ 为每个子载波的用户分组集合， ψ 为效用函数，是用户 $a (\forall a \in S_n)$ 在分组 S_n 的回报价值函数，与卸载时延成反比，即效益越大，卸载时延越小。合作博弈可实现效用函数的最大化，进而优化卸载时延。假设有 N 个子载波，子载波集合为 Y 。分组集合 S 满足 $S_n \cap S_g = \emptyset, n, g \in Y, \forall n \neq g$ 。

效用函数为：

$$\psi(S_n) = \sum_{i=1}^{A_n} \frac{1}{\frac{D_i}{R_{a, z_m}} + \frac{C_i}{F_M}} \quad (4.11)$$

首先针对合作博弈过程，进行相关定义与证明。

定义 1：拆分与合并。在两个分组 S_g 和 S_n 中，如果用户 a 离开 S_n 加入 S_g ，其中 $g \neq n, g, n \in Y$ 。表示如下：

$$\{S_g, S_n\} \rightarrow \{S_g \cup \{a\}, S_n \setminus \{a\}\} \quad (4.12)$$

定义 2： \succ_a 表示车辆用户 a 对于分组 S_g 和 S_n 的偏移度。用户 a 偏移某个分组取决于其效用提升程度。

$$S_g \succ_a S_n \Leftrightarrow \psi(S_n \setminus \{a\}) + \psi(S_g \cup \{a\}) > \psi(S_n) + \psi(S_g) \quad (4.13)$$

定义 3：交换。当定义 2 的条件 $a (\forall a \in S_n)$ 和 $a' (\forall a' \in S_g)$ 都满足后，会产生交换行为，相应地更新分组：

$$\{S_g, S_n\} \rightarrow \{S_g \cup \{a'\} \setminus \{a\}, S_n \setminus \{a'\} \cup \{a\}\} \quad (4.14)$$

在本节基于 NOMA 的合作博弈卸载决策算法中，车辆用户最初随机分配给可用的子载波。在每次迭代中再次随机选择子载波，并且与上次子载波不同，随之发生分组的不同，相继进行上述定义的操作，直到出现最佳的分组，实现多用户卸载时延的优化。

引理 1：从随机分区开始，算法 1 的合作博弈最终分区状态一定收敛。

证明：为了达到效用函数 Ψ 的最优值，车辆用户会不断进行交换，进行合并与离开操作，进而分区集不断的改变。在第 n 次和第 $n+1$ 次迭代中，分区从 S_n 变为 S_{n+1} 。这种操作当且仅当博弈效用 Ψ 严格增加时发生，即：

$$S_n \rightarrow S_{n+1} \Leftrightarrow \Psi(S_n) < \Psi(S_{n+1}) \quad (4.15)$$

因此博弈效用值 Ψ 总是递增的，即：

$$S_{START} \rightarrow S_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_{END} \quad (4.16)$$

其中， S_{START} 和 S_{END} 表示初始分区集和最终分区集。 A 个用户在每个新分区集上的时延将会因为效用函数的升高而降低。因为车辆用户是有限的，所以分组集合数目符合贝尔数^[65]。因此，上述递增顺序会收敛于最终态 S_{END} 。

引理 2：最终态 S_{END} 是趋于稳定的。

证明：假设最终态 S_{END} 不是趋于稳定的。那么，必定会存在用户 $a \in A$ 会趋向于离开当前所在分区加入新的分区，形成新的分区集 S_{NOW} ，即 $S_{NOW} \succ_a S_{END}$ 。这与 S_{END} 是最终态相矛盾，因此最终态 S_{END} 是趋于稳定的。表 4.1 为基于合作博弈算法的卸载决策伪代码，图 4.4 为基于合作博弈算法的卸载决策流程图。

表 4.1 基于合作博弈算法的卸载决策

算法 4.1：基于合作博弈算法的卸载决策

- 1: 用户初始化，采用随机分组 S_{START}
- 2: 定义当前分组 S_{NOW}
- 3: 从分组 $S_n \in S_{NOW}$ 随机挑选用户 a
- 4: 从分组 $S_g \in S_{NOW}$ 随机挑选用户 a'
- 5: **if** 假设交换用户形成分组 S_{new}
- 6: **if** $S_{new} \succ_{a'} S_{NOW}$ **then**
- 7: 用户 a 离开 $S_n \in S_{NOW}$ 加入 $S_g \in S_{NOW}$
- 8: 用户 a' 离开 $S_g \in S_{NOW}$ 加入 $S_n \in S_{NOW}$
- 9: 更新 S_{NOW} ， $S_{NOW} \leftarrow \{S_{NOW} \setminus \{S_n, S_g\}\} \cup \{S_g \cup \{a\} \setminus \{a'\}, S_n \cup \{a'\} \setminus \{a\}\}$
- 10: **else** 假设用户 a' 加入 $S_n \in S_{NOW}$ 形成 S_{new}
- 11: **if** $S_{new} \succ_{a'} S_{NOW}$ **then**
- 12: 用户 a' 离开 $S_g \in S_{NOW}$ 加入 $S_n \in S_{NOW}$
- 13: 更新分组 S_{NOW} ， $S_{NOW} \leftarrow \{S_{NOW} \setminus \{S_n, S_g\}\} \cup \{S_g \setminus \{a'\}, S_n \cup \{a'\}\}$
- 14: 重复迭代，最终趋于稳定

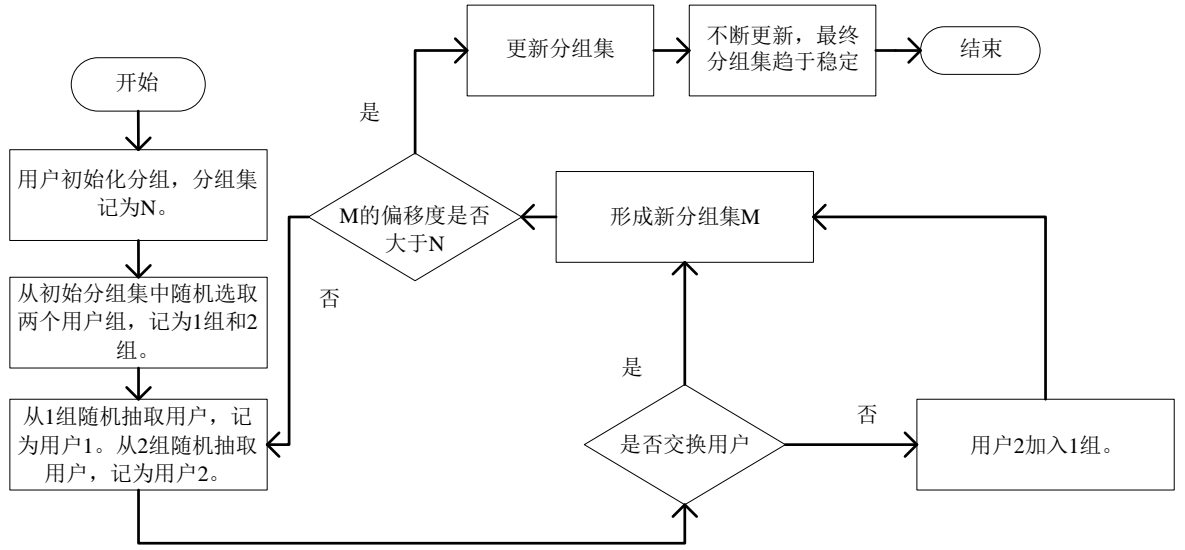


图 4.4 基于合作博弈算法的卸载决策流程图

4.3.2 基于 Q-learning 的任务缓存与迁移优化算法

通过上一小节的合作博弈卸载决策算法，在车辆用户进行任务卸载时，可以得到最佳车辆用户分组，实现卸载时延的优化。车辆用户分组后，用户会将任务卸载给 MEC 服务器，MEC 服务器将会检查缓存池，若任务结果存在，MEC 服务器会将任务结果回传给用户。这一过程中，由于车辆用户的移动性，MEC 服务器同时也会进行任务迁移，本节采用 Q-learning 算法优化服务器上的任务缓存迁移联合时延。Q-learning^[66]作为一种增强学习方法，通过动作集和状态集的建模，可以实现奖励的最大化，将奖励设置为任务缓存迁移联合时延的倒数，通过不断迭代达到收敛后，实现奖励最大化，得出最小的任务缓存迁移联合时延。与第三章 Q-learning 算法有所不同，本章 Q-learning 算法用于单个服务器的内部迭代运算，对附近服务器只发生一次任务迁移。具体情况如下：

任务缓存迁移联合时延函数为：

$$\chi(\lambda, E, \varpi) = \sum_{i=1}^{A_n} \left(\frac{D_i}{L} \times \frac{1-\lambda^{E+1}}{1-\lambda} + \frac{D_i}{L} \times \lambda^E + \frac{D_i}{\varpi P(\Gamma_i)} \right) \quad (4.17)$$

引理：优化目标函数(4.17)为非凸优化问题。

证明：

$$\frac{\partial^2 \chi}{\partial \lambda^2} = \frac{2\lambda^{(E-1)E(E+1)} - \lambda^{E(E-2)(E-1)}}{\lambda-1} - \frac{2(\lambda^E - 2\lambda^{E+1} + 1)}{(\lambda-1)^3} + \frac{2\lambda^{E(E-1)} - 2\lambda^{E(E+1)}}{(\lambda-1)^2} \quad (4.18)$$

$$\frac{\partial^2 \chi}{\partial \lambda \partial E} = 2\lambda^E - \lambda^{E-1} - \lambda^{(E-1)E \ln(\lambda)} + \frac{2\lambda^{E(E+1) \ln(\lambda)}}{\lambda-1} - \frac{2\lambda^{(E+1) \ln(\lambda)} - \lambda^{E \ln(\lambda)}}{(\lambda-1)^2} \quad (4.19)$$

$$\frac{\partial^2 \chi}{\partial E \partial \lambda} = 2\lambda^E - \lambda^{E-1} - \lambda^{(E-1)E \ln(\lambda)} + \frac{2\lambda^{E(E+1) \ln(\lambda)}}{\lambda-1} - \frac{2\lambda^{(E+1) \ln(\lambda)} - \lambda^{E \ln(\lambda)}}{(\lambda-1)^2} \quad (4.20)$$

$$\frac{\partial^2 \chi}{\partial E^2} = -\frac{\lambda^{E \ln \lambda^2} - 2\lambda^{(E+1) \ln \lambda^2}}{\lambda-1} \quad (4.21)$$

$$\frac{\partial^2 \chi}{\partial \varpi^2} = \frac{2}{\varpi^3} \quad (4.22)$$

其 Hessian 矩阵为:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \chi}{\partial \lambda^2} & \frac{\partial^2 \chi}{\partial \lambda \partial E} & 0 \\ \frac{\partial^2 \chi}{\partial E \partial \lambda} & \frac{\partial^2 \chi}{\partial E^2} & 0 \\ 0 & 0 & \frac{\partial^2 \chi}{\partial \varpi^2} \end{bmatrix} \quad (4.23)$$

在 $0 < \lambda < 1$, $0 < \varpi < \varpi_{max}$, $0 < E < E_{MAX}$ 且 E 为整数的条件下, $\frac{\partial^2 \chi}{\partial E^2} < 0$, $\frac{\partial^2 \chi}{\partial \lambda^2} > 0$,

$\frac{\partial^2 \chi}{\partial \lambda \partial E} = \frac{\partial^2 \chi}{\partial E \partial \lambda}$, $\frac{\partial^2 \chi}{\partial \varpi^2} > 0$ 。得出:

$$|\mathbf{H}| = \frac{\partial^2 \chi}{\partial E^2} \times \frac{\partial^2 \chi}{\partial \lambda^2} \times \frac{\partial^2 \chi}{\partial \varpi^2} - \frac{\partial^2 \chi}{\partial \lambda \partial E} \times \frac{\partial^2 \chi}{\partial E \partial \lambda} \times \frac{\partial^2 \chi}{\partial \varpi^2} < 0 \quad (4.24)$$

由上述证明可知,优化目标函数(4.17)为非凸优化问题。可采用 Q-learning 算法来求解此问题,建立状态集 $\xi = \{R, L, S_{END}\}$, 动作集 $A = \{\lambda, E, \varpi\}$ 。其中, R 为迁移期间的脏页率, L 为 VM 所处数据中心的网络带宽, S_{END} 为车辆用户最终分组集, λ 为迁移模型参数, E 为数据拷贝的迭代次数, ϖ 为缓存模型中数据平均传输速率。由上一小节中的数学模型可知,动作集会导致状态集的变化,在算法不断迭代后,可达到收敛,实现奖励最大化。

Q-learning 算法奖励设为:

$$r = \frac{1}{\chi(\lambda, E, \varpi)} \quad (4.25)$$

即任务缓存迁移时延的倒数，这样 Q-learning 算法奖励最大化即为任务缓存迁移联合时延最小化。

Q-learning 算法数学模型为：

$$Q(\xi, a) \leftarrow (1 - \alpha)Q(\xi, a) + \alpha[r(\xi, a) + \gamma \max_{a'} Q(\xi', a')] \quad (4.26)$$

其中， ξ 表示当前的状态， ξ' 表示下一状态。 a 表示当前状态选择的动作， a' 表示下一状态选择的动作。 α 是学习率，且 $0 < \alpha < 1$ ，本章将 α 设置为 0.01。 $r(\xi, a)$ 表示在状态 ξ 下执行动作 a 将得到的回报。 γ 表示折扣因子，决定时间的远近对回报的影响程度，本章将 γ 设置为 0.9。 $Q(\xi, a)$ ， $Q(\xi', a')$ 表示在相应状态动作下的价值回报。

为了提高算法的稳定性，采用状态集和动作集分别收敛的方法。MEC 服务器记录状态动作对应的执行次数及其状态动作回报。当这个状态动作回报收敛时，把之前回报的平均值视为最新的状态动作回报，并停止对这个状态动作回报的更新。即：

$$|Q_n(\xi, a) - Q_{n-1}(\xi, a)| \leq \vartheta \quad (4.27)$$

$$\phi_{n-1}(\xi, a) = \sum_{m=1}^{n-1} Q_m(\xi, a) \quad (4.28)$$

$$Q(\xi, a) = \frac{\phi_{n-1}(\xi, a)}{n-1} \quad (4.29)$$

其中， $Q_n(\xi, a)$ 是第 n 次的价值回报， $\vartheta \approx 1$ 为收敛误差参数。 $Q(\xi, a)$ 为收敛后固定的状态动作回报。 $\phi_{n-1}(\xi, a)$ 为记录前 $n-1$ 次迭代的效益累计值。当某个状态动作回报与上一次回报值的差异足够小时，就认为该状态动作回报达到收敛状态。

由第一个算法得知车辆用户的最佳分组 S_{END} ，在此基础上，建立状态集 ξ ，通过动作集 A 的不断更新，得到奖励集 r 。最后不断迭代学习，系统会得到最佳的动作参数指标，从而优化缓存迁移联合时延。表 4.2 为基于 Q-learning 的任务缓存与迁移优化算法伪代码。

表 4.2 基于 Q-learning 的任务缓存与迁移优化算法

算法 4.2: 基于 Q-learning 的任务缓存与迁移优化算法

- 1: 输入：根据合作博弈算法得到最后分组 S_{END} ，更新状态集 ξ ，加入动作集 A
- 2: 输出：奖励 r
- 3: 初始化 Q 矩阵
- 4: **for** 每一次探索
- 5: 随机选择一个初始状态 ξ ，随机选择动作。

表 4.2 （续）

6:	利用选择的动作进入下一个状态 ξ' ， $\xi \rightarrow \xi'$ 。
7:	多次探索后， Q 矩阵趋于收敛
8: Until	到达最终收敛状态

4.3.3 时间复杂度

本章通过一个两阶段算法来优化任务时延函数。第一阶段利用基于 NOMA 的合作博弈卸载决策算法优化卸载时延，每次迭代进行 n 次计算操作，当算法迭代 n 次时，时间复杂度为 $O(n^2)$ 。第二阶段通过 Q-learning 算法优化分组用户的任务缓存迁移联合时延，算法时间复杂度为 $O(n)$ 。整个算法时间复杂度为 $O(n^2 + n) = O(n^2)$ 。

4.4 仿真分析

本节基于 MATLAB 平台对基于 NOMA-MEC 的车联网任务卸载、缓存与迁移策略进行验证。根据 MEC 白皮书和 IEEE 802.11p 标准，相关参数如表 4.3 所示。

表 4.3 仿真参数表

参数	值
输入数据的大小/kb	1-700
噪声功率/dBm	-114
车辆用户的 CPU 周期频率/GHz	0.5-1
传输带宽/MHz	20
最小脏页率/Mbps	5
车辆最大传输功率/dbm	20
任务所需 CPU 周期数/GHz	0.1-1
最小 λ	0.2
最大迭代次数	30
MEC 服务器 CPU 周期频率/GHz	6
路径损耗指数 α	3
子载波个数 N	4
缓存内容总数 N_f	10-50
数据平均传输速率 ϖ /Mbps	50-100

图 4.5 是本章方案、全部本地计算方案和全部卸载方案的平均时延对比图，由于车辆本地计算不涉及迭代运算，因此不受迭代次数影响。当迭代次数较少时，全部卸载到 MEC 服务器时延最高。这是因为车辆自身有一定计算能力，没有必要进行任务全部卸载。随着迭代次数的增多，车辆用户计算能力的不足会凸显出来。因此，车辆会将部分任务卸载到服务器进行计算，利用服务器强大的计算能力来缩短任务计算时间。所以随着迭代次数的增多，本章方案的平均时延会比其它两个方案少，这是因为本章方案不是单一的全部卸载，通过合作博弈，车辆用户关联会发生变化，迭代次数越多，分区集趋于最优，时延会更小。

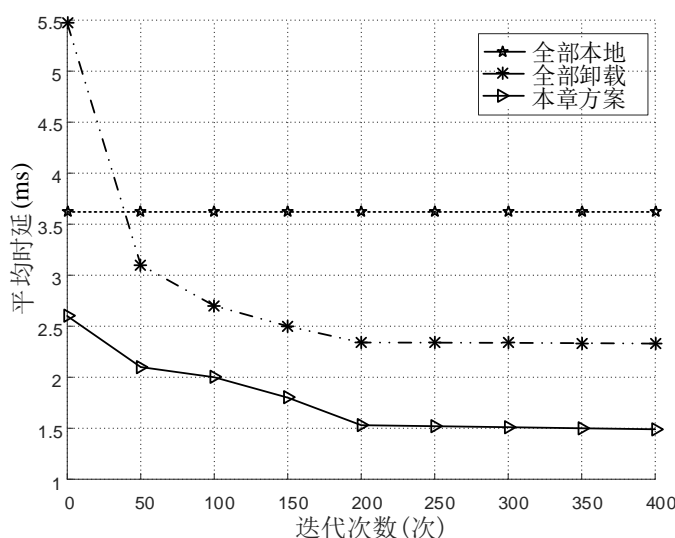


图 4.5 迭代次数与平均时延对比图

图 4.6 显示了基于 NOMA-MEC 系统部分卸载方案（本章方案）、基于 NOMA-MEC 系统全部卸载方案及基于 OFDM-MEC 系统部分卸载和全部卸载方案的性能对比，即时延与数据大小的关系。首先，可以观察到，对于所有方案，计算延迟都随着数据量的增加而增加。在基于 NOMA 系统的方案中，与部分卸载方案相比，将计算完全卸载到 MEC 服务器会导致更大的延迟。因此，将计算任务划分为卸载计算和本地计算有助于更快地完成任务。

图 4.7 是数据大小与平均时延对比图。本章对比方案为无缓存迁移方案，仅迁移无缓存方案以及基于贪婪算法的缓存迁移策略。首先，从图中得知，随着数据大小的增加，计算时延都会增多。无缓存迁移方案时延是最大的，因为它没有合理地利用 MEC 服务器的计算资源。其次，若只迁移无缓存，会增加服务器的计算量进而导致时延增多。本章方案通过 Q-learning 算法优化缓存迁移时延，它通过查看 Q

表来找到最佳策略,使累积奖励最大化。基于贪婪算法(Greedy Algorithm, GA)的方案,选择当前状态下的最佳决策,而不考虑整个系统的最佳决策,容易陷入局部最优,因此性能差于本章方案。

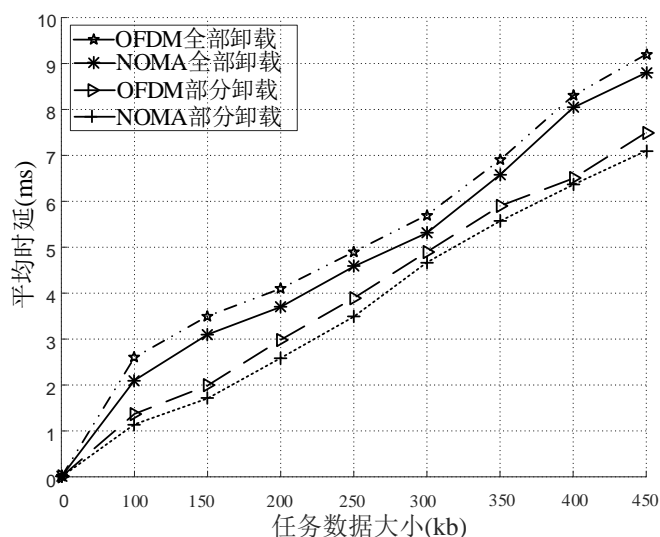


图 4.6 任务数据大小与平均时延对比图

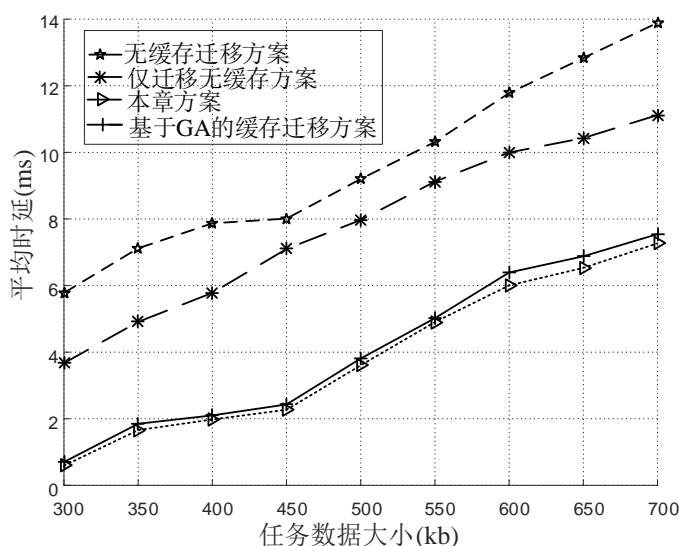


图 4.7 数据大小与平均时延关系图

图 4.8 是 NOMA 模式和 OFDM 模式中缓存内容总数与用户平均时延关系图,从图中得知,随着缓存内容总数的增多,用户平均时延会减少,这是因为缓存内容的增多,可避免重复执行相同任务,从而简化卸载过程,进而减少了 MEC 服务器计算资源的消耗,因此用户的时延会逐渐减少。还可以看出 NOMA 模式的时延会小于 OFDM 模式,这是因为在相同条件下, NOMA 可以承载更多用户,那么相同内容的访问概率会增加,进而减少重复运算,因此其平均时延会小于 OFDM 模式。

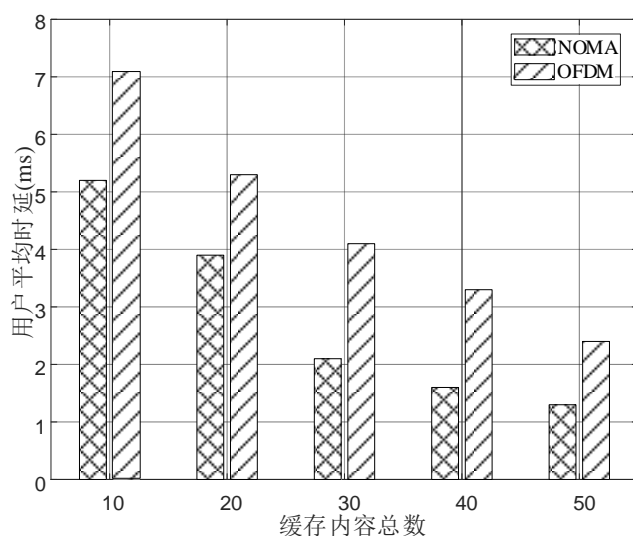


图 4.8 缓存内容总数与用户平均时延关系图

本章与如下策略进行比较，任务缓存卸载方案^[67]、协作边缘计算方案^[68]和安全任务卸载方案^[69]。任务缓存卸载方案将流行的任务缓存于基站，关键任务缓存于就近的 MEC 服务器，提高了缓存命中率。协作边缘计算方案通过 MEC 服务器协作计算来减少能量消耗。安全任务卸载方案利用加密算法对任务进行加密解密，提高了任务的安全性。但是它们都忽略了任务时延的影响。

图 4.9 是本章方案与任务缓存卸载方案、协作边缘计算方案、安全任务卸载方案在不同任务数据大小下的卸载时延对比图。图中可以得知，与其它方案相比，本章方案时延消耗较小，这是因为本章通过合作博弈得到的最佳车辆分组集来进行任务卸载，使得卸载过程更加高效，因此效率会高于对比方案。

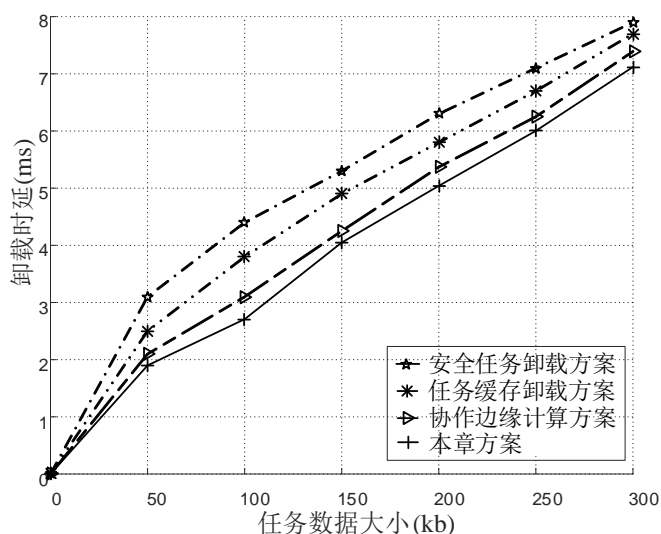


图 4.9 任务数据大小与卸载时延对比图

图 4.10 是本章方案与任务缓存卸载方案、协作边缘计算方案、安全任务卸载方案在不同车辆用户数量下的总体时延对比图。从图中可以得知，随着车辆用户的增加，本章方案与协作边缘计算方案时延会低于其它方案，这是因为协作边缘计算方案采用了 MEC 服务器之间的协作计算技术，在用户增多的情况下，效率会高于其它方案，但是时延依然不是最优。而本章方案在多用户场景下，不仅考虑到基站之间的协作计算，还采用了任务缓存技术，使得服务器计算资源得到充分合理地分配，因此时延会小于其它方案。

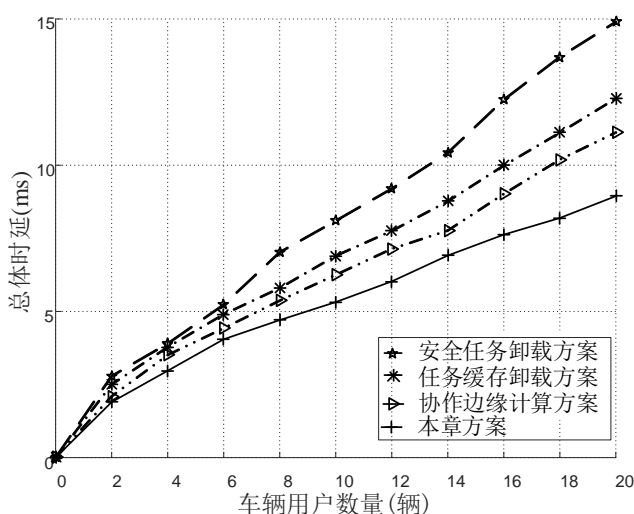


图 4.10 车辆用户数量与总体时延对比图

4.5 本章小结

本章研究了基于 NOMA-MEC 的车联网任务卸载、缓存与迁移策略问题，研究目标是针对基于 NOMA-MEC 的软件定义车联网中任务卸载、缓存与迁移时延的优化。本章通过合作博弈得出最佳的车辆用户分组，从而优化卸载时延，并通过 Q-learning 算法来优化 MEC 服务器中任务缓存迁移联合时延。仿真结果表明本章的算法效果较好，任务时延得到了优化。未来的工作将继续开展车联网中关于 QoS 的研究，优化除时延外关于用户体验的其它因素。

第5章 总结与展望

5.1 全文工作总结

车联网可以减少道路交通事故,并为乘客和驾驶员提供便利,其支持并运行 ITS 中流量优化和紧急警告消息等应用程序。然而这些程序都属于计算密集型和时延敏感型任务,车辆本身计算能力有限,难以满足 QoS。为此引入 MEC 技术,车辆可以将任务卸载到处于网络边缘的 MEC 服务器,进而缓解自身的计算压力。为了满足任务需要,车联网体系结构还必须支持灵活性、可编程性、可靠性和可伸缩性。因此将其与 SDN 架构相结合,可实现在通用硬件上执行控制软件,提高了传统车联网的拓展性和伸缩性。本文在基于 MEC 的软件定义车联网络系统下,对车辆任务卸载与迁移机制的问题进行探究,主要工作创新内容如下:

1. 在基于 MEC 的软件定义车联网架构下,针对用户任务迁移优化问题展开。对系统模型进行数学建模,建立目标函数,通过证明得知该目标函数优化问题为非凸优化问题。首先通过 Dijkstra 路由算法在城市分布模型中找到本次任务迁移的最佳路径,之后利用基于 Q-learning 算法的迁移方案对迁移时延进行优化。仿真结果表明,本策略可以有效降低用户任务迁移时延。

2. 在基于 MEC-NOMA 的软件定义车联网系统中,当多用户同时进行任务卸载、缓存与迁移服务时,会面临任务时延较高的问题,为此提出了一种基于博弈论和 Q-learning 的任务卸载、缓存与迁移优化方案。首先,通过数学建模得出目标函数;其次,当车辆用户竞争频谱资源时,通过合作博弈算法对车辆用户进行分组,提高了卸载效率,实现了卸载时延优化;最后,通过建立动作集和状态集,利用 Q-learning 算法优化缓存迁移联合时延。仿真结果表明,本方案可以有效提高卸载效率,降低任务时延。

5.2 研究工作展望

上述工作主要针对基于 MEC 的车联网中关于任务卸载与迁移机制的研究,帮助车辆在行驶期间获得 QoS,优化用户体验。本文主要考虑了任务卸载、任务迁移、任务缓存、多用户上行传输信道资源,MEC 服务计算资源等约束条件,但是还有很

多影响通信质量的因素没有考虑到。因此，在未来的工作展望中，主要研究方向如下：

1. 基于 MEC 的车联网系统用户能耗的优化研究。随着车辆自动驾驶和电气化的发展，电动汽车相对于传统车辆的优势愈发明显，对于电动车辆，主要动力来源是电池，可是用户自身电池容量有限，所以必须考虑用户能耗的优化问题，通过优化计算资源分配，来降低用户的计算开销。因此，如何制定合理的策略来降低用户的能耗是值得研究的关键问题。

2. 基于车辆行驶轨迹的车联网通信研究。在车辆系统通信环境下，通信及计算资源有限，由于车辆的行驶轨迹导致网络拓扑结构的不断变化，当车辆数目增多时，网络性能会受到极大的影响。因此，在考虑车辆行驶轨迹的情况下，如何制定有效的资源分配策略及车辆任务卸载决策是值得研究的关键问题。

参考文献.

- [1] 李波, 刘雪, 冯菁翠, 等. 5G 蜂窝网辅助的车载自组网数据传输机制与路由算法[J]. 电子科技大学学报, 2021, 50(3): 321-331.
- [2] 李智勇, 王琦, 陈一凡. 车辆边缘计算环境下任务卸载研究综述[J]. 计算机学报, 2021, 44(5): 963-982.
- [3] He Yu, Si P., Sun Enchang, et al. Joint task management in connected vehicle networks by software-defined networking, computing and caching[C]// 2017 International Conference on Network and Information Systems for Computers (ICNISC). Shanghai: IEEE Press, 2017: 47-50.
- [4] Cisco Visual Networking Index: Global mobile data traffic forecast update 2017–2022 White Paper [OL]. URL: [2019-2].
www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.
- [5] 刘明月, 涂崎, 汪洋, 等. 移动云计算卸载技术研究现状及其在电网中的应用[J]. 电力信息与通信技术, 2021, 19(1): 49-56.
- [6] Satyanarayanan M., Bahl P., Caceres R., et al. The case for VM-based cloudlets in mobile computing[J]. IEEE Pervasive Computing, 2009, 8(4): 14-23.
- [7] Barbarossa S., Sardellitti S., Lorenzo P. Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks[J]. IEEE Signal Processing Magazine, 2014, 31(6): 45-55.
- [8] Chiang M., Zhang Tao. Fog and IoT: An overview of research opportunities[J]. IEEE Internet of Things Journal, 2016, 3(6): 854-864.
- [9] 贾维嘉, 周小杰. 雾计算的概念、相关研究与应用[J]. 通信学报, 2018, 39(5): 153-165.
- [10] Mach P., Becvar Z. Mobile edge computing: A survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [11] 曹咪, 徐雷, 陶冶. 雾计算认证与隐私保护研究综述[J]. 信息通信技术, 2018, 12(6): 25-33.
- [12] 王寒松. 车联网中基于 MEC 的计算任务卸载策略研究[D]. 北京: 北京邮电大学, 2019.
- [13] Kekki S., Featherstone W., Fang Y., et al. MEC in 5G networks[R]. ETSI White Paper, 2018, 28: 1-28.

- [14] 张建明, 赵玉娟, 江浩斌, 等. 车辆自组网的位置隐私保护技术研究[J]. 通信学报, 2012, 33(8): 180-189.
- [15] Govindarajulu S., Alwan E. Range optimization for DSRC and 5G Millimeter-Wave Vehicle-to-Vehicle communication link[C]// 2019 International Workshop on Antenna Technology (IWAT). Miami: IEEE Press, 2019: 228-230.
- [16] Campolo C., Molinaro A., Berthet A., et al. On latency and reliability of road hazard warnings over the cellular V2X sidelink interface[J]. IEEE Communications Letters, 2019, 23(11): 2135-2138.
- [17] Toukabri T., Said A., Abdelrahman E., et al. Cellular vehicular networks (CVN): ProSe-based ITS in advanced 4G networks[C]// 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems. Philadelphia: IEEE Press, 2014: 527-528.
- [18] 谷晓会, 章国安, 孔德尚. 基于 SDN 和 MEC 的 5G VANET 架构及其性能分析[J]. 电讯技术, 2019, 59(11): 1332-1337.
- [19] Lee J., Kim D., Lee J. ZONE-based multi-access edge computing scheme for user device mobility management[J]. Applied Sciences-basel, 2019, 9(11): 2308-2316.
- [20] 李景博. 运营商基于 SDN 承载边缘计算业务的设计及演进的思考[J]. 信息通信, 2020(12): 74-76.
- [21] 谷晓会, 章国安. SDN 在车载网中的应用综述[J]. 计算机科学, 2020, 47(1): 237-244.
- [22] Ge Xiaohu, Li Zipeng, Li Shikuan. 5G software defined vehicular networks[J]. IEEE Communications Magazine, 2017, 55(7): 87-93.
- [23] 邵雯娟, 沈庆国. 软件定义的 D2D 和 V2X 通信研究综述[J]. 通信学报, 2019, 40(4): 179-194.
- [24] Liu Jianqi, Wan Jiafu, Zeng Bi, et al. A scalable and quick-response software defined vehicular network assisted by mobile edge computing[J]. IEEE Communications Magazine, 2017, 55(7): 94-100.
- [25] Khan Z., Fan P., Abbas F., et al. Two-level cluster based routing scheme for 5G V2X communication[J]. IEEE Access, 2019, 7(1): 16194-16205.
- [26] Teng Haojun, Liu Wei, Wang Tian, et al. A collaborative code dissemination schemes through two-way vehicle to everything (V2X) communications for urban computing[J]. IEEE Access, 2019, 7(1): 145546-145566.
- [27] 陆忠梅, 陈巍, 魏杰, 等. 车联网极低时延与高可靠通信: 现状与展望[J]. 信号处理, 2019, 35(11): 1773-1783.

- [28] Huang A., Nikaein N., Stenbock T., et al. Low latency MEC framework for SDN-based LTE/LTE-A networks[C]// 2017 IEEE International Conference on Communications. Paris: IEEE Press, 2017: 1-6.
- [29] Tran T., Pompili D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856-868.
- [30] Huang Chungming, Chiang Mengshu, Dao Duytuan, 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(1): 17741-17755.
- [31] Li Meng, Si Pengbo, Zhang Yanhua. Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city[J]. IEEE Transactions on Vehicular Technology, 2018, 67(10): 9073-9086.
- [32] Chen Min, Hao Yixue. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587-597.
- [33] Jiang Yulin, Chen Yashu, Yang Suwei, et al. Energy-efficient task offloading for time-sensitive applications in fog computing[J]. IEEE Systems Journal, 2019, 13(9): 2930-2941.
- [34] Sopin E., Daraseliya A., Correia L. Performance analysis of the offloading scheme in a fog computing system[C]// Proceedings of the 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). Moscow: IEEE Press, 2018: 1-5.
- [35] Gao Lingfang, Moh M. Joint computation offloading and prioritized scheduling in mobile edge computing[C]// Proceedings of the 2018 International Conference on High Performance Computing Simulation (HPCS). Orleans: IEEE Press, 2018: 1000-1007.
- [36] Yi Shanhe, Hao Zijiang, Zhang Qingyang, et al. LAVEA: latency-aware video analytics on edge computing platform[C]// Proceedings of 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). Atlanta: IEEE Press, 2017: 2573-2574.
- [37] Hu Junshi, Zhang Heli, Xi Li, et al. Task-aware joint computation offloading for uav-enabled mobile edge computing systems[J]. Springer International Publishing Communications and Networking, 2020, 312(1): 102-116.
- [38] Alfakih T., Hassan M., Gumaei A., et al. Task offloading and resource allocation for

- mobile edge computing by deep reinforcement learning based on SARSA[J]. IEEE Access, 2020, 8(1): 54074-54084.
- [39] Alam M., Hassan M., Uddin M., et al. Autonomic computation offloading in mobile edge for iot applications[J]. Elsevier Journal on Future Generation Computer Systems, 2019, 90(1): 149-157.
- [40] Yapicioglu T., Oktug S. A Traffic-aware virtual machine placement method for cloud data centers[C]// 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing. Dresden: IEEE Press, 2013: 299-301.
- [41] Chen Y., Tsai S. Optimal provisioning of resource in a cloud service[J]. International Journal of Computer Science Issues, 2010, 7(6): 95-100.
- [42] Piao J., Yan J. A network-aware virtual machine placement and migration approach in cloud computing[C]// 2010 Ninth International Conference on Grid and Cloud Computing. Nanjing: IEEE Press, 2010: 87-92.
- [43] Chang Decheng, Xu Gaochao, Hu Liang, et al. A network-aware virtual machine placement algorithm in mobile cloud computing environment[C]// 2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). Shanghai: IEEE Press, 2013: 117-122.
- [44] Sato K., Sato H., Matsuoka S. A model-based algorithm for optimizing i/o intensive applications in clouds using vm-based migration[C]// 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid. Shanghai: IEEE Press, 2009: 466-471.
- [45] Meng Xiaoqiao, Pappas V., Zhang Li. Improving the scalability of data center networks with traffic-aware virtual machine placement[C]// 2010 Proceedings IEEE INFOCOM. San Diego: IEEE Press, 2010: 1-9.
- [46] Abuhussein A., Bedi H., Shiva S. Towards a stakeholder-oriented taxonomical approach for secure cloud computing[C]// 2013 IEEE Sixth International Conference on Cloud Computing. Santa Clara: IEEE Press, 2013: 958-959.
- [47] Anwar M. Virtual firewalling for migrating virtual machines in cloud computing[C]// 2013 5th International Conference on Information and Communication Technologies. Karachi: IEEE Press, 2013: 1-11.
- [48] Zhang Qi, Liu Ling, Ren Yi, et al. Residency aware inter-VM communication in virtualized cloud: performance measurement and analysis[C]// IEEE International Conference on Cloud Computing CLOUD. Santa Clara: IEEE Press, 2013: 204-211.
- [49] Ristenpart T., Tromer E., Shacham H., et al. Hey you get off of my cloud: exploring

- information leakage in third-party compute clouds[C]// Proceedings of the 16th ACM conference on Computer and communications security. Chicago: ACM Press, 2009: 199-212.
- [50] Khari M., Gupta S., Kumar M. Security outlook for cloud computing: a proposed architectural-based security classification for cloud computing[C]// 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). New Delhi: IEEE Press, 2016: 2153-2158.
- [51] Krebs R., Momm C., Kounev S. Architectural concerns in multi-tenant saas applications[J]. CLOSER, 2012, 12(1): 426-431.
- [52] Bauman E., Ayoade G., Lin Zhiqiang. A survey on hypervisor-based monitoring: approaches, applications, and evolutions[J]. ACM Computing Surveys, 2015, 48(1): 1-33.
- [53] Zheng Junjun, Okamura H., Dohi T. Survivability analysis of VM-based intrusion tolerant systems[J]. IEICE Transactions On Information And Systems, 2015, 98(12): 2082-2090.
- [54] Li Guozhi, Yang Zhihua, Chen Shoufeng, et al. A traffic flow-based and dynamic grouping-enabled resource allocation algorithm for LTE-D2D vehicular networks[C]// 2016 IEEE/CIC International Conference on Communications in China (ICCC). Chengdu: IEEE Press, 2016: 1-6.
- [55] Chen Xu, Jiao Lei, Li Wenzhong, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2015, 24(5): 2795-2808.
- [56] Cerroni W. Multiple virtual machine live migration in federated cloud systems[C]// IEEE Conference on Computer Communications Workshops. Toronto: IEEE Press, 2014: 25-30.
- [57] Zhang Haijun, Feng Mengting, Long Keping, et al. Artificial intelligence-based resource allocation in ultradense networks: applying event-triggered q-learning algorithms[J]. IEEE Vehicular Technology Magazine, 2019, 14(4): 56-63.
- [58] Jang B., Kim M., Harerimana G., et al. Q-learning algorithms: a comprehensive classification and applications[J]. IEEE Access, 2019, 7(1): 133653-133667.
- [59] Ksentini A., Taleb T., Chen M. A markov decision process-based service migration procedure for follow me cloud[C]// 2014 IEEE International Conference on Communications (ICC). Sydney: IEEE Press, 2014: 1350-1354.
- [60] Meneguette R., Boukerche A. An efficient green-aware architecture for virtual machine

- migration in sustainable vehicular clouds[J]. IEEE Transactions on Sustainable Computing, 2020, 5(1): 25-36.
- [61] Gamal M., Rizk R., Mahdi H., et al. Osmotic bio-inspired load balancing algorithm in cloud computing[J]. IEEE Access, 2019, 7(1): 42735-42744.
- [62] Wang Huandong, Li Yong, Zhang Ying, et al. Virtual machine migration planning in software-defined networks[J]. IEEE Transactions on Cloud Computing, 2019, 7(4): 1168-1182.
- [63] Li Jun, Chen Wen, Xiao Ming, et al. Efficient Video Pricing and Caching in Heterogeneous Networks[J]. IEEE Transactions on Vehicular Technology, 2016, 65(10): 8744-8751.
- [64] Wang Chenmeng, Liang Chengchao, Yu F., 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.
- [65] Rota G. The number of partitions of a set[J]. The American Mathematical Monthly, 1964, 71(1): 498-504.
- [66] Huang Yungfa, Tan Tanhsu, Wang Nengchung, et al. Resource allocation for D2D communications with a novel distributed q-learning algorithm in heterogeneous networks[C]// 2018 International Conference on Machine Learning and Cybernetics (ICMLC). Chengdu: IEEE Press, 2018: 533-537.
- [67] Hao Yixue, Chen Min, Hu Long, et al. Energy efficient task caching and offloading for mobile edge computing[J]. IEEE Access, 2018, 6(1):11365-11373.
- [68] Misra S., Saha N. Detour: Dynamic task offloading in software-defined fog for IoT applications[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(5):1159-1166.
- [69] Elgendy I., Zhang Weizhe, Tian Yuchu, et al. Resource allocation and computation offloading with data security for mobile edge computing[J]. Future Generation Computer Systems-The International Journal of Escience, 2019, 100(1): 531-541.

致谢

毕业将至，三年研究生时光转瞬即逝，我的心态从最初考上研究生的兴奋逐渐转变为研究生日常生活中的紧张与充实。落笔刹那，回忆如潮涌。我依然可以清晰的想起录取那时的欢欣雀跃，可以想起初入重邮的校园美景带来的惊艳，也可以想起逸夫楼那彻夜的灯光。这每一个时刻都让我对重庆邮电大学这一名字深感自豪。我对我的人生有三年属于重邮而感到荣幸。

这三年我心存感激，首先我要感谢张海波老师、李方伟老师、朱江老师、陈善学老师。张海波老师年轻有为，不仅具备渊博的学识，也不恪守成规，有着年轻人的创新敢拼精神。张老师作为我的引路人，不仅在科研道路上让我受益良多，而且在为人处世方面也是我的榜样。张老师会了解我们的日常生活，会关心学生的情绪变化并加以引导，让学生可以感受到家人般的温暖。李方伟老师是我们团队的主心骨，是一个严厉有原则的前辈，他为我们实验室制定了规范的规章制度，在大局上把控我们的研究动态，为我们团队注入强心剂。朱江老师豪迈爽朗，和学生没有距离感，亦师亦友，在学术探讨和生活交流中为我们提出了很多建议，让我受益匪浅。陈老师和蔼可亲，学富五车，曾经有幸上过他的教学课，可以将复杂的理论知识化解为简单的公式推导，过程依然历历在目。我很荣幸在研究生生涯遇见这些优秀的老师。

其次我也要感谢荆昆仑师兄、许云飞师兄、王子心师姐和程妍师姐。每当我遇到学术上的问题向他们求助时，他们都会细心地为我讲解，而且也在我生活上给予了巨大的帮助。同时也要感谢代超蓝、袁小扣、鲁佳文、董博志等人对我的鼓励与建议，很开心并且荣幸可以和他们共处一个研究团队，可以一起学习生活，共同进步。

还要感谢我的父母，他们在我的生活给予了巨大的帮助，在我人生中关键的转折中提出宝贵的人生经验，让我不再迷茫，可以从容的面对困难。我以后一定好好努力来报答父母。

最后感谢评阅本论文以及参加论文答辩的各位老师，您的意见对本论文的提升至至关重要！特此向你们致敬！

攻读硕士学位期间从事的科研工作及取得的成果

参与科研项目：

- [1] 车联网中基于 MEC 的任务卸载机制研究, 重庆市基础研究与前沿探索项目, cstc2018jcyjAX0463, 2018.08-2021.07

论文：

- [1] 张海波, **张益峰**, 刘开健. 基于NOMA-MEC的车联网任务卸载、迁移与缓存策略[J]. 计算机科学, 2021. (录用)
- [2] Haibo Zhang, **Yifeng Zhang**, Kaijian Liu. A Service Migration Mechanism in Software Defined Vehicular Network Based on Mobile Edge Computing[J]. China Communications, 2021.(在审)

受理专利：

- [1] 张海波, **张益峰**, 许云飞. 一种基于MEC缓存服务的车联网资源分配方法: 中国, 201911298296.9[P]. 2019.

奖励：

- [1] 重庆邮电大学2018学年学业奖学金
- [2] 重庆邮电大学2019学年学业奖学金