

车辆边缘计算环境下任务卸载研究综述

李智勇 王 琦 陈一凡 谢国琪 李仁发

(湖南大学信息科学与工程学院 长沙 410082)

(嵌入式与网络计算湖南省重点实验室 长沙 410082)

摘 要 计算密集和延迟敏感型车辆应用的出现对车辆设备有限的计算能力提出了严峻的挑战,将任务卸载到传统的云平台会有较大的传输延迟,而移动边缘计算专注于将计算资源转移到网络的边缘,为移动设备提供高性能、低延迟的服务,因此可作为处理计算密集和延迟敏感的任务的一种有效方法。同时,鉴于城市地区拥有大量智能网联车辆,将闲置的车辆计算资源充分利用起来可以提供巨大的资源和价值,因此在车联网场景下,结合移动边缘计算产生了新的计算模式——车辆边缘计算。近年来,智能网联车辆数量的增长和新兴车辆应用的出现促进了对车辆边缘计算环境下任务卸载的研究,本文对现有车辆边缘计算环境下任务卸载研究进展进行综述,首先,从计算模型、任务模型和通信模型三个方面对系统模型进行梳理、比较和分析。然后介绍了最小化卸载延迟、最小化能量消耗和应用结果质量三种常见的优化目标,并按照集中式和分布式两种不同的决策方式对现有的研究进行了详细的归类和比较。此外,本文还介绍了几种常用的实验工具,包括 SUMO、Veins 和 VeinsLTE。最后,本文围绕卸载决策算法复杂度、安全与隐私保护和车辆移动性等方面对车辆边缘计算任务卸载目前面临的挑战进行了总结,并展望了车辆边缘计算环境下任务卸载未来的发展方向与前景。

关键词 车辆边缘计算;移动边缘计算;任务卸载;资源分配;车联网

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2021.00963

A Survey on Task Offloading Research in Vehicular Edge Computing

LI Zhi-Yong WANG Qi CHEN Yi-Fan XIE Guo-Qi LI Ren-Fa

(College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082)

(Key Laboratory of Embedded and Network Computing of Hunan Province, Hunan University, Changsha 410082)

Abstract The emergence of computation intensive and delay sensitive vehicle applications poses a severe challenge to the limited computing capacity of vehicle equipment. Offloading tasks to traditional cloud platforms have large transmission delays, and the cost of upgrading on-board computers is huge, so these two methods have some disadvantages in dealing with computation intensive and delay sensitive tasks. Mobile edge computing is a new computing paradigm, which focuses on transferring computing resources to the edge of the network, providing high performance, high reliability and low latency services for mobile devices. Therefore, it will be a more effective way to process computation intensive and delay sensitive tasks. Meanwhile, vehicles can act as both service requesters and service providers. In view of the large number of intelligent networked vehicles in urban areas, making full use of idle vehicle computing resources can provide huge resources and value. Therefore, combined with the mobile edge computing, a new computing paradigm is generated in the Internet of Vehicles scenario, called vehicular edge computing

收稿日期:2019-07-13;在线发布日期:2020-12-21。本课题得到国家重点研发计划“智能机器人”专项课题(2018YFB1308604)、国家自然科学基金(61976086,61672215,61702172)、湖南省自然科学基金青年科学基金(2018JJ3076)资助。李智勇,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为智能优化算法、智能感知与学习和嵌入式系统 CPS 模型。E-mail: zhiyong.li@hnu.edu.cn。王 琦,硕士研究生,主要研究方向为移动边缘计算和深度强化学习。陈一凡,博士研究生,主要研究方向为移动边缘计算和博弈论。谢国琪,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为嵌入式与信息物理系统、并行与分布式系统和安全关键系统。李仁发,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为计算机体系结构、嵌入式计算系统、信息物理融合系统和物联网。

(VEC). In recent years, the increase of the number of intelligent networked vehicles and the emergence of emerging vehicle applications have promoted the research on task offloading in vehicular edge computing. However, there is no detailed summary and analysis of the problems related to task offloading in VEC at present. This paper summarizes the research progress of task offloading in the existing vehicular edge computing. Firstly, the VEC system model is summarized, compared and analyzed, including computing model, task model and communication model. Specifically, the VEC computing model consists of a three-layer cloud structure of remote cloud (RC), edge cloud (EC), and vehicular cloud (VC), each of which has its own advantages in different aspects. The VEC task model is divided into critical applications (CAs), high-priority applications (HPAs) and low-priority applications (LPAs) according to the degree of application criticality to vehicles, and the dependencies between tasks are summarized. Several main communication protocols for task offloading in the Internet of Vehicles are introduced in the VEC communication model, and the influence of vehicle mobility on communication is analyzed. Secondly, we summarize three common optimization objectives in the task offloading of VEC from existing studies, namely, minimizing offloading delay, minimizing energy consumption and application quality of results. Thirdly, we summarize some models and methods of VEC task offloading, such as semi-Markov decision process, game theory, reinforcement learning, heuristic algorithm and contract theory. At the same time, the existing researches are classified according to centralizing and distributing two different decision-making methods, and the characteristics of these two decision-making methods are compared. Then, we describe several commonly used experiment tools, including SUMO, Veins and VeinsLTE, which can make simulation experiments more realistic and credible. In addition, the current challenges of task offloading in vehicular edge computing are summarized in section 6, including offload decision algorithm complexity, security and privacy protection and vehicle mobility. In view of the above problems, we have proposed some possible future research directions and development prospects. Finally, we summarize the whole paper in section 7.

Keywords vehicular edge computing; mobile edge computing; task offloading; resource allocation; Internet of Vehicles

1 引 言

随着智能交通系统和智能网联汽车的发展,车辆已经被赋予更多的计算、存储、传感和通信能力,许多新型的车载应用也正在涌现,如增强现实^[1]、虚拟现实、模式识别等. 根据 Gartner, Inc. 的预测,在 2020 年将有约 2.5 亿辆智能网联车辆^①,而根据 Intel 的预计,未来一个智能网联车辆为了做出安全的决策,需要计算和分析来自传感器的大量数据(大约 1 GB/s)^②. 传统的车辆已经不能满足大量数据对计算、存储资源的需求,应对复杂度高、数据密集、延迟敏感的应用时,车辆面临着巨大的挑战^[2].

云计算^[3] (Cloud Computing) 使用云服务器或者一组远程服务器提供集中式的计算和存储服务,被广泛地应用到华为、谷歌、亚马逊等公司的数据中

心,吸引了很多用户. 将车辆的计算任务通过卸载的方式交由远程的云服务器计算是一个解决方案,但是卸载到云服务计算的方式有几个关键的问题需要考虑. 首先,智能网联车辆中有许多数据密集和延迟敏感的计算任务,它们对延迟有较高的要求,如果将任务传输到远程云处理,即使云服务器的执行时间比较短,但由于卸载到云需要较高通信延迟,因此对于延迟敏感的任务是不可接受的. 其次,车辆与云服务器之间的通信需要高带宽,大量用户直接与云服务器通信,占据了带宽资源,严重影响了传输速率. 因此,对于车辆数据密集和延迟敏感的计算任务,传

① <https://www.gartner.com/en/newsroom/press-releases/2015-01-26-gartner-says-by-2020-a-quarter-billion-connected-vehicles-will-enable-new-in-vehicle-services-and-automated-driving-capabilities>
② <https://www.intel.com/content/www/us/en/automotive/driving-safe-ty-advanced-driver-assistance-systems-self-driving-technology-paper.html/>

统的云计算不再占据优势. 另外, 升级车载计算机是另一种选择, 但是升级硬件成本很高, 因此也不是很好的选择.

当数据在网络边缘产生时, 云计算不再是一种有效的数据处理方式, 在计算模式的领域内, 边缘计算^[4-5] (Edge Computing) 变得越来越重要, 尤其是移动网络中的移动边缘计算^[6-8] (Mobile Edge Computing, MEC). 与边缘计算相似的概念在一些研究中被提出, 例如: 微数据中心^[9]、Cloudlet^[10]、雾计算^[11-12] (Fog Computing), 它们都认为将位于网络边缘设备产生的数据交由边缘计算处理比在云端处理更有效, 这里将“边缘”定义为数据源和云数据中心之间通道中的任何计算和网络资源. 边缘计算将移动设备的部分或全部计算任务迁移到网络边缘设备执行, 弥补了移动设备计算能力不足以及电量有限的缺点, 提高了数据的传输性能, 增强了数据处理的实时性和可靠性, 同时又降低了云计算中心的计算和带宽负载. 图 1 是一种典型的移动边缘计算框架. 车辆应用产生的计算任务可以卸载到路侧单元 (Road Side Unit, RSU)、基站 (Base Station, BS)、边缘服务器等固定基础设施. 同时由于智能网联车辆具备了通信、存储和计算资源, 可以将车辆未充分利用的存储和计算资源看作可利用的“边缘”基础设施, 为其他用户提供服务^[13-15]. 因此在车联网场景下产生了新的计算模式, 车辆边缘计算 (Vehicular Edge Computing, VEC).

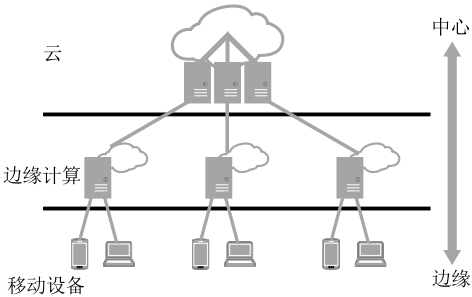


图 1 移动边缘计算框架

计算卸载是 VEC 的一个重要应用, 资源受限的车辆可以通过计算卸载的方式将任务交由其他资源充足的节点计算, 以加快处理速度、提高服务质量. 计算卸载主要包括卸载决策和计算资源分配两大问题, 其中卸载决策主要研究移动终端是否需要卸载以及卸载多少的问题, 而计算资源分配主要研究将需求卸载的计算任务卸载到何处的问

题^[16]. 车辆边缘计算相对于传统的云计算而言, 虽

然与数据源更近, 但是边缘设备的资源是受限的, 因此需要利用有限的计算资源, 满足大量车辆用户的请求, 协调多个车辆之间的计算卸载, 这对车辆计算任务的卸载决策和资源分配策略提出了更高的要求.

近年来 MEC 和 VEC 的任务卸载引起了研究人员的广泛关注. Mach 等人在文献[8]中从 MEC 计算卸载的三个重点研究领域: 计算卸载的决策、MEC 内计算资源的分配和移动性管理详尽地总结了 MEC 的架构和计算卸载问题. 在研究^[16]中, Xie 等人介绍了 MEC 的网络架构及其部署方案, 并分析和对比了不同的部署方案; 然后从卸载决策、资源分配和系统实现这 3 个方面对 MEC 计算卸载关键技术进行了研究. 文献[17]中, Li 等人从优化计算卸载策略、无线资源分配、计算资源分配和系统可扩展性四个方面对目前 MEC 系统中计算卸载方案的研究现状和成果进行了总结. Li 和 Gu 等人分别在文献[18]和文献[19]中对 MEC 在车联网中的应用进行了分析研究, 在文献[19]中还介绍了 MEC 在车联网中的应用、基于软件定义网络的车联网 MEC 研究现状以及车联网 MEC 应用实例, 并提出了车联网中部署 MEC 所面临的问题和挑战. Raza 等人在文献[20]中, 结合智能汽车的概念、服务、通信和应用, 阐述了 VEC 的体系结构. 但是目前还没有针对 VEC 环境下任务卸载相关问题进行详细总结和分析的研究. 本文通过对现有研究的分析, 对 VEC 环境下任务卸载的相关问题(包括系统模型、优化目标和卸载技术等)进行了全面的总结.

本文在第 1 节中介绍 VEC 计算任务卸载的背景和意义以及相关的研究现状; 第 2 节分别从计算模型、任务模型和通信模型三个方面对 VEC 的系统模型进行分析; 第 3 节总结 VEC 计算任务卸载三种常见的优化目标; 第 4 节按照集中式和分布式两种不同的决策方式对现有的研究进行详细地介绍、归类和总结; 第 5 节介绍 VEC 几种常用的实验工具; 第 6 节总结目前 VEC 任务卸载面临的几个挑战, 并提出相应未来研究方向; 第 7 节对全文进行总结.

2 VEC 系统模型

2.1 VEC 计算模型

VEC 计算模型由远程云、边缘云和车辆云三层云结构组成^[21], 如图 2 所示. 其中远程云是传统云

计算中的云资源,边缘云位于网络的边缘,可以提供更低延迟和更高带宽的服务,车辆云与边缘云通过无线通信连接. 本小节介绍了 VEC 环境下计算资源的分层结构,并对三类资源进行了详细的对比和分析.

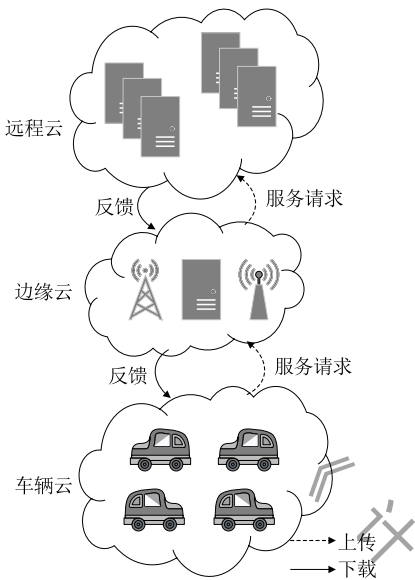


图 2 VEC 计算资源分层结构图

2.1.1 远程云

远程云(Remote Cloud, RC)通常具有超大的规模,能赋予用户超强的计算能力,对大规模数据聚合、数据挖掘、分析、优化和存储复杂数据,RC可以在短的时间内计算得到结果. 车辆应用产生的计算任务可以通过网络传输到远程云,在 RC 计算并将结果返回给车辆. 由于 RC 超强的计算能力,因此能够计算和分析复杂的计算任务,并能够提供较高的计算性能,但是因为 RC 位于网络的远端,将车辆计算任务传输到 RC 并得到相应的返回结果需要很大的传输时间开销. 因此对于数据密集型 and 延迟敏感型的计算任务,即使计算延迟较小,较大的传输延迟仍会使任务总卸载延迟较大. 同时,车辆产生的大量数据全部上传到 RC 会占用大量的网络带宽,从而造成网络性能和稳定性的下降.

2.1.2 边缘云

边缘云(Edge Cloud, EC)位于远程云和车辆云之间,由拥有计算能力的边缘设备组成,它是移动边缘计算中的重要组成部分,是解决 RC 传输延迟高的一种有效替代方案. 在 VEC 中,EC 可由边缘服务器、RSU、BS、无线接入点(Access Point, AP)以及移动设备等其他具备计算、存储和通信能力的边

缘设备组成. 因为 EC 更靠近车辆,处在网络的边缘,车辆到 EC 的传输延迟可以大大降低. 因此 EC 可以为车辆提供低延迟计算、高速缓存、位置感知、紧急管理等服务,并能用于实时交互. 对于数据密集型和延迟敏感型的车辆计算任务,例如:增强现实,实时视频分析,人类行为识别等任务,它们需要极低的延迟以便快速响应和决策,此时 EC 比 RC 具有更大的优势. 最新的文献综述^[22-23]详细列举了在物联网和无线网络迅速普及的背景下传统云计算固有的缺陷和瓶颈,并说明了边缘云在面向边缘设备海量数据时的独特优势. 但边缘计算作为云计算的延伸,未来将与云计算协同配合,为用户提供更高质量的服务.

2.1.3 车辆云

随着汽车工业中技术的演变和智能网联车辆的发展,车辆已被赋予更多的计算、存储、通信和传感资源. Hou 等人在文献[13]中提出,鉴于城市地区有大量的车辆,将空闲的车辆资源加以利用,可以为移动设备提供巨大的资源 and 价值. 通过整合车辆空闲的计算资源形成车辆云(Vehicular Cloud, VC),为其他车辆或者移动用户提供服务,可以大大提高服务和应用的质量. 智能网联车辆包含使用无线通信协议的设备,车辆能以较低的通信延迟接发计算任务,充分利用车辆上空闲的计算、存储资源,在保证车辆安全行驶的前提下,降低任务处理延迟,提高用户的驾驶体验. VC 中车辆的角色不是固定的,当车辆资源空闲时可以作为服务提供者,接受来自其他车辆或是移动设备的任务请求;当车辆需要卸载任务时,可以作为服务请求者,将任务卸载至其他拥有空闲计算资源的车辆.

表 1 从计算和存储能力、位置、延迟、移动性、通信和部署方式等方面详细对比了远程云、边缘云和车辆云的三层计算资源的异同. 由表 1 可知,虽然 EC 和 VC 在延迟方面表现更优,但是与 RC 相比,EC 和 VC 的计算和存储资源相对较低,当任务量较大时可能无法满足所有的卸载请求,对此需要制定合理的计算卸载决策和资源分配方案以最大限度满足用户需求. 同时,车辆具有较高的移动性,由于网络的动态性,将计算任务卸载至 VC 可能会造成卸载失败. 因此,远程云、边缘云和车辆云在不同方面各具优势,采用“云-边-端”相结合的计算模型,可以依据不同的优化目标和任务需求选择三层云资源中的一种或多种进行卸载.

表 1 VEC 计算资源对比

种类	远程云	边缘云	车辆云
计算和存储能力	强(无限)	较强(有限)	较强(有限)
位置	远	靠近车辆	靠近车辆
延迟	高	低	低
移动性	固定	固定	移动
通信	受带宽限制	实时	实时
部署方式	集中式	分布式	分布式

将 VC 直接与 RC 相结合提出了车辆云计算^[24-27] (Vehicular Cloud Computing, VCC), 而 VC 与 EC、RC 结合提出了车辆雾计算^[13,28-29] (Vehicular Fog Computing, VFC). VCC 与 VFC 的对比在文献[30]中给出,但这两者都将车辆视为能够提供计算、通信和存储资源的基础设施,以减少延迟、提高服务质量,这与本文所述 VEC 的本质特性类似. 3.3 节的表 3 列举了目前 VEC 环境下任务卸载研究^[31-56]对三类资源的使用情况. 由表 3 可知,目前车联网场景下车辆计算任务卸载主要采用 VC 和 EC 两种模型,RC 由于其固有的缺点对于 VEC 而言不是最优选择,VC 和 EC 是未来 VEC 的主力军.

2.2 VEC 任务模型

随着车辆计算能力的提升,越来越多的车载应用部署在车辆上. 按照计算任务的不同特点,我们对车辆计算任务进行了分类,分类方式有如下两种: (1) 按照车辆应用程序的关键程度分类; (2) 按照任务之间是否有依赖性进行分类.

2.2.1 车辆应用程序的关键程度分类

在文献[46]中,Feng 等人将车辆应用程序按照其关键程度分为三类:关键应用程序(Crucial Applications, CAs)、高优先级应用程序(High-Priority Applications, HPAs)和低优先级应用程序(Low-Priority Applications, LPAs). CAs 是和车辆安全相关的应用,是保证车辆和乘客安全的关键应用程序,如车辆控制、系统监控和事故预防等. 由于 CAs 和安全紧密相关,因此享有最高的优先级,车辆制造商必须保留充足的计算资源给这部分应用,不能因为 HPAs 和 LPAs 的存在而影响 CAs 的正常运行,同时这类任务也不允许卸载,只允许在本地执行,不属于计算任务卸载的范畴.

HPAs 包括与驾驶相关的应用和可选的安全增强应用,这类应用程序对车辆而言是重要但不是必须的,拥有较高的执行优先级,例如实时路径规划和路况提醒等. 这类应用允许出现延迟或卸载失败的情况,但不会影响车辆安全. LPAs 是一类为乘客提供娱乐服务的应用程序,它的优先级较低,例如语音

识别^[57],它允许驾驶员发出各种声音命令,通过语音识别命令计算机做一些响应,而不会使驾驶员分心. HPAs 和 LPAs 已经被部署到越来越多的车辆上,由于 HPAs 和 LPAs 不会影响到车辆的安全,因此可以将其进行卸载. 车辆会优先卸载优先级较高的任务,当计算资源充足时才会卸载低优先级的任务. 表 2 对常见的车辆应用程序按照其关键程度进行了分类.

表 2 常见车辆应用关键程度分类

应用关键程度	应用实例	卸载
CAs	车辆控制、碰撞预警、红绿灯警告、网上车辆诊断、道路湿滑检测等	不允许
HPAs	地图导航、平视显示器 ^[58] 、视野增强 ^[59] 、车辆传感 ^[60] 等	允许
LPAs	虚拟现实、语音识别、视频处理 ^[61] 、在线游戏等	允许

2.2.2 任务之间依赖性分类

按照计算任务之间的依赖关系可以将计算任务分为独立任务和依赖任务. 如果计算任务之间无联系则为独立任务,可以将任务并行地卸载,无需考虑任务之间的输入和时间约束,如文献[62-65]等的研究. 除独立任务外,应用程序的计算任务在数据输入和计时优先级方面常常相互依赖^[66]. 例如,一个视频导航应用程序可以分为四个部分^[67]:图像处理、人脸检测、相机自动定焦拍照和视频处理,其中图像处理任务之间的关系是依赖的,而人脸检测任务之间的关系是并行的. 当任务之间存在依赖关系时,如果需要在不同的计算节点上执行这些任务,就需要进行任务之间的传输数据,从而产生通信延迟和通信成本,此时需要考虑到任务之间的依赖性对卸载决策和资源分配方案的影响. 依据文献[49,66-68]等对依赖任务的研究,我们将相互依赖的计算任务按照其基本逻辑拓扑分为三类,分别为线形、树形和图形依赖. 图 3(a)~(c)分别代表了三种类型的依赖性任务,其中一个圆圈代表一个任务,箭头末尾的任务必须在箭头起始位置的任务完成之后才能执行. 任务都按顺序执行,输出数据作为其后续任务的输入. 为了详细地描述任务(任务数为 n),我们将三元组 $\varphi_k = (\omega_k, \alpha_k, \beta_k)$ 用于描述计算任务^[69],其中 $k = 1, 2, \dots, n$, ω_k 是第 k 个任务的计算量, α_k 和 β_k 分别是第 k 个任务的输入大小和输出大小. 显然总的计算量为 $\sum_{k=1}^n \omega_k$. 在图 3(a)的线形依赖任务中,有 $\alpha_k = \beta_{k-1}$; 在图 3(c)的图形依赖任务中,有 $\alpha_6 = \beta_3 + \beta_5$.

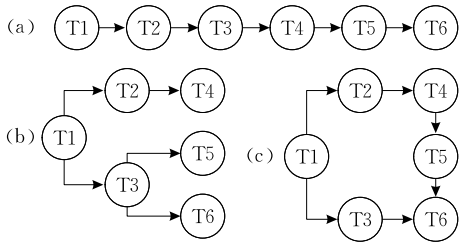


图 3 任务逻辑拓扑分类

我们利用数据结构有向无环图(Directed Acyclic Graph, DAG)来表示依赖任务之间的关系. 更具体的, 用一个下三角矩阵 $I=(I_{i,j})_{n \times n}$ 来表示任务 i 和 j 之间的依赖关系, 其中 $I_{i,j}=\{0,1\}, 1 \leq j < i \leq n$. 如果任务 i 需要任务 j 的输出作为输入, 则有 $I_{i,j}=1$, 否则 $I_{i,j}=0$. 例如, 图 3(c) 的任务依赖关系可以表示成式(1), $I_{6,3}, I_{6,5}$ 表示任务 T6 需要任务 T3 和 T5 的输出作为输入.

$$I=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad (1)$$

独立任务例如: 实时目标检测^[70], 从视频流中识别对象并将其标注, 视频不同帧的目标识别之间并无依赖关系. 依赖任务例如: 增强现实, 在文献[71]中研究了增强现实所包括的 5 个计算组件以及各个组件之间的依赖关系. 但不同应用计算任务之间的依赖关系应该按照该应用具体特征进行分析, 不同应用的计算任务之间具有不同的依赖关系, 相同的应用按照不同的计算任务划分粒度也会有不同的依赖关系, 因此计算任务之间的依赖关系不是绝对的、固定的, 可以按照不同的标准对其进行相应的改变.

2.3 VEC 通信模型

VEC 计算任务卸载过程主要通过无线通讯网

络将计算任务及其相关参数从服务请求者传输至服务提供者. 随着车联网的发展, 车载网络技术已经相对成熟, 车辆之间 (Vehicle-To-Vehicle, V2V) 的通信可以基于专用短程通信 (Dedicated Short-Range Communications, DSRC)^[72] 来实现, DSRC 是基于 IEEE 802.11p 设计的, 在 300 m 覆盖内 DSRC 的数据速率可以达到 27 Mb/s 左右. 车辆与固定基础设施之间 (Vehicle-To-Infrastructure, V2I) 的通信则可以使用 Long-Term Evolution (LTE)^[73-74] 和 DSRC, 与 DSRC 相比, LTE 的覆盖范围更广^[75], 服务质量更具有保障, LTE 还支持 350 km/h 的高移动性的用户设备, 可以很好地适应高速移动的车辆. 此外, 迅速发展的 5G 通信网络提供了一种全新的网络架构, 提供 10 Gbps 以上的峰值速率、更佳的移动性能、毫秒级时延和超高密度连接, 国际电信联盟无线电通信局定义了 5G 的三大典型应用场景为增强型移动宽带 (eMBB)、超低时延高可靠通信 (uRLLC) 和海量大规模连接物联网 (mMTC)^[76]. 其中 uRLLC 由于其即时、可靠、高效数据传输的特点, 如车联网、自动驾驶、自动化无人机等都是其主要实际应用, 基于 5G 的车联网将是未来 VEC 计算卸载的通信技术之一^[77-81]. DSRC、LTE、5G 等车联网络技术为 VEC 计算任务卸载提供了相对可靠的网络传输通道, 为高速移动下车辆的计算任务卸载提供了可靠的通信保障. 图 4(a) 为 VEC 中计算卸载 V2I 和 V2V 两种通信模式.

除了移动车辆外, 城市内还存在大量的停泊车辆 (Parked Vehicles, PV), 如图 4(b) 所示, 它们广泛地分布在街道停车场、室外停车场和室内停车场内^[13]. 与移动车辆不同, 这些停泊车辆在一定的较长时间内的位置相对不变, 因此通信链路更加稳定. 由于停泊车辆数量多、停留时间长、地理分布广、位置特定等特点, 在适当的通信条件下连接在一起时, 它们可以相互协作以完成大型计算任务. 因此, 这些停泊车辆成为计算基础设施, 提供大量

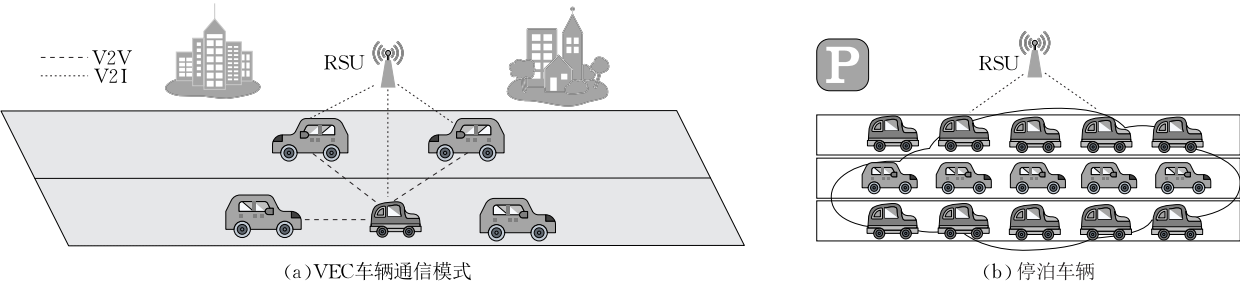


图 4 VEC 车辆通信模式和停泊车辆

计算资源,可以很好地解决单个车辆计算资源有限的问题,在更短的时间内完成计算任务,效率更高.例如,在文献[82-83]等研究中,都提出了将停泊车辆作为计算资源基础设施,为用户提供更好的服务.

3 VEC 优化目标

优化目标是 VEC 环境下任务卸载的研究目标,要根据优化目标来确立卸载模型,并使用适当的方法使目标最优(较优),本节将讨论在 VEC 环境中计算任务卸载几种主要的优化目标:(1)最小化卸载延迟;(2)最小化能源消耗;(3)应用结果质量.

3.1 最小化卸载延迟

计算任务卸载决策通常可以分为本地执行、部分卸载、全部卸载三种.本地执行是将车辆所产生的计算任务全部在本地进行计算,不需要进行计算卸载,所消耗的时间全部为任务在本地运行的时间.部分卸载是将计算任务部分在本地执行,另一部分通过计算迁移的方式卸载到其他资源更加充足的地方执行.全部卸载则是将车辆产生的计算任务毫无保留地卸载,不在本地执行.部分卸载和全部卸载所消耗的时间包括三个部分:将计算任务通过网络传输到边缘的时间、在边缘执行任务的时间和将计算结果返回给车辆的时间.随着智能车辆的发展,诸如视频流处理、增强现实等应用已经在车辆普及,它们具有数据密集和延迟敏感的特点,但是由于本地的计算能力有限,升级硬件成本又太高,如果任务计算延迟太高,会影响应用的服务质量,因此任务卸载是减少任务处理延迟的一种可行手段.如何降低在车辆应用程序中任务卸载的处理延迟是 VEC 卸载领域主要研究内容之一.卸载延迟通过具体的卸载策略和资源分配策略来决定,在该策略下使计算任务处理延迟最小化,从而提升应用的服务质量.

3.2 最小化能源消耗

移动设备的能耗通常是移动边缘计算中经常考虑的问题,能耗包括两种:执行能耗和传输能耗.计算任务本地执行消耗的能量只有本地处理器处理任务的执行能耗,而通过卸载消耗的能量则包含将计算任务通过网络传输到服务提供者的传输能

耗以及车辆接收来自服务提供者返回结果消耗的传输能耗.卸载耗能通常不包括边缘处理任务的执行能耗,车辆可以通过付费的方式使用 RC、EC 和 VC 的计算和存储资源,因此不必考虑服务提供者的能量消耗.目前一些研究认为和车辆运行消耗的能量相比用于计算和传输的消耗的能量很少,甚至可以忽略不记.但是随着电动汽车的发展与普及,车辆的动力将更多由电池提供,在车辆执行大量高复杂度的计算任务会消耗大量能量,这对电动汽车的续航问题提出了巨大的挑战^[84].因此以节能为优化目标是未来车辆边缘计算任务卸载中的一大研究热点问题.

3.3 应用结果质量

超低响应时间对于目标检测和图像处理算法至关重要,这些算法广泛使用于增强现实和虚拟现实等应用中.使用高质量的原始数据进行计算会得到更精确、完美的计算结果,但是高质量数据会带来更高的传输和执行开销,从而会造成更高的响应时间和能源消耗,这与目标检测和图像处理算法对超低延迟的要求并不相符.然而在诸多车载应用中有一些能够容忍计算结果中一定程度的质量损失的应用,包括媒体处理、机器学习和数据挖掘等,它们通常不需要一个完美的结果,一个低质量或不太理想的结果就足够了.在文献[85]中,Li 等人提出了一种移动边缘计算中新的权衡方法—结果质量(Quality of Results, QoR),并提出在应用程序中放松 QoR 可以减轻计算工作量,并使移动边缘计算的响应时间和能源消耗大大减少. QoR 的衡量标准取决于应用领域和相应的算法,例如:目标检测算法, QoR 可以定义为正确检测目标的百分比.依据文献[85-86],通过放松对 QoR 的容忍度,可以在一定程度上降低服务延迟,但如果一味地降低 QoR,则计算结果会没有可信度.因此如何权衡 QoR 和服务延迟之间的关系是研究卸载决策和资源分配策略的一个重要问题.

本节列出了三种 VEC 环境下任务卸载的优化目标,表 3 详细列举了现有研究的优化目标,由表 3 分析可知,最小化卸载延迟是 VEC 任务卸载的主要优化目标,能耗和 QoR 也被一部分研究考虑在内.此外如系统效用、服务质量等优化目标,也主要靠延迟、能耗等来衡量.

表 3 车辆计算任务卸载技术分类

相关文献	计算资源	优化目标	卸载类型	任务依赖性	车辆移动性	决策方式
[31]	VC,RC	最小化卸载延迟和能量消耗	全部卸载	独立	移动	集中决策
[32]	VC,EC,RC	最小化卸载延迟和能量消耗	全部卸载	独立	移动	集中决策
[33]	VC	最小化移动设备卸载延迟和能量消耗(车辆形成 VC 提供服务)	部分卸载	依赖	移动	自主决策
[34]	VC,RC	优化车辆计算和能源利用的同时将任务延迟最小化	—	独立	停泊	自主决策
[35]	EC	最小化车辆任务延迟和租用 MEC 价格	部分卸载	独立	移动	自主决策
[36]	VC	最小化请求车辆卸载成本	—	独立	停泊	自主决策
[37-38]	VC	最小化卸载延迟	全部卸载	独立	移动	自主决策
[39]	VC	最小化卸载延迟并提高服务的可靠性	全部卸载	独立	移动	自主决策
[40]	VC	保证任务处理的时效性	全部卸载	独立	移动	集中决策
[41]	VC,EC	最小化卸载延迟	部分卸载	依赖	移动	自主决策
[42]	EC	考虑能耗和延迟的同时提高用户体验质量	全部卸载	独立	移动	集中决策
[43]	VC,EC	满足任务延迟约束的同时降低整体能耗	全部卸载	独立	移动和停泊	集中决策
[44]	VC,EC	最大化 VEC 网络的长期效用	部分卸载	独立	移动	集中决策
[45]	VC,EC,RC	降低处理时延和系统能量消耗	—	独立	移动	集中决策
[46]	VC	最小化卸载延迟	部分卸载	独立	移动	自主决策
[47]	VC	卸载延迟和服务质量	全部卸载	独立	移动	集中决策
[48]	VC	最小化系统所有车辆卸载延迟	部分卸载	独立	移动	集中决策
[49]	VC	最小化卸载延迟	—	依赖	移动	集中决策
[50]	VC,EC	最小化系统卸载延迟	部分卸载	独立	移动	集中决策
[51-52]	EC	最大化服务提供商收益,同时提高车辆的效用	部分卸载	独立	移动	集中决策
[53]	VC,EC	最大化服务提供商效益,并提高停靠车辆效益	全部卸载	独立	停泊	集中决策
[54]	EC	最小化卸载延迟同时 VEC 服务器负载均衡	部分卸载	独立	移动	自主决策
[55]	EC	同时优化车辆和 MEC 服务器的成本	部分卸载	独立	移动	集中决策
[56]	VC,EC	最小化任务超时概率	全部卸载	独立	移动	集中决策

4 VEC 卸载技术

由 2.1 节对 RC、EC 和 VC 的对比分析可知, EC 和 VC 可以减少传输成本并快速响应,与具有大计算容量的传统远程云服务器相比,EC 和 VC 的计算资源有限,特别是在交通流量密集的情况下,通常需要一定的时间来执行计算任务.这要求对车辆计算任务有合理的卸载决策和计算资源分配方案,在尽量满足卸载优化目标的同时使 EC 和 VC 保持良好的服务能力.本节首先总结了 VEC 计算任务卸载常用的模型与方法,然后按照计算任务卸载的集中式决策和分布式决策方式两种方式详细介绍了现有的 VEC 计算卸载决策和资源分配方案的研究状况.

VEC 计算任务卸载常用的模型与方法如下:

(1) 半马尔可夫决策过程

马尔可夫决策过程常常用于建模决策模型. 在一个动态系统中,系统状态是随机的,在每个 epoch 系统必须做出决定,且代价由决策决定.然而,在许多的决策问题中,决策阶段之间的时间不是恒定的,而是随机的.半马尔可夫决策过程(Semi-Markov Decision Process, SMDP)作为马尔可夫决策过程的扩展,用于对随机控制问题进行建模,不同于马尔可夫决策过程,半马尔可夫决策过程的每个状态都具

有一定的逗留时间,并且逗留时间不是常数,而是随机的^[87].因此半马尔可夫决策过程可以用于对卸载决策问题进行建模,文献[31-33, 88]都以半马尔可夫决策过程进行建模求解.

(2) 博弈论

博弈论是一种建立资源分配问题模型的经典工具,在移动边缘计算卸载中得到了广泛的研究与应用^[89-90].许多卸载决策和计算资源分配问题可以归结为分布式决策问题,博弈论是一个理想的选择,它关注的是玩家之间的战略互动,从而消除了对集中管理器的需求,在提高系统协作效率的同时,降低问题建模、算法设计与分析的复杂度.博弈论有两个主要分支:非合作博弈和合作博弈.“非合作博弈”指的是博弈各方相互冲突、独立行动;而“合作博弈”指的是各方虽然有竞争但也有合作,在各方合作中达到一个有效的平衡点.博弈论就是研究博弈行为中斗争各方是否存在最合理的行为方案,以及如何找到这个合理的行为方案的数学理论和方法,例如,文献[34-36]利用博弈论建模并对 VEC 计算卸载进行分布式决策.

(3) 强化学习

强化学习是另一种用于分布式决策的方法^[89].在强化学习中,自主的 Agent 通过在每一轮游戏中获得的奖励和惩罚来学习最佳行为,由于 Agent 不知道哪个操作是最好的,所以它们通过平衡未

知操作的探索 (Exploration) 和现有知识的利用 (Exploitation) 来学习。也就是说, Agent 以“试错法”进行学习, 通过与环境进行交互获得的奖赏指导行为, 目标是使 Agent 获得最大的奖赏, 一些常见的强化学习方法包括多摇臂赌博机 (Multi-Armed Bandit, MAB)、Q-learning 等, 如文献[37-40]利用 MAB 自主地计算卸载的决策。此外, 深度强化学习 (Deep Reinforcement Learning, DRL) 可用于解决资源管理问题^[91], 目前有一些使用深度强化学习对车辆计算任务进行卸载决策和资源分配, 如文献[41-45]。

(4) 启发式算法

启发式算法是一种比传统方法更快、更有效地解决问题的算法, 它通过牺牲最优性、准确性、精密度或完整性来提高速度。计算卸载和资源分配依据任务、通信和计算资源的基本信息, 根据优化目标进行卸载建模, 这通常可以将其认为是一个组合最优化问题。这些问题多数都是使用精心设计的启发式算法来解决的, 典型的设计流程是: ①对问题的模型提出巧妙的启发式; ②在实践中对启发式进行测试和调优, 以获得良好的性能。常见的启发式算法有遗传算法、蚁群算法、模拟退火、粒子群算法等。如文献[46]使用了蚁群算法、文献[47]使用了二进制粒子群优化算法、文献[48]使用蝙蝠算法对目标函数进行了优化求解。

(5) 契约理论

契约理论是经济学的分支之一, 主要研究在特定交易环境下来分析不同合同人之间的经济行为与结果, 往往需要通过假定条件在一定程度上简化交易属性, 建立模型来分析并得出理论观点。契约理论主要包括激励理论、不完全契约理论和新制度交易成本理论, 近年来在经济控制、资源配置和运营管理等领域也得到了广泛应用。契约理论在 VEC 计算卸载领域近年也取得了一定的进展, 例如文献[50-53]等, 研究者采用契约理论设计 VEC 计算卸载策略, 并取得了一定的成果。

4.1 VEC 集中式决策

车辆计算任务卸载集中式决策是在某个通信区域内, 由一个固定的或移动的节点 (称为管理节点) 来对该区域内的计算资源进行协调管理, 通过统一的调度对区域内的服务请求车辆进行卸载决策和计算资源的分配。集中式决策需要管理节点掌握管辖区域内的所有节点的信息, 包括计算资源的使用情况、通信条件以及计算任务的基础信息, 在这些信息

都已知的情况下才能进行集中式的决策和管理。计算任务卸载集中式决策追求对管理节点管辖范围内的所有用户利益的最大化, 如使管理区域内所有车辆的计算任务卸载的处理延迟最小、所有车辆的耗能最小。这也意味着, 集中式决策通常不考虑管理区域内其中一辆车是否达到了最优目标。可能出现这样的情况, 集中式卸载决策对车辆 A 而言是比较好的选择, 但对于车辆 B 而言, 该决策是差的。但由于集中式决策目标是让管理区域内的整体最优, 因此一些车辆较差并不会对整体造成较大影响。本小节将对 VEC 计算卸载集中式决策一些最新的研究进行总结。

启发式算法常常用于 VEC 中计算任务卸载集中式决策, 例如在参考文献[47]中提出了 Folo 概念, 将被蜂窝网络覆盖的地理区域进行划分, 每个区域都有一个固定节点 (区域头) 负责监控和协调管理该区域内所有的车辆。车辆离开或者进入该区域都需要向区域头汇报, 车辆需定期报告其移动方向、位置和可用资源容量, 以及将任务请求发送给区域头。区域头并不执行卸载的计算任务, 只负责任务卸载的决策工作, 任务根据调度结果卸载至其他的车辆节点执行。该文章将任务分配问题制定为联合优化问题, 同时优化平均服务时间和特定应用的服务质量, 且证明该问题为 NP-hard 问题, 并使用基于线性规划的优化 (LBO) 和二进制粒子群优化 (BPSO) 算法进行协同优化。实验使用 VeinsLTE 工具进行仿真, 结果表明, Folo 提供的计算任务卸载方案根据服务延迟和质量的实际需求进行调整, 与天真 (Naive) 算法和随机选择相比, Folo 将平均服务延迟缩短了 27%, 同时将质量损失降低了 56%。

在文献[48]中, Sun 等人将车辆分为用户车辆、MEC 车辆和 CH 车辆 (系统管理者), 首先 CH 车辆定期广播其状态信息, 其他车辆接收到信号后, 会对信道的质量进行估计, 并更新其信道状态信息。其次, 用户车辆将其信道状态信息和任务卸载请求发送到 CH 车辆, 并且 MEC 车辆将其信道状态信息和资源使用信息发送到 CH 车辆。CH 车辆根据收集到的信息, 做出最优任务决策, 并通过信道广播给其他车辆。最后用户车辆向 MEC 车辆发送任务数据, MEC 车辆执行任务后返回结果。Sun 等人提出了一种基于蝙蝠算法的多目标 VEC 任务调度算法, 通过观察 VEC 系统的状态, 对通信和计算资源的分配做出最优决策, 实验结果表明, 该算法能有效地缩短任务执行时间, 具有较高的可靠性。

在文献[49]中, Sun 等人提出充分利用 VC 的计算资源, 将 EC 上的计算任务通过基站卸载到 VC (任务可能由车辆产生, 也可能从其他的边缘设备产生). 文献[49]通过联合考虑资源的不稳定性、车辆计算能力的异构性以及计算任务的相互依赖性, 提出了 VC 中计算卸载的协同任务调度方案来最小化任务执行延迟, 并将该方案证明为 NP-hard 调度问题. 其车辆运动模型使用 Hyper-Erlang 分布来表示车辆在 VC 中的停留时间, 并将连续的时间离散化, 分成基础时间片, 执行时间、传输时间和排队时间都是基础时间片的整数倍. 最终通过改进的遗传算法(MGA)来最小化计算任务在不同车辆中执行总体响应时间. 实验结果表明对五个计算任务, 基于 MGA 的调度方案的时间效率比贪婪方案减少了 12.5%.

半马尔可夫过程同样可以用于集中式决策计算卸载建模, 例如在文献[31]中, Zheng 等人基于半马尔可夫过程, 为车辆计算任务制定资源分配方案. 文献[31]将计算资源分为 RC 和 VC 两层, VC 层将车辆的计算资源构成虚拟资源池(卸载时不考虑将任务卸载到具体哪一辆车计算, 只将虚拟资源池中一定数量的资源分配给一个任务). 系统中车辆的进入和离开服从相同的泊松分布, 且设定每辆车提供相同大小的虚拟资源. 其目标是最大化系统的长期预期回报, 回报由节约的能量、时间以及失败任务的惩罚构成, 然后利用系统的状态空间、动作空间、奖励模型和转移概率, 将优化问题表示为无限时域半马尔可夫决策过程. 文献[31]使用基于值迭代的方法对优化目标进行求解, 并将结果与基于贪婪的方法(GA)和模拟退火算法(SA)比较, 实验结果表明与 GA 方案相比, 基于值迭代的方法提升了近 7% 的收益, 此外, 基于 SMDP 的方案复杂度低于 SA 方案. Lin 等人^[32]在文献[31]的基础上将 VC 扩展为车辆和 RSU 形成的异构环境, 并将车辆的移动模型扩展为不同的泊松分布, 不同种类的车辆提供不同大小的虚拟资源. 因此, 与文献[31]相比, 文献[32]的异构环境增加了大量的状态数量, 问题求解更加复杂. 文献[32]也使用了值迭代的方法对优化目标进行求解, 该模型在长期预期回报(能量和时间消耗)方面表现良好.

此外, DRL 也可用于 VEC 集中式计算卸载决策, 例如在文献[42]中, Ning 等人将 DRL 方法与 VEC 相结合, 研究了车辆网络中任务调度和资源分配的优化问题, 在考虑能耗和执行延迟的同时, 最大

限度地提高用户的 QoE(Quality of Experience). 文章考虑一个 BS、 K 个 RSU 和 U 个移动车辆, 首先, 所有车辆广播它们的位置信息, 并更新它们可用的 RSU 列表. 然后, 车辆计算效用值, 构造相应的偏好列表, 并将卸载请求发送到 BS. 第三, BS 执行任务调度和资源分配, 并将任务分配发送给 RSU. 最后, 所有车辆将其卸载任务发送到相应的 RSU. 由于所构造问题的复杂性, 作者将其划分为两个子优化问题, 分别提出了一种双边匹配方案和 Double DQN (DDQN)方法进行求解, 实验证明了其有效性和优越性. 在文献[43]中, Ning 等人在车联网中构造了一个三层结构, 分别由 Cloudlet、RSU 和 VC(停泊和移动)构成, 车辆可以将卸载任务发送到附近的 RSU, RSU 不将所有任务上传到 RC 或基站, 而是将这些任务分配给 Cloudlet 或附近的车辆进行处理. 作者构造了一个在满足延迟约束的前提下使总功耗最小化的卸载问题, 并将其分为两部分: 流量重定向和卸载决策, 对第一部分提出了一种基于 Edmonds-Karp 的 RSU 流量均衡方案, 而对第二部分研究了一个基于 DRL 的模型. 基于上海出租车实际轨迹的数值结果表明, 与基线算法相比平均能耗可以减少约 60%. 研究^[44]提出了一种 VEC 网络体系结构, 在此结构中, VC 和 EC 由 VEC 管理者统一管理, 为 UEs(User Equipments)提供付费服务. 在第 t 时刻, UE i 生成一个计算任务, 它向 VEC 管理者发送一条消息, 管理者依据任务信息分别计算本地、VC 和 EC 的效用, 通过求解效用最大化的优化问题, VEC 管理者将得到使用哪种计算决策方案, 以及通信和计算资源分配策略. 考虑到交通的随机性和通信条件的不确定性, 研究中将提出的问题重新表述为半马尔可夫过程, 提出了基于 Q-learning 和 DRL 方法寻找计算卸载和资源分配的策略. 最后, 通过数值分析了该方案在 VEC 网络中的有效性.

VEC 现有大部分研究都假设 VEC 中的计算资源可以无条件共享, 然而在实际中, MEC 服务是由运营商提供的, 车辆使用 MEC 服务需要付费; 同时车辆用户希望自己的利益最大化, 因此车辆无条件地共享空闲的计算资源是不现实的. 为了解决这一挑战, 一些研究工作利用契约理论设置激励方案, 以满足服务提供商或车辆的收益需求. 例如, 在文献[50]中, Zhou 等人通过契约-匹配集成的方法解决了现有 VEC 计算资源和任务分配问题中缺乏有效的激励机制使车辆充当零节点以及缺乏低复杂度任务分配机制所带来的挑战, 利用附近车辆的未充

分利用的资源为 UEs 提供更高的服务质量。在第一阶段,提出了一种基于契约的机制来激励车辆资源共享,契约规定了每辆车所需的计算资源和奖赏之间的关系。第二阶段,为了降低复杂度,提出了一种基于定价的稳定匹配算法(Pricing-Based Stable Matching Algorithm)来解决任务分配问题,该算法通过迭代执行建议和价格上涨过程,得到基于动态更新的偏好列表的稳定匹配。数值结果表明,该方案可以显著地改善系统的性能,并拥有更低的复杂度。在文献[51]中,Zhang 等人采用契约理论的方法为 VEC 服务供应商设计最优的卸载策略,使供应商的收益最大化,同时提高车辆的效用。为了进一步提高 VEC 服务器计算资源的利用率,Zhang 等人将任务优先级和额外资源结合到卸载方案的设计中,提出了一种高效的 VEC 服务器选择和计算资源分配算法。结果表明,该方案极大地提高了 VEC 提供商的收入,同时也提高了云计算资源的利用率。为解决边缘服务器超负荷的问题,Wu 等人在文献[53]中提出一个基于停车协同的边缘计算任务卸载框架,利用契约理论激励停靠车辆执行卸载的计算任务。服务提供商通过网络边缘服务器执行计算任务,当计算任务数量过多时,服务提供商需要把超负荷的计算任务卸载给附近停靠车辆。基于真实数据集的实验仿真结果证明该方案不仅能够满足停靠车辆的效益,而且能够最大化服务提供商的效益。

4.2 VEC 分布式决策

集中式决策的决策者需要大量的通信开销来获得各类异构计算资源的全局信息,尤其是在超密集车辆网络中是不现实的,并且由于车辆的随机性和动态性,无限通信信道质量、计算资源的非确定性以及任务的随机到达都对计算卸载的集中式决策方案提出了挑战。与集中式决策由一个统一资源管理中心进行协调决策相比,分布式决策通常由车辆自行地对卸载决策和资源分配方案做决定,而不需要资源管理者以很大的复杂度去做决策。因此,使用分布式和自主的决策方法是至关重要的,在这种方法中,各个移动设备和移动边缘服务器为系统做出决策,使整个系统在高效和稳定的运转。本小节将对车辆计算任务卸载分布式决策一些最新的研究进展和方法做出总结。

博弈论是一种解决分布式决策问题的常用工具,例如在研究^[34]中,Klaimi 等人提出了一种 VFC 的计算模式,使位于网络边缘的车辆资源能够为资源匮乏的移动应用程序提供高质量服务。同时,由于

VFC 资源有限,当 VFC 中缺少资源时,一些应用程序任务必须重定向到云执行。文献[34]将移动用户产生的任务分为高(HP)、低(LP)优先级两类,两类任务都需要在特定的时限内完成,其到达率均服从泊松分布。从车辆的角度来看,该文章考虑的是将车辆停放在一个最大容量为 n 辆的停车场,这些车辆都是电池驱动的电动车辆,每个车辆都是一个计算资源服务器。车辆进入 VFC 系统服从泊松分布,而离开率则服从指数分布。作者使用势博弈(Potential Game)来建模,将车辆看作博弈者,目标是优化计算和能源利用的同时,将延迟最小化。作者证明了纳什均衡(Nash equilibrium),并提出了一种动态资源分配的调度算法进行求解。实验结果证明了该方法的有效性,并表明所提出的贡献在保持车辆最低能耗水平的前提下,取得了较好的性能。

在文献[35]中,Zhang 等人为了在交通流量密集的道路上为车辆提供可靠和高效的服务,提出了一种分层的车辆边缘计算框架。该框架引入一个邻域备份计算服务器(BCS)来弥补 VEC 服务器计算资源的不足,BCS 放置在路边,当 VEC 服务器资源不足时可以付费租用 BCS 的计算服务。车辆的目标是在满足任务时延约束的前提下,最小化卸载延迟和卸载的费用,而 VEC 服务器的目标是通过向车辆出售计算资源来赚取更多利润,对此 Zhang 等人提出了一种基于 Stackelberg 博弈的车辆计算卸载过程。由于每辆车都可以在指定的任务延迟约束下选择任意一台 VEC 服务器作为其卸载目标,因此各服务器之间进行非合作的价格确定博弈,以确定各自的最优资源价格。因此,在卸载过程中服务器之间存在竞争。此外,当 VEC 服务器自身资源不能满足车辆的需求时,可以从 BCS 购买计算资源。由于可用的计算资源总量影响着每台 VEC 服务器的收益,从 BCS 购买的资源数量也是每台 VEC 服务器在卸载竞争中的策略。同时由于 VEC 服务器计算资源的限制,车辆在选择卸载服务器时可能会产生竞争。因此,使用 Stackelberg 博弈方法建立多级卸载系统模型,在博弈中,VEC 服务器是领导者,车辆是追随者。文献[35]证明了博弈的纳什均衡,并使用一种高效的分布式算法,在满足车辆任务时延约束的前提下,使服务提供商的收益最大化,从而达到最优的计算卸载。实验数值结果表明,该方案大大提高了卸载服务提供商的效用。

在参考文献[36]中,Huang 等人提出了一种停泊车辆边缘计算(Parked Vehicle Edge Computing,

PVEC)的计算范式,利用闲置的 PV 的计算资源,与 VEC 服务器协同执行计算任务.该文献介绍了一种可行的系统架构,包括请求车辆、租用 VEC 服务器的服务提供商和 PV,并设计了一种提供基本请求和响应操作的交互式协议,保证用户的安全和隐私.请求车辆首先将任务请求通过 RSU 发送到 VEC 服务器,VEC 服务器将任务划分为多个子任务后选择一些 PV 执行任务,子任务执行完成后由 VEC 聚集并返回到请求车辆.文献中利用 Stackelberg 博弈方法,建立并求解资源调度优化问题,其中服务提供者是领导者,而 PV 作为追随者来响应它,并提出了一种迭代算法求解 PVs 之间工作量分配的 Stackelberg 均衡分析方法,目标是使请求车辆的总成本最小化.通过对实际数据集的大量仿真表明,PVEC 在提高网络容量和降低计算卸载服务成本方面具有较好的性能.

除博弈论外,MAB 也被用于 VEC 计算卸载中的自主决策.例如,Sun 等人在文献[37]中提出了一种车辆边缘计算系统中基于学习的自适应任务卸载方法.该文章将需要任务卸载的车辆称为 TaVs (Task Vehicles),将能够帮助执行任务的车辆称为 SeVs (Service Vehicles).由于车载任务卸载环境是动态的、不确定的,网络拓扑结构、无线信道状态和计算工作量变化很快,这些因素很难建模或预测,因此 TaV 事先不知道哪个 SeV 在延迟性能方面表现得最好.因此 Sun 等人提出了一种卸载时学习的方案,即 TaV 能够在卸载任务时学习延迟性能,采用 MAB 框架设计了一种基于学习自适应任务卸载算法 (Adaptive Learning-Based Task Offloading, ALTO),使 TaVs 的平均卸载延迟最小化. ALTO 算法以分布式方式工作,使 TaV 能够学习相邻 SeVs 的延迟性能,而不需要在车辆之间交换信道状态、计算工作量等精确的状态信息. ALTO 算法采用了基于上置信界 (Upper Confidence Bound, UCB) 的算法,同时加入了两种自适应性:输入感知和事件感知,分别通过任务的大小和 SeVs 的出现时间调整探索权重.实验分别在综合场景下 (MATLAB) 和现实公路场景 (Veins) 进行,结果表明,该算法具有较低的时延性能,与现有的基于上置信界的学习算法相比,平均时延降低了 30%.

在研究^[39]中,在文献[37-38]工作的基础上,Sun 等人又提出高度动态的车辆网络中,对延迟性能和服务可靠性有重要的影响.因此提出了一种基于学习的组合多摇臂赌博机 (Combinatorial Multi-Armed

Bandit, CMAB) 的任务复制算法 (Learning-Based Task Replication Algorithm, LTRA),以最小化平均卸载延迟.文献[39]采用任务复制的方式,将同一个任务复制后卸载到多个 SeVs,以保证服务的可靠性.基于 CMAB 理论,设计了 LTRA 来解决 TaVs 相邻 SeVs 计算量的问题.通过对现实城市道路交通的仿真,并将 LTRA 与文献[38]单卸载算法进行了比较,结果表明,当有充足的 SeVs 时,以 0.6 s 为时限的任务,完成率从 80% 提高到了 98%,通过任务复制可以显著降低车辆的平均卸载延迟和提高服务可靠性.

其他方法也用于 VEC 分布式决策,例如,Wang 等人在文献[33]中利用城市内公交车总是遵循相对固定的路线、位置比其他车辆更容易预测的特点,提出了一种为 SMDs (Smart Mobile Devices) 服务的基于 SMDP 的 cloudlet 协作策略.该策略考虑到任务之间的顺序逻辑关系,以及实际车辆网络中无线信道的移动性和影响,并基于 SMD 的状态确定本地执行或是卸载到由公交车构成的 cloudlet,以实现最小能耗和延迟.作者将问题转化为状态转移图上的时滞约束最短路径问题,并通过使用值迭代算法求解 SMDP 模型,实验结果表明,提出的基于 SMDP 的协作策略可以降低 SMD 成本 (能源消耗和应用延迟) 并提高卸载率. Qi 等人在文献[41]中采用了 DLR 对 VEC 的任务卸载进行决策.文章考虑使用多种类型的边缘计算节点为智能车辆服务提供异构资源,将卸载决策制定为具有单个或多个目标函数和约束的资源调度问题,并制定了启发式算法. Qi 等人将多个依赖任务的卸载决策制定为长期规划问题,并使用深度强化学习以获得最优解.此外,该框架支持在边缘计算节点进行预训练和车辆任务执行时进行持续在线学习,从而可以适应环境变化,并可以学习有远见的策略.仿真结果表明,基于深度强化学习的卸载决策收敛速度快,适应不同条件,优于贪心卸载决策算法.在文献[46]中,Feng 等人提出了一种用于车辆边缘计算的自主车辆边缘 (Autonomous Vehicular Edge Computing, AVE) 框架,并通过该框架对车辆无依赖计算任务的执行延迟进行优化.车辆都安装有由作业队列模块、资源管理模块和调度模块构成的管理器模块,由管理器决定是否进行任务卸载或者接收来自其他车辆的卸载请求.车辆之间通过广播 beacon 的方式来相互交换信息,由车辆依据自己与其两跳范围内的邻近车辆的速度、计算资源的空闲状态来自主决定任务卸载和资源分配

的策略. 文献[46]介绍了 AVE 框架任务卸载的工作流程,提出了有效的工作缓存,将最终的优化目标确立为车辆的任务处理延迟,将其证明为 NP-hard 问题,并使用蚁群算法对问题进行优化. 作者使用 Veins 对实验进行仿真,结果表明,在典型的城市和高速公路两种场景中,该方法都优于现有的其他方案.

从以上介绍可知,近年来研究者在 VEC 集中式决策和分布式决策都投入了大量精力,也取得了一定的进展. 表 4 给出 VEC 集中式决策和分布式决策方式的技术对比,需要注意的是,两种决策方式各有特点,在不同的场景下可采用不同的决策方式,以更少的代价获得更高的收益.

表 4 VEC 集中式和分布式决策方式技术对比		
决策方式	优点	缺点
集中式决策	了解全局信息,有助于做出对全局更有利的决策,使整体效益最大化	掌握全局信息需要额外通信开销和延迟;对个体而言决策不一定最优
分布式决策	自主决策,减少通信开销,决策更快,减轻管理者计算压力	只考虑个体最优,全局不一定最优;占用自身的计算资源用于做决策

5 VEC 实验环境

MATLAB 是一套功能强大的工程计算软件,研究人员常使用 MATLAB 对所研究的问题进行建模仿真,通过使用 MATLAB 提供的工具箱,可以高效地解决复杂的问题,用强大的图形功能对数值结果进行显示. MATLAB 同样用于车联网下计算任务卸载的仿真实验,例如文献[37-39]等. 但是由于车辆的通信通常受到多种因素的影响,如车辆的高移动性、建筑物干扰, MATLAB 通常只能忽略这些因素,只计算理想情况下的通信延迟. 为了使车联网场景下车辆的移动及其任务卸载仿真更加接近实际,研究人员们使用了如下几个实验工具: SUMO、Veins、VeinsLTE,本节将对这几个工具的特点及使用作简单的介绍.

5.1 SUMO

SUMO^[92-94] (Simulation of Urban MObility) 是一个开源的、微观的、多模态的、空间连续的、时间离散的交通流仿真平台,用于模拟大型道路网络. 在过去的十年里, SUMO 已经发展成为一个全功能的交通建模工具,包括能够读取不同源格式的道路网络、使用不同的输入源生成车辆路由、高性能地模拟单路口以及整个城市交通流量,并提供了 TraCI 接口^[95]

(Traffic Control Interface),通过访问正在运行的 SUMO,在线操作和获取 SUMO 中的行为,例如设置车辆的速度、获取车辆的位置等. 使用 SUMO 仿真平台对交通流量进行仿真的步骤如下: (1) 生成静态的道路网络. SUMO 提供了使用 Open Street Map 导入道路网络的功能,使用真实的地图数据构建道路网络,保证仿真的真实性,使仿真结果更具说服力; (2) 对车辆的路由进行模拟. 每辆车都有其出发时间、行驶速度、行驶路径等基本信息,这些信息可以根据实际的交通数据生成,如在文献[96]中,利用真实交通信息生成了卢森堡市 24 h 车辆路由信息; (3) 仿真. 将静态道路网络和车辆路由结合,使车辆在道路网络上运行,在运行过程中即可使用 TraCI 接口获取仿真过程的具体数据. 图 5 为 SUMO 生成的埃朗根(德国城市)道路网络,其中矩形块为建筑物,线条为道路.

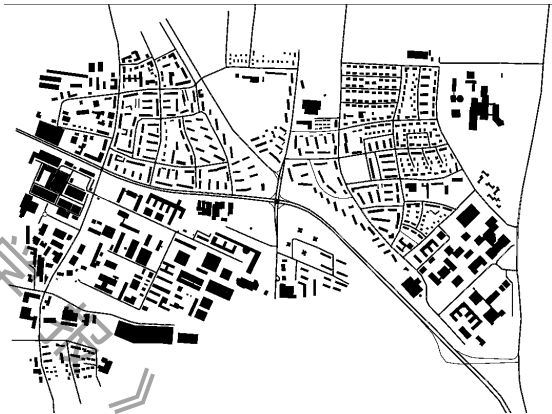


图 5 SUMO 道路网络

5.2 Veins

SUMO 为实验仿真提供了真实的道路网络和车辆移动模型, Veins^[97-98] (Vehicles in Network Simulation) 在网络模拟器 OMNeT++^[99] 和 SUMO 的基础上,开发了一套全面的车辆网络模型,在尽可能真实的前提下保证了仿真的速度. Veins 的道路交通仿真由 SUMO 进行的,网络仿真由 OMNeT++ 和物理层建模工具包 MiXiM 共同完成,是一套开源车辆网络仿真框架,支持 IEEE 802.11p 和 IEEE 1609.4 DSRC/WAVE. Veins 对车辆通信的无线电干扰以及静止和移动障碍物的遮挡建立精确的模型,更好地提供真实场景下车辆通信的仿真. 同时 Veins 提供了 TraCI 接口,方便研究人员编写特定的应用,依据网络数据包在线重新配置车辆和重新制定车辆路由,以便于确定车辆任务卸载和资源分配方案.

5.3 VeinsLTE

VeinsLTE^[100] 是 Veins 的延伸,是一个开源的异构车辆网络通信模拟器.由于 Veins 目前不支持异构网络,特别是 LTE,因此对于异构车辆网络下的通信 Veins 还不足以提供足够的支持. VeinsLTE 由 Veins、INET 和 SimuLTE 构成,其中 Veins 继续提供 DSRC 车辆网络仿真模型,而 INET 框架提供了通用的 IP 连接支持和 IEEE 802.11 Wi-Fi 的链路模型,SimuLTE 则提供了使用 LTE 模拟蜂窝网络的方法.因此 VeinsLTE 支持异构车辆网络的仿真,例如:IEEE 802.11p DSRC、Wi-Fi 和 LTE.

表 5 对三种实验工具的优缺点进行了比较.由于 Veins 或 VeinsLTE 都基于 SUMO,实验时可以先使用 SUMO 生成实验所需的道路网络和交通流量,然后再选择 Veins 或 VeinsLTE 进行网络通信. VeinsLTE 比 Veins 支持更多的通信协议,可以优先考虑选择 VeinsLTE 作为 VEC 计算卸载的主要实验工具.但是上述工具目前均不支持 5G 通信,因此暂时不能基于 5G 网络进行车辆网络仿真.使用上述仿真工具使得车辆计算任务卸载的仿真更具说服力,尤其是 Veins 和 VeinsLTE,它们将 SUMO 与 OMNeT++ 完美连接,可以真实地模拟车辆任务卸载,为研究者提供可靠的实验工具.例如在文献[37,46]中,作者利用 Veins 工具对实验进行了仿真,文献[47]则利用 VeinsLTE 在异构车辆网络环境下进行了实验.

表 5 VEC 常见实验工具比较

实验工具	优缺点
SUMO	SUMO 可以进行交通建模,生成城市道路网络、模拟城市交通流量.但是 SUMO 没有提供网络通信工具,单纯使用 SUMO 无法模拟车辆 V2V 和 V2I 通信.
Veins	Veins 在 SUMO 基础上增加了车辆网络仿真框架,并模拟了真实场景下的各种干扰.但 Veins 仅支持 IEEE 802.11p 和 IEEE 1609.4 DSRC/WAVE,不支持由 LTE 等构成的异构网络.
VeinsLTE	VeinsLTE 基于 Veins 增加了对 Wi-Fi 和 LTE 网络的支持,兼容多种类型的异构网络.

6 挑战与展望

VEC 计算任务卸载充分利用边缘设备低延迟的特点,为计算资源受限的车辆获得更好的服务质量提供了有效的途径.通过上述分析可知,我们看到 VEC 计算任务卸载已经进行了较为深入的研究,并取得了一定的研究成果.但无论集中式决策还是分

布式决策,都还不能完全满足人们的应用需求,这意味着 VEC 计算任务卸载在当前是一件极具挑战性的工作.经过前几节的介绍发现,现存的 VEC 卸载技术仍存在诸多的问题需要研究和解决.

(1) 卸载决策算法复杂度

卸载是满足车辆应用程序对响应时间需求的一个有吸引力的解决方案,在卸载过程中除正常的计算和通信时间开销外,决策算法的时间复杂度会影响应用程序的整体响应时间.时间复杂度是衡量决策算法好坏的一个重要指标,时间复杂度越高,决策就需要更多的时间开销,这不适用于实时性要求高的计算任务卸载.特别是随着城市内智能车辆增多以及日益复杂的车辆应用程序和无线网络体系结构,时间复杂度高的算法都很难适应 VEC 的计算卸载.无论是集中式决策还是分布式决策,理想的情况是决策者接收到任务请求立刻进行卸载决策和资源分配,决策算法所消耗的时间越少,计算卸载的实时性越高.而现有的 VEC 卸载方案其主要目的是为了得到更好的卸载方案,缺乏对实际场景下决策算法高实时性要求的考虑.目前有些新兴的研究工作^[101-102]基于深度强化学习来解决计算卸载高实时性的问题,深度强化学习具有较强的处理复杂问题的能力,能够有效地从经验中学习,因此能够在 VEC 各种复杂场合下获得更好的卸载策略,这为复杂模型的计算卸载决策提供了一个良好的思路,基于深度强化学习的 VEC 计算卸载将是下一个研究的热点问题.

(2) 安全与隐私保护

边缘节点的移动性和车辆网络拓扑的动态性给车辆网络的安全性和保密性带来了新的挑战.在 VEC 中,包括车辆、RSU、MEC 服务器等在内的计算节点以分布式的方式部署,单点保护能力低,可能遭到恶意用户的网络攻击,发送恶意消息和非法命令,破坏网络服务的可靠性,原本适用于云计算中的安全机制可能对 VEC 不再适用.在 VEC 中的攻击可以分为被动攻击和主动攻击,被动攻击不会破坏 VEC 系统的功能,但会攻击系统的保密性,如窃听个人隐私信息等,造成隐私数据的泄露.主动攻击是指故意破坏 VEC 系统运行,例如篡改计算卸载的决策信息、分布式拒绝服务攻击等,情节严重时可能会影响车辆的控制系统,威胁车辆行驶安全.被动攻击由于只监听不修改信息是很难被检测到的,而主动攻击对系统影响较大,很容易检测到.预防被动攻击可以采用加密技术来保护信息的安全

性,然而大多数安全机制只能有效地预防被动攻击,没有万无一失的安全解决方案来预防主动攻击,一旦某个边缘节点遭到攻击,在 VEC 控制系统内发生破坏性的行为,就需要更复杂的安全机制来阻止此类攻击。目前已有部分对于 VEC 计算卸载中安全和隐私问题的研究,例如, Mtibaa 等人在文献[103]中设计了 HoneyBots 来检测和跟踪执行计算卸载的 D2D(Device-To-Device)网络中的恶意通信活动。但总的来说 VEC 计算卸载的安全研究相对较少,制定更加合理的安全策略以保证 VEC 系统安装仍然是下一步研究的一大热点。

(3) 车辆的移动性

车辆的高移动性是 VEC 计算任务卸载的一大难题。首先,车辆的高速移动导致通信节点移动速度快,网络拓扑变化快,通信链路持续时间非常短,通信链路频繁地连接和断开,容易导致 V2I 和 V2V 通信和计算任务传输失败。其次,车辆移动导致周围复杂环境的快速变化,通信链路会受到不同程度的干扰,导致数据传输速率和传输功耗的变化、通信质量的下降,进而造成较大的传输延迟。此外,当车辆正在执行卸载任务或正在等待卸载结果时离开通信范围,会导致计算任务卸载失败。针对车辆高移动性造成的通信质量问题暂时无法完全避免,但可以尽量减少因此带来的问题。例如,车辆在行驶时相对于固定基础设施速度差比较大,对向行驶的车辆之间的速度差则要更大,而同向行驶的车辆速度差则会比较小,因此形成的通信链路也会相对稳定,在文献[46]中,作者将同向行驶的车辆作为任务卸载的候选集,对向行驶的车辆由于速度差较大,即使在通信范围内也不会卸载任务。而对于车辆离开通信范围造成的计算任务失败,现有研究中提出了一些解决方案,例如在文献[31-32]中,作者在优化目标中对失败任务加入惩罚项,来尽量避免任务因车辆离开而失败,但是仍然不能完全避免任务的失败。在文献[39]中,Sun 等人通过任务复制卸载的方式,将同一个任务同时卸载至多个车辆,来增强任务卸载的可靠性,但这需要数倍的通信开销,极大地浪费了网络资源。在文献[47]中,由一个固定的基站来统一协调管理其通信范围内的车辆,当有携带有计算任务的车辆离开此区域时,重新进行任务调度,但重新调度会造成额外的时间开销。基于上述研究,由于车辆离开通信范围而造成的卸载失败仍然是 VEC 计算卸载中的一个待解决的问题。

7 总 结

车辆边缘计算作为移动边缘计算在车联网场景中的一个重要应用而得到了广泛的研究,VEC 系统集成各类边缘计算资源,为计算资源受限车辆和其他用户提供低延迟、高带宽、高可靠的计算服务。本文对 VEC 下的任务卸载背景和意义进行了详细的介绍,针对于现有的研究,对 VEC 的系统模型分别从计算模型、任务模型和通信模型进行了详细的介绍。然后对于 VEC 中常见的优化目标如最小化卸载延迟、最小化能量消耗和应用结果质量进行了归纳。随后分别从任务卸载的集中式决策和分布式决策两个角度对现有的研究做了详细的讨论,并从计算资源、优化目标、卸载类型、任务依赖性、车辆的移动性和决策方式等方面对相关研究进行了归类。此外还介绍了几种车辆边缘计算卸载的仿真工具,便于研究者使用更加真实的仿真环境进行研究。由于车联网下的计算任务卸载还处于起步阶段,卸载决策算法的复杂度、计算任务的安全和隐私保护以及车辆移动性带来的卸载难题还有待探索和研究。

随着 5G 通信的发展和普及,5G 更高的传输速率、更佳的移动性能、毫秒级延迟以及超高密度连接等特性,将为车辆计算密集和延迟敏感的计算任务提供更高速度、更可靠的通信支持,使 VEC 环境下任务卸载更加令人期待。

参 考 文 献

- [1] Qiu H, Ahmad F, Govindan R, et al. Augmented vehicular reality: Enabling extended vision for future vehicles//Proceedings of the International Workshop on Mobile Computing Systems and Applications. New York, USA, 2017: 67-72
- [2] Zhang Ke, Mao Yu-Ming, Leng Su-Peng, et al. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. IEEE Vehicular Technology Magazine, 2017, 12(2): 36-44
- [3] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing. Communications of the ACM, 2010, 53(4): 50-58
- [4] Shi Wei-Song, Cao Jie, Zhang Quan, et al. Edge computing: Vision and challenges. IEEE Internet of Things Journal, 2016, 3(5): 637-646
- [5] Khan W, Ahmed E, Hakak S, et al. Edge computing: A survey. Future Generation Computer Systems, 2019, 97: 219-235
- [6] Abbas N, Zhang Yan, Taherkordi A, Skeie T. Mobile edge computing: A survey. IEEE Internet of Things Journal, 2018, 5(1): 450-465

- [7] Mao Yu-Yi, You Chang-Sheng, Zhang Jun, et al. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys Tutorials*, 2017, 19(4): 2322-2358
- [8] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1628-1656
- [9] Greenberg A, Hamilton J, Maltz D A, Patel P. The cost of a cloud: Research problems in data center networks. *ACM SIGCOMM Computer Communication Review*, 2009, 39(1): 68-73
- [10] Satyanarayanan M, Bahl P, Caceres P, Davies N. The case for VM-based Cloudlets in mobile computing. *IEEE Pervasive Computing*, 2009, 8(4): 14-23
- [11] Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the Internet of Things//*Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing*. New York, USA, 2012: 13-16
- [12] Al-Doghman F, Chaczko Z, Ajayan A R, Klempous R. A review on fog computing technology//*Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics*. Budapest, Hungary, 2016: 1524-1530
- [13] Hou Xue-Shi, Li Yong, Chen Min, et al. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 2016, 65(8): 3860-3873
- [14] Abdelhamid S, Hassanein H, Takahara G. Vehicle as a resource (VaaR). *IEEE Network*, 2015, 29(1): 12-17
- [15] Jang I, Choo S, Kim M, et al. The software-defined vehicular cloud: A new level of sharing the road. *IEEE Vehicular Technology Magazine*, 2017, 12(2): 78-88
- [16] Xie Ren-Chao, Lian Xiao-Fei, Jia Qing-Min, et al. Survey on computation offloading in mobile edge computing. *Journal on Communications*, 2018, 39(11): 142-159 (in Chinese)
(谢人超, 廉晓飞, 贾庆民等. 移动边缘计算卸载技术综述. *通信学报*, 2018, 39(11): 142-159)
- [17] Li Qiu-Ping, Zhao Jun-Hui, Gong Yi. Computation offloading and resource management scheme in mobile edge computing. *Telecommunications Science*, 2019, 35(3): 36-46 (in Chinese)
(李邱平, 赵军辉, 贡毅. 移动边缘计算中的计算卸载和资源管理方案. *电信科学*, 2019, 35(3): 36-46)
- [18] Li Zuo-Zhao, Liu Jin-Xu. Application of mobile edge computing in Internet of Vehicles. *Modern Science & Technology of Telecommunications*, 2017, 47(3): 37-41 (in Chinese)
(李佐昭, 刘金旭. 移动边缘计算在车联网中的应用. *现代电信科技*, 2017, 47(3): 37-41)
- [19] Gu Xiao-Hui, Zhang Guo-An. Survey of mobile edge computing applications in vehicular networks. *Application Research of Computers*, 2019, 37(6): 1-9 (in Chinese)
(谷晓会, 章国安. 移动边缘计算在车联网中的应用综述. *计算机应用研究*, 2019, 37(6): 1-9)
- [20] Raza S, Wang Shang-Guang, Ahmed M, Anwar M R. A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wireless Communications and Mobile Computing*, 2019, 2019: 1-19
- [21] Lin Fu-Hong, Zhou Yu-Tong, Pau G, Collotta M. Optimization-oriented resource allocation management for vehicular fog computing. *IEEE Access*, 2018, 6: 69294-69303
- [22] Shi Wei-Song, Zhang Xing-Zhou, Wang Yi-Fan, Zhang Qing-Yang. Edge computing: State-of-the-art and future directions. *Journal of Computer Research and Development*, 2019, 56(1): 69-89 (in Chinese)
(施巍松, 张星洲, 王一帆, 张庆阳. 边缘计算: 现状与展望. *计算机研究与发展*, 2019, 56(1): 69-89)
- [23] Zhou Yue-Zhi, Zhang Di. Near-end cloud computing: Opportunities and challenges in the post-cloud computing era. *Chinese Journal of Computers*, 2019, 42(4): 677-700 (in Chinese)
(周悦芝, 张迪. 近端云计算: 后云计算时代的机遇与挑战. *计算机学报*, 2019, 42(4): 677-700)
- [24] Gu Lin, Zeng De-Ze, Guo Song. Vehicular cloud computing: A survey//*Proceedings of the 2013 IEEE Globecom Workshops*. Atlanta, USA, 2013: 403-407
- [25] Gerla M. Vehicular cloud computing//*Proceedings of the 2012 the 11th Annual Mediterranean Ad Hoc Networking Workshop*. Ayia Napa, Cyprus, 2012: 152-155
- [26] Whaiduzzaman M, Sookhak M, Gani A, Buyya R. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 2014, 40: 325-344
- [27] Boukerche A, Grande R. Vehicular cloud computing: Architectures, applications, and mobility. *Computer Networks*, 2018, 135: 171-189
- [28] Xiao Yu, Zhu Chao. Vehicular fog computing: Vision and challenges//*Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Kona, USA, 2017: 6-9
- [29] Huang Chen, Lu Rong-Xing, Choo K. Vehicular fog computing: Architecture, use case, and security and forensic challenges. *IEEE Communications Magazine*, 2017, 55(11): 105-111
- [30] Menon V. Moving from vehicular cloud computing to vehicular fog computing: Issues and challenges. *International Journal on Computer Science and Engineering*, 2017, 9(2): 14-18
- [31] Zheng Kan, Meng Han-Lin, Chatzimisios P, et al. An SMDP-based resource allocation in vehicular cloud computing systems. *IEEE Transactions on Industrial Electronics*, 2015, 62(12): 7920-7928
- [32] Lin Chun-Cheng, Deng Der-Jiunn, Yao Chia-Chi. Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units. *IEEE Internet of Things Journal*, 2018, 5(5): 3692-3700
- [33] Wang Zhe, Zhong Zhang-Dui, Ni Min-Ming. A semi-Markov decision process-based computation offloading strategy in vehicular networks//*Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Montreal, Canada, 2017: 1-6
- [34] Klaimi J, Sidi-Mohammed S, Mohamed-Ayoub M. Theoretical game approach for mobile users resource management in a vehicular fog computing environment//*Proceedings of the 2018 14th International Wireless Communications & Mobile*

- Computing Conference (IWCMC). Limassol, Cyprus, 2018; 452-457
- [35] Zhang Ke, Mao Yu-Ming, Leng Su-Peng, et al. Optimal delay constrained offloading for vehicular edge computing networks//Proceedings of the 2017 IEEE International Conference on Communications (ICC). Paris, France, 2017; 1-6
- [36] Huang Xu-Min, Yu Rong, Liu Jian-Qi, Shu Lei. Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications. IEEE Access, 2018, 6; 66649-66663
- [37] Sun Yu-Xuan, Guo Xue-Ying, Song Jin-Hui, et al. Adaptive learning-based task offloading for vehicular edge computing systems. IEEE Transactions on Vehicular Technology, 2019, 68(4); 3061-3074
- [38] Sun Yu-Xuan, Guo Xue-Ying, Zhou Sheng, et al. Learning-based task offloading for vehicular cloud computing systems//Proceedings of the 2018 IEEE International Conference on Communications. Kansas City, USA, 2018; 1-7
- [39] Sun Yu-Xuan, Song Jin-Hui, Zhou Sheng, et al. Task replication for vehicular edge computing: A combinatorial multi-armed bandit based approach//Proceedings of the 2018 IEEE Global Communications Conference. Abu Dhabi, United Arab Emirates, 2018; 1-7
- [40] Chen Li-Xing, Xu Jie. Task replication for vehicular cloud: Contextual combinatorial bandit with delayed feedback//Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications. Paris, France, 2019; 748-756
- [41] Qi Qi, Wang Jing-Yu, Ma Zhan-Yu, et al. Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach. IEEE Transactions on Vehicular Technology, 2019, 68(5); 4192-4203
- [42] Ning Zhao-Long, Dong Pei-Ran, Wang Xiao-Jie, et al. Deep reinforcement learning for vehicular edge computing: An intelligent offloading system. ACM Transactions on Intelligent Systems and Technology, 2019, 10(6); 1-24
- [43] Ning Zhao-Long, Dong Pei-Ran, Wang Xiao-Jie, et al. Deep reinforcement learning for intelligent Internet of Vehicles: An energy-efficient computational offloading scheme. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(4); 1060-1072
- [44] Liu Yi, Yu Hui-Min, Xie Sheng-Li, Zhang Yan. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. IEEE Transactions on Vehicular Technology, 2019, 68(11); 11157-11168
- [45] Qiao Guan-Hua, Leng Su-Peng, Liu Hao, et al. Task collaborative offloading scheme in vehicle multi-access edge computing network. Chinese Journal on Internet of Things, 2019, 3(1); 51-59(in Chinese)
(乔冠华, 冷甦鹏, 刘浩等. 面向车辆多址接入边缘计算网络的任务协同计算迁移策略. 物联网学报, 2019, 3(1); 51-59)
- [46] Feng Jing-Yun, Liu Zhi, Wu C, Ji Yu-Sheng. AVE: Autonomous vehicular edge computing framework with ACO-based scheduling. IEEE Transactions on Vehicular Technology, 2017, 66(12); 10660-10675
- [47] Chao Zhu, Jin Tao, Paster G, et al. Folo: Latency and quality optimized task allocation in vehicular fog computing. IEEE Internet of Things Journal, 2019, 6(3); 4150-4161
- [48] Sun Jia-Nan, Gu Qing, Zheng Tao, et al. Joint communication and computing resource allocation in vehicular edge computing. International Journal of Distributed Sensor Networks, 2019, 15(3); 1550147719837859; 1-13
- [49] Sun Fei, Hou Fen, Cheng Nan, et al. Cooperative task scheduling for computation offloading in vehicular cloud. IEEE Transactions on Vehicular Technology, 2018, 67(11); 11049-11061
- [50] Zhou Zhen-Yu, Liu Peng-Ju, Feng Jun-Hao, et al. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. IEEE Transactions on Vehicular Technology, 2019, 68(4); 3113-3125
- [51] Zhang Ke, Mao Yu-Ming, Leng Su-Peng, et al. Contract-theoretic approach for delay constrained offloading in vehicular edge computing networks. ACM/Springer Mobile Networks and Applications, 2019, 24(3); 1003-1014
- [52] Zhang Ke, Mao Yu-Ming, Leng Su-Leng, et al. Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks//Proceedings of the 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). Halmstad, Sweden, 2016; 288-294
- [53] Wu Zhen-Quan, Ye Dong-Dong, Yu Rong, et al. Edge computing offloading with parked vehicular collaboration in Internet of Vehicles. Journal of Beijing University of Posts and Telecommunications, 2019, 42(2); 108-113(in Chinese)
(吴振铨, 叶东东, 余荣等. 车联网中基于停车协同的边缘计算卸载方法. 北京邮电大学学报, 2019, 42(2); 108-113)
- [54] Dai Yue-Yue, Xu Du, Maharjan S, Zhang Yan. Joint load balancing and offloading in vehicular edge computing and networks. IEEE Internet of Things Journal, 2019, 6(3); 4377-4387
- [55] Du Jian-Bo, Yu F, Chu Xiao-Li, et al. Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization. IEEE Transactions on Vehicular Technology, 2019, 68(2); 1079-1092
- [56] Jiang Zhi-Jiang, Zhou Sheng, Guo Xue-Ying, Niu Zhi-Sheng. Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic. IEEE Internet of Things Journal, 2018, 5(1); 93-107
- [57] Feng Xue, Richardson B, Amman S, Glass J. On using heterogeneous data for vehicle-based speech recognition: A DNN-based approach//Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Brisbane, Australia, 2015; 4385-4389

- [58] Park H, Park M, Won K, et al. In-vehicle AR-HUD system to provide driving-safety information. *ETRI Journal*, 2013, 35(6): 1038-1047
- [59] Lin C, Lin Yu-Chen, Chen Long-Tai, Wang Yuan-Fang. Enhancing vehicular safety in adverse weather using computer vision analysis//Proceedings of the 2014 IEEE 80th Vehicular Technology Conference. Vancouver, Canada, 2014: 1-7
- [60] Hull B, Bychkovsky V, Zhang Yang, et al. CarTel: A distributed mobile sensor computing system//Proceedings of the 4th International Conference on Embedded Networked Sensor Systems. Boulder, USA, 2006: 125-138
- [61] Russo G, Baccaglini E, Boulard L, et al. Video processing for V2V communications: A case study with traffic lights and plate recognition//Proceedings of the 2015 IEEE 1st International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow(RTSI). Turin, Italy, 2015: 144-148
- [62] Chen Xu, Jiao Lei, Li Wen-Zhong, Fu Xiao-Ming. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 2016, 24(5): 2795-2808
- [63] Chen Wei-Wei, Wang Dong, Li Ke-Qin. Multi-user multi-task computation offloading in green mobile edge cloud computing. *IEEE Transactions on Services Computing*, 2019, 12(5): 726-738
- [64] Liu Chu-Bo, Li Ken-Li, Liang Jie, Li Ke-Qin. COOPER-SCHED: A cooperative scheduling framework for mobile edge computing with expected deadline guarantee. *IEEE Transactions on Parallel and Distributed Systems*, 2019(Early Access)
- [65] Tran T X, Pompili D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, 2019, 68(1): 856-868
- [66] Sundar S, Liang Ben. Offloading dependent tasks with communication delay and deadline constraint//Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Honolulu, USA, 2018: 37-45
- [67] Mahmoodi S E, Uma R N, Subbalakshmi K P. Optimal joint scheduling and cloud offloading for mobile applications. *IEEE Transactions on Cloud Computing*, 2019, 7(2): 301-313
- [68] Guo Song-Tao, Xiao Bin, Yang Yuan-Yuan, Yang Yang. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing//Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. San Francisco, USA, 2016: 1-9
- [69] Zhang Wei-Wen, Wen Yong-Gang, Wu D O. Collaborative task execution in mobile cloud computing under a stochastic wireless channel. *IEEE Transactions on Wireless Communications*, 2015, 14(1): 81-93
- [70] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 6517-6525
- [71] Al-Shuwaili A, Simeone O. Energy-efficient resource allocation for mobile edge computing-based augmented reality applications. *IEEE Wireless Communications Letters*, 2016, 6: 398-401
- [72] Ren Yi, Liu Fu-Qiang, Liu Zhi, et al. Power control in D2D-based vehicular communication networks. *IEEE Transactions on Vehicular Technology*, 2015, 64(12): 5547-5562
- [73] Araniti G, Campolo C, Condoluci M, et al. LTE for vehicular networking: A survey. *IEEE Communications Magazine*, 2013, 51(5): 148-157
- [74] Cox C. An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications. London, UK: John Wiley and Sons, 2012
- [75] Bey T, Tewolde G. Evaluation of DSRC and LTE for V2X//Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas, USA, 2019: 1032-1035
- [76] Shafi M, Molisch A F, Smith P J, et al. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE Journal on Selected Areas in Communications*, 2017, 35(6): 1201-1221
- [77] Tao Ming, Li Jia-Xing, Zhang Jie, et al. Vehicular data cloud platform with 5G support: Architecture, services, and challenges//Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). Guangzhou, China, 2017: 32-37
- [78] Masini B, Bazzi A, Natalizio E. Radio access for future 5G vehicular networks//Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall). Toronto, Canada, 2017: 1-7
- [79] Xiang Wei, Zheng Kan, Shen Xue-Min. 5G Mobile Communications. Switzerland: Springer, Cham, 2017
- [80] Molina-Masegosa R, Gozalvez J. LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 2017, 12(4): 30-39
- [81] Hu Yu-Chao, Patel M, Sabella D, et al. Mobile edge computing: A key technology towards 5G. *ETSI White Paper*, 2015, 11(11): 1-16
- [82] Olariu S, Eltoweissy M, Younis M. Towards autonomous vehicular clouds. *ICST Transactions on Mobile Communications and Applications*, 2011, 11(7-9): 1-11
- [83] Olariu S, Khalil I, Abuelela M. Taking VANET to the clouds. *International Journal of Pervasive Computing and Communications*, 2011, 7(1): 7-21
- [84] Zhang Qing-Yang, Wang Yi-Fan, Zhang Xing-Zhou, et al. OpenVAS: An open vehicular data analytics platform for CAVs//Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). Vienna, Austria, 2018: 1310-1320

[85] Li Yong-Bo, Chen Yu-Rong, Lan Tian, Venkataramani G. MobiQoR: Pushing the envelope of mobile edge computing via quality-of-result optimization//Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems(ICDCS). Atlanta, USA, 2017: 1261-1270

[86] Zhu Chao, Pastor G, Xiao Yu, et al. Fog following me: Latency and quality balanced task allocation in vehicular fog computing//Proceedings of the 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). Hong Kong, China, 2018: 1-9

[87] Tijms H C. A First Course in Stochastic Models. New York, USA: John Wiley and Sons, 2003

[88] Liu Yan-Chen, Lee M, Zheng Yan-Yan. Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. IEEE Transactions on Mobile Computing, 2016, 15(10): 2398-2410

[89] Ranadheera S, Maghsudi S, Hossain E. Mobile edge computation offloading using game theory and reinforcement learning. arXiv preprint/1711.09012, 2017

[90] Moura J, David H. Game theory for multi-access edge computing: Survey, use cases, and future trends. IEEE Communications Surveys & Tutorials, 2019, 21(1): 260-288

[91] Mao Hong-Zi, Alizadeh M, Menache I, Kandula S. Resource management with deep reinforcement learning//Proceedings of the 15th ACM Workshop on Hot Topics in Networks. New York, USA, 2016: 50-56

[92] Behrisch M, Bieker-Walz L, Erdmann J, Krajzewicz D. SUMO—Simulation of Urban MObility: An overview//Proceedings of the SIMUL 2011, the Third International Conference on Advances in System Simulation. Barcelona, Spain, 2011: 63-68

[93] Krajzewicz D, Erdmann J, Behrisch M, Bieker-Walz L. Recent development and applications of SUMO-Simulation of Urban Mobility. International Journal on Advances in Systems and Measurements, 2012, 5(3&4): 128-138

[94] Krajzewicz D, Hertkorn G, Rssel C, Wagner P. SUMO: An open-source traffic simulation//Proceedings of the 4th Middle East Symposium on Simulation and Modelling. Sharjah, United Arab Emirates, 2002: 183-187

[95] Wegener A, Piórkowski M, Raya M, et al. TraCI: An interface for coupling road traffic and network simulators//Proceedings of the 11th Communications and Networking Simulation Symposium. New York, USA, 2008: 155-163

[96] Codeca L, Frank R, Engel T. Luxembourg SUMO traffic (LuST) scenario: 24 hours of mobility for vehicular networking research//Proceedings of the 2015 IEEE Vehicular Networking Conference(VNC). Kyoto, Japan, 2015: 1-8

[97] Sommer C, German R, Dressler F. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. IEEE Transactions on Mobile Computing, 2011, 10(1): 3-15

[98] Sommer C, Yao Zheng, German R, Dressler F. Simulating the influence of IVC on road traffic using bidirectionally coupled simulators//Proceedings of the IEEE INFOCOM Workshops. Phoenix, USA, 2008: 1-6

[99] Varga A, Visual M, Omnet R, Method S. The OMNeT++ discrete event simulation system//Proceedings of the European Simulation Multiconference (ESM'2001). Prague, Czech Republic, 2001: 319-324

[100] Hagenauer F, Dressler F, Sommer C. Poster: A simulator for heterogeneous vehicular networks//Proceedings of the 2014 IEEE Vehicular Networking Conference(VNC). Paderborn, Germany, 2014: 185-186

[101] Wang Jin, Hu Jia, Min Ge-Yong, et al. Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning. IEEE Communications Magazine, 2019, 57(5): 64-69

[102] Huang Liang, Bi Su-Zhi, Zhang Ying-Jun. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. IEEE Transactions on Mobile Computing, 2019(Early Access)

[103] Mtibaa A, Harras K, Alnuweiri H. Friend or foe? Detecting and isolating malicious nodes in mobile edge computing platforms//Proceedings of the 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom). Vancouver, Canada, 2015: 42-49



LI Zhi-Yong, Ph. D. , professor, Ph.D. supervisor. His research interests include intelligent optimization algorithms, intelligent perception and learning, and CPS model in embedded systems.

WANG Qi, M.S. candidate. His research interests include mobile edge computing and deep reinforcement learning.

CEHN Yi-Fan, Ph.D. candidate. Her research interests include mobile edge computing and game theory.

XIE Guo-Qi, Ph. D. , associate professor. His research interests include embedded and cyber-physical systems, parallel and distributed systems, and safety-critical systems.

LI Ren-Fa, Ph. D. , professor, Ph.D. supervisor. His research interests include computer architectures, embedded computing systems, cyber-physical systems, and Internet of Things.

Background

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1308604), the National Natural Science Foundation of China (Nos. 61976086, 61672215, 61702172), and the Hunan Provincial Natural Science Foundation Youth Science Foundation (No. 2018JJ3076). Currently, the emergence of computation-intensive and delay-sensitive on-board applications makes it quite a challenge for vehicles to be able to provide high level of computation capacity and performance. Resource-constrained vehicles can limit these applications and make it difficult to guarantee the quality of service required. Traditional cloud computing is an option, but remote cloud has a large transmission delay. While mobile edge computing focuses on transferring computing resources to the edge of the network, providing high-performance, low-latency services for mobile devices, it can be possible to deal with computation-intensive and delay-sensitive tasks. Vehicular edge computing (VEC) is a new paradigm that has received much attention recently, as it can extend the computation capability to vehicular network edge. With the advent of VEC, edge devices such as RSU, MEC server and vehicles can provide high quality service for vehicle applications. At

present, some researches have summarized task offloading of mobile edge computing, but have not studied the detailed analysis of task offloading in vehicular edge computing.

This paper summarizes the previous related studies on vehicular edge computing task offloading. Specifically, from the existing research, the system model of VEC is introduced in detail from three aspects; computing model, task model and communication model. Then some optimization objectives in VEC, such as minimizing offload delay, minimizing energy consumption and application quality of results, are summarized. Subsequently, the existing researches are discussed in detail from the perspectives of centralized decision-making and distributed decision-making of computing task unloading, and have been classified from the aspects of computing resources, optimization objectives, offloading types, task dependencies, vehicle mobility and decision-making methods. In addition, several simulation tools for computing offloading in VEC are introduced, which is convenient for researchers to use a more realistic simulation environment for research. Finally, some problems faced by VEC computing offloading are proposed, and possible developing trends are discussed.