

边缘计算网络中基于排队论的通信和计算资源联合优化

薛建彬, 郁柏文, 徐小凤, 豆俊

引用本文

薛建彬, 郁柏文, 徐小凤, 豆俊. 边缘计算网络中基于排队论的通信和计算资源联合优化[J]. 计算机科学, 2024, 51(11A): 240100103-9.

XUE Jianbin, YU Bowen, XU Xiaofeng, DOU Jun. Queueing Theory-based Joint Optimization of Communication and Computing Resources in Edge Computing Networks [J]. Computer Science, 2024, 51(11A): 240100103-9.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于强化学习考虑电池损耗的电动汽车充放电控制算法](#)

Reinforcement Learning Algorithm for Charging/Discharging Control of Electric Vehicles Considering Battery Loss

计算机科学, 2024, 51(11A): 231200147-7. <https://doi.org/10.11896/jsjx.231200147>

[基于改进近端策略优化算法的智能渗透路径研究](#)

Intelligent Penetration Path Based on Improved PPO Algorithm

计算机科学, 2024, 51(11A): 231200165-6. <https://doi.org/10.11896/jsjx.231200165>

[基于自供电无人机远距离中继通信与计算卸载策略优化研究](#)

Study on Optimization of Long-distance Relay Communication and Computational Offloading Strategy Based on Self-powered UAVs

计算机科学, 2024, 51(11A): 240300069-7. <https://doi.org/10.11896/jsjx.240300069>

[基于深度强化学习的云边协同任务迁移与资源再分配优化研究](#)

Cloud-Edge Collaborative Task Transfer and Resource Reallocation Optimization Based on Deep Reinforcement Learning

计算机科学, 2024, 51(11A): 231100170-10. <https://doi.org/10.11896/jsjx.231100170>

[基于端边协同的节点部署和资源分配联合优化方法](#)

Joint Optimization Method for Node Deployment and Resource Allocation Based on End-EdgeCollaboration

计算机科学, 2024, 51(11A): 240200010-7. <https://doi.org/10.11896/jsjx.240200010>

边缘计算网络中基于排队论的通信和计算资源联合优化

薛建彬 郁柏文 徐小凤 豆俊

兰州理工大学计算机与通信学院 兰州 730050

(volvoxuejb@126.com)

摘要 高可靠和低延迟是目前车联网边缘计算网络中最重要的研究方向之一。为了满足车联网网络中复杂多变的任务请求,有效并且高效地分配通信资源和计算资源,提出了一种基于任务排队论模型和边缘计算模型相结合的智能通信和计算资源分配的多目标强化学习策略。该策略将通信资源和计算资源的分配相结合,以降低由延迟和可靠性组成的系统总成本。该策略可以被分解成3种算法,首先联合计算卸载与协作算法是该策略的一个通用框架,它首先使用KNN方法为生成的任务请求选择卸载层,如边缘计算层和本地计算层;然后,当选择本地计算层执行任务时,使用称为协作车辆选择的算法来查找执行协作计算的目标车辆;最后,通信和计算资源的分配被定义为两个独立的目标,称为多目标资源分配的算法在移动边缘计算层使用强化学习来实现问题的最优解。仿真结果表明,与随机计算、全部边缘计算和全部本地计算相比,所提策略有效地降低了系统的总成本。KNN方法和随机卸载方法相比,节省了系统的总成本,强化学习算法在系统总成本的控制上也优于传统的粒子群算法。

关键词: 车联网;边缘计算;排队论;KNN;资源分配;强化学习

中图分类号 TN929.5

Queueing Theory-based Joint Optimization of Communication and Computing Resources in Edge Computing Networks

XUE Jianbin, YU Bowen, XU Xiaofeng and DOU Jun

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730053, China

Abstract High reliability and low latency is one of the most important research directions in edge computing networks for vehicular networking. In order to meet the complex and variable task requests in vehicular networking networks, communication and computation resources are allocated effectively and efficiently. A multi-objective reinforcement learning strategy for intelligent communication and computation resource allocation based on the combination of task queueing theory model and edge computing model is proposed. The strategy combines the allocation of communication and computation resources to reduce the total system cost consisting of latency and reliability. The strategy can be decomposed into three algorithms, firstly, the joint computational offloading and collaboration algorithm is a generic framework for the strategy which first selects the offloading layer for the generated task requests such as the edge computing layer and the local computing layer using the KNN method. Then, when the local computing layer is selected to perform the task, an algorithm called collaborative vehicle selection is used to find the target vehicle to perform the collaborative computation. Finally, the allocation of communication and computational resources is defined as two independent objectives and the algorithm called multi-objective resource allocation uses reinforcement learning at the mobile edge computing layer to achieve an optimal solution to the problem. Simulation results show that the proposed strategy effectively reduces the total cost of the system compared to random computing, all edge computing and all local computing. The KNN approach saves the total cost of the system compared to the random offloading approach and the reinforcement learning algorithm outperforms the traditional particle swarm algorithm in controlling the total cost of the system.

Keywords Internet of Vehicles, Edge computing, Queueing theory, KNN, Resource allocation, Reinforcement learning

1 引言

车联网(IOV)可以处理许多由车辆生成的高级任务,比如有关车辆的道路安全、交通效率和自动驾驶以及车上乘客移动设备所产生的娱乐信息等^[1]。这说明必须在高可靠性和

低延迟性的双重约束下,对于车辆产生的不同任务请求,给车辆提供更加多样化的服务。然而,近年来,由于车辆数目的爆炸性增长以及车辆有限的计算能力,传统的IOV无法满足高可靠性和低延迟性的双重约束^[2]。为了克服这些困难和挑战,移动边缘计算(MEC)方法应运而生^[3-4]。与传统云计算

基金项目:甘肃省科技资助计划(23YFGA0062);基于5G移动边缘计算的无人机应急场景中绿色通信方案研究(2022A-215)

This work was supported by the Gansu Provincial Science and Technology Program Funding(23YFGA0062) and Research on Green Communication Scheme in UAV Emergency Scenarios Based on 5G Mobile Edge Computing(2022A-215).

通信作者:郁柏文(13639329628@163.com)

相比,在 MEC 网络中,计算能力在无线网络的边缘被扩展放大,因此一些计算密集型任务可以被卸载到 MEC 服务器进行处理,并以此作为终端设备向车辆提供边缘化任务处理与本地化云服务,进而可以进一步减少因为传输和计算所产生的时延,同时也可以提高系统整体的可靠性^[5]。此外,与传统云计算相比,MEC 服务器以其所特有的灵活性和可扩展部署性,用一种经济高效的方式保证了 IOV 网络中的异构服务类型^[6-7]。

同时,由于车辆数目的激增,车辆所产生的任务请求也在随之大幅度增加,对于任务请求处理的需求量也在不断提升。这些车辆的任务在无形中产生了排队和拥挤的现象,在这样的环境下,排队论作为一种研究排队系统(又称随机服务系统)的一门数学理论和方法,越来越多地被人们所关注。它是在对各种排队系统概率规律性进行研究的基础上,解决排队系统的最优设计和最优控制问题。70 年代以来,人们开始研究排队网络和复杂排队问题的渐近解等,成为了研究现代排队论的新趋势。

此外,大量研究已经证明机器学习是在计算机视觉、自然语言处理和机器人技术等领域的一种有效的问题解决工具^[8],它可以在无线网络中的信道预测、安全架构和设备对设备的通信等方面发挥极大的作用^[9-11]。因此,机器学习可以作为智能交通系统发展的重要组成部分并应用于车辆网络。

可靠性和延迟是评估车载网络性能的两个重要指标。但是这两个指标在之前的研究中通常被视为单独的目标去进行优化。文献[12]考虑了可靠性和排队等待时间约束下的功率最小化问题,然后提出了一个 Lyapunov 框架来解决这个问题。文献[13]提出了一种资源分配策略,该策略可以在平均功耗和可靠性之间得到权衡。文献[14]提出了一种基于深度强化学习的方法来自适应地分配通信和计算资源,从而最大程度地减少卸载延迟。文献[15]提出了一个多用户非协作计算卸载游戏模型来调整每辆车的卸载概率,并构建了一种基于计算卸载游戏的分布式最佳响应算法,以最小化任务执行的延迟。文献[16]将可靠性定义为瞬时速率的函数,并提出了一种风险敏感方法来保证可靠性。此外,还利用了一种风险敏感的方法来最大限度地减少车载边缘计算网络中的端到端任务卸载延迟^[17]。一些文献,如文献[18],提供了关于异构网络中计算和通信的资源分配的调查,以保持极高的服务质量(QoS)。文献[19]将车辆通信的高可靠低延迟要求转换为优化约束,然后提出了一种单独的资源块和功率分配算法,以最大化该约束下的用户和速率。文献[20]提出了一种效用延迟方法,以在高可靠性和低延迟性的约束下最大化网络效用。然而,这些工作仅在缓慢变化的信道中进行了验证,它们可能不适合时变网络拓扑。文献[21]使用 Lyapunov 优化和匹配理论来关联用户 MEC 服务器,以保证可靠性和延迟性能。文献[22]开发了一种元启发式方法来求解 URLLC 的混合整数规划。文献[23]采用了短分组传输,并将其考虑到资源分配算法设计中,以减少延迟并提高可靠性。文献[24]提出了一种帧设计算法和半持久调度算法来实现最优的帧设计和资源分配方案,这可以极大地满足车载网络的高可靠性和低延迟性要求。对于车载边缘计算网络,一种有效且有前景的资源分配是一种统一的策略,它调度所有可用的资源(即云计算资源、移动边缘计算资源、本地计算资源和通信资源),以

满足不同应用程序的各种任务要求^[25]。文献[26]提出了一种 COCA 在线算法,优化了数据中心运营成本,同时满足了碳中和要求,限制了潜在的碳中和破坏。文献[27]提出了一种基于 Lyapunov 优化的方案,在具有多核 CPU 的移动设备中运行的多个应用程序的云卸载调度以及云执行输出的下载进行了调度优化,得到了对于排队系统中平均功耗和平均队长这两个优化目标之间的基本权衡。文献[28]使用排队模型为软件定义卫星网络提供了一个分析模型,使其能够准确地评估软件定义卫星网络的性能。文献[29]通过建立合理的排队模型,采用数学中函数求极值的方法,得出了最佳的启动门限,进而得到总成本最小的最优策略。文献[30]引入了排队论模型,以便于评估各服务器的实时负载情况,根据评估所得的负载指标,将任务分配给服务器处理,有效地平衡各服务器的负载,减少了任务请求的平均队长。文献[31]将对偶内点法(PDIPM)和填充函数法(FFM)相结合,得到了 OPF 问题的全局最小值,并且有效地避免了局部最小值。文献[32]通过在排队系统中采用概率提升方案,有效地控制了发送到 MEC 的不同优先级分组的分组拥塞,并且提出了一种改进的磷虾群元启发式优化算法,从而获得了在能耗和排队拥塞方面对于 MEC 系统总开销最小化的最优结果。文献[33]采用 M/M/1 排队模型推导了最优重试限制、数据包溢出丢弃率和总丢失率的闭式表达式,并研究了一种自适应 MAC 重试限制方案。文献[34]在排队论系统中,引入云辅助计算框架,利用 ORP 算法得到了更高的系统灵活性和成本效率。

以上文献主要关注车载边缘计算网络中用户和 MEC 服务器之间的计算卸载问题,只有少数文献关注此类网络中 MEC 服务器之间的协作计算。然而,MEC 服务器之间的协同计算将有效提高车辆的计算能力,有助于车辆网络提高可靠性和降低延迟。通常,解决 NP 难组合优化问题的传统数值优化方法包括 max-min 算法^[35]、迭代算法^[36]等。如今,由于无线信道的动力学和车辆网络的网络拓扑结构,研究人员开始更加关注强化学习。RL 可以利用历史数据来解决不确定性下的决策^[37]。当新的任务请求到达时,RL 可以自适应地分配资源,以最大限度地提高车辆网络的长期效益。文献[38]提出了基于 RL 算法的集群协同调度,通过最大化容量来实现高效可靠的通信。文献[39]研究了一种更有效的基于强化学习的资源分配算法,以自适应地改变 5G 车载网络中的时分双工(TDD)配置。为了降低处理请求延迟的概率,文献[40]在车载云系统中提出了一种新的基于半马尔可夫决策过程和 RL 算法的自适应云资源分配模型。文献[41]通过利用停放的车辆,从所提出的启发式算法和 RL 算法中获得了短期和长期资源分配的资源分配算法。

基于上述背景,本文提出了一种基于排队论 M/M/1 任务队列的多目标强化学习策略,以降低系统的总成本,包括延迟成本和可靠性成本。主要贡献总结如下:所提出的策略考虑了联合层和跨层的计算和通信资源分配,以保证各种任务需求。将所提出的策略分解为 3 种算法。算法 1 是为车辆任务请求选择卸载层的通用框架;算法 2 是当车辆选择任务请求在本地执行时,为任务请求生成车辆寻找辅助计算车辆;算法 3 用于分配 MEC 的通信和计算资源。

本文第 2 章介绍了系统模型,包括网络模型、任务模型和计算模型;第 3 章联合通信和计算资源分配被公式化为一个

优化问题;第4章提出了一种多目标强化学习策略,以保证IoV网络中多样化服务的可靠性和延迟;第5章说明了数值结果,并进行对比;最后总结全文。

2 系统模型

如图1所示,为了最小化系统总成本,我们考虑了一个移动边缘计算模型并且在其中引入了M/M/1队列模型。假设该系统由车辆、RSU以及MEC服务器组成。车辆 $N=\{1,2,3,\dots,n\}$ 行驶在道路上,道路两侧部署了RSU,每一个RSU配备了一个MEC服务器 $M=\{1,2,3,\dots,m\}$,来为车辆生成的任务请求提供计算服务。车辆和MEC处的任务请求处理模型都视为M/M/1队列。车辆所产生的任务可以在本地处理,也可以卸载至MEC服务器进行处理。对于每个车辆 n ,它可以通过卸载决策决定任务请求是在本地执行还是卸载到RSU上的MEC服务器执行。为了便于研究,本文假设车辆通过vehicle-to-infrastructure(V2I)链路被最近的RSU覆盖,vehicle-to-vehicle(V2V)链路被用来执行车辆间的辅助计算。

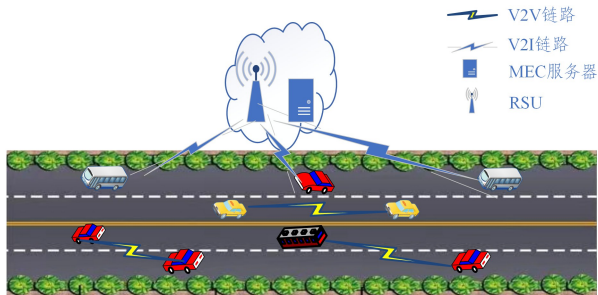


图1 系统模型图

Fig. 1 System model diagram

2.1 任务模型

如图2所示,假设车辆 $N=\{1,2,3,\dots,n\}$ 会产生一系列任务请求,生成的任务请求遵循泊松分布,平均到达率为 λ ,平均服务率为 μ 。然后进行任务请求的卸载决策,引入卸载决策向量 $\mathbf{A}=[\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n]$,其中 $\alpha_n=\{\alpha_n^l, \alpha_n^m\}$, $\alpha_n^l, \alpha_n^m \in [0,1]$,并且 $\alpha_n^l + \alpha_n^m = 1$ 。这里, α_n 是车辆 n 的卸载决策向量,它包含两个二进制指示变量 α_n^l 和 α_n^m 。 $\alpha_n^l = 1$ 表示车辆将在本地处理所产生的任务请求, $\alpha_n^m = 1$ 表示车辆将会把所产生的任务请求卸载到MEC服务器上进行处理。对于 α_n^l 和 α_n^m ,每次只能有一个指示变量等于1,也就是说,车辆产生的任务请求只能在一个地方进行处理。卸载决策执行任务请求卸载后,任务请求根据卸载决策结果进入到本地任务请求队列或MEC任务请求队列中进行处理。在本地或MEC处理任务请求时,亦可选择辅助车辆来进行协同计算。由于车辆和MEC处的任务队列为M/M/1模型,则在车辆处和MEC处的任务平均队长为:

$$L_s = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda} \quad (1)$$

平均排队长为:

$$L_q = L_s - \rho = \frac{\lambda^2}{\mu(\mu-\lambda)} \quad (2)$$

任务请求在排队系统中的平均响应时间为:

$$T_s = \frac{L_s}{\lambda} = \frac{1}{\mu-\lambda} \quad (3)$$

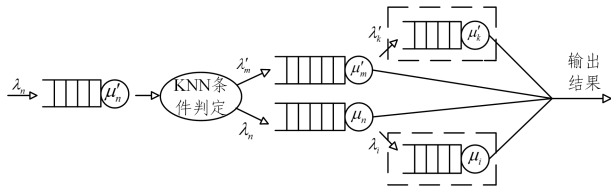


图2 任务模型图

Fig. 2 Task model diagram

本文把车辆产生的每个任务请求表示为 $\phi_n = \{K_n, X_n, \tau_n\}$,其中 K_n 表示任务请求输入数据的大小, X_n 表示任务请求在被执行后输出数据的大小, τ_n 被定义为任务请求的最大可容忍延迟,并且以上3个参数的值对于车辆生成的每个任务请求都是不同的。

与卸载决策向量类似,我们将带宽分配向量定义为 $B=[\beta_1, \beta_2, \beta_3, \dots, \beta_N]$, $0 < \beta_N \leq 1$,并且 $\sum_{n=1}^N \beta_n \leq 1$ 。对于每一个车辆来说,所分配的带宽将会影响传输速率,而传输速率将会直接影响IoV网络的可靠性,因此在总带宽的约束下,适当的带宽分配将会提高IoV网络的可靠性。

2.2 计算模型

本节将描述本地计算模型和MEC计算模型,其中本地计算模型中还包括辅助车辆协同计算。

2.2.1 本地计算模型

由于车辆处的队列模型为M/M/1队列,平均服务率为 μ ,车辆 $N=\{1,2,3,\dots,n\}$ 生成的任务请求遵循泊松过程,平均到达率为 λ ,且每个请求包含的任务大小为 K 。当车辆生成的任务请求 Ψ_n 被选择在本地执行时,我们将车辆 n 在本地进行处理任务请求的平均响应时间定义为 T_n^{comp} ,将 Z_n 定义为车辆 n 的标准化工作负载,它表示车辆已经被占用的CPU百分比。因此车辆在本地处理任务请求的平均响应时间为:

$$T_n^{\text{comp}} = \frac{1}{\mu_n - \lambda_n} = \frac{1}{\mu_n(1 - Z_n) - \lambda_n} \quad (4)$$

当车辆产生的任务请求被卸载决策要求在本地执行时,恰好此时车辆 n 没有足够的计算资源进行任务请求的处理,车辆 n 将选择附近具有可用计算资源的车辆 i ,将任务请求卸载至车辆 i 进行辅助计算。此时车辆 i 的辅助计算的平均响应时间包括3部分:上行链路传输延迟、辅助计算延迟和下行链路传输延迟。

考虑到车辆之间的辅助计算,上行链路的平均传输延迟可以表示为:

$$T_{n,i}^{\text{UL}} = \frac{\sum_{n=1}^n K_n}{nR_{n,i}^{\text{UL}}} \quad (5)$$

其中, $R_{n,i}^{\text{UL}}$ 是车辆 n 将任务请求传输到车辆 i 的上行链路传输速率,它可以表示为:

$$R_{n,i}^{\text{UL}} = \omega_{n,i}^{\text{UL}} \log_2 \left(1 + \frac{P^{\text{VE}} h_{n,i}^{\text{UL}}}{\omega_0} \right) \quad (6)$$

其中, $\omega_{n,i}^{\text{UL}}$ 是车辆 n 和车辆 i 之间V2V通信的上行链路信道带宽, P^{VE} 是车辆的发射功率, $h_{n,i}^{\text{UL}}$ 是V2V通信的上行链路信道增益, ω_0 是高斯噪声。

在车辆 i 上处理任务请求的平均响应时间为:

$$T_i^{\text{comp}} = \frac{1}{\mu_i - \lambda_i} = \frac{1}{\mu_i(1 - Z_i) - \lambda_i} \quad (7)$$

计算结果需要从车辆 i 回传到车辆 n , 与上行链路相似, 下行链路的平均传输时延可以被表示为:

$$T_{n,i}^{DL} = \frac{\sum_{n=1}^n X_n}{nR_{n,i}^{DL}} \quad (8)$$

其中, $R_{n,i}^{DL}$ 是车辆 i 将任务处理结果回传到车辆 n 的下行链路传输速率, 它可以表示为:

$$R_{n,i}^{DL} = \omega_{n,i}^{DL} \log_2 \left(1 + \frac{P^{VE} h_{n,i}^{DL}}{\omega_0} \right) \quad (9)$$

其中, $\omega_{n,i}^{DL}$ 是车辆 n 和车辆 i 之间 V2V 通信的下行链路信道带宽, P^{VE} 是车辆的发射功率, $h_{n,i}^{DL}$ 是 V2V 通信的下行链路信道增益。在本地计算中, 由于车辆的发射功率参数不在优化范围内, 为了便于计算, 默认所有车辆发射功率相同, 用 P^{VE} 表示。

因此, 任务请求在本地执行的平均响应时间为:

$$T_n^l = \begin{cases} T_n^{\text{comp}}, & \text{local} \\ T_m^{\text{UL}} + T_i^{\text{comp}} + T_m^{\text{DL}}, & \text{collaborative} \end{cases} \quad (10)$$

2.2.2 MEC 计算模型

由于 MEC 处的队列模型依然为 $M/M/1$ 队列, 平均服务率为 μ 。任务请求到达 MEC 服务器 $M = \{1, 2, 3, \dots, m\}$ 遵循泊松过程, 平均到达率为 λ , 且每个任务请求包含的任务大小为 K 。如果车辆 n 产生的任务请求决定在 MEC 执行, 那么任务请求执行过程可以分成 4 部分。首先, 车辆 n 将产生的任务请求上传到最近的 RSU, 其次 RSU 将任务请求转发给 MEC 服务器 m , 然后 MEC 将会分配其计算资源来执行车辆上传的任务请求, 并且如果在 MEC 服务器 m 中没有足够的剩余计算资源来处理车辆的任务请求时, MEC 服务器 k 将会被选择用来进行协作计算。我们将 Z_m 定义为 MEC 服务器 m 的标准化工作负载, 它表示 MEC 服务器已经被占用的 CPU 百分比。辅助计算完成后 MEC 服务器 k 将会把辅助计算结果反馈给 MEC 服务器 m 。最后, MEC 服务器将会把计算结果返回给车辆。

根据以上步骤, 车辆 n 上传任务请求的平均传输时延可以表示为:

$$T_{n,m}^{\text{UL}} = \frac{\sum_{n=1}^n K_n}{nR_{n,m}^{\text{UL}}} \quad (11)$$

其中, $R_{n,m}^{\text{UL}}$ 是车辆 n 将任务请求上传到 RSU 的上行链路传输速率, 它可以表示为:

$$R_{n,m}^{\text{UL}} = \omega_{n,m}^{\text{UL}} \log_2 \left(1 + \frac{P^{VE} h_{n,m}^{\text{UL}}}{\omega_0} \right) \quad (12)$$

其中, $\omega_{n,m}^{\text{UL}}$ 是车辆 n 和 RSU 之间 V2I 通信的上行链路信道带宽, P^{VE} 是车辆的传输功率, P^{VE} 是 V2I 通信的上行链路信道增益, ω_0 是高斯噪声。

在 MEC 服务器上执行任务请求的处理延迟, 即处理任务请求的平均响应时间可以表示为:

$$T_{n,m,k}^{\text{comp}} = \frac{1}{\mu_m (1 - \chi_m) + \lambda_m} + \frac{1}{\mu_k (1 - \chi^k) - \lambda_k} \quad (13)$$

其中, χ_m 为 MEC 服务器 m 和 MEC 服务器 k 之间所分配执行的请求的数据大小比。也就是说, $\chi_m = 0$ 意味着 MEC 服务器 m 将独立完成所上传的任务请求。

计算结果从 MEC 服务器 m 回传到车辆 n , 与上行链路相似, 下行链路的平均传输时延为:

$$T_{n,m}^{\text{DL}} = \frac{\sum_{n=1}^n X_n}{nR_{n,m}^{\text{DL}}} \quad (14)$$

其中, X_n 为任务请求处理结果的数据大小, $R_{n,m}^{\text{DL}}$ 是下行链路的传输速率, 它可以被表示为:

$$R_{n,m}^{\text{DL}} = \omega_{n,m}^{\text{DL}} \log_2 \left(1 + \frac{P^{\text{RSU}} h_{n,m}^{\text{DL}}}{\omega_0} \right) \quad (15)$$

其中, $\omega_{n,m}^{\text{DL}}$ 是车辆 n 和 RSU 之间 V2I 通信的下行链路信道带宽, P^{RSU} 是 RSU 的传输功率, $h_{n,m}^{\text{DL}}$ 是 V2I 通信的下行链路信道增益, ω_0 是高斯噪声。

因此, 当生成的任务请求由卸载决策决定在 MEC 执行时, 任务请求执行的平均响应时间为:

$$T_n^m = T_{n,m}^{\text{UL}} + T_{n,m,k}^{\text{comp}} + T_{n,m}^{\text{DL}} \quad (16)$$

当任务请求被卸载到 MEC 执行时, 由于 MEC 服务器之间任务请求传输的延迟值较小, 同时为了便于计算, 我们通常将此延迟忽略。

3 问题公式化

本章中, 车辆生成的任务请求排队系统的任务卸载和 MEC 服务器中的资源分配将作为一个优化问题进行详细阐述。目的是为了获得系统中所有车辆任务请求的由延迟(任务请求排队系统的平均响应时间)和传输速率组成的最小总系统成本。

对于车辆 h , 其延迟成本可以定义为:

$$C_n^{\text{del}} = \alpha_n^l T_n^l + \alpha_n^m T_n^m \quad (17)$$

传输速率的成本可以定义为:

$$C_n^{\text{re}} = R_{n,m}^{\text{UL}} + R_{n,m}^{\text{DL}} \quad (18)$$

传输速率和所分配的带宽之间的关系可以表示为:

$$R_n = \omega_n \log_2 \left(1 + \frac{Ph}{\omega_0} \right) = \beta_n \omega \log_2 \left(1 + \frac{Ph}{\omega_0} \right) = \beta_n R \quad (19)$$

然后将式(12)和式(15)带入到式(18)中, 就可以得到:

$$C_n^{\text{re}} = \beta_n (R^{\text{UL}} + R^{\text{DL}}) \quad (20)$$

其中, R^{UL} 和 R^{DL} 分别指在没有进行资源分配情况下的上行链路传输速率和下行链路传输速率。

因此, 在最大容忍延迟 τ_n 、MEC 层中总的计算资源 U 和总带宽 ω 的共同约束下, 本文的优化问题可以被公式化为:

$$\min_{A,B,F} \sum_{n=1}^N I_d C^{\text{del}} + I_r \frac{1}{C^{\text{re}}} = \min_{A,B,F} \sum_{n=1}^N I_d (\alpha_n^l T_n^l + \alpha_n^m T_n^m) +$$

$$I_r \frac{1}{\beta_n (R^{\text{UL}} + R^{\text{DL}})}$$

$$\text{s. t. C1: } \alpha_n^l, \alpha_n^m \in \{0, 1\}, \alpha_n^l + \alpha_n^m = 1,$$

$$\text{C2: } \alpha_n^l T_n^l + \alpha_n^m T_n^m \leq \tau_n, \forall n \in N,$$

$$\text{C3: } 0 \leq Z_m \leq \alpha_n^m U, \forall m \in M,$$

$$\text{C4: } \sum_{m=1}^M Z_m \leq U, \forall m \in M,$$

$$\text{C5: } 0 < \beta_n \leq 1, \sum_{n=1}^N \beta_n = 1,$$

$$\text{C6: } \sum_{n=1}^N (\omega_n^{\text{UL}} + \omega_n^{\text{DL}}) \leq \omega,$$

$$\text{C7: } R_n^{\text{min}} \leq \beta_n R \leq R_n^{\text{UL}} \quad (21)$$

其中, I_d 是延迟成本的权重, I_r 是可靠性成本的权重。我们用传输速率去衡量系统的可靠性。对于不同车辆产生的不同任务请求, 其对可靠性和延迟的要求是不同的, 因此根据不同要求, 可以将 I_d 和 I_r 设置成不同的值, 但必须满足条件 $I_d + I_r = 1, I_d < I_r$ 表示此任务请求的延迟要求小于可靠性要求,

$I_d > I_r$ 表示此任务的延迟要求大于可靠性要求。在式(21)中,我们把传输速率的倒数定义为可靠性的成本,也就是说可靠性越大,可靠性成本越低,反之亦然。 A 和 B 分别是卸载决策向量和带宽分配向量,在此定义 $F = [f_1, f_2, f_3, \dots, f_N]$ 是在 MEC 层中的资源分配向量。

接下来对给出的 7 个约束条件进行解释。C1 表示计算任务 ψ_n 可以在任意一个层中执行,但是只能在一个层中执行。C2 表示延迟成本应该小于最大容忍延迟。C3 表示给每一个车辆生成的任务请求所分配的计算资源不可以超过 MEC 层中总的计算资源。C4 表示给所有车辆生成的任务请求所分配的计算资源不可以超过 MEC 层中总的计算资源。C5 表示所分配的带宽不可以超过总的可用带宽。C6 表示上行链路和下行链路的联合带宽约束。C7 表示传输速率必须满足 QoS 要求,并且不能超过可实现的传输速率。

问题(16)将通过获得最优卸载决策向量 A 、最优带宽分配向量 B 和最优计算资源分配向量 F 来得到解决。卸载决策向量 A 决定了生成的任务请求将要卸载到何处进行处理。带宽分配向量 B 决定了将要分配的带宽量用于计算任务的传输,所分配的带宽量与传输的可靠性有很大的关系。资源分配向量 F 决定了将分配多少的计算资源来进行任务请求的处理。由于 A 是一个二元向量,因此问题(16)的可行集和目标函数是非凸的。此外,最优的 A, B 和 F 的大小会随着任务请求数量的增加而迅速增加。由于 IOV 网络不稳定的拓扑结构,这一问题将更加严重。以往的方法是将非凸问题转化为凸问题,然而这种方法不适合多目标优化问题,因此使用 KNN 方法来获得最优的卸载决策向量 A ,通过强化学习来获得最优的带宽分配向量 B 和资源分配向量 F ,从而获得问题(21)的最优解。

4 算法实现

本章使用了一种多目标强化学习算法,用于获得最优的卸载决策向量 A 、带宽分配向量 B 和资源分配向量 F 。

所使用的算法主要由 3 种算法组成。算法 1 使用 KNN 方法选择出任务将要在何处执行。算法 2 用来在任务请求被选择在本地执行时,需要辅助车辆计算时,通过距离、运动方向和车辆计算能力等参数选择出辅助车辆。算法 3 用来当任务请求被选择在 MEC 执行计算时分配计算和通信资源,以此来保证延迟约束和可靠性的要求。

4.1 任务请求执行选择

算法 1 是在本地计算和 MEC 计算联合通信和计算资源分配的一个总体框架。首先选择任务请求在何处执行,然后嵌入算法 2 和算法 3 用来在本地或者 MEC 处理任务请求。

KNN 方法是一个简单高效的分类器,它可以用来选择生成的任务请求将要在何处执行。首先对于延迟 τ 和任务大小 K 进行归一化处理。算法 1 定义 E 为车辆和所选择计算位置的欧几里得距离,它可以被表示为:

$$E = \sqrt{(\tau_{\psi_j} - \tau_j)^2 + (K_{\psi_j} - K_j)^2} \quad (22)$$

算法 1 整体算法框架

1. 初始化

- 1.1. 计算任务集 $\{\psi_1, \psi_2, \dots, \psi_n\}$
- 1.2. 任务集的参数 $\psi_n = \{K_n, X_n, \tau_n\}$
- 1.3. 任务的当前位置 $P=1$, MEC 层的任务集 ζ

1.4. 本地计算层 L 的一般延迟 τ_1 , MEC 层 τ_2 , 允许的任务大小为 K_1 和 K_2

2. 选择计算层

计算 $(\tau_{\psi_n}, K_{\psi_n})$ 和 (τ_j, K_j) 之间的欧几里得距离 E , $E = \sqrt{(\tau_{\psi_n} - \tau_j)^2 + (K_{\psi_n} - K_j)^2}$, $P=1$

for $j=2$ do

计算 $(\tau_{\psi_n}, K_{\psi_n}, \tau_{\psi_n}, K_{\psi_n})$ 和 τ_j, K_j 之间的欧几里得距离 d_j

if $d_j < E$ then

$E = d_j, P = j$.

end if

end for

3. 任务卸载

For 每个计算任务 ψ_n do

if $P=1$ and $T_n^L < \tau_{\psi_n}$ then

在本地层进行任务计算,执行算法 2.

else

将任务 ψ_n 放置在任务集 ζ .

end if

if $P=2$ then

将任务 ψ_n 放置在任务集 ζ

end if

end for

if $\zeta \neq \emptyset$ then

执行算法 3

end if

首先进行初始化,被选择在 MEC 执行任务请求的任务集定义为 ζ 。然后进入到任务卸载决策阶段, $P=1$ 表示任务请求将要在本地进行处理, $P=2$ 表示任务请求将要在 MEC 进行处理。

在任务请求处理位置选择阶段,通过欧几里得距离选择出任务请求合适的处理位置,然后执行任务卸载决策,此时嵌入算法 2 和算法 3。如果 $P=1$ 且满足延迟约束,任务请求就会在本地进行处理,然后执行算法 2。如果由于不满足延迟约束条件,那么任务请求将会被卸载到 MEC 进行处理,此时执行算法 3。

4.2 协作车辆选择

算法 2 用来在任务请求被选择在本地执行时,选择辅助车辆帮助目标车辆进行任务请求的处理。利用距离、运动方向和车辆的计算能力,来寻找合适的辅助车辆。如果没有合适的辅助车辆,任务请求将被卸载到 MEC 进行处理。

在初始化阶段,车辆 i 生成计算任务,车辆 j 是候选的辅助车辆用来进行车辆间协同计算。在辅助车辆选择阶段,我们利用距离、运动方向和车辆计算能力等参数来选择出辅助车辆。如果有合适的辅助车辆,那么任务请求就在辅助车辆的帮助下在本地处理任务请求。如果没有合适的辅助车辆,任务请求将会被卸载到 MEC 进行处理。

算法 2 辅助车辆选择

1. 初始化

- 1.1. 计算车辆 i 和车辆 j 之间的距离 L_{ij} , 运动方向 d_i, d_j , 可用的计算能力 Z_i, Z_j
- 1.2. 任务大小 O_i , 每比特所需要的计算能力 O_i
- 1.3. 距离阈值 O_i
- 1.4. 辅助车辆 O_i

2. 车辆选择

```

if  $Z_i$ 是可用的 then
  计算任务  $\Psi_i$ 
else
  for 候选车辆  $j$  do
    if  $L_{ij} < \epsilon_1$  and  $d_i = d_j$  then
      if  $Z_j > K_{\Psi_i} E_{\Psi_i}$  then
         $\Pi = j$ , 车辆  $j$  辅助计算任务  $\Psi_i$ , 然后反馈给车辆  $i$ 
      break
    else
      continue
  end if
end if
end for
end if
if  $\Pi = 0$  then
  将任务卸载至 MEC 层.
end if

```

4.3 多目标资源分配

如果任务请求被选择在 MEC 进行处理, 此时算法 3 将会被执行。我们使用强化学习的多目标资源分配算法来保证系统的可靠性和延迟。延迟成本和可靠性成本分别被表示为 O_1 和 O_2 , 因此问题(21)的目标函数可以重新表示为:

$$\min_{A, B, F} \sum_{n=1}^N I_d O_1 + I_r O_2 \quad (23)$$

其中, O_1 和 O_2 分别是延迟成本和可靠性成本。

然后, 在问题(18)中, 使用多目标强化学习来分配通信和计算资源, 从而最小化系统的总成本。 $I = [I_d, I_r]$ 是权重向量。如果 $I_d > I_r$, 表示系统对延迟要求高于可靠性要求, 反之亦然。 $I_d = 1$ 表示系统成本只考虑延迟成本, $I_r = 1$ 表示系统成本只考虑可靠性成本。

算法 3 联合优化

1. 初始化

- 1.1. Q 矩阵 $Q(s, a)$, 学习参数 γ , 奖励矩阵 R .
- 1.2. 本地计算层的系统总成本 tc_{local} .
- 1.3. 多目标函数集合 $O = \{O_1, O_2, \dots, O_m\}$.
- 1.4. 多目标权重集合 $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_m\}$.
- 1.5. 多目标 Q 矩阵 $Q = \{Q_1, Q_2, \dots, Q_m\}$.

2. 资源分配

```

for 每一阶段 do
  选择一个随机状态  $s_t$ .
  for 每一步 do
    从状态  $s_t$  的可能动作中选择一个动作  $a$ .
    执行所选调作  $a$ , 观察每个动作  $O_i$  的奖励  $r_i$ ,  $r_i = \frac{1}{tc'(s, a)}$ , 更新  $Q_i$  矩阵.
  end for
end for

```

更新 Q 对应的 Λ , $Q = \sum_{i=1}^m \lambda_i Q_i$.

计算 $Q(s, a) \leftarrow R(s, a) + \gamma \cdot \max_{a'} Q(s', a')$.

更新状态 $s \leftarrow s'$.

直到 Q 矩阵覆盖.

end for

end for

1) 状态

对于车辆 n , 在时间 $t \in \{0, 1, 2, \dots, T-1\}$ 的状态向量可以表示为 $s = (tc, ac)$, 其中 tc 表示系统成本, ac 表示可用资源。对于延迟成本 O_1 , ac 表示可用的计算资源, 它可以被表示为 $ac = U - \sum_{m=1}^M Z_m$; 对于可靠性成本 O_2 , ac 是可用的通信带宽。

2) 行动

行动被定义为带宽分配向量 $B = [\beta_1, \beta_2, \beta_3, \dots, \beta_N]$ 和计算资源分配向量 $F = [f_1, f_2, f_3, \dots, f_N]$ 。对于延迟成本 O_1 来说, 不同的计算资源分配向量决定了不同的计算资源分配, 也就导致了不同的延迟成本。对于可靠性成本 O_2 来说, 不同的带宽分配向量决定了不同的带宽分配, 也就导致了不同的可靠性成本。

3) 奖励

问题式(16)的目的是为了获得最小的总系统成本, 在强化学习中这一目的可以转化成获得最大的奖励。在此处, 我们把奖励定义为 $r = \frac{1}{tc(s, a)}$, 这里 $tc(s, a)$ 表示系统的总成本, 这就表明系统的总成本越高, 所获得的奖励越小; 系统的总成本越低, 所获得的奖励越大。

对于可靠性成本 O_2 , $tc(s, a) = \frac{1}{\sum_{n=1}^N R_n^{UL} + R_n^{DL}}$ 。对于延迟

成本 O_1 , 我们要考虑两种情况。

当 $ac = 0$ 时, 表明 MEC 服务器 m 中没有可用的计算资源, 这时要选择 MEC 服务器 k 来进行任务请求的处理。在这种情况下, 延迟成本 O_1 可以被表示为:

$$tc(s, a) = \sum_{n=1}^N T_{n,m}^{UL} + T_{n,k}^{comp} + T_{k,n}^{DL} = \left(\sum_{n=1}^N T_{n,m}^{UL} + T_{k,n}^{DL} \right) + \left(\frac{n}{\mu_k (1 - Z_k) - \lambda_k} \right) \quad (24)$$

当 $ac \neq 0$ 时, 表明在 MEC 服务器 m 中有可用的计算资源, 因此任务请求可以在 MEC 服务器 m 上进行处理, 或者在 MEC 服务器 m 和 MEC 服务器 k 之间协作处理。在这种情况下, 延迟成本 O_1 可以表示为:

$$tc(s, a) = \sum_{n=1}^N T_{n,m}^{UL} + T_{n,m,k}^{comp} + T_{m,n}^{DL} = \left(\sum_{n=1}^N T_{n,m}^{UL} + T_{m,n}^{DL} \right) + \frac{n}{\mu_m (1 - \chi^n) - \lambda_n} + \frac{n}{\mu_k \chi^n - \lambda_k} \quad (25)$$

其中, χ^n 表示协助 MEC 服务器 k 的数据大小比, $\chi^n = 0$ 表示不需要 MEC 服务器 k 的协助。

在多目标强化学习中, 使用 Q 学习并且记录 Q 值来进行问题的优化。每一个状态和行动对应一个 Q 值, 每一步计算并且将计算的结果储存在 Q 矩阵中, 这个值可以被视为一个奖励, 它可以公式化为:

$$Q(s, a) = R(s, a) + \gamma \cdot \max_{a'} Q(s', a') \quad (26)$$

其中, s 和 a 表示当前状态和当前行动; s' 和 a' 表示下一个状态和下一个行动。 γ 为学习参数, 并且有 $0 \leq \gamma \leq 1$ 。如果 γ 趋近于 0, 则表示主要考虑直接奖励; 如果 γ 趋近于 1, 则表示主要考虑未来奖励, 并且每一步都会迭代该值。

对于多目标强化学习, 首先分别计算延迟成本 O_1 和可靠性成本 O_2 的 Q 矩阵。然后, 用延迟成本和可靠性成本的权重向量计算总系统成本的 Q 矩阵, 可以表示为:

$$Q = I_d O_d + I_r O_r \quad (27)$$

其中, Q_d 和 Q_r 分别是延迟成本 O_1 和可靠性成本 O_2 的 Q 矩阵。在每一步 Q 值迭代并且覆盖后, 我们就可以得到最优的带宽分配向量 B 和计算资源分配向量 F 。

5 仿真结果分析

本章将通过仿真分析来验证所提出策略和算法的可行性。

在仿真中,将带宽设置为 $w=10$,给每一个 RSU 都配备一个 MEC 服务器。并使用 WINNER II D2a 信道模型,并且在此模型中充分考虑了实际生活场景中车辆的移动性,具体系统参数如下:车辆数目为 1~4;RSU 数目为 1;本地排队系统服务率 μ^l 为 1GHz;MEC 排队系统服务率 μ^m 为 5GHz;所要处理的任务大小为 200~1000 kbits;车辆和 RSU 之间的距离为 0~1000 m;车辆的传输功率为 0.2 W;RSU 传输功率为 0.3 W;高斯噪声为 10^{-13} W。

将本文所提出的策略和以下 3 个策略进行仿真对比。

1)全部本地计算:车辆生成的任务请求全部在本地进行计算,即车辆自身进行计算处理。

2)全部边缘计算:车辆生成的任务请求全部卸载至 RSU,由 MEC 服务器机型计算处理。

3)随机计算:车辆生成的任务请求随机地选择在本地进行计算处理,或者卸载至 MEC 服务器进行计算处理。

图 3 给出了在车辆和 RSU 距离不同的情况下系统总成本的变化情况。将车辆任务请求数设置为 2,假设任务大小随机生成。从图中可以看到,当车辆与 RSU 的距离小于 890 m 时,所提策略的总成本始终小于其他 3 种策略,这表明所提策略在给定范围内是优于其他 3 种策略的;当距离大于 890 m 时,全部本地计算的系统总成本低于其他 3 种策略,这表明当车辆与 RSU 的距离大于 890 m 时,全部本地计算或许是最优的选择。

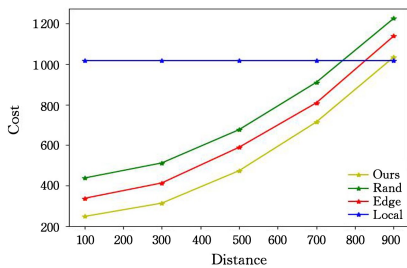


图 3 不同距离下系统成本对比图

Fig. 3 System cost comparison chart at different distances

图 4 给出了在车辆生成的任务请求数不同的情况下,系统总成本的变化情况。将车辆与 RSU 之间的距离设置为 700 m,假设任务大小随机生成。

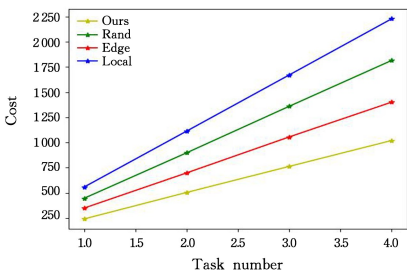


图 4 不同任务数下系统成本对比图

Fig. 4 System cost comparison chart with different number of tasks

从图中可以看到,4 种策略的任务总成本都在随着生成的任务请求数的增加而增加,并且 4 条线的斜率不同,随着任务请求数的不断增大,各策略之间的系统总成本的差距也在不断增大。从图中不难看出,所提的策略对于系统总成本的控制随着任务请求数不断增大的情况下始终优于其他 3 种

策略。例如,当车辆任务请求数量为 3 时,所提出策略的系统总成本为 730,所有边缘计算、随机计算以及所有本地计算的系统总成本分别为 1150,1350 和 1630。因此,通过计算得出所提策略与其他 3 种对比策略相比分别可以节省约 37.2%,46.4% 和 55.3% 的系统总成本,平均节省 46.3% 的系统总成本。

图 5 给出了在车辆数目不同的情况下,系统总成本的变化情况。将车辆任务的数量设置为 3,距离设置为 700 m,假设任务大小随机生成。由图中可以看出,所有策略的系统总成本随着车辆数目的增加而增加。所提策略的系统总成本始终低于其他 3 种策略,且随着车辆数目的不断增加,所提策略的系统总成本与其他 3 种策略的系统总成本的差距越来越大。这表明,在面对实际生活中复杂多变的车联网环境时,有效的资源分配设计是至关重要的,所提策略可以为控制系统成本提供一种很好的解决方法。

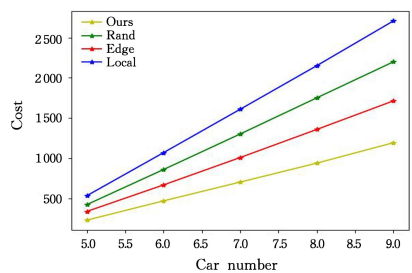


图 5 不同车辆数下系统成本对比图

Fig. 5 System cost comparison chart with different number of vehicles

图 6 给出了 KNN 任务卸载方法和随机卸载在任务请求数量不同的情况下所产生的任务总成本比较。从图中可以看出,KNN 任务卸载方法在控制系统总成本方面始终优于随机卸载方法,并且随着任务数量的不断增加,KNN 任务卸载方法和随机卸载方法之间的差距越来越大,这表明在任务复杂度较高的实际车联网应用场景中,KNN 卸载方法可以很好地控制系统成本。图 7 给出了强化学习算法和粒子群算法在任务请求数不同的情况下所产生的系统总成本比较。从图中可以得出,我们所采用的 Q-Learning 算法对于系统总成本的控制始终优于粒子群算法,并且两条线斜率不同,随着任务请求数量的增加,Q-Learning 算法和粒子群算法对于成本控制的差距也越来越大,这得益于 Q-Learning 算法中对奖励的设置,这使我们可以通过设置对应的系统奖励来得到不同的系统成本要求。

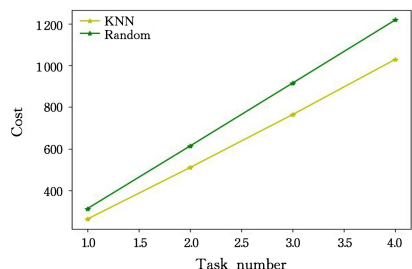


图 6 不同卸载方法下系统成本对比图

Fig. 6 System cost comparison chart with different unloading methods

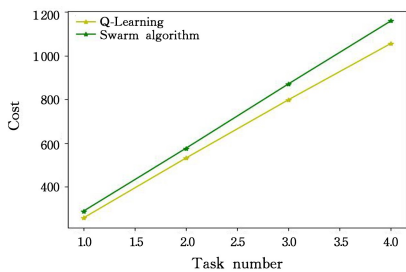


图 7 不同算法下系统总成本对比图

Fig. 7 System cost comparison chart with different algorithms

结束语 本文提出了一种基于任务排队论模型和边缘计算模型相结合的智能通信和计算资源分配的多目标强化学习策略。该策略将通信资源和计算资源的分配相结合,以降低由延迟和可靠性组成的系统总成本。考虑到现实生活中在不同场景下对于可靠性和延迟的要求不同,设置权重系数,利用强化学习算法使该策略能够适应不同场景下对于系统总成本的控制需求,通过仿真分析证明了所提算法在不同场景下在系统成本控制方面有很大的改善,并且降低了任务处理的时延,提高了可靠性。

但是在实际场景中,车载设备所产生的任务请求不会完全符合 M/N/1 队列模型,之后的工作会考虑优化模型中的排队方式,以更好地为车辆提供计算服务。同时会考虑使用深度强化学习来解决更复杂的资源分配问题。

参考文献

- [1] ELAYOUBI S E, BROWN P, DEGHEL M, et al. Radio resource allocation and retransmission schemes for URLLC over 5G networks[J]. *IEEE Journal on Selected Areas in Communications*, 2019, 37(4): 896-904.
- [2] NI Y, HE J, CAI L, et al. Delay analysis and message delivery strategy in hybrid V2I/V2V networks[C]// 2016 IEEE Global Communications Conference (GLOBECOM). IEEE, 2016: 1-6.
- [3] MAO Y, YOU C, ZHANG J, et al. A survey on mobile edge computing: The communication perspective[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(4): 2322-2358.
- [4] 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.
- [5] XIE R, TANG Q, WANG Q, et al. Collaborative vehicular edge computing networks: Architecture design and research challenges[J]. *IEEE Access*, 2019, 7: 178942-178952.
- [6] JIA M, CAO J, LIANG W. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks[J]. *IEEE Transactions on Cloud Computing*, 2015, 5(4): 725-737.
- [7] WANG Y, TAO X, ZHANG X, et al. Cooperative task offloading in three-tier mobile computing networks: An ADMM framework[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(3): 2763-2776.
- [8] YE H, LIANG L, LI G Y, et al. Machine learning for vehicular networks: Recent advances and application examples[J]. *IEEE Vehicular Technology Magazine*, 2018, 13(2): 94-101.
- [9] LUO C, JI J, WANG Q, et al. Channel state information prediction for 5G wireless communications: A deep learning approach

- [J]. *IEEE Transactions on Network Science and Engineering*, 2018, 7(1): 227-236.
- [10] WANG R, YANG H, WANG H, et al. Social overlapping community-aware neighbor discovery for D2D communications[J]. *IEEE Wireless Communications*, 2016, 23(4): 28-34.
- [11] WANG R, LIU H, WANG H, et al. Distributed security architecture based on blockchain for connected health: Architecture, challenges, and approaches[J]. *IEEE Wireless Communications*, 2019, 26(6): 30-36.
- [12] ASHRAF M I, LIU C F, BENNIS M, et al. Dynamic resource allocation for optimized latency and reliability in vehicular networks[J]. *IEEE Access*, 2018, 6: 63843-63858.
- [13] FENG L, LI W, LIN Y, et al. Joint computation offloading and URLLC resource allocation for collaborative MEC assisted cellular-V2X networks[J]. *IEEE Access*, 2020, 8: 24914-24926.
- [14] WANG J, ZHAO L, LIU J, et al. Smart resource allocation for mobile edge computing: A deep reinforcement learning approach [J]. *IEEE Transactions on emerging topics in computing*, 2019, 9(3): 1529-1541.
- [15] WANG Y, LANG P, TIAN D, et al. A game-based computation offloading method in vehicular multiaccess edge computing networks[J]. *IEEE Internet of Things Journal*, 2020, 7(6): 4987-4996.
- [16] KIEN VU T, BENNIS M, DEBBAH M, et al. Ultra-Reliable Communication in 5G mmWave Networks: A Risk-Sensitive Approach[J]. *arXiv: 1802. 03878*, 2018.
- [17] BATEWELA S, LIU C F, BENNIS M, et al. Risk-sensitive task fetching and offloading for vehicular edge computing[J]. *IEEE Communications Letters*, 2019, 24(3): 617-621.
- [18] XU Y, GUI G, GACANIN H, et al. A Survey on Resource Allocation for 5G Heterogeneous Networks: Current Research, Future Trends, and Challenges[J]. *IEEE Communications Surveys and Tutorials*, 2021, 23(2): 668-695.
- [19] SUN W, STRÖM E G, BRÄNNSTRÖM F, et al. Radio resource management for D2D-based V2V communication [J]. *IEEE Transactions on Vehicular Technology*, 2015, 65(8): 6636-6650.
- [20] VU T K, LIU C F, BENNIS M, et al. Ultra-reliable and low latency communication in mmWave-enabled massive MIMO networks[J]. *IEEE Communications Letters*, 2017, 21(9): 2041-2044.
- [21] LIU C F, BENNIS M, DEBBAH M, et al. Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing[J]. *IEEE Transactions on Communications*, 2019, 67(6): 4132-4150.
- [22] KHERRAF N, SHARAFEDDINE S, ASSI C M, et al. Latency and reliability-aware workload assignment in IoT networks with mobile edge clouds [J]. *IEEE Transactions on Network and Service Management*, 2019, 16(4): 1435-1449.
- [23] GHANEM W R, JAMALI V, SUN Y, et al. Resource allocation for multi-user downlink MISO OFDMA-URLLC systems[J]. *IEEE Transactions on Communications*, 2020, 68(11): 7184-7200.
- [24] YANG H, ZHANG K, ZHENG K, et al. Joint frame design and resource allocation for ultra-reliable and low-latency vehicular networks[J]. *IEEE Transactions on Wireless Communications*,

2020,19(5):3607-3622.

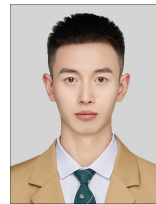
- [25] LIN C C, DENG D J, YAOC C. Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units[J]. *IEEE Internet of Things Journal*, 2017, 5(5): 3692-3700.
- [26] REN S, HE Y. COCA: Online distributed resource management for cost minimization and carbon neutrality in data centers[C]// *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 2013:1-12.
- [27] JIANG Z, MAO S. Energy delay tradeoff in cloud offloading for multi-core mobile devices[J]. *IEEE Access*, 2015, 3: 2306-2316.
- [28] LI T, ZHOU H, LUO H, et al. Modeling software defined satellite networks using queueing theory[C]// *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017:1-6.
- [29] SHEN Y H, CAO J, CHEN X J. Cost optimization problem in communication systems based on queueing theory [J]. *Applied Mathematics*, 2007(S1): 95-98.
- [30] WANG W B, YE Q W, ZHOU Y, et al. A dynamic load balancing algorithm based on queueing theory comprehensive index evaluation [J]. *Telecommunications Science*, 2018, 34(7): 86-91.
- [31] WU W, HU Z, SONG Y. A new method for OPF combining interior point method and filled function method[C]// *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016:1-5.
- [32] YANG Y, MA Y, XIANG W, et al. Joint optimization of energy consumption and packet scheduling for mobile edge computing in cyber-physical networks [J]. *IEEE Access*, 2018, 6: 15576-15586.
- [33] BOBARSHAD H, SHIKH-BAHAEI M. M/M/1 queueing model for adaptive cross-layer error protection in WLANs[C]// *2009 IEEE Wireless Communications and Networking Conference*. IEEE, 2009:1-6.
- [34] MA X, WANG S, ZHANG S, et al. Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing[J]. *IEEE Transactions on Cloud Computing*, 2019, 9(3): 968-980.
- [35] XU Y, XIE H, HU R Q. Max-min beamforming design for he-

terogeneous networks with hardware impairments [J]. *IEEE Communications Letters*, 2020, 25(4): 1328-1332.

- [36] XU Y, GUI G. Optimal resource allocation for wireless powered multi-carrier backscatter communication networks [J]. *IEEE Wireless Communications Letters*, 2020, 9(8): 1191-1195.
- [37] SUTTON R S, BARTO A G. *Reinforcement learning: An introduction*[M]. MIT press, 2018.
- [38] XIA Y, WU L, WANG Z, et al. Cluster-enabled cooperative scheduling based on reinforcement learning for high-mobility vehicular networks[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(11): 12664-12678.
- [39] ZHOU Y, TANG F, KAWAMOTO Y, et al. Reinforcement learning-based radio resource control in 5G vehicular network [J]. *IEEE Wireless Communications Letters*, 2019, 9(5): 611-614.
- [40] LIANG H, ZHANG X, ZHANG J, et al. A novel adaptive resource allocation model based on SMDP and reinforcement learning algorithm in vehicular cloud system[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(10): 10018-10029.
- [41] LEE S, LEE S K. Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information[J]. *IEEE Internet of Things Journal*, 2020, 7(10): 10450-10464.



XUE Jianbin, born in 1973, Ph.D, associate professor. His main research interests include wireless communication theory and technology, mobile edge computing technology, etc.



YU Bowen, born in 1998, postgraduate. His main research interests include edge computing resource allocation and automatic driving of vehicles queuing up on the Internet of Vehicles.