

分类号 TN915.5

学 号 1808211505

西安建筑科技大学
学 位 论 文

基于移动边缘计算的车联网

资源分配与任务卸载方法

作者 程鹏

指导教师姓名 张文柱 教授

申请学位级别 硕士 专 业 名 称 智能建筑

论文提交日期 2021.06 论文答辩日期 2021.06

学位授予单位 西安建筑科技大学

答辩委员会主席 冯兴乐

评 阅 人 盲审

盲审

学位论文独创性声明

本人郑重声明：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，除论文中明确注明引用的内容外，论文内容不包含其他个人或集体完成或已发表的研究成果，也不包含本人或其他人已用于其他学位申请的论文和成果。对论文的研究工作做出重要贡献的个人和集体，均已在论文中作了明确的说明并已致谢。

申请学位论文及其相关资料若有不实之处，本人承担一切相关责任。

论文作者签名：程朋

日期：2021年6月11日

学位论文知识产权声明

我们所提交的学位论文及相关的职务作品是在西安建筑科技大学攻读学位期间完成的成果，知识产权归属西安建筑科技大学。论文作者离校或论文指导教师因故离职后使用或发表学位论文或与学位论文直接相关的研究内容时，应取得西安建筑科技大学同意；取得同意后使用或发表时，应署名作者、指导教师及西安建筑科技大学。

我们完全了解学校有关保存、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交学位论文的复印件和电子文档，可以采用影印、缩印或者其它复制手段保存和汇编学位论文，允许本论文被查阅和借阅，学校可以公布论文的全部或部分内容，可在☐一年/☐两年/☐三年（此未勾选的，默认为即时公开）以后授权中国学术期刊（光盘版）杂志社、中国科学技术信息研究所等单位将本学位论文全部内容收录到有关“学位论文数据库”之中，在网络上全文发布并通过网络向社会公众提供信息服务。

保密论文在保密期内遵守有关保密规定，解密后适用于此声明。

论文作者签名：程朋

日期：2021年6月11日

指导教师签名：李之桂

日期：2021年6月11日

基于移动边缘计算的车联网资源分配与任务卸载方法

专 业：智能建筑

硕 士 生：程鹏

指导教师：张文柱教授

摘 要

随着移动边缘计算的逐渐普及，各种对时延有苛刻需求的应用相继产生。虽然云端能为终端用户提供强大的计算服务，但是由于距离遥远无法提供实时的服务。边缘计算是很好的解决办法，移动边缘计算通过将终端用户所产生的任务卸载到边缘服务器上，通过运用虚拟化、软件定义网络等技术，使终端用户能够灵活的获取计算资源等服务。将移动边缘计算应用到车联网架构中能够有效的解决诸多问题。为了能够在接近用户端利用边缘设备的计算资源为人们提供高质量的服务，文本对车联网环境下的计算卸载方法进行了研究和分析，车载应用产生的任务需求可以卸载到配备边缘服务器的路侧单元进行处理，也可以卸载到接近终端用户的车辆中进行处理，从而使车联网的服务效率最大化。

本文根据 K-means 聚类算法对任务卸载进行优化，并对比随机选择算法对系统平均完成时间的影响。本文在 Matlab R2018b 平台上进行仿真实验，结果表明基于 K-means 聚类算法优于随机选择算法，并且验证了聚类算法在本文中的优势。最后针对在车联网环境下的移动边缘计算架构设计了一种联合计算卸载和资源分配机制。在该机制中，将原多目标问题化解为复杂度较低的比例资源分配算法和改进的粒子群算法两个问题分别求解，仿真结果证明了该机制相较于其他算法大大降低了算法的复杂度，并且用户效用始终较高。

关键词：移动边缘计算；车联网；聚类；资源分配；卸载决策

Internet of Vehicles Resource Allocation and Task Offloading Method Based on Mobile Edge Computing

Specialty: Intelligent Building

Candidate: Cheng Peng

Supervisor: Prof.Zhang Wenzhu

Abstract

With the gradual popularity of mobile edge computing, various applications with demanding delay requirements have emerged one after another. Although the cloud can provide powerful computing services for end users, it cannot provide real-time services due to the long distance. Edge computing is a good solution. Mobile edge computing offloads tasks generated by end users to edge servers, and uses technologies such as virtualization and software-defined networking to enable end users to flexibly obtain computing resources and other services. Applying mobile edge computing to the Internet of Vehicles architecture can effectively solve many problems. In order to be able to use the computing resources of edge devices to provide people with high-quality services close to the user side, the text researches and analyzes the computing offloading methods in the Internet of Vehicles environment. The task requirements generated by in-vehicle applications can be offloaded to the roadside unit equipped with edge servers. The processing by the side unit can also be offloaded to a vehicle close to the end user for processing, thereby maximizing the service efficiency of the Internet of Vehicles.

This paper optimizes task offloading based on the K-means clustering algorithm, and compares the effect of random selection algorithms on the average completion time of the system. This paper conducts simulation experiments on the Matlab R2018b platform. The results show that the K-means clustering algorithm is better than the random selection algorithm, and the advantages of the clustering algorithm in this paper are verified. Finally, a joint computing offloading and resource allocation mechanism is designed for the mobile edge computing architecture in the Internet of

Vehicles environment. In this mechanism, the original multi-objective problem is solved into two problems, the less complex proportional resource allocation algorithm and the improved particle swarm algorithm. The simulation results prove that this mechanism greatly reduces the complexity of the algorithm compared to other algorithms, and user utility is always high.

Keywords: Mobile edge computing; Internet of vehicles; Clustering; Resource allocation; Offloading decision

目录

1 绪论.....	1
1.1 研究背景及意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	3
1.2 国内外研究现状.....	4
1.2.1 移动边缘计算研究现状.....	4
1.2.2 车联网研究现状.....	5
1.3 面临的挑战.....	6
1.4 论文主要内容及结构安排.....	6
2 相关方法和概念概述.....	9
2.1 车联网及相关技术介绍.....	9
2.1.1 车联网体系架构.....	9
2.1.2 车联网关键技术.....	10
2.1.3 车联网特点及应用.....	11
2.2 MEC 介绍.....	11
2.2.1 MEC 特点.....	13
2.2.2 MEC 的应用场景.....	14
2.2.3 MEC 系统中理论模型分析.....	17
2.2.4 MEC 关键技术.....	17
2.3 计算卸载及相关知识介绍.....	20
2.3.1 计算卸载的概念.....	21
2.3.2 计算卸载决策.....	21
2.3.3 计算卸载流程.....	22
2.3.4 计算卸载分类.....	24
2.4 本章小结.....	25
3 基于 K-means 聚类的计算卸载算法设计.....	27
3.1 引言.....	27
3.2 计算卸载算法设计.....	28
3.3 实验分析.....	33

3.3.1 仿真环境建立.....	34
3.3.2 仿真参数设置.....	34
3.3.3 对比分析.....	34
3.4 本章小结.....	37
4 联合计算卸载和资源分配算法.....	39
4.1 引言.....	39
4.2 系统模型和优化问题的制定.....	40
4.2.1 系统模型.....	40
4.2.2 通信模型.....	40
4.2.3 计算模型.....	41
4.2.4 最优化问题制定.....	42
4.3 解决方案.....	42
4.4 仿真分析.....	46
4.4.1 平均 CPU 执行时间.....	47
4.4.2 平均用户效用.....	47
4.5 本章小结.....	49
5 总结与展望.....	51
5.1 本文总结.....	51
5.2 工作展望.....	52
参考文献.....	53
致谢.....	61
攻读硕士学位期间取得的研究成果.....	63

1 绪论

1.1 研究背景及意义

1.1.1 研究背景

随着移动边缘计算(Mobile Edge Computing,MEC)技术的逐渐成熟,车联网行业同时也发生了巨大的改变,目前大多数学者研究的方向是关于系统稳定性,其中车辆用户与路侧单元(Road Side Unit,RSU)的信息碎片化问题是需要解决的难题。在车联网环境下的车辆用户主要采用两种通信方式进行通信,即通过专用短距离通信技术在车辆与基础设施通信与车对车的通信^[1]。车辆与基础设施之间(Vehicle-to-Infrastructure,V2I)通信方式主要是与 RSU 进行通信从而达到资源共享的目的。车对车(Vehicle-to-Vehicle, V2V)的通信是通过全球导航定位系统和智能传感器等设备发送车辆用户的位置以及速度信息给邻近的车辆,当发生紧急事件时可以及时的告知对方的车辆以便做出应答。因此,V2V 和 V2I 通信是构成车联网系统的主要技术之一^[2]。相关机构专门为车联网环境下的无线接入要求定制了 IEEE 802.11p 标准^[3]。随着车联网通信网络的大范围普及,其在车载娱乐,交通安全监管以及事故预测警告等方面发挥着重要作用。为了使自动驾驶,增强现实等新型业务的有序开展,在网络边缘侧进行任务处理成为了车联网很有发展前景的目标之一。使用网络边缘侧的计算资源将大大提高终端用户获取服务的效率,并且降低了数据传输和计算过程中所产生的时延以及能耗。

车联网技术的完善与否是走向信息化时代重要因素之一,各大院校以及企业已经预测了车联网的潜在价值。车联网是由车辆中的车载单元产生的信息组成的庞大的信息化网络。通过车辆中的传感器、全球导航定位系统、摄像头等终端设备完成对周边事物中的信息以及路况的探测与感知,并且车辆通过专用短程通信技术与周边的 RSU 等基础设施进行信息共享从而达到统筹全局的目的。随着信息化时代的到来,车辆所扮演的角色已经不仅仅局限于交通出行,车载娱乐、自动驾驶、紧急事件处理、高精度地图等服务是车辆用户追求的方向。然而这些服务对时延与电池寿命有了更加苛刻的需求,以及 RSU 部署的成本、计算资源的匮乏等因素会减缓车联网的发展脚步。车联网环境有许多不确定性,例如,雾霾能见度极低,能见度都达不到 70m,这时车辆用户可以通过 RSU 以及其他的基础设施获取到实时动态信息避免交通事故的产生。当乘客在长途奔波的时候可以通过车载娱乐设备度过漫长的旅途。当然随着车联网技术的拓展,终端设备所产

生的数据以好几个数量级的增长，云数据中心已经无法满足如此多用户的同时接入与访问。最重要的是车辆的高动态性造成通信状态不稳定，以及传统车辆与云数据中心长距离通信产生的不可估量的经济开销和计算资源的浪费，如今传统车联网系统已经无法满足车辆用户的需求。

在 2016 年，高校与通信运营商合作创立了车联网实验室来攻克车联网通信以及边缘计算等相关技术。《智能汽车创新发展战略》中阐述了车联网技术的战略意义与愿景，指出了车联网未来发展的重要性。此外，有关单位划分大约 100MHz 作为车联网无线电通信频段，并且在国内各地成立了车联网边缘计算试验项目。为了进一步推动我国车联网行业的标准化进程，工信部联合国家监管委发表了有关促进车联网体系标准指导方案，其中包含车联网产业链涉及到的部门以及领域。

MEC 标准在 2018 年正式完成，车联网作为低时延高可靠性场景的代表是 MEC 技术典型应用场景之一。《车联网发展创新行动计划》中明确指出要积极推动基于 MEC 的车联网标准化进程，车联网作为 MEC 技术的垂直行业能加快 MEC 基础设施的部署，MEC 技术的融入能满足车辆终端对带宽、时延、可靠性等方面的需求，有助于推动“车路云”协同管理的车联网系统架构的发展。3GPP 提出，要使 MEC 技术能满足各应用场景的需求，需要实现以下性能指标：10GB/s 以上的峰值速度、毫秒级别的时延、99.99%以上的稳定性等。完成上述指标对于 MEC 技术来说无疑是一个巨大的挑战，而多接入移动边缘计算技术是可以有效的协助车联网应用场景提升带宽、降低时延的关键技术之一^[4,5]。MEC 将云服务器的一部分功能迁移到更接近于用户的网络端，因此，大大降低了端到端的时延并且实现了存储、计算、管理的多方面协调管理，MEC 技术的产生为车联网行业提供了前所未有的帮助。本文针对基于移动边缘计算的车联网(Internet of Vehicle, IoV)架构展开了研究。相较于传统的车联网，本文将 MEC 技术引入到车联网环境中解决了云数据中心长距离传输导致的高时延问题并且承担了云中心由于大量智能终端的接入造成的网络堵塞、资源分配不均匀等一部分压力。MEC 技术的发展为 5G 标准的实现提供了强有力的技术支持，并且车辆用户的安全得到保障、出行舒适度大大提升。但理论层面的研究是远远不够的，车联网特有的性质以及 RSU 的部署成本使系统性能的降低，目前学者需要从理论层面到项目落地，从现实层面去思考问题对车联网边缘计算(Vehicle Edge Computing, VEC)的实现是至关重要的，如图 1.1 所示为车联网边缘计算系统架构图。

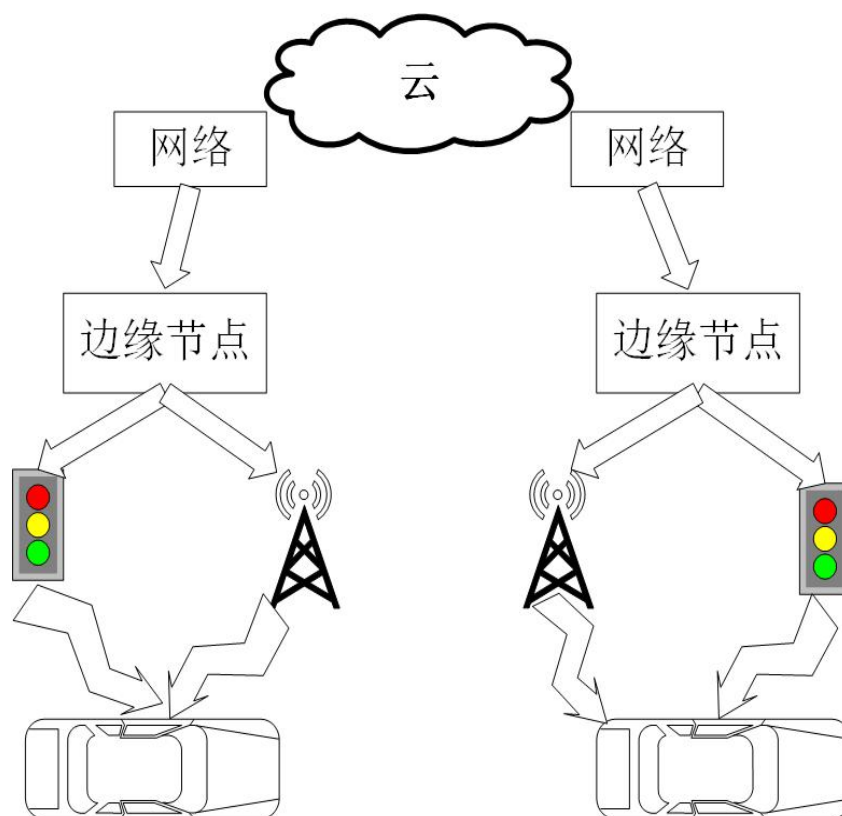


图 1.1 车联网边缘计算系统架构图

随着车载设备的多样化,各种智能传感器设备安装在车辆中,车辆变成了一个可以监控感知周围道路信息和车辆状态的小型信息体,伴随着车辆终端的数据量以指数级上升^[6],并且数据类型的多样化给大数据技术提出了新的挑战^[7]。另一方面由于车辆的高机动性对通信链路产生了较大的困扰。例如,传输干扰、计算迁移等。智能传感器已经成为 MEC 的核心部分之一^[8],基于云端的车联网架构被提出为海量的智能传感器提供技术支持^[9]。基于云端的车联网结合了自组织网络和云计算,终端设备可以将任务卸载到有强大资源的云服务器进行计算,为那些对时延没有苛刻需求的任务提供服务^[10]。过去大部分研究是关于移动车辆通过资源共享,相互协作从而达到互利共赢的目的,例如信息传播和任务分割。然而,由于传输网络的时延、通信链路的干扰以及传输速率的限制,距离云端遥远的车辆用户无法满足计算密集型和时延敏感型的任务需求,从而达不到车辆用户的预期效果。

1.1.2 研究意义

近几年来,车联网行业正进行着重大的改革,许多新型的车载应用和服务相继产生,例如,休闲娱乐、无人驾驶、安全预警、智慧交通、环境感知和智能导

航等。休闲娱乐和智慧交通会涉及到视频和图像处理技术，这对车载终端的处理能力提出了新的挑战。例如，智能导航系统采用数据流处理技术以及增强现实（Augmented Reality, AR），通过智能传感器收集周围车辆的信息以及分析道路交通状况从而为用户做出最合理的决策。这样能减轻司机的一些压力尤其是长期驾驶的车辆用户。因此，将任务迁移到云数据中心去处理会提高车联网的服务质量。

云数据中心提供全球集中式服务，而 MEC 提供的是低时延的本地化服务。MEC 服务层进一步解释了边缘-云之间的关系。在边缘服务器收到来自终端用户的任务请求后将计算任务请求信息根据地理位置以及时效性进行高效的调度，忽略了在本地计算的任务之后，其余的任务卸载到边缘服务器进行处理。MEC 支持新型应用程序和资源管理，并且在云中心和终端用户之间建立了协同管理体系从而进行高效通信。特别是在资源管理和数据分析方面，MEC 为实时数据收集处理和资源统筹提供了良好的环境，同样也支持离散分布和移动性的计算任务请求^[11]，MEC 也被当做云数据中心的延伸。从 IoV 的角度来看，由于网络拓扑结构的不确定性和车辆的移动性给通信链路带来了巨大的挑战，而将车辆自身当作微型服务器可以缓解云端的压力，并且在边缘服务器达到服务阈值时可以在本地处理任务，这样可以充分的利用资源^[12,13]。由于 MEC 计算资源有限、计算任务的异质性以及通信环境的不确定性^[14]，使得资源分配成为 VEC 场景必须考虑的因素之一。

1.2 国内外研究现状

1.2.1 移动边缘计算研究现状

从 2013 年 IBM 与诺基亚联合推出 MEC 以来受到了大众的关注，由于其高带宽、低时延、感知度灵敏等优势是传统技术无可比拟的。在 MEC 环境下计算卸载以及资源分配是研究的热点问题。与云计算有充足资源不同的是，边缘服务器资源有限，由于地理位置的差异性使得终端用户与 MEC 服务器之间负载失衡，有的处于闲置状态而有些处于饱和状态。当多个终端用户同时接入一个 MEC 服务器时会出现任务等待以及资源竞争的现象，因此，有效的计算卸载策略可以提高资源的利用率，均衡负载是 MEC 需要攻克的难题之一。

在异构无线网络中，终端用户通过多种无线接入技术与 MEC 进行通信，如 V2V、V2I、蓝牙以及 zigbee 等技术，信道间的干扰以及通信链路间的损耗是国

内外热点研究问题。移动终端间缺乏有效的通信，在卸载过程中导致通信链路中断将影响车辆用户的服务体验，如何确保车辆终端与 MEC 服务器有效的通信是不得不考虑的问题。

1.2.2 车联网研究现状

近几年来车联网行业发展迅速已经成为全球范围内新的经济增长点。因此，发展 IoV 有助于推动国民生产总值，加快汽车领域的创新与研究。其中，交通基础设施建设以及无人驾驶技术是关键突破口，许多机构正完善 IoV 应用标准以及规范。我国当前已批准天津、江苏等地区作为 IoV 先导区，北京、长沙等地已经对 IoV 基础设施的建设展开了研究^[15]。

在 IoV 环境下的数据传输率与用户和服务器之间的连接稳定性有密切关系，因此，MEC 服务器的选择直接影响到车辆用户与服务器之间的通信效率。由于车联网环境的异构性，稳定的通信链路难以持续，而且部署服务器成本高昂，大范围的在 RSU 中安装 MEC 服务器是个浩大的工程，因此，对 RSU 的部署策略是国内外学者研究的问题之一^[16]。文献[17]针对公路环境下，综合考虑车的数量、链路间干扰以及车速等对通信质量的影响，以直接或者多跳传输方式实现与车辆终端的通信，并且建立最优化模型实现吞吐量最大化。文献[18]提出了一种基于停泊车辆的 MEC 架构，文章将容器技术与 VEC 相结合确保停泊车辆在执行任务时能够快速响应。文献[19]将 MEC 作为技术支撑融入到 IoV 架构中将任务请求进行分组处理，通过与传统车-云架构进行对比分析得出了所提方案的优越性。文献[20]论述了将 MEC 融入 IoV 架构的优势，对如何满足多接入移动边缘计算展开了论述，并且针对多接入移动边缘计算造成的负载不均衡问题提出了一种资源分配和任务卸载方案。文献[21]构造了车辆雾计算的架构，文章表明通过聚集单个车辆的资源能够有效提高车联网的服务效率。文献[22]考虑了一个两用户 MEC 网络，其中每个无线设备都有一系列要执行的任务。通过计算设备间的任务依赖关系，制定了最佳任务卸载策略和资源分配方案使无线设备能耗与任务执行时间的加权总和最小。文献[23]提出了一种自主 VEC 架构，通过收集和分析邻近车辆的信息运用蚁群算法来合理的安排作业。然而这些研究都没有考虑到任务的特殊性，没有充分考虑到任务卸载过程中由于资源分配不均匀导致资源浪费的问题。

1.3 面临的挑战

当前对 VEC 的研究还存在以下不足：

(1) 由于车联网环境的高机动性，以及网络拓扑结构的异构性，对终端用户与 MEC 服务器建立稳定实时的通信连接是一个较大的挑战。

(2) 车联网中通信方式种类繁多，车辆在移动过程中，和其他车辆或者 RSU 进行任务卸载的通信过程中传输速率随着移动过程中相互距离的变化以及车路设备间通信的干扰需要深入研究。

(3) 车联网和一般的移动网络或者静态网络最大的区别是网络拓扑变化快，节点间链路持续时间短暂，节点移动速度快，移动模型限制在路网拓扑这种网状结构中，在任务卸载过程中很容易卸载失败，在部分卸载以及任务间的关联性问题需要进一步研究。

1.4 论文主要内容及结构安排

本文主要研究关于 IoV 环境下的计算卸载和资源分配问题，本论文的结构如下所示：

第一章，本文首先论述了车联网边缘计算的背景以及意义；然后介绍了 MEC 以及 IoV 的发展近况；接着论述了当下基于边缘计算的车联网系统所面临的挑战；最后介绍了 VEC 系统架构所面临的挑战。

第二章，对 IoV 及相关技术进行了阐述，主要包括 IoV 体系架构、主要的通信技术、IoV 的特点及应用；然后论述了 MEC 技术，主要包括 MEC 技术的系统架构、MEC 的特点、MEC 的主要应用场景、MEC 系统中理论模型分析以及 MEC 关键技术。最后对任务卸载以相关知识进行了阐述，主要包括计算卸载的概念、计算卸载决策、计算卸载流程、计算卸载分类；最后对本章进行了总结。

第三章，对 VEC 的通信模式进行了研究，通过分析移动车辆位置以及速度建立优化问题，然后采用 K-means 算法对本文所提问题进行求解，仿真结果证明，相较于随机选择算法该算法能有效的将车辆用户进行分类，从而最大化 IoV 系统的服务效率。

第四章，针对 MEC 环境下的 IoV 架构设计了一种联合计算卸载和资源分配机制。在该机制中，将原始多目标问题化解为复杂度较低的比例资源分配算法和改进的粒子群算法两个问题对其进行求解，仿真结果表明，该机制所得出的系统效用与其他算法对比始终较高。

第五章，总结与展望。对论文进行总结概括，归纳论文所完成的工作和取得的研究成果，并且提出了进一步的研究方向。如图 1.2 所示为本论文结构安排流程图。

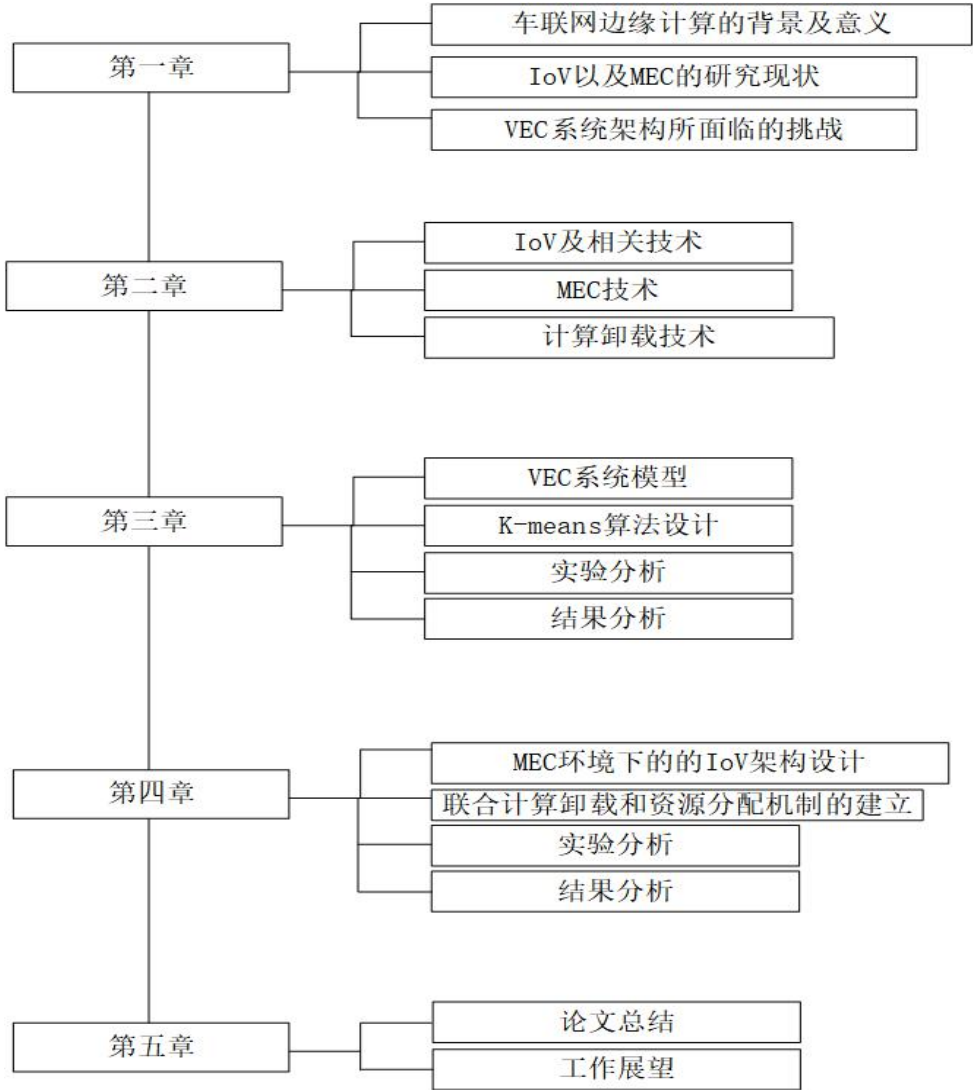


图 1.2 车联网边缘计算系统架构图

2 相关方法和概念概述

2.1 车联网及相关技术介绍

在 2018 年,我国划分了 20MHZ 带宽资源为 V2I 提供支撑。V2I 是车辆与 RSU 信息交互的核心技术,是实现“舒适、安全、高效”的车联网信息交互网络的基础。车联网作为信息化新时代最具代表性的应用场景之一,其将 GPS、微型计算机、射频识别技术(Radio Frequency Identification,RFID)等技术相结合,逐步向“路-边-云”庞大的无线通信系统架构迈进^[24]。

2.1.1 车联网体系架构

IoV 的体系架构主要由感知层、传输层、应用层三部分组成,如图 2.1 所示。通过感知层的智能设备采集外部信息,然后初始数据通过传输层与边缘设备进行交互,最后将处理后的数据回传到车联网中相对应的车辆中。



图 2.1 车联网体系架构

感知层相当于 IoV 的神经末梢,通过车辆终端的智能设备,对道路信息(例如,拥堵情况、交通事故等)和车辆终端的动静态信息(例如,行驶速度、车辆位置等)进行实时采集。

传输层相当于 IoV 的大脑,主要由接入网络(例如,卫星通信、4G 以及 5G 等)和通信网(例如,交通信息专用网络、互联网等)构成,其主要任务是通过通信技术(例如, wifi、zigbee、802.11p 等)和定义专用的协议以及传输感知层收集到的数据^[25]。传输层会综合利用云计算、MEC 等技术,实现 IoV 与其他异构无线网络间的通信,使得 IoV 应用层的各种应用得到低时延、高效的服务。

应用层是其它子系统的接口,可以利用处理、计算后的数据,提供诸如交通安全、娱乐休闲等服务,这些服务不仅要依靠 MEC 等作为技术支撑,而且要适

应 IoV 高动态性的复杂场景^[26]。

2.1.2 车联网关键技术

(1)RFID 技术, RFID 利用射频信号对高速移动的车辆进行识别, 因此, 被广泛应用于 IoV 中, RFID 可提供道路识别、收费站收费等交通服务, 是 IoV 系统的核心技术之一。

(2) 专用短距离通信 (Dedicated Short-Range Communications, DSRC), 在 IEEE802.11p 通信标准下展开, 其时延在 50~100ms 之间, 通信距离在 300-500m 之间, 数据传输速率在 3-27Mbps 之间, 可支持高达 200km/h 的移速, 适用于 V2I 和 V2V 等应用场景^[27,28]。DSRC 技术虽已十分成熟了, 但由于其成本高昂, 容易受环境的影响等, 因此, 无法在车辆中做到全面普及。

(3)控制器局域网络 (Controller Area Network, CAN), 车辆内部通信时就是利用车载设备与智能设备进行信息交互后对车辆进行有效管理与控制。车辆内部通信范围狭小且相对静止, 某些应用对延迟的敏感度高, CAN 技术满足车联网对于稳定性、实时性的需求。

(4)4G, 4G 技术主要应用于 IoV 的传输层, 采用 3GPP 长期演进标准, 4G 有效的弥补了 3G 技术的无线控制器节点多的缺陷^[29]。同时 4G 引入了诸如 OFDMA、MTMO 等先进技术, 与 3G 相比 4G 的频谱效率以及吞吐量大大提高。

(5)激光雷达, 传统的全球导航卫星系统的更新频率、时延、精度等性能已满足不了目前 IoV 的需求, 激光雷达在全球导航卫星系统的基础上加入了惯性测量单位技术和激光束, 激光束向物体发射激光, 然后将发射信号与反射回来的目标回波进行对比, 能够对目标进行高精度的定位; 惯性测量单位技术的更新频率可达到 200Hz, 能够实现车辆速度、终端坐标等信息的实时更新^[30]。因此, 激光雷达被广泛应用于智能地图等服务。

(6)干扰管理, 在车联网环境下通信间的干扰会影响车辆用户的体验, 如果不能解决此问题将影响数据的传输以及降低网络的吞吐量。车联网中通信间干扰包括邻道、同频、互调、多径、以及多普勒效应产生的干扰。邻道干扰是指在传输数据时超过了带宽阈值, 对相邻载波上的信号造成的干扰。同频干扰指在相同通信区域内的车辆通过相同载波频率进行数据传输所产生的信号干扰。多径干扰是指无线信号在在在某些物体上产生折射或者反射等现象使得无线信号通过不同路径到达电子器件所引起的干扰。多普勒效应会出现在车联网场景中使得无线信号

的频率发生改变,收发端无线信号的频率有所不同,因此会产生多普勒效应所引起的干扰。干扰管理的基本方式是通过频域、时间以及空间等方面的功率以及信道资源进行调配,例如,时分多址(Time Division Multiple Access,TDMA)、码分多址(Code Division Multiple Access,CDMA)、正交频分多址(Orthogonal Frequency Division Multiple Access,OFDMA)等通过信号处理技术产生比较高效的分配方案,以此降低车联网用户通信时所产生的干扰问题。

2.1.3 车联网特点及应用

车联网的特点主要有:

(1)车联网的高动态性造成车辆与车辆间的通信链路存活时间相对较短。

(2)数据量大,据统计,智能车辆的传感器每小时会收集到 20GB 左右的数据,其中仅仅一个摄像头每秒就会产生 25-65MB 的数据量^[31]。

(3)车辆终端在道路上是随机分布的,不同的区域和时间点,车辆终端的数目是不同的,但是车辆在道路上行驶有一定的范围并呈现规律性。

车联网的应用可分为安全出行、高效出行以及休闲娱乐,其中安全出行和高效出行是车辆用户对车联网最基本的要求,而休闲娱乐则是车联网不断发展、用户需求不断提升的趋势^[32]。

(1)安全出行,车联网通过车辆终端和 RSU 获取实时的信号灯、路况信息以及车辆的位置、速度信息,提供紧急制动、高速预警、前方事故路况提醒等功能,进而降低在复杂道路、能见度低、暴雪天气、疲劳驾驶等情况下交通事故发生的可能性。

(2)高效出行,如高德地图等,车联网可以对车辆用户的出行线路做出最优规划,避免临时封闭路段或者当前时刻拥堵路段。

(3)休闲娱乐,车联网提供的服务不仅限于安全、高效的出行,车辆用户对车联网提出了更加苛刻的需求,例如 3D 游戏、网页浏览、蓝光电影等功能以此提高车辆用户的出行体验,同时运营商可以通过车联网向车辆终端发送广告信息以此获利。

2.2 MEC 介绍

至 2014 年 MEC 这一概念被提出以来受到大众的广泛关注,MEC 的目的是在接近用户端为运营商和终端用户提供计算资源以及服务载体^[33,34]。MEC 将计算服务下沉到终端用户的网络边缘对数据进行实时处理,大大降低了时延以及长距离传

输产生的费用，因此，MEC 是目前公认的可以辅助 5G 达到性能指标的关键技术之一^[35,36]。在 2020 年，物联网设备将增加 500 亿左右，如果只采用传统云计算的服务，大量终端用户与距离遥远的云中心进行通信会造成带宽资源的浪费、并且产生较高的时延甚至使服务器瘫痪，单单依赖云中心已经无法满足 5G 的性能指标^[37]。

MEC 系统架构阐明了边缘服务器的内部工作模式，并且含有 MEC 所涉及到的业务范围以及 MEC 的硬件部分，将其划分为主体部分，内部核心和 MEC 网络的硬件部分。移动边缘计算主体部分是 MEC 架构的主干，由主机和管理平台构成。移动边缘计算主体部分提供虚拟化基础平台，使应用程序能够正常运行。MEC 网络的硬件部分位于架构的底部，提供各种网络访问连接接口，包括传统的宽带连接、3G、4G 网络以及外部访问。内部核心充当 MEC 系统架构决策者对系统进行管理。MEC 的内部核心对系统进行统一性管理，因此，运营商和智能设备可以随时进行访问。

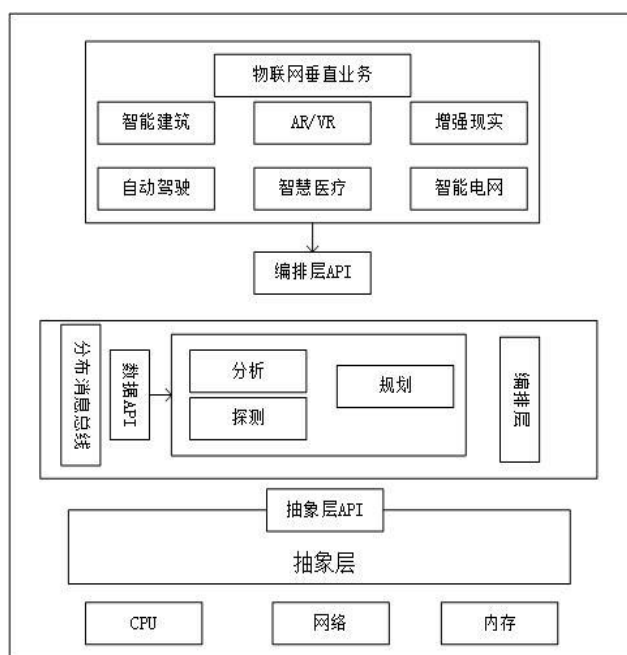


图 2.2 移动边缘计算组件

如图 2.2 所示，抽象层提供通用的应用程序编程接口(API)，通过解耦平台的异构性，实现对内存、CPU、网络等资源的管理，并且提出了一套可编程的、统一的接口，实现了对资源的全程监管。其支持网络功能虚拟化，允许多个操作系统同时在计算机上工作，保证了计算资源的有效利用。编排层为统筹管理边缘服务器提供动态的、基于策略的生命周期。作为底层的边缘基础设施，编排层通过分布式方法来运作。编排层由一个小型的软件代理组成，通过使用分布式存储、

软件代理、可分布式策略引擎和分布消息总线等组件提供边缘化功能，并且用于管理监控边缘节点的当前状态。编排层 API 主要分为四个部分，即探测数据、分析数据、规划和分配资源来管理请求以及执行决策。边缘计算平台下的垂直业务逐渐增多，如智能建筑和自动驾驶最具代表性。

与 MEC 相关的一些特性使它成为云计算的一个重要延伸。下面将介绍关键的一些特性：

互操作性：边缘计算组件需要相互操作，为了保障对数据流等服务的兼容。

传感器网络：在大型监控场景或智能建筑中的分布式系统，这些场景需要分布式计算和监控管理。

低延迟：例如，AR/VR、车辆紧急制动等一些对时延要求高的应用场景。

实时交互：大量 MEC 应用程序(如车辆位置、速度预测)需要实时处理，而不是分批次处理。

动态兼容性：对于某些应用程序来说，动态兼容性是非常重要的，这使得使用诸如 Cisco 的 Locator/ID 分离等协议可以与移动设备直接通信。

无线接入：蜂窝移动网关和无线接入点是经典的光纤陀螺网络节点。

地理分布：为了向移动终端和固定终端提供高效的服务，MEC 进行了大范围的分布式部署。

异构性：MEC 是一个可以由多种设备和系统构成的虚拟平台，具有可兼容性，可以支持在不同协议上工作。其异构性可以作为构建块，它以不同的形式存在，并且可以部署在特殊的场景中。

2.2.1 MEC 特点

MEC 对带宽、时延、可靠性等指标有着近乎苛刻的需求，要真正实现 MEC 在车联网行业的落地，仅仅依赖传统的云计算是无法满足 MEC 的指标要求的。从 2015 年起，3GPP 等研究机构就表明，MEC 是 5G 技术发展道路上必不可少的关键支撑技术之一。表 2.1 对边缘计算和云计算的特点进行比较、并对需求进行了总结，这样有助于理解 MEC 被称为支撑 5G 通信最有力技术的原因。

表 2.1 边缘计算和云计算的比较

性能	边缘计算	云端
上下文感知	强	不支持
安全性	强	弱
时延	低	高
位置分布	边缘侧	云数据中心
部署模式	分布式	集中式
移动性	支持	不支持
计算能力	强	强

(1)上下文感知, MEC 服务器利用接近用户端这一特性可以为用户提供实时信息(如行为、位置、环境等)监测,从而为用户提供精确导航、智能推送等服务。

(2)隐私/安全,云计算的集中式部署使得用户数据信息高度集中,相对于 MEC 分布式部署、小规模的特点,使得 MEC 服务器相较于云数据中心信息少,不容易被攻击。MEC 对用户终端的隐私、安全得到进一步加强、更加稳定可靠。

(3)低延迟,传统意义上的延迟主要由计算时延、通信时延以及传输时延三部分构成。对于传输延迟, MEC 被安装在用户侧,相比于云数据中心,大大降低了与终端用户的距离,因此,有较小的传输延迟;对于通信延迟,传输距离与请求和回传数据的传输和转发的次数成正比, MEC 服务器处理数据无需经过核心网、回程网等,因此,有较小的通信延迟;最后,对于计算延迟来说,虽然云数据中心的计算能力比 MEC 服务器强,但云中心的计算资源要被比 MEC 大好几个数量级的用户共享,而 MEC 只需为附近的终端用户提供服务即可, MEC 服务器不但提高了用户的服务体验更重要的是为云数据中心缓解了一部分压力,因此, MEC 更加符合 5G 通信对延迟的需求。

(4)带宽资源, MEC 作为局部管理者,可以对终端用户所产生的时延敏感型任务进行处理并回传。已被处理过得任务就不需要经过多跳上传给云中心,因此,大大缓解了云中心的压力、避免了带宽资源的浪费。MEC 的引入为 5G 通信实现了带宽方面的性能指标。

2.2.2 MEC 的应用场景

(1)车联网:边缘服务器可以在收费口、交通拥堵路段等区域的 RSU 中进行安装,从而实现本地信息实时共享;利用图像识别等技术对超车、逆行等违章车辆

进行识别，向下对违章车辆进行警告、向上将相关信息发送给数据中心进行资源整合；根据采集到的车流量、车速、堵塞路段等信息，对堵塞路段附近的车辆进行提醒并规划路径。VEC 技术可以实现娱乐、交通信息的实时广播，事故多发路段预警，使得智慧出行、绿色出行成为现实。文献[38]中针对虚拟机移动性和资源管理提出了一个车辆间通信和支持车载应用的边缘服务器体系架构研究。文献[39]将 MEC 服务器安装在基站处，数据通过基站实时共享信息。基站范围内的车辆和路边传感器上感知并收集到本地信息，在 MEC 服务器中对数据信息进行分析计算后，及时将危险信息或其他信息回传给目标车辆，这使得区域内的车辆可以实时收发数据避免事故的发生，MEC 在提供燃料补给和停车信息以及自动驾驶等领域同样发挥着巨大的作用，如图 2.3 所示为基于 MEC 的车联网架构平台。

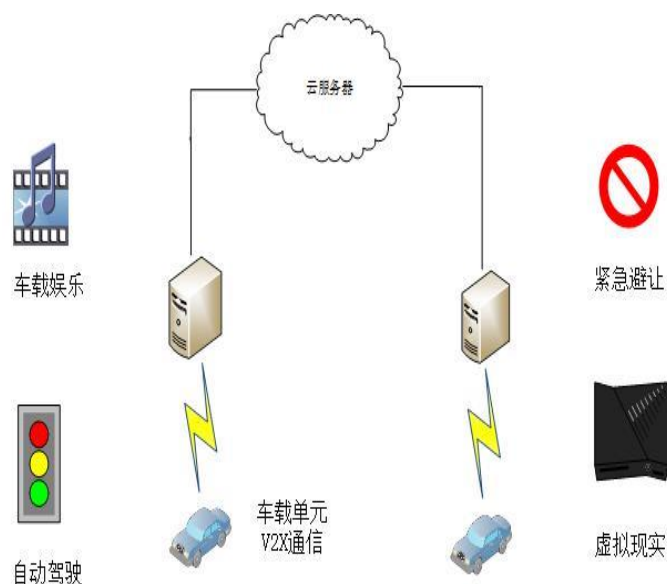


图 2.3 基于 MEC 的车联网架构平台

(2)VR/AR：虚拟和增强现实已经广泛应用到了工业控制以及休闲娱乐等领域。VR 所涉及到的程序对于时延有近乎苛刻的需求，现在大部分的无线通信技术已经无法满足 VR 应用程序的低延迟和高带宽的需求，使用智能设备在刷新率方面达到超高的画质是个艰巨的任务。虽然这些智能设备已经被广泛使用，但由于其能耗较大并且体积较小，对于硬件的设计有一定局限性。使用配备了图形服务器来处理图形，并用基础的方式回传处理后的视频文件是一种很好的解决方法。当一个服务代理商为多个智能设备提供服务时将会产生资源竞争的现象。例如，设计一种服务器关于实时交互式射击类游戏，游戏中的动作与硬件的操作之间的时延时毫秒级的，否则将影响游戏的体验感流畅性。如果使用服务器为此游

戏提供服务将产生超高的时延以及能耗。MEC 服务器能够在接近游戏服务器端来完成云服务器的功能，从而最大限度地降低延迟并防止海量数据流入核心网，因此，MEC 为虚拟增强现实提供了有利的支撑。

(3)视频分析：视频直播、短视频等业务成为当代人娱乐消遣的重要方式，因此，动态自适应流传输标准应运而生，动态自适应流传输使得智能设备兼容多种比特率，因此，在观看视频时变得更加流畅清晰，并且可以根据当前网络状况调节和控制视频设备。当智能设备使用相同的网络进行通信时，动态自适应流传输协议可能会变得不稳定，并且会产生资源竞争的现象。因此，需要完善智能设备的视频传输网络、通信链路的选择以及上下文感知。MEC 在用户侧提供服务将加快接入网的信息传播，并且能够将移动网络代理商集成到视频传递价值链中以此得到更多的价值。用于开启移动边缘计算的核心设备是通信运营商的大型网络服务器，移动边缘计算为许多代理商提供了商业价值，运营商可以灵活的部署应用程序在移动边缘计算服务器中，因此便于处理来自终端用户的数据。

(4)无线传感网络：其用于感知、监视、监测信息，例如大气压、光强度、pm2.5 等，在矿井中，MEC 执行器通过反馈电路自动监测矿井的情况，例如气压传感器会自动监测矿井内的空气压力差，从而推断矿井内的状况防止矿下人员遇到危险。在移动边缘计算的辅助下无线传感网络所产生的能耗以及数据量相较于传统传感器是非常小的。

(5)大数据：大数据作为信息化时代不可或缺的纽带，对于经济的发展具有长远的影响。智能设备中的应用以及传感器所产生的数据，如测速仪、全球导航定位系统、蓝牙等设备，都将产生大量的数据对云服务器是一种巨大的负担。大数据包括由手机、电脑以及智能传感器中应用程序中的数据所产生的异构数据集和，这些数据集由于结构多种多样无法用单个处理器进行解析。大数据是信息化时代的产物，因为它首先是建立在大量数据集的基础上，对有用信息进行提取分析，从而得到有价值的信息这有助于购物推送、个人视频喜好预测等一系列通过分析个人以及社会群体某些特性而做出决策的行为。移动边缘计算可以很好的为大数据服务，首先对底层用户的数据进行收集以及筛选，过滤掉一些无价值的信息，然后上传到云服务器再进行大数据分析，从而大大提高分析的效率以及准确性，此方案同样也适用于多台智能设备的数据。物联网一个非常重要特性是运行不同种类的协议时设备的异构性，数据任务应该由低延迟节点来完成，也就是 MEC 服务器。

(6)医疗保健：将 MEC 技术运用到医学是许多学者的重要课题内容。例如，高血压患者，高血压可能会成为其它疾病的诱因，例如心肌梗塞，脑梗塞等，这些疾病一旦发作将会影响到生命危险。为了监测和预防这类疾病的发生，国内外大量学者投入了研究。例如，通过引入智能交互设备，但仍然无法有效的解决问题。最近，一位学者提出了一种智能医疗设备，该设备基于 MEC 技术通过实时监测血压的情况和人体温度分析设计了一种危险预测算法。该设备借助智能传感器，例如，血压和心率等感应检测设备，通过 MEC 来实时管理监测传感器的运行状态以便及时预测风险的产生。MEC 的三层架构可以有效的协助医护人员，无需在病人身边就能随时得知病人的信息，避免病人因为距离以及就医高峰期所遇到的困难。智能传感器通过 MEC 能够实时的收集患者的日常身体指标，并通过大数据进一步分析、共享以及更新同步信息，有权访问云端的医护人员可以及时作出判断。

2.2.3 MEC 系统中理论模型分析

MEC 的组件包括 MEC 服务器和移动设备。MEC 服务器是由云端和电信代理商部署在接近终端设备的微基站。MEC 服务器一般安装在无线接入集合节点上并通过云端与网关相连。移动设备与 MEC 服务器通过空中接口隔离开，并运用无线通信技术和计算机网络技术建立可靠的通信链路。因此，在将实际问题建模前，需要先确定计算模型、卸载模型、通信模型^[40]。输入输出数据，可容忍时延阈值以及计算能力，这些参数与任务的类型相关，可以通过任务分析器进行估算^[41,42]。输入输出数据的大小将影响计算卸载的效率，任务必须在可容忍时延阈值之前完成，超过该期限任务将处理失败。

2.2.4 MEC 关键技术

(1)软件定义网络技术（Software Defined network, SDN）：SDN 是一种兼容多种网络的，设计灵活且易于维护的网络。SDN 将网络功能转移到了基于软件的网络控制器来推动网络实体向软件化的转变。SDN 无需使用供应商特定的黑盒硬件，从而促进使用商品服务器和交换机专用设备。尽管如此，将网络控制功能转移到基于软件的集中式实体仍需要数据平面设备与 SDN 控制器频繁通信。因此，SDN 控制器的位置需要更靠近数据平面以此减少数据包处理过程中的延迟。MEC 将控制功能定位在更靠近数据平面的设备位置上。此外，MEC 补充了 SDN 从移动宽带网络向电信网络转变的进程，为高效的网络运营和服务交付打下了基础。

因此，SDN 已经在 MEC 的各种应用场所得普及。SDN 的网络控制和转发功能是分开的，并且支持可编程。传统的网络技术将网络控制功能与网络设备结合在一起，SDN 将控制功能迁移到智能设备上，上层服务将网络看做虚拟化实体，网络基础设施看做抽象的。因此，网络相当于应用程序的逻辑交换机。如图 2.4 为 SDN 基本网络结构，通过 SDN 控制器，代理商可以统一的对系统进行控制以及转发，相较于传统分布式系统来说，不需要遍历每个网络设备。SDN 兼容目前大多数设备，且易于更新，避免不兼容等原因造成的巨大损失，并且兼容各种网络协议和标准，对外提供南向和北向接口，可以实现网络通用服务，包括路由选择、多播、访问权限控制、链路资源分配等服务。例如，SDN 可以在无线和有线连接上定义和实施一致的策略。SDN 也可以智能的配置系统来协调整个网络。SDN 的研究已经十分全面，例如，多路径路由和实时流转码等技术已经逐渐成熟。

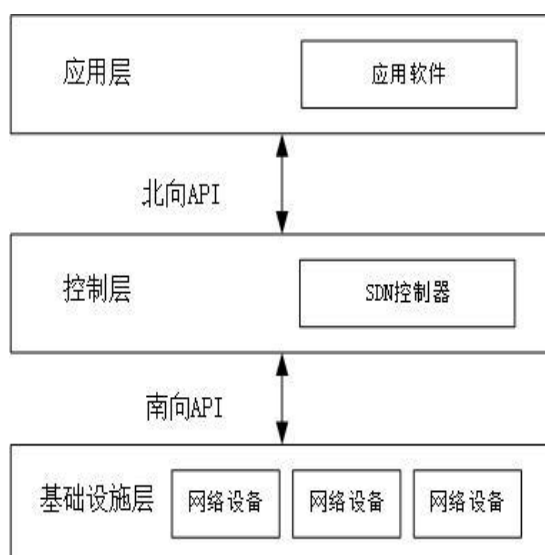


图 2.4 SDN 基本网络结构

(2)网络功能虚拟化：现在的智能设备中安装着许多特有的硬件装置。这些特有的硬件装置性价比较低，而且更新换代较快，这些硬件设备已经无法满足科技飞速发展的今天，通过智能设备中的软件来代替硬件设备的功能成为了一种比较高效的方法。网络功能虚拟化（Network Function Virtualization, NFV）的目的在于通过软件功能的抽象化在硬件上实现软件的功能，这样将提高资源的利用率，并且能够高效的利用某些硬件所特有的功能。图 2.5 为 NFV 的基本架构，NFV 控制和调度系统中网络资源。NFV 管理和协调负责框架中所特有的虚拟化事物。网络功能虚拟化设施含有多种类型的硬件设备以及这些硬件设备功能的软件化。

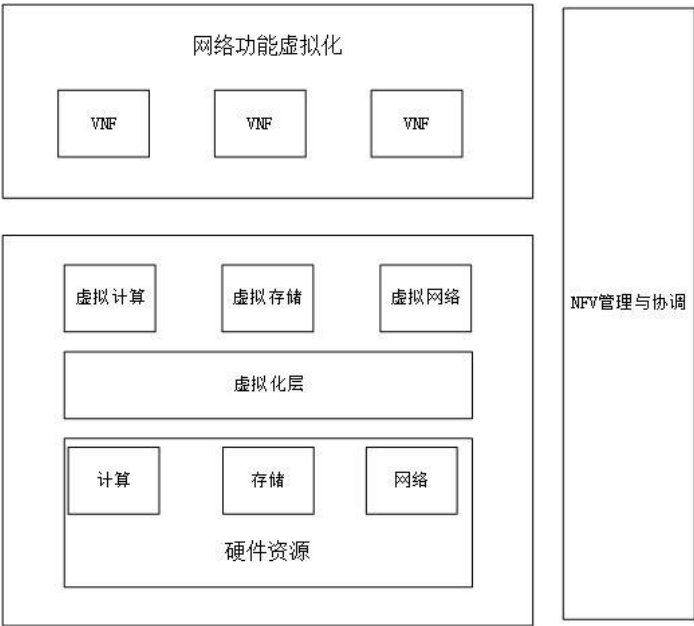
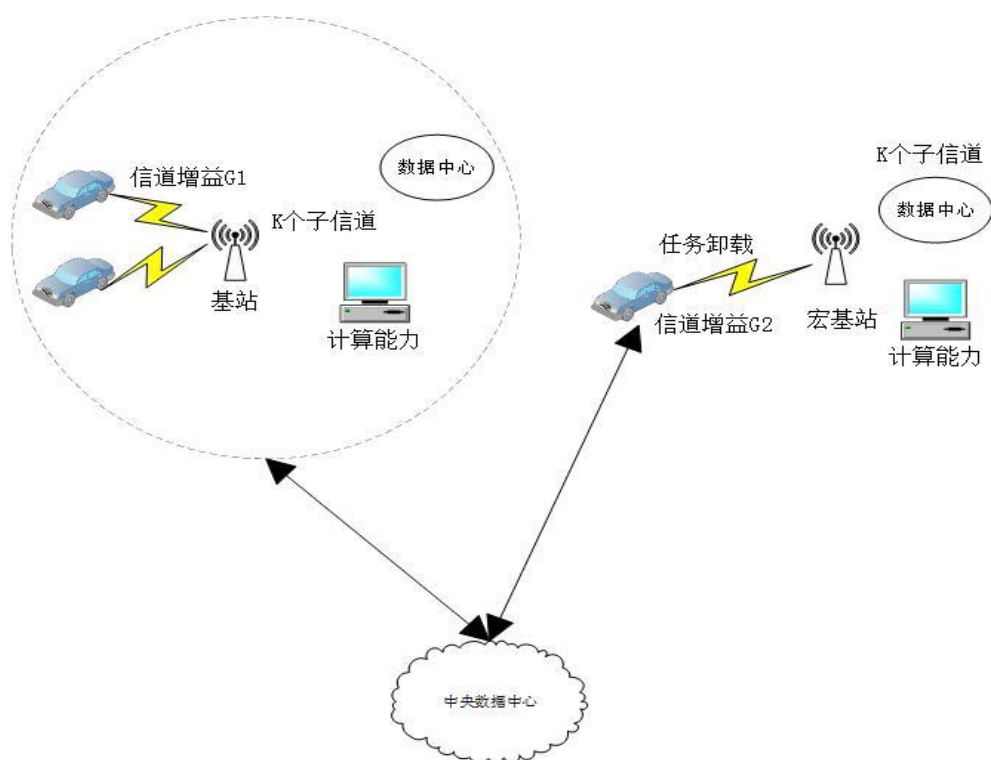
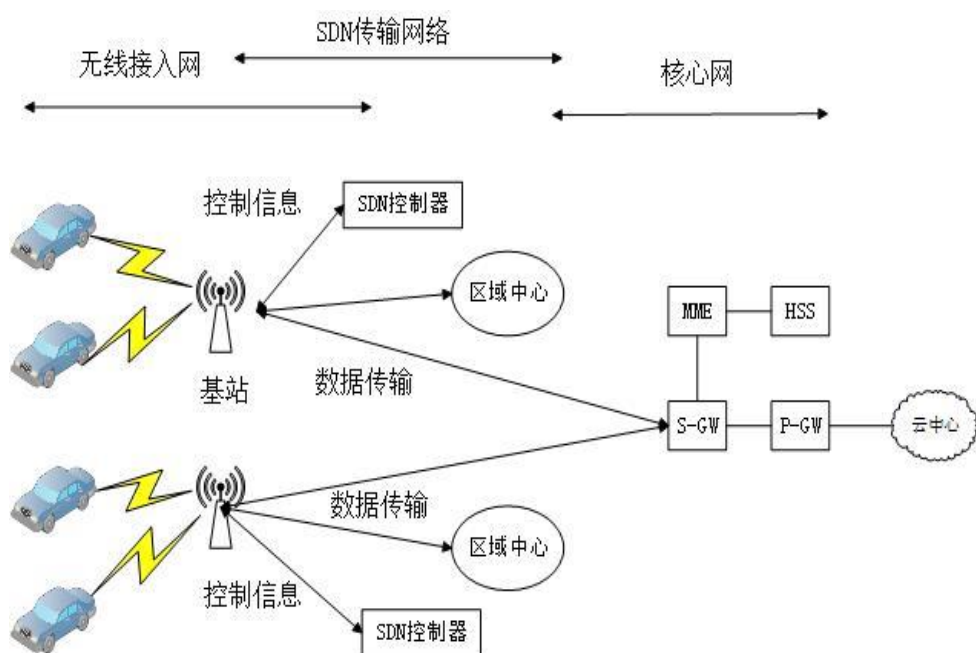


图 2.5 NFV 基本架构

MEC 以虚拟化平台作为载体利用 NFV 和 SDN 的功能。NFV 是从单一的 MEC 服务器中创建多个虚拟化软件，以便管理不同的软件设备，从而达到同时为多台智能化设备提供网络功能的服务。SDN 通过软件化来管理网络功能，提供灵活和动态的计算资源。计算卸载过程只需要在配备边缘服务器的设备上就能执行，因此，如何根据移动终端的任务类型和数据大小来处理任务是需要攻克的问题。软件定义网络技术可以保证边缘服务器中资源的合理利用并有效地进行优化调度。如图 2.6 所示为 MEC 系统架构，SDN 控制器可以统筹管理系统中所有器件。软件定义网络在资源分配、规则生成以及流动性管理等一些系统管理及调度发挥着重要作用，也负责某些智能化处理方式（例如，大数据分析、数据预处理以及机器学习）。如图 2.7 所示为 MEC 资源调度图，软件定义网络可以用来协调云服务器与终端用户之间的资源分配问题，以及计算卸载流程中的卸载决策的制定也需要 SDN 的协调整合。软件定义网络通过网络传输技术传输到 MEC 服务器，MEC 服务器分析整合信息并且将信息回传至云服务器，云中心由很多服务器组成，每台服务器都通过 NFV 技术创建多台虚拟机同时处理多个终端用户的任务请求。



2.3 计算卸载及相关知识介绍

计算卸载是满足应用程序时延要求的一种方案，其中最大容忍时延内完成任务卸载是最基本的约束条件。例如，机器人导航系统需要与物体碰撞之前识别出物体，如果导航系统的处理时延太高，可能需要计算卸载^[43,44]。另一个案例是上

下文感知^[45]，其中收集到多种设备的数据流，例如，GPS、陀螺仪、加速度计、温湿度传感器等需要一起计算处理，为了得到有关设备上下文的实时信息，可以将终端设备所产生的任务卸载到边缘设备以此得到较高的计算能力。能耗一直以来是移动设备关注的问题，移动用户现在不仅仅局限在语音通话，大多数人想要获得高清直播、VR 游戏等高级服务，智能终端极大地缩短了电池的寿命。虽然电池续航技术在逐渐进步，但却无法满足现在用户的需求，计算卸载将任务的高能耗部分卸载到 MEC 服务器是延长电池寿命的解决方案之一^[46]。

2.3.1 计算卸载的概念

计算卸载是将终端设备产生的计算任务迁移到微服务器的方法。虽然终端设备受限于计算资源、散热以及电池寿命，但是通过将终端设备中高耗能的计算任务迁移到 MEC 服务器可以有效地解决诸多问题。计算卸载主要的问题包括是否执行卸载，执行全部卸载还是部分卸载，以及卸载计算的方式和内容。卸载决策取决于计算任务的类型，其可以通过三个标准进行划分分别是：计算任务是否含有不可卸载的部分；是否能够估计某些连续执行计算任务的数据量；每个子任务间的递进顺序以及关系。调试机、系统集成器和控制器是计算卸载的主要组成部分。调试机用来控制卸载的内容；系统集成器用来监测传输的数据大小、可用带宽等一些影响迁移效率的参数；控制器用来决策卸载策略。目前大多数研究的目的是优化时延以及用电量在终端设备进行任务卸载时^[47,48]，大多数计算任务可以接受最大容忍时延内的延迟影响，或者有些研究旨在找到时延与能耗之间的最佳权衡点，例如加权系数法、帕累托法。由于在云端执行任务的过程中会存在会话状态产生的延迟，MEC 处理任务时的能耗低，主要因为不存在会话状态产生的延迟，并且有较低的传输时延。因此，用户能够能耗更低的服务^[49]。

2.3.2 计算卸载决策

计算卸载是提高计算效率，降低能耗的有效解决方案。卸载决策优化主要关注两方面内容，即时延（计算时延、传输时延、等待时延）以及能耗。应用程序的响应时间是衡量服务质量的重要指标之一。卸载决策包括单任务决策以及多任务组合决策。单任务决策是基于单个任务的简单应用场景，确定是否执行计算卸载。多任务组合决策会产生多个具有关联性的计算任务的复杂应用场景，在该场景中任务请求需要通过多次计算卸载才能完成交付，且这些任务之间存在关联性，即某项任务的执行必须依赖其他任务的计算结果。多任务组合决策的目标是在计算时延和通信时延之间取得权衡，类似于车间调度问题需要遍历每一个可能

发生的结果，此过程类似于图像分割。

当某个任务在执行过程需要服务器切换时，最应该保证服务的连续性。从图 2.8 可以看出，车辆终端的位置改变后，如果将任务迁移到另一个 MEC 服务器，则需要进行虚拟机迁移或者使用容器技术。决定虚拟机是否迁移取决于服务的要求、类型以及数据大小，例如，具有严格时延要求的制动装置或者是 AR/VR 服务的延迟敏感型任务总是被迁移到最佳的 MEC 服务器。迁移过程需要考虑是否完全迁移或者部分迁移，同时要考虑虚拟机迁移成本，包括由于计算节点之间的切换而导致的虚拟机迁移和回传资源所消耗的能量。文献[50]提出以一种利润最大化迁移策略，通过选择最佳的 MEC 服务器从而优化成本和增益之间的权衡。文献[51]提出一种虚拟机动态规划策略来为用户找到最佳的通信路径的预测算法。

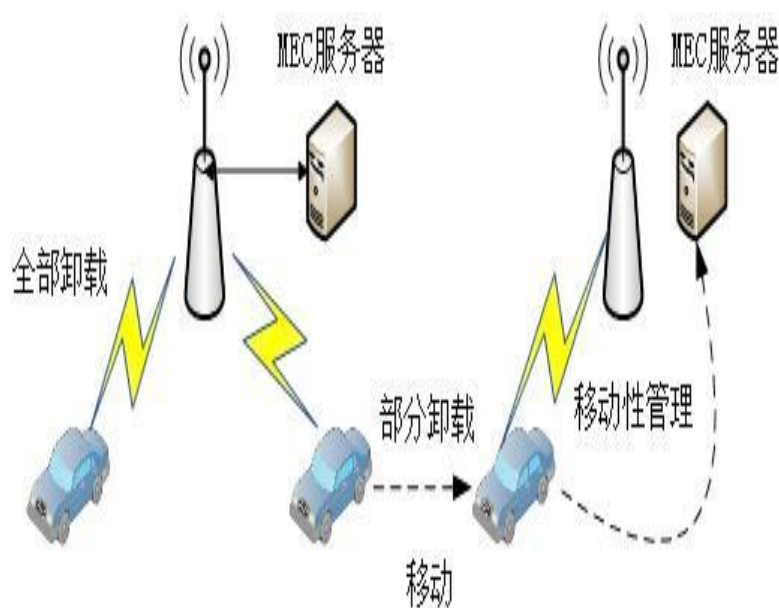


图 2.8 全部卸载与部分卸载示意图

2.3.3 计算卸载流程

如图 2.9 所示，计算卸载由任务环境感知、任务分割、迁移决策、任务上传、卸载计算、任务回传六部分组成：

(1) 任务环境感知：此过程为任务卸载提供相关参数，也是计算卸载流程的第一步，这里的任务环境对应于 MEC 系统中的边缘节点。当终端用户发送卸载任务请求时，MEC 服务器进行卸载环境感知，并监测 MEC 服务器的资源状况。同样，远程云数据中心的高性能服务器也可以为终端用户提供服务。

(2) 任务分割：任务分割属于计算卸载的初始阶段，显而易见，任务分割就是利用任务划分策略将一个完整的任务分割为具有互相依赖性的子任务集合，任务分割将任务分成两类，可迁移任务与不可迁移任务，其中不可迁移任务只能在本地进行计算；可迁移任务表示可以更进一步的细粒度划分，运用并行方式进行处理，为 MEC 服务器执行阶段做好预备工作。值得注意的是，计算卸载任务中大部分是可迁移任务。

(3) 迁移决策：迁移决策是计算卸载过程中的关键步骤^[52,53]，首先进行任务抽象化处理使得建模过程简单化，然后创建目标优化函数，并且根据所建模型设计算法找到最优的卸载策略。迁移决策过程中需要注意的是哪些任务需要在本地处理，哪些任务需要迁移到 MEC 服务器进行处理以及卸载到哪一个 MEC 服务器，而做出迁移决策的目的是降低能量消耗和时延等因素。若卸载到 MEC 服务器执行，需要对信道传输功率、计算和存储资源进行合理化分配。

(4) 任务上传：任务上传是将任务通过蜂窝网络或者 wifi 上传至 MEC 服务器。

(5) 卸载计算：任务迁移到服务器后会对接收到的任务进行确认，然后根据事先设计好的卸载策略为任务动态的分配资源并执行任务。

(6) 任务回传：卸载流程的最后一步是任务回传，MEC 服务器将计算结果回传给终端设备，任务回传完成后表明任务被成功的计算卸载。

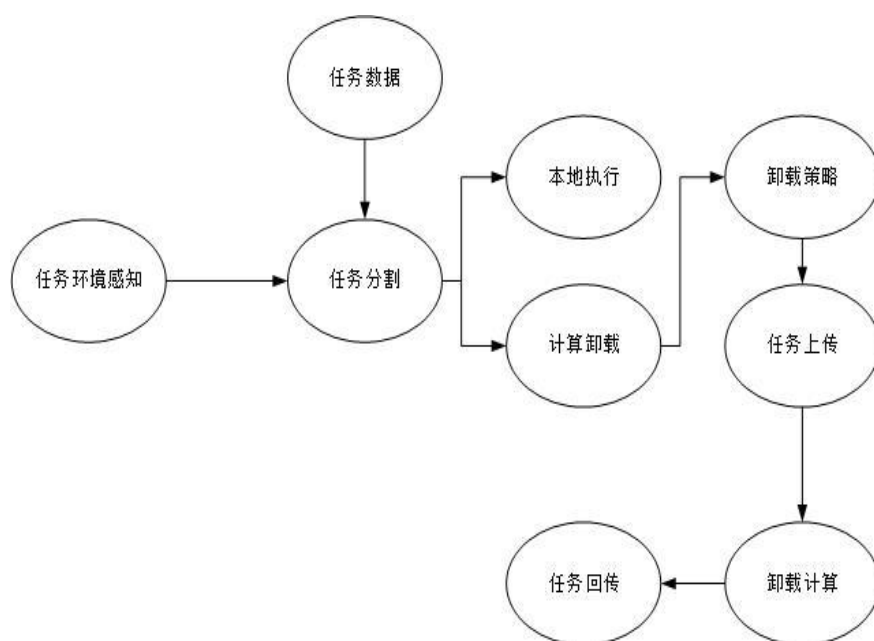


图 2.9 MEC 卸载流程

2.3.4 计算卸载分类

计算卸载缓解了数据传输过程中带来的网络负载，如图 2.10 所示为计算卸载模型。

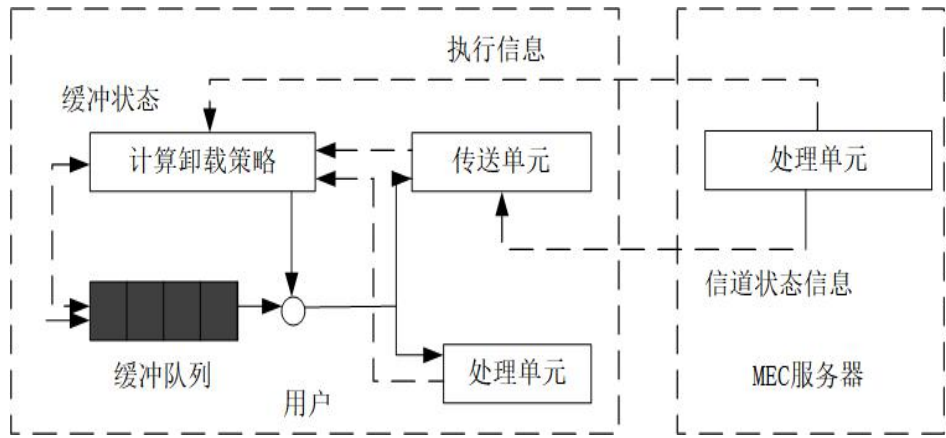


图 2.10 计算卸载模型

根据不同的划分标准，可以将计算卸载分为粗粒度与细粒度计算卸载以及全部与部分计算卸载，下面将一一阐述：

（1）粗粒度卸载与细粒度卸载：粗粒度卸载是将全部或部分的应用程序封装在虚拟机实体中，不加以细粒化地卸载到云中心执行，这种卸载方式不需要修改代码和提前标注可以减轻程序员的负担，但是对于那些频繁交互的应用程序中，粗粒度卸载优势就没有那么明显。细粒度卸载则是将应用任务进行细粒度分割或划分成块，细粒度卸载必须提前标注和修改代码，提前划分好程序，避免了额外的数据传输。此外，细粒度卸载还可进一步划分为静态策略和动态策略。静态策略是为了预先设置哪些任务需要上传到服务器处理速度较快，但是由于执行任务时未对环境进行准确判断导致静态策略的效率较低；动态策略主要通过实时感知移动设备终端运行环境、云中心状态以及实时网络状态，动态的调整任务是否需要卸载，这样可以充分利用服务器上的计算资源，但是动态卸载策略对设备要求较高，需要获取精确网络实时运行状态。

（2）全部卸载与部分卸载：全部卸载是将终端设备上所有的计算任务通过传输链路迁移到移动边缘计算服务器上进行计算，全部卸载在卸载前不仅需要考虑计算任务的卸载执行问题，同时要考虑处理任务所产生的时延和能耗。如果卸载计算所产生的系统效用小于本地执行时系统效用，则说明任务卸载产生的效率更低。而部分卸载则更具灵活性，通过将计算密集型任务划分为多个子任务的集合，

子任务一部分在本地执行，另一部分则在 MEC 服务器上进行处理，许多学者致力于研究本地执行与 MEC 服务器执行任务分配比例间的权衡问题。部分卸载的优势在于多个子任务并行执行，效率较高且具有灵活性，但是增加了问题的复杂性，因此，需要进一步研究。

2.4 本章小结

本章简述了车联网的体系架构以及车联网中涉及到的关键技术及其应用；接着对 MEC 技术进行了详细介绍，简述了 MEC 的特点、关键技术以及在车联网中的应用；最后对计算卸载技术进行了相关说明，简述了计算卸载流程以及分类。

3 基于 K-means 聚类的计算卸载算法设计

3.1 引言

近几年,国内外学者对解决车辆与 RSU 的信息孤岛化问题进行了深入研究。VEC 系统架构主要采用两种通信方式进行通信。V2I 通信模式主要是与 RSU 进行通信从而达到资源共享的目的。V2V 通信模式主要是通过 GPS 和传感器等设备发送车辆速度以及位置信息给周围的车辆,当发生紧急事件时可以及时的告知对方的车辆以便做出快速的应答。相关机构专门为车联网环境下的无线接入规范定制了 IEEE 802.11p 标准。随着车联网通信网络的大范围普及,其在车载娱乐,交通安全监管以及事故预测警告等方面起着重要作用。在车联网架构中的 MEC 服务器的资源分配是分布式计算领域的重要问题之一。在每个 MEC 服务器中,处理器周期、内存带宽、通信宽带和计算存储被多个应用程序和用户所共享,由于具有不同资源块之间的分配组合特性,多个资源的优化分配是一个需要攻克的难题^[54-56]。考虑到 V2I 通信链路与 V2V 通信链路的差异性,本章将以时延为评价指标,通过聚类算法对车辆进行聚类,从而为车辆选择最佳的卸载策略。本章构建了 MEC 的车联网服务模型,并采用正交频分多址技术来实现通信过程。该模型中任务车辆服从泊松分布。本章将任务车辆分为两种类型,在交通拥堵地带和停车场中的车辆对卸载效率有更多的需求。而在道路畅通的情况下,任务车辆默认匀速行驶,因此对时延有了更加苛刻的需求。此外当任务车辆离 RSU 太远时,不得不考虑选择 V2V 通信模式。当任务车辆在任务计算完成之前驶离 RSU 覆盖范围 MEC 要考虑计算迁移的问题,由于计算迁移代价高昂并且会降低车联网的服务质量,因此尽量避免这种情况的发生。VEC 系统架构中车辆的计算能力和 MEC 服务器功能的差异性以及某些应用程序对时延苛刻的需求,使得问题进一步复杂化,从运营商的角度来看,车联网环境的异构性给提供实时、高质量的服务带来了巨大的挑战^[57]。

V2V 模式是以车与车之间提供智能化服务,在本章系统中选择最优的车辆作为服务终端为其它车辆提供服务。因此,通过聚类来选择最优服务终端成为车联网的热点研究问题^[58],聚类可以保持 VEC 系统架构中通信车辆之间的协调性。本章中 MEC 服务器和任务车辆由簇头和簇成员组成;簇成员选择簇头来管理并协调簇成员之间的关系。因此,K-means 聚类可以为车联网提高通信能力和服务稳定性。由于车联网的异构拓扑结构使得车辆用户频繁地驶入和驶离服务范围。

频繁的聚类重组可能会导致通信系统不稳定从而产生额外开销。此外，在车联网中车辆之间的相关干扰会影响通信质量。因此，聚类思想的引入可以很大程度上解决一部分问题。

3.2 计算卸载算法设计

3.2.1 计算卸载模型

如图 3.1 所示为 VEC 系统架构，每辆车都被视为集合元素作为实验对象。该架构中的每个元素都能通过智能传感器获取当前的地理位置以及共享周围智能终端的数据。

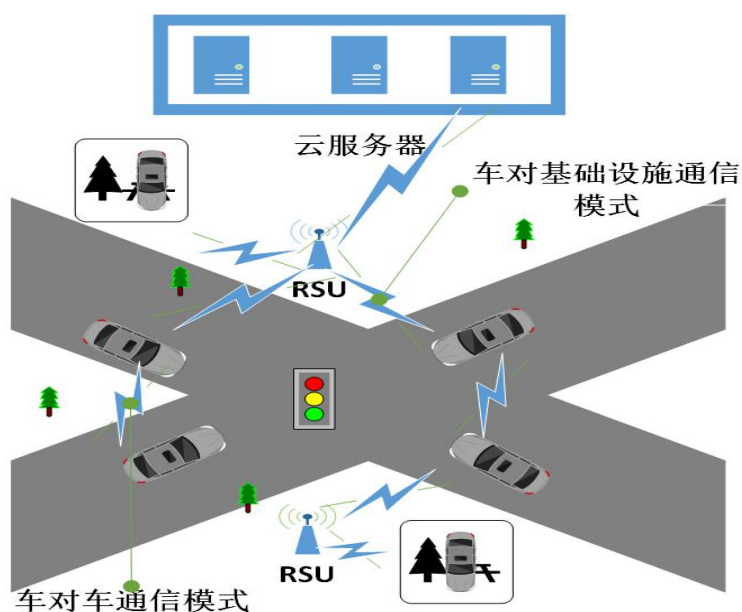


图 3.1 VEC 系统架构

该模型是一个多边缘服务器的模型，RSU 随机分布在道路两侧，并且 RSU 都配备有边缘服务器。本文将任务车辆集合用 $V=\{v_1, v_2, \dots, v_N\}$ 来表示， $v_i (i \in [1, N])$ ，每辆车都配备有车载单元、GPS、传感器等设备。本章将服务节点统一用 $G=\{g_1, g_2, \dots, g_M\}$ 来表示。每个 RSU 的无线信道和计算资源只作用该覆盖范围内的车辆。实验仿真区域，由 x-y 平面定义，x 和 y 的定义范围如下所示：

$$S = \{(x, y) | 0 \leq x \leq L, 0 \leq y \leq W\} \quad (3-1)$$

在设备区域内第 i 个任务车辆的初始位置表示为：

$$vp_i^0 = (vp_{x_i}^0, vp_{y_i}^0) \quad (3-2)$$

第 j 个服务节点的位置表示为：

$$gp_j = (gpx_j, gpy_j) \quad (3-3)$$

第 j 个服务节点的候选任务车辆集合表示为:

$$gv_j = \{v_i | dis(v_i, g_j) \leq R, 1 \leq i \leq N, 1 \leq j \leq M\} \quad (3-4)$$

其中, 任务车辆到服务节点的最短距离表示为:

$$dis(v_i, g_j) = \sqrt{(vpx_i - gpx_j)^2 + (vpy_i - gpy_j)^2} \quad (3-5)$$

(1) 平均完成时间

本章根据计算卸载的平均完成时间来衡量服务质量, 任务车辆在服务节点的完成时间包括传输时延、处理时延以及等待时间 t_i^{que} 。

传输时延为:

$$t_i^{trans} = \frac{o_i}{\gamma_{i,j}} \quad (3-6)$$

计算时延为:

$$t_i^{cal} = \frac{d_i}{c_j} \quad (3-7)$$

其中, o_i 是输入的数据量; $\gamma_{i,j}$ 是服务节点 j 分配给任务车辆 v_i 的传输速率; d_i 是 CPU 所需的周期数。

任务车辆 v_i 的总时延为:

$$t_i = t_i^{trans} + t_i^{cal} + t_i^{que} \quad (3-8)$$

模型中所有车辆在服务节点的平均完成时间称为系统的平均完成时间, 即:

$$\bar{t} = \frac{1}{N} \sum_{i=1}^N t_i \quad (3-9)$$

(2) 通信模型与负载均衡模型

本文计算任务需要通过无线信道传输到服务节点, 因此, 任务车辆的上行传输速率可以定义为:

$$\gamma = \frac{B}{S} \log_2 \left(1 + \frac{p_i \eta^2}{\sigma^2} \right) \quad (3-10)$$

其中, B 为系统带宽, p_i 为传输功率, σ^2 为白色高斯噪声, S 为子载波的数量, η^2 为任务车辆到 MEC 服务器的信道增益。

为了提高计算卸载的效率, 本章引入了负载均衡模型。建立模型需要求得系统的平均资源利用率。第 j 个服务节点的资源利用率 $ru_j(\Lambda)$ 如下所示。

$$ru_j(\Lambda) = \begin{cases} \frac{1}{S} \sum_{s=1}^S \sum_{i=1}^N i_{i,s}, I_j = 1 \\ 0, I_j = 0 \end{cases} \quad (3-11)$$

其中, $I_j=1$ 表示第 j 个服务节点的资源被占用, 否则 $I_j=0$ 。 $i_{i,s} \in \{0,1\}$ 表示子载波分配器, 当值为 1 时表示子载波 s 分配给车辆 v_i , 本章用 $\Lambda = (a_1, a_2, \dots, a_N)$ 来表示卸载决策集合。

占用服务节点的数量可以用式(3-12)来表示:

$$Ra(\Lambda) = \sum_{j=1}^G I_j(\Lambda) \quad (3-12)$$

平均资源利用率用式(3-13)来表示:

$$Aru(\Lambda) = \frac{\sum_{j=1}^G ru_j(\Lambda)}{Ra(\Lambda)} \quad (3-13)$$

负载均衡率用式(3-14)来表示:

$$lb_j(\Lambda) = \begin{cases} |ru_j(\Lambda) - Aru(\Lambda)|, I_j(\Lambda) = 1 \\ 0, \text{否则} \end{cases} \quad (3-14)$$

因此, 得出了平均负载均衡率:

$$Alb(\Lambda) = \frac{\sum_{j=1}^G lb_j(\Lambda)}{Ra(\Lambda)} \quad (3-15)$$

负载均衡率表示被占用服务节点的资源利用率围绕着平均资源利用率的波动情况, $L(\Lambda)$ 越小, 说明卸载策略的负载越均衡。因此, 总的负载均衡率 $L(\Lambda)$ 如式(3-16)所示:

$$L(\Lambda) = \frac{Alb(\Lambda)}{Aru(\Lambda)} \quad (3-16)$$

每个任务车辆的任务需求不一样, 因此服务节点能够接收任务的数量是不同的, 因此, 有必要平衡每个服务节点所容纳任务的数量, 本章用阈值 τ 来判断服务节点的最大容量。单个服务节点可以服务的数据量如式(3-17)所示:

$$\sum_{v_i \in g v_j} o_i \leq \tau \quad (3-17)$$

(3) 时间约束

以 RSU 为圆心, 能够卸载到某个 RSU 的条件应该满足 $-R \leq v p_i \leq R$, 其中 R 为 RSU 的覆盖范围。任务车辆 v_i 的速度用 v_i 来表示。因此, 任务车辆 v_i 在 t 时刻

的位置如下所示：

$$vp_i = \mathbf{v}_i t + vp_i^0 \quad (3-18)$$

其中， \mathbf{v}_i 的符号代表方向。因此，根据式(3-19)推导出 v_i 离开次 RSU 的时间，表示为：

$$t_i^{out} = (vp_i - vp_i^0) / \mathbf{v}_i \quad (3-19)$$

任务车辆访问服务节点的网络时延与边缘服务器的部署位置密切相关。为了保证在规定时间内完成任务卸载，整个卸载过程的时间不能超过任务车辆的最大容忍时延，即：

$$t_i^{trans} + t_i^{cal} + t_i^{que} \leq t_i^{out}, \forall i \in N \quad (3-20)$$

(4) 目标函数

本章以最小化平均完成时间为目标，以负载均衡、最大可容忍时延、平均完成时间为约束条件，建立优化模型如下所示：

$$\begin{aligned} & \min t \\ & s.t. \\ & C1: \sum_{v_i \in gv_j} o_i \leq \tau \\ & C2: t_i^{trans} + t_i^{cal} + t_i^{que} \leq t_i^{out}, \forall i \in N \\ & C3: t = \frac{1}{N} \sum_{i=1}^N t_i \end{aligned} \quad (3-21)$$

3.2.2 K-means 聚类算法

K-means 聚类可以应用到各个研究领域中，例如，神经网络、自然语言处理等，如图 3.2 所示为 K-means 聚类算法过程，将样本聚类成 3 个类别。本文根据任务车辆的位置进行聚类，最终选择最优服务节点对任务进行处理。本章将任务车辆重新分配到多个簇中，最小化簇内平方和，公式如下：

$$\sum_{j=1}^k \sum_{x \in B_i} (x - b_i)^2 \quad (3-22)$$

其中， k 是簇的数量， x 是服务节点的位置， B_i 是分配给第 i 个服务节点的任务车辆集合。在簇更新完成后每个车辆集和将清零。随机选择 k 个服务节点作为簇的中心。根据选取的服务节点作为最初的聚类中心，一直迭代直至完成聚类，具体算法如下所示。

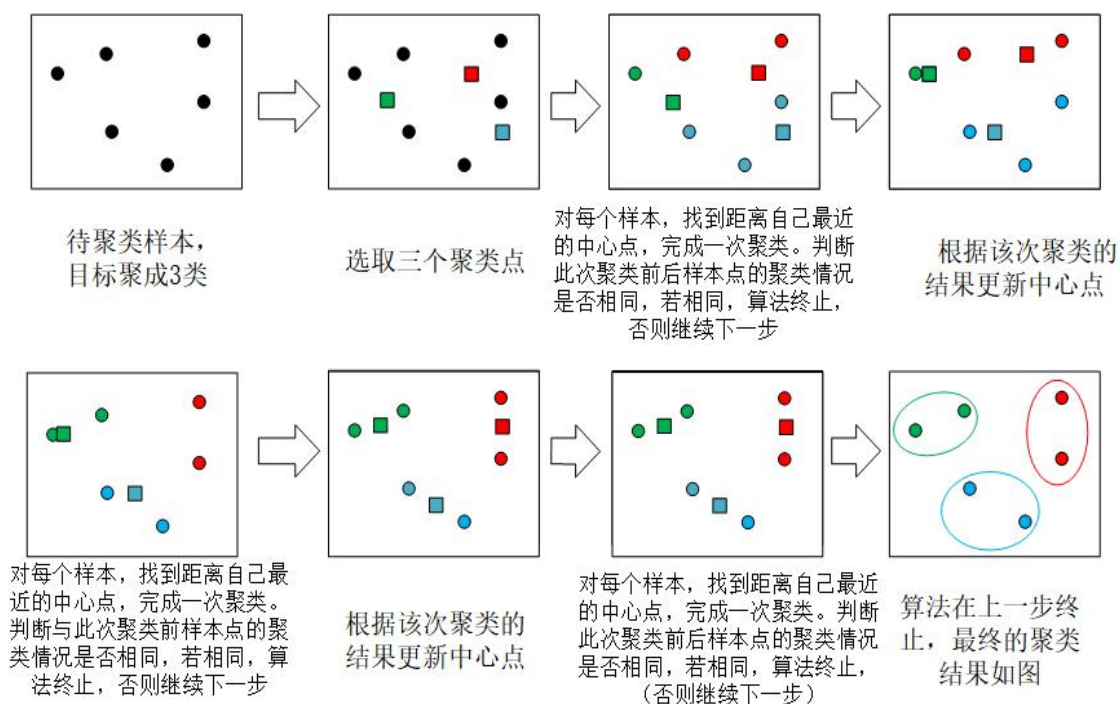


图 3.2 K-means 聚类算法过程

第一步将每个车辆分配给一个服务节点 B_i ，如果 b_i 是最接近 x 的簇心，那么 x 就会被分配给 B_i ：

$$B_i = \{x: \|x - b_i\|^2 \leq \|x - b_j\|^2, 1 \leq i, j \leq k\} \quad (3-23)$$

其中， b_i 是 B_i 中心， $(x - b_i)^2$ 的值是任务车辆与 B_i 的中心的平方。利用欧式距离更新任务车辆的中心位置，并计算每一对车辆与节点中心间的距离。

第二步是更新每个任务车辆集合的中心。

$$b_i = \frac{1}{|B_i|} \sum_{x \in B_i} x \quad (3-24)$$

其中， $|B_i|$ 是 B_i 中任务车辆的个数。在簇心更新完成后每个集和将被清零，然后该算法按照式(3-23)和(3-24)进行下一次迭代。具体的步骤如下所示：

算法一.K-means 聚类算法

输入： 初始车辆集合 $V = \{v_1, v_2, \dots, v_N\}$ 。

输出： 聚类集合 $\{B_1, B_2, \dots, B_k\}$ 。

- 1 while $i \leq k$ do
- 2 while $i \leq N$ do
- 3 计算任务车辆集合 U 中车辆之间的差值，找到距离最近的两个车辆，形成集合 B_i ，从任务车辆集合中删除 B_i ；。
- 4 计算 U 中每辆车与 b_i 的距离，找到 U 中与 b_i 位置最近的车辆，将其从 U 中移除，并

存储到 B_i 中。

5 重新计算聚类群 $\{B_1, B_2, \dots, B_k\}$ 中任务车辆位置的中心位置 $\{b_1, b_2, \dots, b_k\}$ ；。

6 步骤7：计算最终的每个中心位置与每个服务节点之间的距离，依次选择每个中心位置距离最近的服务节点；

7 end while

在迭代过程中不断查找未分配的任务车辆以及有空闲计算资源的服务节点。为了降低算法复杂度，该算法优先从满负荷的服务节点上选择任务车辆，并将所选任务车辆分配到更接近满负荷的服务节点上。

为了防止超出单个服务节点的容量阈值，当服务节点接受的任务车容量阈值超过 τ 时，服务节点满负荷工作时，此时，用 $Cap(j)$ 表示在服务节点处的候选任务车辆集合的容量，那么：

$$Cap(j) \leq \sum_{v_i \in g v_j} o_i \quad (3-25)$$

该算法重新查找未分配的任务车辆以及未超载的服务节点。为了减少运行时间，该算法优先考虑从超载的服务节点上选择任务车辆，并将所选择的任务车辆分配到更接近超载服务节点上。基于服务节点的存储容量和任务终端的存储需求，可以确定服务节点的最大任务车辆存储需求，并且允许多个任务车辆在同一个服务节点上卸载任务。

3.3 实验分析

本章以 Matlab 为实验平台对所提出的算法进行仿真，对随机卸载算法（Random Offloading Algorithm, ROA）、基于 K-means 聚类卸载算法（K-means Clustering Offloading Algorithm, KCOA）在 VEC 系统中的平均完成时间进行比较，三组仿真实验具体为：

(1)任务车辆数量一定，随着 MEC 服务器数量的增加，对比分析两种算法的系统的平均完成时间。

(2)MEC 服务器的数量一定，随着任务车辆数量的增加，对比分析两种算法的系统的平均完成时间。

(3)任务车辆和 MEC 服务器的数量一定，随着网络最大容忍时延的改变，分析对比两种算法的系统的平均完成时间。

3.3.1 仿真环境建立

本文使用正方形作为实验部分的设备区域，并且该区域的 x 轴和 y 轴的范围设置在 $[0, 200]$ m，并且保证所有的任务车辆都在 MEC 服务器的覆盖范围内，任务车辆与 MEC 服务器之间采用对跳最短路由的方式建立连接。

3.3.2 仿真参数设置

本次仿真实验涉及到两种算法：KCOA、ROA。通过对比实验中两种算法的结果，找到 VEC 系统架构下的最优卸载策略。当 MEC 服务器数量数值较低时，单个 MEC 服务器的处理负荷增大，计算卸载时的等待时间也会相应上升，这样就会使网络延迟严重增加以及 MEC 服务器可能会出现超荷的情况不能达到车载应用的需求。当 MEC 服务器数量数值较高会产生资源浪费的现象，运营商也会收取费用增大了成本，但是网络延迟以及等待时间相应的减少。综上所述，MEC 服务器的位置和数量对于任务车辆是非常重要的。本章将迭代次数设置为一百次，最后的计算结果进行求和取均值，以此增加实验结果的可靠性。表 3.1 是实验过程所涉及到的参数。

表 3.1 仿真参数

参数	描述	取值
o_i	任务计算大小/MB	400~500
d_i	任务所需 CPU 周期数/Megacycle	1200~2000
v_i	任务车辆的数量（个）	20,30,40,50,60,70
c_i	车辆计算能力/GHZ	1~2
c_j	边缘节点计算能力/GHZ	5~6
g_j	MEC 服务器的数量（个）	20
R	RSU 的覆盖半径（m）	150
τ	MEC 服务器存储容量阈值（MB）	2000

3.3.3 对比分析

本节针对两种算法的不同性能指标进行分析归纳，实验结果分析的性能参数包括系统的平均完成时间、任务车辆可容忍的时延阈值。

(1) 系统的平均完成时间

计算卸载的总时延为处理时延、传输时延以及等待时间相加。系统的平均完

成时间为单个计算任务的时延的累加求和然后除以车辆数。通过系统的平均完成时间的大小可以判断出基于 K-means 聚类卸载算法的优势。

1) MEC 服务器的数量对平均完成时间影响

任务车辆的数量为 20 个，MEC 服务器的数量增加直到趋于平稳或者到达了预设的迭代次数。图 3.3 表明，随着 MEC 服务器数量的增加，单个服务节点的计算负担将有所缓解，此时，计算能力逐渐增强，当曲线趋于平稳是无需考虑资源竞争的现象，系统的平均完成时间逐渐趋于平稳。其中，在 MEC 服务器的数量相同的情况下，基于 KCOA 的时延始终较低，ROA 相较于 KCOA 的性能较差；在 MEC 服务器的数量为 4 到 14 时，KCOA 一直比 ROA 表现的更优；当 MEC 服务器的数量超过 14 时，KCOA 和 ROA 的系统平均完成时间的差距变得不太明显。因为 ROA 没有考虑任务车辆的位置和数量以及任何约束条件，车辆在选择目标 MEC 服务器时总是会优先选择与其距离最近的 MEC 服务器，因此，导致有的 MEC 服务器出现超载和有的边缘服务器闲置的状态，这样会增加任务的排队时间导致系统的平均完成时间最大。KCOA 根据对任务车辆集合的聚类结果，将任务车辆更均匀的分配给模型中的服务器上。因此，任务的平均排队时间也会相应的减少。

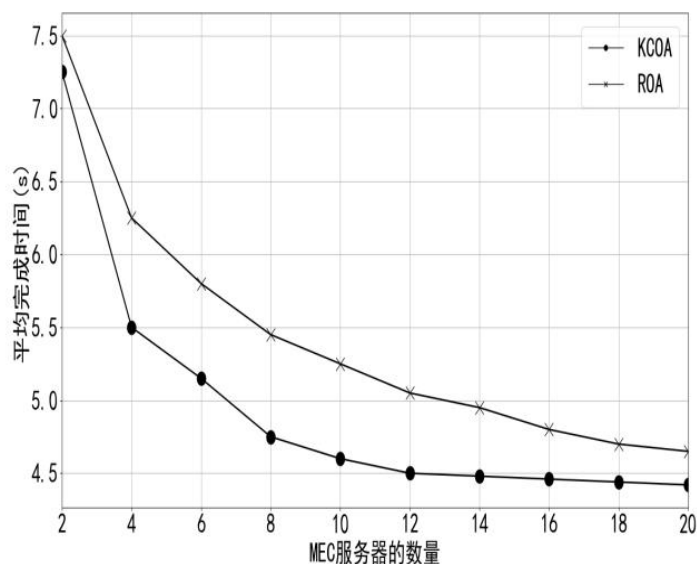


图 3.3 MEC 服务器数量和平均完成时间关系图

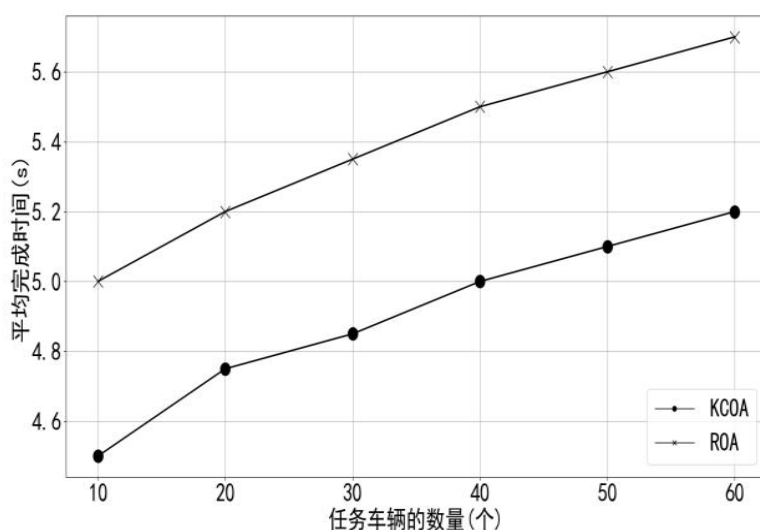


图 3.4 任务车辆数量和平均完成时间关系图

2)任务车辆的数量对平均完成时间的影响

MEC 服务器的数量为两种算法趋于稳定时的 MEC 服务器的个数，任务车辆数量被设置从 10 到 60 间隔 10 递增。图 3.4 表明，在任务车辆数量的增加的同时，MEC 服务器上的工作负载也会增加，两种算法的平均完成时间均有所增加，KCOA 的性能表现最佳。因为该算法在满足任务车辆资源需求和服务质量的要求上给予了足够的考虑，保证了最稳定通信方式，即最接近 MEC 服务器，系统的平均完成时间最小，在约束条件下最能满足车载应用的资源需求。

(2)任务车辆可容忍的时延阈值

车辆内的智能设备有对实时性需求较高的车载应用，同时也存在一些对时延不敏感的车载应用程序。MEC 服务器覆盖范围内的任务车辆数不同，在覆盖范围外的任务车辆与目标 MEC 服务器之间的网络时延也不一样，所以通过改变任务车辆可容忍的时延阈值来判断具体哪个算法更优。

本节设置任务车辆数量为 20 个，MEC 服务器的数量设置为两种算法趋于稳定时的个数。图 3.5 表明，随着最大容忍时延阈值的增加，每个 MEC 服务器的候选任务车辆集合最终将包括网络中的所有任务车辆，这将会使 MEC 服务器出现严重的超荷现象。因此在选取最合适的最大容忍时延阈值是非常重要的问题。一般移动速度较快的车辆对时延要求比较苛刻，而对于车速较慢的车辆影响不大。KCOA 充分的考虑的车辆移动性为其匹配最优的服务节点，因此能够在时延阈值允许的范围内成功的进行卸载计算。而 ROA 没有考虑车辆的高动态性，导致某些车辆由于超过最大容忍时延阈值而失败的去处理任务。因此，KCOA 充

分的考虑了车联网的高动态性，性能始终优于 ROA。

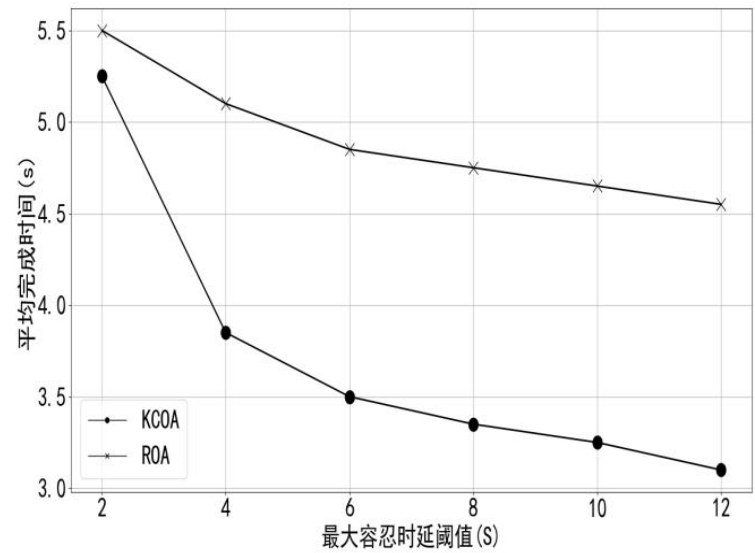


图 3.5 最大容忍时延阈值和平均完成时间关系图

3.4 本章小结

本章通过具体的仿真实验验证了基于 K-means 聚类卸载算法和随机卸载算法的性能优劣。相较于随机卸载算法来说，K-means 聚类卸载算法考虑了车辆的高动态性，更加符合现实中的情况。本章对车辆的位置、速度以及时延进行了建模，并通过算法对其进行优化。

4 联合计算卸载和资源分配算法

4.1 引言

MEC可以被看作是一种微型云服务平台,通过在接近用户端来完成计算、存储和通信等功能^[59,62]。因此一些对带宽和时延有更加苛刻需求的计算任务可以在边缘服务器上处理。随着MEC技术的逐渐成熟,VEC成为了研究的热点问题,车载应用对IoV技术提出了新的挑战。例如,增强现实、自动驾驶等新型技术。VEC能够解决车辆网络中计算能力不足的缺陷^[63-64]。在城市交通拥堵的区域以及停车场有大量闲置的车辆,这些车辆有大量的计算资源未被利用。将车辆作为边缘节点来提供计算服务,用户将任务卸载到网络的边缘去计算成为一种可行的方案^[65-67]。同样RSU、微基站都可以作为服务节点去使用,其中用户和服务端通过专用短程通信技术和设备对设备的通信技术进行通信。

目前国内外有大量学者针对 MEC 有了很多研究成果。文献[68]提出了一种基于李亚普诺夫的任务卸载优化算法该方法根据当前的参数动态调整所有任务的卸载决策,并对本地执行还是在 MEC 服务器上执行任务做出决策。文献[69]作者通过分类的多任务调度移动边缘计算降低了车辆网络的损失。该文章将问题建模为多目标帕累托优化,使车辆避免了移动时任务间的冲突。同样是利用闲置车辆的资源文献.[70]采用动态定价策略以最小化约束条件下 MEC 系统的平均成本通过不断调整价格来提高服务质量。文献[71]作者将车辆中可用资源用来运行分布式移动应用程序,并且通过博弈论来解决资源调度问题得到了很好的效果。文献[72]通过 SDN 控制上下文感知,为任务车辆选择合适的通信接入方案从而达到最优的卸载策略,仿真证明了该机制有效的降低了网络部署开销。文献[73]对终端用户卸载决策进行建模,大大降低了时延与能耗。文献[74]考虑到任务排队时间、移动车辆卸载延迟以及 RSU 有限的计算资源等因素,通过迭代算法对目标函数进行寻优结果相较于其他算法有较大的提升,这篇论文的主要贡献如下:

本章提出了一种在VEC环境下的联合卸载决策与资源分配机制,并将该机制建模为最优化问题。以往的方法都是离线优化的迭代非线性方法,本文提出了一种低复杂度非迭代的按照比例进行资源分配的算法,该方法无需考虑子信道有较高信噪比的约束,显著降低了复杂度。最后采用改进的粒子群算法(improved particle swarm optimization,IPSO)得出了最优的卸载决策。

4.2 系统模型和优化问题的制定

4.2.1 系统模型

图4.1是本章所提出的VEC系统场景。该场景中道路两侧部署路测单元RSU。因此，按照RSU的覆盖范围将道路划分为若干区域。此时，RSU充当接近用户端的微型云。本文 $V=\{v_1, v_2, \dots, v_N\}$ 表示任务车辆集合， $v_i (i \in [1, N])$ 可以将任务卸载到附近的RSU或者附近闲置的车辆上处理，最后将计算结果返回给 v_i 。由于车辆和路测单元RSU都可提供计算资源，因此，将它们统一表示为 $G=\{g_1, g_2, \dots, g_M\}$ 服务节点，这样能够节约时间减少不必要的能量损耗。每个RSU的无线信道和计算资源都专门用于相应的区域。本文主要研究的是具有代表性的区域，因此将本文的体系结构应用到其它场景也是非常容易的。 v_i 的计算任务可以描述为 $OT_i=\{o_i, d_i, t_i^{\max}\}$ ，所有要卸载的任务是原子性的，不能分离。本章在计算卸载过程中网络拓扑结构保持不变^[75]。

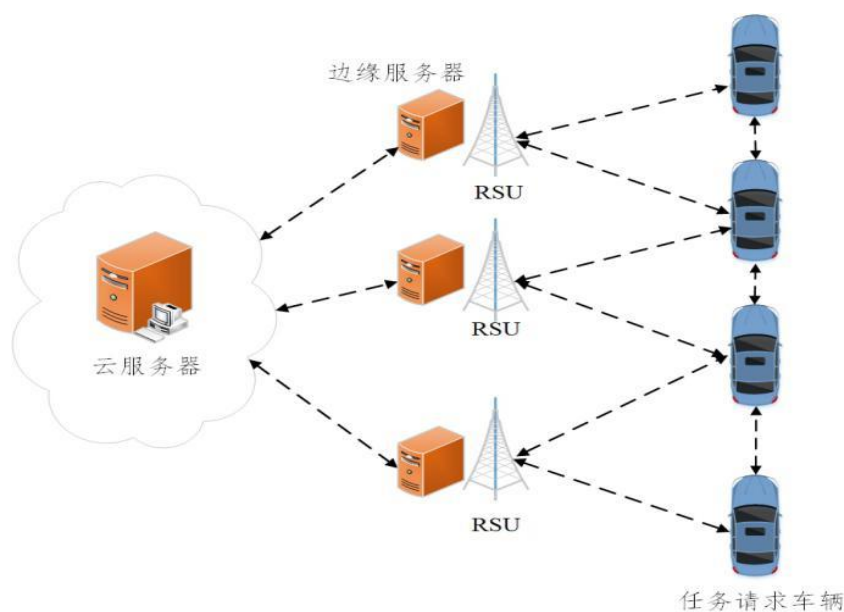


图 4.1 VEC 系统场景

4.2.2 通信模型

本文采用正交频分多址技术在用户和基站之间进行通信。用户数据区域的每个车辆 v_i 均可将其计算任务 OT_i 卸载到邻近的服务节点进行处理。且输入数据需要通过无线信道传输到服务节点，请求车辆的上行传输速率可以定义为：

$$\gamma = \frac{B}{S} \log_2 \left(1 + \frac{p_i \eta^2}{\zeta \sigma^2} \right) \quad (4-1)$$

为了有效地复用频谱，车辆与基础设施通信模式和车辆与车辆通信模式在同一频段工作。该系统还为RSU分配了正交频谱，从而可以忽略RSU间的干扰。设B为系统带宽， p_i 为分配给车辆 v_i 的传输功率，白色高斯噪声 $\sigma^2 = (\mathfrak{g}_0 B)/S$ ，其中， \mathfrak{g}_0 是噪声谱密度， s 个子载波用 $s(1 \leq s \leq S)$ 来表示， η^2 表示 v_i 到服务节点的信道增益。因此，任务请求车辆 v_i 的信噪比(signal noise ratio, SNR)可以表示为 $SNR_{i,s} = \eta^2 / (\zeta \sigma^2)$ ，其中 ζ 是恒定的信噪比差距。

4.2.3 计算模型

在进行卸载计算时会涉及到处理时延、传输时延以及传输能耗；在本地计算涉及处理时延和计算能耗，本节将逐一说明在不同情况下的计算模型。

车辆 v_i 的计算能力为 c_i ，不同车辆所拥有的计算资源是不同的，则本地执行时延为：

$$t_i^{local} = \frac{d_i}{c_i} \quad (4-2)$$

根据文献[76]，本地计算能量消耗为：

$$e_i^{local} = \kappa d_i (c_i)^2 \quad (4-3)$$

式中， κ 为能耗参数^[77-78]，设置为 $\kappa = 10^{-28}$ 。

当 v_i 将其任务卸载到 $g_j (j \in [1, M])$ 进行处理时的传输时延为：

$$t_i^{trans} = \frac{o_i}{\gamma_{i,j}} \quad (4-4)$$

因为输出这些计算密集型任务的数据大小相对于输入数据非常小，因此本文中不考虑下行传输时延。

传输能耗为：

$$e_i^{trans} = p_i t_i^{trans} \quad (4-5)$$

在本文中卸载计算的能耗只考虑与传输能耗有关的能耗。

卸载完成后，服务节点 g_j 将对任务进行处理。计算过程所消耗的时间为：

$$t_i^{cal} = \frac{d_i}{c_j^i} \quad (4-6)$$

其中， c_j^i 表示 g_j 分配给 v_i 的CPU数量， o_i 是数据量， d_i 是所需CPU周期数， t_i^{\max} 是可容忍的时延阈值。

4.2.4 最优化问题制定

本文用 $\chi_i \in \{0,1\}$ 代表卸载决策，当 $\chi_i=1$ 时计算任务 OT_i 将被卸载，否则 OT_i 将在本地执行。因此，执行 OT_i 所产生的时延和能耗如下式所示：

$$t_i = \chi_i(t_i^{trans} + t_i^{cal}) + (1 - \chi_i)t_i^{local} \quad (4-7)$$

和：

$$e_i = \chi_i e_i^{trans} + (1 - \chi_i) e_i^{local} \quad (4-8)$$

本文中定义了用户效用作为 v_i 所产生的时延和能耗的权衡，如下所示：

$$u_i(\chi_i, p_i) = \alpha \frac{t_i^{local} - t_i}{t_i^{local}} + \beta \frac{e_i^{local} - e_i}{e_i^{local}} \quad (4-9)$$

其中， α 为时延权重因子， β 为能耗权重因子，且满足 $\alpha + \beta = 1, 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$ 。如果任务很紧急，用户可以适当提高时延权重。否则，车辆运行在低电量情况下，可以适当提高能耗权重因子。由此，可将上式定义为一个最优化问题，如下所示：

$$\begin{aligned} \text{P1: } & \max_{\Lambda, \mathbf{P}} \sum_{i=1}^N u_i(\chi_i, p_i) \\ \text{s.t.} & \\ \text{C1: } & \chi_i \in \{0,1\}, \forall i \in N \\ \text{C2: } & 0 < p_i < p_{\max}, \forall i \in N \\ \text{C3: } & \sum_{i=1}^N p_i = p_{\text{total}}, \forall i \in N \\ \text{C4: } & (1 - \chi_i)t_i^{local} + \chi_i(t_i^{trans} + t_i^{cal}) \leq t_{\max}, \forall i \in N \\ \text{C5: } & \psi_{i,s} \in \{0,1\}, \forall i \in N, s \in S \end{aligned} \quad (4-10)$$

其中： $\Lambda = (\chi_1, \chi_2, \dots, \chi_N)$ 表示卸载决策集合， $\mathbf{P} = \{p_1, p_2, \dots, p_N\}$ 表示车辆传输功率分配集合。约束条件 C1 表示决策是否进行卸载计算。约束条件 C2 表示车辆传输功率约束条件。C3 为总功率约束条件。约束条件 C4 表示完成任务所需的时间不能超过任务所能容忍的最大时延。在 C5 中 $\psi_{i,s} \in \{0,1\}$ 是子载波分配指示器，当 $\psi_{i,s} = 1$ 将子载波 s 分配且仅分配给车辆终端。

4.3 解决方案

本文将原问题分解成两个子问题：资源分配和卸载决策。根据式(4-9)可知当任务在本地执行时不提供任何用户效用，即 $\chi_i = 0$ 。因此，得到 P2 如下所示：

$$\begin{aligned}
 \text{P2: } & \max_{\Lambda, \mathbf{P}} \sum_{i=1}^N u_i(1, p_i) \\
 \text{s.t.} & \\
 & C1, C2, C3, C4, C5
 \end{aligned} \tag{4-11}$$

4.3.1 资源分配

首先本文根据初始的卸载决策集集合 Λ ，将问题 P2 简化成了系统容量最大化的资源分配问题 SP1。

$$\begin{aligned}
 \text{SP1: SysCap}(\Lambda) = & \max_{\mathbf{P}} \frac{B}{S} \sum_{i=1}^N \sum_{s=1}^S \psi_{i,s} \log_2 \left(1 + \frac{p_i \eta^2}{\zeta \sigma^2} \right) \\
 \text{s.t.} & \\
 & C2, C3, C5
 \end{aligned} \tag{4-12}$$

上式的优化问题很难用多项式时间算法来解决它。因此，需要对原目标函数进行进一步处理。为每个 v_i 分配的功率可以用拉格朗日乘子法求解^[79]，得出的结果是：

$$\begin{aligned}
 \frac{1}{\Omega_1} \frac{S_1}{S} (\log_2(1 + \text{SNR}_{i,1} \frac{p_i - R_i}{S_1}) + \log_2 W_i) = \\
 \frac{1}{\Omega_k} \frac{S_i}{S} (\log_2(1 + \text{SNR}_{i,1} \frac{p_i - R_i}{S_i}) + \log_2 W_i)
 \end{aligned} \tag{4-13}$$

其中：

$$R_i = \sum_{s=2}^{S_i} \frac{\text{SNR}_{i,s} - \text{SNR}_{i,1}}{\text{SNR}_{i,s} \text{SNR}_{i,1}} \tag{4-14}$$

$$W_i = \left(\prod_{s=2}^{S_i} \frac{\text{SNR}_{i,s}}{\text{SNR}_{i,1}} \right)^{\frac{1}{S_i}} \tag{4-15}$$

其中， S_i 为分配给请求车辆 v_i 的子载波数， $\text{SNR}_{i,s}$ 代表第 i 个用户在子载波 s 上接收到的信噪比。 $\Omega_1 : \Omega_2 : \dots : \Omega_N$ 是归一化的比例常数并且 $\sum_{i=1}^N \Omega_i = 1$ 。由参考文献[80]可知：

$$S_1 : S_2 : \dots : S_N \approx \Omega_1 : \Omega_2 : \dots : \Omega_N \tag{4-16}$$

因此，在式(4-13)中，我们可以用 S_i 代替 Ω_i ，然后成立方程组，写成矩阵形式为：

$$\begin{bmatrix} a_{N,N} & 0 & \cdots & 1 \\ 0 & a_{N-1,N-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_N \\ p_{N-1} \\ \vdots \\ p_1 \end{bmatrix} = \begin{bmatrix} y_N \\ y_{N-1} \\ \vdots \\ p_{\max} \end{bmatrix} \quad (4-17)$$

对稀疏矩阵进行三角分解后得出：

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{N,N}} & \frac{1}{a_{N-1,N-1}} & \cdots & 1 \end{bmatrix} \quad (4-18)$$

$$\mathbf{U} = \begin{bmatrix} a_{N,N} & 0 & \cdots & 1 \\ 0 & a_{N-1,N-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 - \sum_{i=2}^N \frac{1}{a_{i,i}} \end{bmatrix} \quad (4-19)$$

其中：

$$a_{i,i} = -\frac{S_1}{S_i} \frac{SNR_{i,1} W_i}{SNR_{1,1} W_1} \quad (4-20)$$

$$y_i = \frac{S_1}{SNR_{1,1}} (W_i - W_1 + \frac{SNR_{1,1} R_1 W_1}{S_1} - \frac{SNR_{i,1} R_i W_i}{S_i}) \quad (4-21)$$

由上式此可得：

$$p_1 = (p_{total} - \sum_{i=2}^N \frac{y_i}{a_{ii}}) / (1 - \sum_{i=2}^N \frac{1}{a_{ii}}) \quad (4-22)$$

$$p_i = (y_i - p_1) / a_{ii} \quad (4-23)$$

算法一.比例资源分配算法

输入：初始卸载决策集合 Λ 。

输出：功率分配集合 \mathbf{P}_{best} 。

1 while $i \leq N$ do

2 将式 (4-23) 带入式 (4-11) 中计算初始功率分配集合 \mathbf{P} ，形成最大系统效用集合 u_i ，将 u_i 所对应的功率 p_i 从 \mathbf{P} 中删除。

3 将 \mathbf{P} 中剩余元素重新分配给任务车辆，找到 \mathbf{P} 中所能产生的最大的系统效用所对应的功率，将其从 \mathbf{P} 中移除，并存储到 u_i 中。

4 依次为任务车辆分配最优功率，计算最终每辆车所产生的系统效用。

5 end while

6 return P_{best}

4.3.2 卸载决策

当用上述方法求得在初始卸载集合情况下资源分配方案，然后将其数值解带入式(4-11)简化后得出最优化问题SP2，即：

$$\begin{aligned} \text{SP2: } \max_{\Lambda} Z(\Lambda) \\ \text{s.t.} \\ C1, C4 \end{aligned} \quad (4-24)$$

本文采用 IPSO 求解上述问题。迭代直到实验现象趋于平稳或者达到迭代次数终止，其中 IPSO 的速度以及位置公式如下式所示：

$$u_i^{k+1} = \omega u_i^k + \tau_1 \text{rand}_1 * (pbest_i - x_i^k) + \tau_2 \text{rand}_2 * (gbest - x_i^k) \quad (4-25)$$

$$x_i^{k+1} = x_i^k + u_i^k \quad (4-26)$$

$$g_i^k = l_i^k \bmod + 1 \quad (4-27)$$

其中， ω 为惯性权重系数， τ_1 和 τ_2 为学习因子， rand_1 和 rand_2 是在[0,1]区间上的随机数， $pbest_i$ 为粒子 i 的最佳位置， $gbest$ 为群体中所有粒子的最佳位置， u_i^k 表示第 k 次迭代粒子 i 的速度矢量， x_i^k 表示第 k 次迭代粒子 i 的位置矢量。由于PSO的位置变量 x_i 是连续的， v_i 中任务分配是离散的集合，因此本文运用最小位置规则(small position value, SPV)把连续的位置矢量 $\mathbf{x}_i^k = [x_1^k, x_2^k, \dots, x_N^k]$ 转化成离散的矢量 $\mathbf{l}_i^k = [l_1^k, l_2^k, \dots, l_N^k]$ ，然后将矢量 \mathbf{l}_i^k 的每个元素映射到服务节点矢量 $\mathbf{G} = [g_1^k, g_2^k, \dots, g_M^k]$ ，具体通过式(4-27)来实现。

4.3.3 改进的粒子群算法

粒子群算法在迭代前期的效率较高，到达迭代后期时，效率会逐渐变差，并且将会陷入局部最优和早熟现象。因此，本章采用粒距来改进算法，如式(4-28)所示。

$$D = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{m=1}^M (pbest_{im} - \overline{pbest_m})^2} \quad (4-28)$$

其中： D 为粒距，表示粒子间的分散程度， D 越大粒子间越分散； N 为种群大小； M 为空间维数； $pbest_{im}$ 表示粒子 i 在 m 维的最优值， $\overline{pbest_m}$ 表示所有粒子在 m 维的均值。通过 D 来决定种群是否变异，变异操作加强了局部搜索的能力，本文设定阈值 $h=10$ 与 D 进行比较，若 D 小于 $h=10$ ，则基本位变异，变异概率为0.3，从而

避免早熟现象的发生。

算法二. 联合卸载决策与资源分配机制(joint offloading decision and resource allocation, JODRA)

输入: 算法迭代次数 $Iteration$, 计算任务 V , 服务节点 G , 粒子个数 N 。

输出: 最优卸载决策集合 Λ_{best} 。

```

1  start
2  生成个体的初始种群。
3  for k=1 to Iteration do
4      for i=1 to N
5          更新粒子的速度和位置。
6      end for
7      求解式(4-12)基于当前卸载决策集合得出每个请求用户的传输功率。
8      变异条件判断, 根据  $D$  和阈值判断是否进行变异操作。
9      对所有个体进行评估, 选择产生下一代。
10 end for
11 return  $\Lambda_{best}$ 
12 end procedure

```

4.4 仿真分析

本文使用Matlab平台对JODRA机制进行仿真实验, 通过实验仿真结果来推断JODRA机制的性能。本节在IEEE 802.11p标准背景下展开。考虑路边配备了RSU其覆盖范围是100m, 系统总功率为1W, 系统带宽为1MHZ, 信噪比差距 $\zeta = 3.3$, 详细仿真参数如表4.1所示。

表 4.1 仿真参数

参数	描述	取值
o_i	任务计算大小/KB	400~500
d_i	任务所需 CPU 周期数/Megacycle	1200~2000
α, β	权重因子设置	0.5
c_i	车辆计算能力/GHZ	1~2
c_j	边缘节点计算能力/GHZ	5~6
p_{max}	车辆最大发射功率/dBm	30

4.4.1 平均 CPU 执行时间

本文将JODRA与遗传算法（Genetic algorithm,GA）进行比较，遗传算法中用户是以分布式的方式进行卸载决策，当有可能获得比本地计算更高的用户效用时，用户将进行卸载计算^[81]。

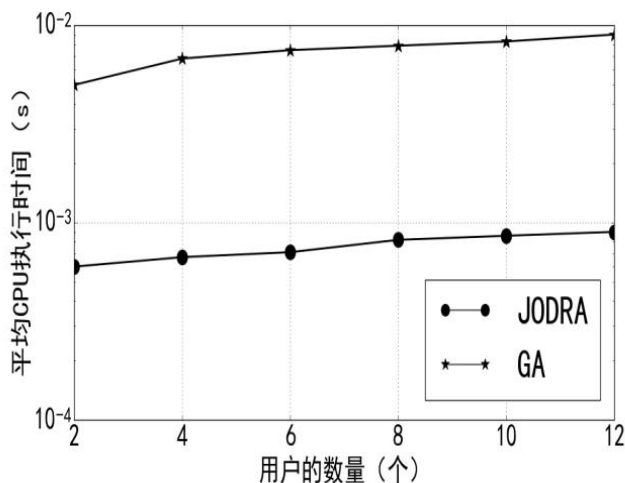


图 4.2 用户的数量与平均 CPU 执行时间关系

由图 4.2 可知，随着请求车辆数目的增加，两种算法之间的性能差异逐渐明显。GA 的执行时间比 JODRA 慢了一个数量级，并且 JODRA 为本实验提供了更高的实现能力。JODRA 机制将原问题转化为俩个子问题的相互迭代求解。由式 (4-22)和(4-23)可知该过程需要一次除法和 $2(N-1)$ 次乘法和 $3(N-1)$ 次减法，因此复杂度是 $O(N)$ 。IPSO 算法的时间复杂度为 $O(N^2)$ 。因此，JODRA 机制总体的时间复杂度为 $O(N^2+N)$ 。而 GA 首先经过 $O(N \log^2 N)$ 迭代后得到最优拉格朗日乘数，然后完成逆向归纳的复杂度取决于节点的数量,即 $O(N)$ 。因此，GA 整体的时间复杂度为 $O(N^2 \log^2 N)$ ，这比本文所提出的算法复杂的多。

4.4.2 平均用户效用

本节中将JODRA与基本的粒子群算法(particle swarm optimization,PSO)以及全部卸载计算（All Offload Computing,AOC）进行比较。如图4.3所示，平均用户效用随着传输功率的增加而增加，这是由于多个用户分集增益的影响，当用户数量较多时此增益更为明显。JODRA赋予了每个任务请求车辆最大程度的数据传输速率。在子载波数 $S=64$ 的情况下，比较在不同传输功率限制下平均用户效用的变化情况,可以看出JODRA所得到的用户效用始终较高。PSO由于自身算法的缺陷，局部搜索能力较差迭代搜索能力逐渐降低，因次，无法得出最优的卸载策略。

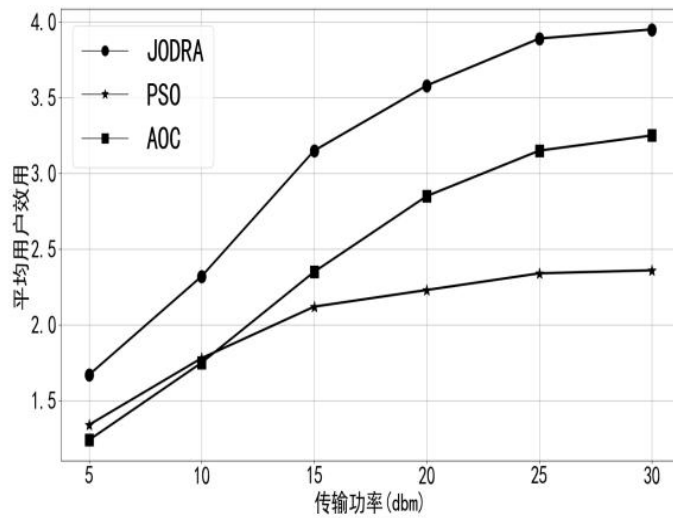


图 4.3 传输功率与平均用户效用关系

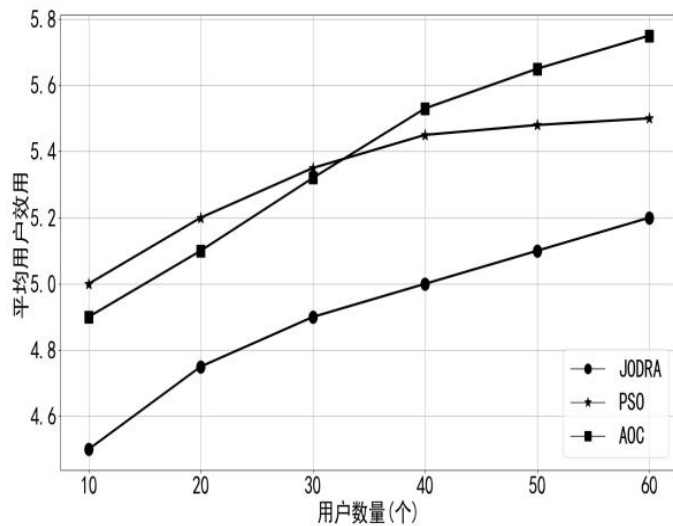


图 4.4 用户数量与平均用户效用关系

图4.4展示了在不同用户数量的情况下每种算法的平均用户效用。当用户数量小于30时JODRA的性能与AOC产生的最优解非常相近。这是因为当用户数量较少时，有充足的服务节点去处理数据，随着用户数量的增多，AOC体现出了明显的劣势，AOC没有充分的考虑时间约束问题，某些程序在处理过程中超过了预期的时间导致卸载失败。JODRA机制改善了传统等功率分配系统效率低的缺点并通过IPSO提高用户效用，随着用户数量的增加JODRA的优势越发突出。

根据任务卸载所得出的实验结果与传统本地处理任务进行的性能指标对比。经过本文所提出的JODRA机制实现的任务卸载决策与功率分配，得到了较优的卸载策略。通过JODRA机制所得到的产生的时延与能耗与全部卸载方法所得到的时延与能耗对比分别降低了6.8%、8.1%，说明JODRA能够有效的降低车载系统所

产生的时延以及能耗。

4.5 本章小结

本文设计了一种VEC系统架构，构建了联合计算卸载和资源分配机制，在此基础上，本文提出了一种低复杂度的比例资源分配算法，并结合改进的粒子群算法对卸载决策进行优化。实验表明该方案收敛速度快，大大降低了计算复杂度，解决了IoV计算能力不足的缺陷。在达到系统容量最大化的同时以最优卸载策略使车辆中的计算资源得到充分利用。通过多组实验对比表明该机制有效的降低了系统时延和能耗，并且具有可行性和高效性。在边缘环境下由于环境的不确定性，未来边缘设备间的通信干扰是需要攻克的热点问题。

5 总结与展望

5.1 本文总结

基于移动边缘计算的车联网架构是一种创新的网络范例，可以满足当下大部分车载应用的服务需求。MEC 目的是在运用云端的功能通过将大量的计算和存储资源推向网络边缘。车辆用户之间的直接交互以及 MEC 服务器的无线通信技术为需要超低延迟的应用程序带来了可能性，延长终端设备的电池寿命并为网络运营商带来了客观的收益。但是，由于车联网复杂的无线环境等原因和 MEC 有限的计算能力为实际工作带来了很多困扰。在本论文中，我们对 MEC 以及其研究前景进行了全面的概述。首先，我们总结了基于 MEC 的车联网架构的关键组件及其建模方法，例如计算任务，通信以及车辆的移动模型和 MEC 服务器的计算方式，这将为表征 MEC 的延迟和能耗性能作为依据。根据系统建模，我们总结了近几年有关在 MEC 环境下车联网最新研究成果，这些架构利用了任务分割、无线通信、资源联合分配、MEC 服务器调度以及多服务器选择和协作等方法。车载终端是车联网通信网络中重要的一个环节，然而车载设备有限的计算资源和存储能力是限制车联网发展的关键因素，本文建立了和车辆移动模型、位置模型、计算与通信模型，在此基础上以时延和能耗为评价标准，并推出了一种新型的联合计算卸载与资源分配机制。本文主要研究了 VEC 系统架构中的计算卸载策略，从时延和能耗角度，进行了相关分析和对比，具体研究内容如下：

(1) 对 MEC 环境下车辆计算卸载模式进行了研究和分析，车载应用产生的服务需求可以卸载到 MEC 服务器进行处理，也可以卸载到邻近的车辆终端进行处理。通过对车辆的位置以及速度进行建模然后根据 K-means 聚类算法对车辆通信模式进行合理选择，并对比随机卸载算法对系统平均完成时间的影响。

(2) 针对基于 MEC 的车联网架构设计了一种联合计算卸载和资源分配机制。在该机制中，将原始多目标问题化解为复杂度较低的比例资源分配算法和改进的粒子群算法两个问题分别求解。本文缓解了 IoV 系统计算能力不足的缺陷，在达到系统容量最大化的同时以最优卸载策略使车辆中的计算资源得到充分利用。通过多组实验对比表明该机制有效的降低了系统时延和能耗，并且具有可行性和高效性。

5.2 工作展望

本文将 MEC 引入到车联网中，主要从时延、能耗两方面进行研究，在此基础上，还可以对以下内容进行探索，为车联网行业进一步发展具有重要的研究意义。

（1）在本文的研究中，是以时延和能效固定的权重系数来作为评价标准，从个性化以及车载应用的多样化来讲，无法达到用户满意度的预期效果。在后续的研究中，可以引入关于时延和能耗的用户个性化模型，通过博弈论进行优化从而可以满足车载应用的多样化需求。

（2）事实上，此类研究还停留在理论层面真正达到落地还需要段路程，由于运营商之前的竞争等因素，MEC 的部署以及统一化、规范化、标准化是我们需要解决的问题，考虑到车辆中的计算任务卸载到 MEC 服务器过程中会产生高昂的费用，包括通信费用以及计算费用，因此，接下来的工作应该将费用代价考虑到模型当中。

（3）本文第四章用到了改进的粒子群算法，由于其局限性无法得出最优的卸载方案，只能得到近似最优解。在边缘环境下由于环境的不确定性，未来边缘设备间的通信干扰是需要攻克的热点问题。

参考文献

- [1]M. Gerla, E. K. Lee, G. Pau, et al. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds[C]. IEEE World Forum on Internet of Things, Seoul, 2014, 241-246.
- [2]S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, et al. A comprehensive survey on vehicular ad hoc network[J]. Journal of network and computer applications, 2014, 37(1): 380-392.
- [3]吴振华,胡鹏. VANET 中路由协议分析[J]. 通信学报, 2015, 36(11): 75-84.
- [4]Kaur R, Singh T P, Khajuria V. Security issues in Vehicular Ad-Hoc Network (VANET)[C]//2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). Tirunelveli: IEEE Press, 2018: 884-889.
- [5]Zhang Ning, Yang Peng, Ren Ju, et al. Synergy of big data and 5G wireless networks: opportunities, approaches, and challenges[J]. IEEE Wireless Communications, 2018, 25(1): 12-18.
- [6]Yang Fangchun, Wang Shangguang, Li Jinglin, et al. An overview of Internet of Vehicles[J]. China Communications, 2014, 11(10): 1-15.
- [7]Kolios P, Panayiotou C, Ellinas G. ExTraCT: Expediting offloading transfers through intervehicle communication transmissions[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(3): 1238-1248.
- [8]Bitam S, Mellouk A, Zeadally S. VANET-cloud: a generic cloud computing model for vehicular Ad Hoc networks[J]. IEEE Wireless Communications, 2015, 22(1): 96-102.
- [9]Dastjerdi A V, Buyya R. Fog Computing: Helping the Internet of Things Realize Its Potential[J]. Computer, 2016, 49(8): 112-116.
- [10]Aazam M, Huh E N. Dynamic resource provisioning through Fog micro datacenter[C]. IEEE International Conference on Pervasive Computing and Communication Workshops, 2015,105-110.
- [11]Morenovozmediano R, Montero R S, Huedo E, et al. Cross-Site Virtual Network in Cloud and Fog Computing[J]. IEEE Cloud Computing, 2017, 4(2): 46-53.
- [12]M. Wei, M. Long and J. Yang. Research on Energy Internet Collaborative Sche-

- duling Considering the Complex Interaction of Power System and Transportation System[C]. IEEE Conference on Energy Internet and Energy System Integration (EI2), 2018, 1-4.
- [13] K. M. Alam, M. Saini and A. E. Saddik. Toward Social Internet of Vehicles: Concept, Architecture, and Applications[J]. IEEE Access, 2015, 3, 343-357.
- [14] Chang Z, Zhou Z, Ristaniemi T, et al. Energy Efficient Optimization for Computation Offloading in Fog Computing System[C]. IEEE Global Communications Conference, 2017,1-6.
- [15] 中国信通院.车联网白皮书（C-V2X 分册）[EB/OL].[2019,12,26]
- [16] 赵冠宇.基于进化计算的车联网路测单元部署优化方法[D]. 长春：吉林大学,2019.
- [17] WU T,LIAO W, CHANG C.A Cost-Effective Strategy for Road-Side Unit Placement in Vehicular Networks[J]/IEEE Transactions on Communications,2012,60(8):2295-2303.
- [18] Huang X, Li P and Yu R. Social Welfare Maximization in Container-Based Task Scheduling for Parked Vehicle Edge Computing[J]. IEEE Communications Letters, 2019, 23(8): 1347-1351.
- [19] HUANG X, YU R,KANG J, et al. Exploring Mobile Edge Computing for 5G-Enabled SoftwareDefined Vehicular Networks[J]. IEEE Wireless Communications, 2017, 25(6): 55-63.
- [20] CHEN X,LIU Z,CHEN Y,et al. Mobile Edge Computing Based Task Offloading and Resource Allocation in 5G Ultra-Dense Networks[J].IEEE Access, 2019, 7: 184172-184182.
- [21] Hou X, Li Y, Chen M, et al. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures[J]. IEEE Transactions on Vehicular Technology, 2016, 65(6): 3860-3873.
- [22] Yan J, Bi S Z, Ying J and Tao M. Optimal Task Offloading and Resource Allocation in Mobile-Edge Computing With Inter-User Task Dependency[J]. IEEE Transactions on Wireless Communications, 2020, 19(1): 235-250.
- [23] Feng J, Liu Z, Wu C and Ji Y. AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling[J]. IEEE Transactions on Vehicular

- Technology, 2017, 66(12): 10660- 10675.
- [24] 张天, 汤利顺, 王彦聪, 等. C-V2X 标准演进及产业化综述[J]. 汽车文摘, 2020(02):22-28.
- [25] 顾海燕. 车联网环境下高速公路车辆跟驰模型及仿真研究[D].南京:东南大学,2017.
- [26] 王群, 钱焕延. 车联网体系结构及感知层关键技术研究[J]. 电信科学, 201228(12):1-9.
- [27] 缪立新, 王发平. V2X 车联网关键技术研究及应用综述[J]. 汽车工程学报, 2020, 10(01):1-12.
- [28] DTANGE V, KAT Z, KUN J, et al. Intelligent and connected vehicles: Current status and future perspectives [J], 中国科学, 2018, 061(010): 1446-1471.
- [29] LIU S, LIU L, TANG J, et al. Edge Computing to Autonomous Driving: Opportunities and Challenges[J]. Proceedings of the IEEE, 2019, 107(8): 1697-1716.
- [30] 于琴. 4G 通信技术在车联网中的应用研究[J]. 安徽电子信息职业技术学院学报, 2015, 14(06): 3-12.
- [31] HUANG C, LU R, CHOOKK R. Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges[J]. and Security and Forensic Challenges[J]. IEEE Communications Magazine, 2017, 55(11): 105-111.
- [32] PAPADIMITRATOS P, FORTELLE A D L, EVENSSEN K, et al. Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intelligent Transportation[J]. IEEE Communications Magazine, 2019, 47(11): 84-95.
- [33] MACH P, BECVAR Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J], IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [34] SABELLA D, VATLLANT A, KUURE P, et al. Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things[J]. IEEE Consumer Electronics Magazine, 2016, 5(4): 84-91.
- [35] TRAN T X, HAJISAMI A, PANDEY P, et al. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges[J]. IEEE Communications Magazine, 2017, 55(4): 54-61.

- [36] 贾庆民. 5G 移动通信网络中缓存与计算关键技术研究[D]. 北京: 北京邮电大学, 2019.
- [37] ZHANG K, MAO Y, LENG S, et al. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks[J]. IEEE Access, 2016, 4: 5896-5907.
- [38] Sun X, Ansari N. EdgeIoT: Mobile Edge Computing for the Internet of Things[J]. IEEE Communications Magazine, 54(12): 22-29.
- [39] Yu R, Zhang Y, Gjessing S, et al. Toward cloud-based vehicular networks with efficient resource management[J]. IEEE Network, 27(5): 48-55.
- [40] Mahmoodi S E, Subbalakshmi K P, Sagar V. Cloud offloading for multiradio-enabled mobile devices[C]//2015 IEEE international conference on Communications(ICC). London, UK. IEEE Press, 2015: 5473-5478.
- [41] Miettinen A P, Nurminen J K. Energy efficiency of mobile clients in cloud computing[J]. Hoc-Cloud, 2010, 10: 4-4.
- [42] Melendez S, McGarry M P. Computation offloading decisions for reducing completion time[C]//2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC). Las Vegas, NV, USA. IEEE Press, 2017: 160-164.
- [43] Nimmagadda Y, Kumar K, Lu Y H, et al. Real-time moving object recognition and tracking using computation offloading[C]. Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, 2010.
- [44] Seshasayee B, Nathuji R, Schwan K. Energy-Aware Mobile Service Overlays: Cooperative Dynamic Power Management in Distributed Mobile Systems[C]. Autonomic Computing, 2007. ICAC'07. Fourth International Conference on, 2007.
- [45] Hong J, Landay J. An Infrastructure Approach to Context-Aware Computing[J]. Human-computer Interaction, 16(2): 287-303.
- [46] Wang S, Zhang X, Zhang Y, et al. A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications[J]. IEEE Access: 1-1.
- [47] You C, Huang K, Chae H, et al. Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading[J]. IEEE Transactions on Wireless Communications, 2017, 16(3): 1397-1411.

- [48] Chen X. Decentralized Computation Offloading Game for Mobile Cloud Computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2015, 26(4): 974-983.
- [49] 焦捷. 移动边缘计算(MEC)中任务协同调度策略[D]. 电子科技大学.
- [50] Sun Xiang, Ansari N. Primal: profit maximization avatar placement for mobile edge computing[C]. 2016 IEEE International Conference on Communications, Kuala Lumpur: IEEE Press, 2016: 1-6.
- [51] Plachy J, Becvar Z, Strinati E C. Dynamic resource allocation exploiting mobility prediction in mobile edge computing[C]. IEEE International Symposium on Personal. Valencia: IEEE Press, 2016: 1-6.
- [52] Al-Sayed M M, Khattab S, Omara F A. Prediction mechanisms for monitoring state of cloud resources using Markov chain model[J]. Journal of Parallel & Distributed Computing, 2016, (96): 163-171.
- [53] Chiang M, Zhang T. Fog and IoT: An Overview of Research Opportunities[J]. IEEE Internet of Things Journal, 2016, 3(6): 854-864.
- [54] Abedin S F, Alam M G R, Tran N H, et al. A Fog based system model for cooperative IoT node pairing using matching theory[C]. IEEE Network Operations and Management Symposium, 2015, 309-314.
- [55] Chih-Hsiu Zeng, Kwang-Cheng Chen. Social Network Analysis Facilitates Cognition in Large Wireless Networks: Clustering Coefficient Aided Geographical Routing[J]. Cognitive Communications and Networking IEEE Transactions, 2018, 4(3): 618-634.
- [56] 左利云, 曹志波, 董守斌. 云计算虚拟资源的熵优化和动态加权评估模型[J]. 软件学报, 2013, (8): 247-256.
- [57] Zhang H, Zhang Y, Gu Y, et al. A Hierarchical Game Framework for Resource Management in Fog Computing[J]. IEEE Communications Magazine, 2017, 55(8): 52-57.
- [58] Zhang H, Xiao Y, Bu S, et al. Fog computing in multi-tier data center networks: A hierarchical game approach[C]. IEEE International Conference on Communications, 2016, 1-6.
- [59] Shi W S, Zhang X Z, Wang Y F, et al. Edge computing: current situation and

- outlook[J]. Journal of Computer Research Development, 2019, 56(1): 69-89.
- [60] Zhao Z M, Liu F, Cai Z P, et al. Edge computing: Platforms, applications and challenges[J]. Journal of Computer Research Development, 2018, 55(2): 327-337.
- [61] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading [J].IEEE Communication Survey & Tutorials, 2017, 19(3): 1628-1656.
- [62] Hu Y C, Patel M, Sabella D, et al. Mobile edge computing-A key Technology towards 5G[J]. ETSI White Paper, 2015, 11(11): 1-16.
- [63] Huang C M, Chen Y F, Xu S, et al. The Vehicular Social Network (VSN)-Based Sharing of Downloaded Geo Data Using the Credit-Based Clustering Scheme[J]. IEEE Access, 2018, 6:58254-58271.
- [64] Zhou H, Wang H, Chen X, et al. Data offloading techniques through vehicular ad hoc network: A Survey[J]. IEEE Access, 2018: 65250–65259.
- [65] Chen X. Decentralized computation offloading game for mobile cloud computing[J]. IEEE Trans on Parallel and Distributed Systems, 2015, 26(4): 97-4983.
- [66] You C S, Huang K B, et al. Energy-efficient resource allocation for mobile-edge computation offloading[J]. IEEE Trans on Wireless Commnications, 2017, 16(3): 1397-1411.
- [67] Malandrino F, Casetti C, Chiasserini C F, et al. The Role of Parked Cars in Content Downloading for Vehicular Networks[J]. Vehicular Technology IEEE Transactions on, 2014, 63(9): 4606-4617.
- [68] Hu Y, Cui T, Huang X, et al. Task Offloading Based on Lyapunov Optimization for MEC -assisted Platooning[C]//IEEE 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, Oct. 23-25, 2019. Piscataway: IEEE, 2019: 1-5.
- [69] Zheng X, Chen Y F, Alam M, et al. Multi-Task Scheduling Based on Classification in Mobile Edge Computing[J]. Electronics, 2019, 8(9): 938.
- [70] Han D, Chen W, Fang Y. A Dynamic Pricing Strategy for Vehicle Assisted Mobile Edge Computing Systems[J]. IEEE Wireless Communications Letters, 2019, 8(2):

420-423.

- [71] Huang X, Yu R, Liu J and Shu L. Parked Vehicle Edge Computing: Exploiting Opportunistic Resources for Distributed Mobile Applications[J]. IEEE Access, 2018, 6: 66649-66663.
- [72] 张海波, 王子心, 贺晓帆. SDN 和 MEC 架构下 V2X 卸载与资源分配[J]. 通信学报, 2020,41(01):114-124.
- [73] CHEN M,LIANG B,DONG M.Joint of Floading decision and resource allocation for multi-usermulti-task mobile cloud[c]. 2016 IEEE International Conference on Communications(ICC), Kuala Lumpur, 2016: 1-6.
- [74] 温琦丽, 洪高风, 苏伟. 一种基于车联网的边缘云资源分配优化方案[J]. 电信科学, 2019, 35(S2): 185-190.
- [75] Wang C, Yu F R, Liang C, 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.
- [76] Guo S T, Xiao B, Yang Y, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[J]. IEEE Transactions on Mobile Computing, 2019, 18(2): 319-333.
- [77] Pham Q, Leanh T, Tran N H, et al. Decentralized Computation Offloading and Resource Allocation for Mobile-Edge Computing: A Matching Game Approach[J].IEEE Access, 2018, 6: 75868-75885.
- [78] Wen Y, Zhang W, Luo H. Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones[C]//2012 Proceedings IEEE INFOCOM, Orlando, FL, Mar. 25-30, 2012. Piscataway: IEEE, 2012: 2716-2720.
- [79] D. Bertsekas, Nonlinear Programming. Athena Scientific, 1995.
- [80] Wong I C, Shen Z, Evans B L, et al. A low complexity algorithm for proportional resource allocation in OFDMA systems[C]//2004 IEEE Workshop on Signal Processing Systems Design and Implementation, Austin, TX, USA, Oct. 13-15, 2004. Piscataway: IEEE, 2004:1-6.
- [81] W. Zhang, Y. Wen, and D. O. Wu. Collaborative task execution in mobile cloud computing under a stochastic wireless channel[J]. IEEE Trans. Wireless Community, 2015, 81-93.

致谢

时间过得很快又到了离别的时刻，短短的研究生生活就这样画上了句号，不舍也有些许遗憾，想起来无数个在实验室日日夜夜，点点滴滴，有好多感慨与不舍。我想这就是我人生中最宝贵的财富。从大学的懵懵懂懂，再到研究生历练和熏陶，在知识不断的积累过程中，同样也重新审视了自己。我觉得三年的研究生生活，不止是在学术上面的造诣，更重要的是使精神层面更加充实。此时此刻，纵然有千言万语也难以言表，我想感谢这些日子中，默默支持我的父母和老师。首先，我要将最真挚的感谢给我的导师张文柱教授！张文柱老师是我学习的榜样，他学风严谨，为人师表，在生活中给予了我莫大的鼓励与帮助，真正设身处地为学生着想。在科研道路中，张文柱老师更是给了我许多中肯的建议，让我避开很多弯路，找到了非常适合我而且感兴趣的研究领域。在老师的指导下使我更加全面严谨的看待问题，从中得到了莫大的财富。张老师为我们提供了完善的科研设备和舒适的科研环境，无论我们遇到什么困难在张老师的帮助下都能游刃有余，我只想说师恩难忘。同时，我要感谢实验室的高鹏师兄，孔维鹏师兄，孙瑞华师姐感谢他们在科研上对我的指导以及生活中对我的关照；感谢我的同学谢书翰，杨子轩，以及王楠，井小浩，张梦豪在我的研究生生活中的帮助与陪伴。在我们亦师亦友的关系中开心的度过了这三年。最后我要感谢我的父亲母亲，他们对我无私的支持与鼓励，为我学习与科研的道路清除了诸多的障碍，我觉得自己非常幸运能在这个家庭中长大。最后再次感谢所有支持、关心和照顾我的人，此刻把最真挚的祝愿送给你们！

攻读硕士学位期间取得的研究成果

1. 发表论文情况

[1]车联网边缘计算的多目标均衡任务卸载方法研究[J].小型微型计算机系统.已录用.

本人署名情况：本人第一、导师第二

发表/录用日期：2021-3-1

[2]基于移动边缘计算的车联网任务卸载研究与仿真[J].系统仿真学报.已发表.

本人署名情况：本人第一、导师第二

发表/录用日期：2021-3-9

2. 参与的科研项目

[1]西安建筑科技大学. 面向 5G 的移动边缘计算关键技术研究.陕西省重点研发计划项目.