

SDN 和 MEC 架构下 V2X 卸载与资源分配

张海波¹, 王子心¹, 贺晓帆²

(1. 重庆邮电大学通信与信息工程学院, 重庆 400065; 2. 武汉大学电子信息学院, 湖北 武汉 430072)

摘 要: 针对车到万物 (V2X) 场景下复杂的网络状态与海量的计算数据为车载网络带来的时延能耗增加和服务质量下降的严峻问题, 构建了移动边缘计算 (MEC) 和软件定义网络 (SDN) 相结合的车载网络框架。MEC 将云服务下沉至无线网络边缘从而弥补了远程云计算所带来时延抖动, SDN 控制器可从全局角度感知网络信息, 灵活地调度资源, 控制卸载流量。为了进一步降低系统开销, 提出一种联合任务卸载与资源分配机制, 对基于 MEC 的 V2X 卸载与资源分配进行建模, 给出了最优卸载决策、通信和计算资源分配方案。考虑到问题的 NP-hard 属性, 利用 Agglomerative Clustering 匹配初始卸载节点, 并采用 Q-learning 进行资源分配; 将卸载决策建模为完全势博弈, 通过势函数构造证明纳什均衡。仿真结果表明, 相比于其他机制, 该机制能有效降低系统开销。

关键词: 车联网; 移动边缘计算; 软件定义网络; 资源分配

中图分类号: TN929.5

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2020023

V2X offloading and resource allocation under SDN and MEC architecture

ZHANG Haibo¹, WANG Zixin¹, HE Xiaofan²

1. School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

2. School of Electronic Information, Wuhan University, Wuhan 430072, China

Abstract: To address the serious problem of delay and energy consumption increase and service quality degradation caused by complex network status and huge amounts of computing data in the scenario of vehicle-to-everything (V2X), a vehicular network architecture combining mobile edge computing (MEC) and software defined network (SDN) was constructed. MEC sinks cloud serviced to the edge of the wireless network to compensate for the delay fluctuation caused by remote cloud computing. The SDN controller could sense network information from a global perspective, flexibly schedule resources, and control offload traffic. To further reduce the system overhead, a joint task offloading and resource allocation scheme was proposed. By modeling the MEC-based V2X offloading and resource allocation, the optimal offloading decision, communication and computing resource allocation scheme were derived. Considering the NP-hard attribute of the problem, Agglomerative Clustering was used to select the initial offloading node, and Q-learning was used for resource allocation. The offloading decision was modeled as an exact potential game, and the existence of Nash equilibrium was proved by the potential function structure. The simulation results show that, as compared to other mechanisms, the proposed mechanism can effectively reduce the system overhead.

Key words: vehicular network, mobile edge computing, software defined network, resource allocation

收稿日期: 2019-04-11; 修回日期: 2019-12-11

通信作者: 王子心, 377698527@qq.com

基金项目: 国家自然科学基金资助项目 (No.61801065, No.61601071); 长江学者和创新团队发展计划基金资助项目 (No.IRT16R72); 重庆市基础与前沿基金资助项目 (No.cstc2018jcyjAX0463)

Foundation Items: The National Natural Science Foundation of China (No.61801065, No.61601071), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT16R72), The Basic Research and Frontier Exploration Projects in Chongqing (No.cstc2018jcyjAX0463)

1 引言

随着无线通信技术的飞速发展, 联网的车辆可以通过车辆对基础设施 (V2I, vehicle-to-infrastructure) 和车辆对车辆 (V2V, vehicle-to-vehicle) 通信方式形成车载自组网 (VANET, vehicular Ad Hoc network)^[1-4]。在 VANET 中, 车辆均配备车载单元 (OBU, on board unit), 因此具有计算能力和存储能力。然而车辆有限的计算和存储资源难以支持计算密集型的车辆服务, 这种能力限制给 VANET 提出了新的问题。为了解决这一难题, 欧洲电信标准化协会 (ETSI, European Telecommunications Standards Institute) 于 2014 年提出移动边缘计算 (MEC, mobile edge computing), 使云计算服务靠近无线网络边缘, 在移动车辆附近提供基于云的服务^[5-7]。

MEC 服务器部署在无线网络的边缘, 缩短了计算服务器与移动车辆的距离, 使移动车辆可以将计算任务卸载至 MEC 服务器进行计算。MEC 不仅满足了车辆计算能力扩展的要求, 而且弥补了远程云计算导致的长时延的不足^[8]。汽车工业的快速发展让车辆也有了计算和缓存能力, 车辆计算任务既可以卸载到 MEC 服务器进行 V2I 通信, 也可以卸载到相邻车辆进行 V2V 通信。

车辆的移动性和网络拓扑结构的动态变化导致网络稳健性变差。不断发展的 5G 接入技术已经应用到 V2X 通信中, 但是 5G 中的设备对设备 (D2D, device-to-device) 通信技术还不成熟, 目前最可行的方案是结合不同的访问技术来保持可伸缩性和灵活性^[9-11]。幸运的是, 软件定义网络 (SDN, software defined network) 具有集成不同访问技术的潜力^[12]。SDN 使数据平面与控制平面解耦, 其可编程性使网络更加灵活, 可以更好地收集车辆动态信息, 应用程序可以选择合适的无线接入接口来传输数据^[13]。MEC 异构车载网络体系结构中引入的 SDN 控制器允许在同一网络体系结构下集成多个网络协议、标准以及灵活管理网络资源。

目前, 已存在许多关于 MEC 的研究。Zhang 等^[14]为了降低计算卸载的时延和传输成本, 提出了一种基于云的 MEC 车载网络卸载框架, 减少了计算任务执行的时间消耗和车辆移动性带来的影响。Li 等^[15]提出了一种基于 Ad Hoc 的多设备任务调度策略。Kan 等^[16]通过将计算密集型任务转移到 MEC

服务器来提升服务质量 (QoS, quality of service), 将优化问题表述为一个成本最小化问题, 并提出了一种启发式算法来解决该问题。You 等^[17]考虑了多用户 MEC 系统中的资源分配问题, 在时延约束下使系统能耗最小化。Deng 等^[18]考虑一个多小区 MEC 场景, 并提出了自适应顺序卸载博弈方法来解决卸载问题。Zhang 等^[19]利用拍卖理论提出了一种资源分配机制。Zheng 等^[20]考虑动态环境下的多用户计算卸载问题, 将移动用户卸载决策过程建模为随机博弈。Mao 等^[21]为了降低执行时延和计算能耗, 对具有多个独立任务的 MEC 系统联合优化任务卸载调度和传输功率分配。但是以上文献未将 V2I 模式扩展至 V2V 模式, 同时忽略了复杂的 V2X 接入方式和初始卸载节点的合理选择对于系统性能的影响。

SDN 代表了一种新兴的网络范式, 它具有整合 VANET 的潜力^[22]。Liu 等^[10]将 SDN 引入 MEC 异构车载网络, 该网络能够在 V2X 通信中提供所需的数据速率和可靠性, 同时满足应用的具体需求, 保持良好的可扩展性和响应性。Huang 等^[23]将 SDN 与 MEC 相结合, 提出了一种可编程、灵活、可控的 5G 软件定义车载网络架构。Huang 等^[24]针对复杂的 VANET V2V 卸载问题, 设计了基于 MEC 和 SDN 的卸载框架, 为车辆寻找最佳卸载路径。李萌等^[25]提出了一种基于移动边缘计算的新型网络架构, 将软件定义的车载网络中的容错数据传输和计算节点选择过程建模为部分可观测马尔可夫决策过程, 使系统成本最小化。然而, 以上场景并未进行有效的资源分配, 并且 V2I 通信模式下的 MEC 服务器和 V2V 通信模式下的车辆所能提供的计算资源都是有限的。因此, 在移动车辆任务交付过程中, 仍然存在与任务计算相关的挑战, 即如何选择恰当的交付节点; 如何在资源有限的情况下合理分配资源, 使车辆以最小成本完成任务计算。

针对以上问题, 本文构建了一种基于 SDN 辅助的 MEC 车载网络架构, 提出一种联合任务卸载与资源分配 (JTORA, joint task offloading and resource allocation) 机制, 联合优化车载网络中的卸载策略、上行子信道分配、传输功率分配和计算资源调度。将 SDN 扩展到车辆网络, SDN 控制器可在全局视野下感知网络状态、收集设备信息, 同时使车辆可以选择合适的接入方式进行任务卸载。将优化问题转化为最优卸载决策、传输功率分配、子

信道分配和计算资源分配问题。优化目标是在约束条件下使系统开销最小化。由于优化问题的 NP-hard 性质, 本文将原优化问题转化为 3 个子算法的相互迭代, 进行有效求解。通过分布式势博弈对卸载决策进行建模和分析, 并且证明了纳什均衡的存在性。为了在初始阶段得到一个较为满意的匹配对象, 采用 Agglomerative Clustering 匹配初始卸载节点。利用 Q-learning 子算法对传输功率、上行传输信道以及计算资源进行合理分配。仿真结果表明, 该机制能有效降低系统总开销。

2 基于 SDN 与 MEC 的系统模型和优化问题制定

2.1 系统模型

本文构建了基于 SDN 辅助的 MEC 车载网络模型, 网络模型架构如图 1 所示。在这个模型中, 道路一侧部署路侧单元 (RSU, road side unit), 每个 RSU 都配备一个 MEC 服务器。MEC 服务器中的 SDN 控制模块将控制平面与数据平面解耦, 基于 SDN 的车载网络可以实现 VANET 与 5G 蜂窝网络的集成, 同时 SDN 控制器可以收集移动设备和 MEC 的信息, 从全局视野感知网络状态。该网络架构分为三层, 具体如下。在用户数据区域, 车辆可以通过 RSU 将计算任务卸载至 MEC 服务器进行 V2I 模式通信, 也可以将计算任务卸载至相邻车辆进行 V2V 模式通信。道路上有 N 个服从泊松分布的车辆, 表示为 $\mathcal{N}=\{v_1, v_2, v_3, \dots, v_N\}$ 。由于车辆和 MEC 服务器均可提供计算服务, 因此将它们统一定义为服务节点, 表示为 $\mathcal{G}=\{g_1, g_2, g_3, \dots, g_M\}$ 。在网络控制区域部署 SDN 控制器, 可灵活管理网络资源, 全局掌握网络信息, 集成多种网络协议和标准, 如车载网络、物联网网络和数据网络^[25]。网络应用区域由多种网络应用填充而成, 通常涉及数据和资源的交换, 网络设计的目标就是为了满足这些应用的要求。

为了有效地复用频谱, V2I 模式和 V2V 模式在同一频段工作。系统总频谱被分成 K 个正交子信道, 表示为 $\mathcal{K}=\{1, 2, 3, \dots, K\}$, 每个子信道带宽为 B Hz。假设每个车辆都有一个计算密集型的任务需要完成, 且每个车辆最多分配一个子信道。车辆 v_i 的计算任务可以被描述为 $\mathcal{T}_i=\{d_i, d_i^{\text{out}}, s_i, t_i^{\text{max}}\}$, $i \in N$, 其中, d_i 表示计算任务 \mathcal{T}_i 的输入数据大小, 包括程序

代码和输入参数; d_i^{out} 表示任务计算完成后的输出大小; s_i 表示完成计算任务 \mathcal{T}_i 所需的 CPU 周期数; t_i^{max} 计算任务 \mathcal{T}_i 所能容忍的最大时延。车辆卸载策略集合表示为 $\mathcal{A}=\{a_1, a_2, a_3, \dots, a_N\}$, 若 $a_i=1$, 则表示 v_i 将任务卸载到服务节点进行计算; 若 $a_i=0$, 则表示 v_i 在本地执行计算任务。类似于之前在移动边缘计算领域的许多工作, 本文考虑一个静态网络场景, 即车辆集合 \mathcal{N} 以及网络拓扑结构在计算卸载期间保持不变^[26]。

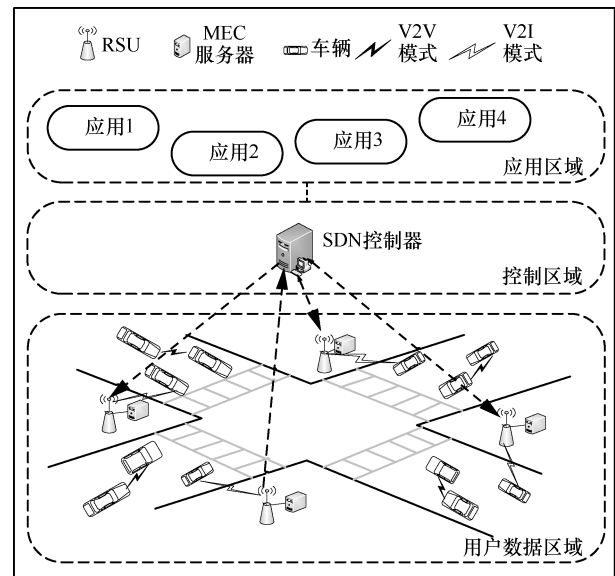


图 1 基于 SDN 辅助的 MEC 车载网络模型架构

2.2 通信模型

在基于 SDN 辅助的 MEC 车载网络架构中, 用户数据区域的每个车辆 v_i 均可将其计算任务 \mathcal{T}_i 卸载到 MEC 服务器处理, 也可卸载到邻近车辆进行计算, 或者在本地执行其计算任务。为了提高频谱利用率, V2V 通信模式复用 V2I 模式的上行传输信道。因此, v_i 在子信道 k 上的信噪比 (SINR, signal to interference plus noise ratio) 可以表示为

$$\text{SINR}_{i,j}^k = \frac{p_i h_{i,j}^k}{N_0 + I_i^C + I_i^D} \quad (1)$$

其中, p_i 表示 v_i 的上行传输发送功率, $h_{i,j}^k$ 表示 v_i 与服务节点 g_j 在子信道 k 上的信道增益, N_0 表示白高斯噪声功率, I_i^C 表示 V2I 用户对 v_i 的干扰, I_i^D 表示 V2V 用户对 v_i 的干扰。

为了更好地描述信道分配状况, 引入信道连接矩阵 \mathbf{C} 与 \mathbf{C}' , 分别为 $N_C \times K$ 和 $N_D \times K$ 的二元变量

矩阵, 其中 N_C 和 N_D 分别代表 V2I 模式和 V2V 模式的车辆数量, 其二元变量 $c_{x,k}$ 和 $c'_{y,k}$ 表示子信道 k 是否被分配给用户 x 或 y 。若 $c_{x,k}=1$ ($c'_{y,k}=1$), 则表示子信道 k 被分配给用户 x (y) 以进行上行数据传输; 若 $c_{x,k}=0$ ($c'_{y,k}=0$), 则表示子信道 k 未被分配给用户 x (y)。

在 V2X 车载网络中, 干扰环境复杂, 必然存在跨层干扰。对 v_i 来说, 若选择的服务节点为 MEC 服务器, 则 I_i^C 和 I_i^D 可以分别表示为 $I_i^C = \sum_{x=1, x \neq i}^{N_C} c_{x,k} p_x h_{x,j}^k$ 和 $I_i^D = \sum_{y=1}^{N_D} c'_{y,k} p_y h_{y,j}^k$; 若选择的服务节点为车辆, 则进行 V2V 模式通信, I_i^C 和 I_i^D 可以分别表示为 $I_i^C = \sum_{x=1}^{N_C} c_{x,k} p_x h_{x,j}^k$ 和 $I_i^D = \sum_{y=1, y \neq i}^{N_D} c'_{y,k} p_y h_{y,j}^k$ 。

根据上述信息, 车辆 v_i 的上行传输速率表示为

$$r_{i,j} = B \ln(1 + \text{SINR}_{i,j}^k) \quad (2)$$

2.3 计算模型

假设每个车辆 v_i 均有一个计算任务 T_i 需要完成, 并且 v_i 可以通过文献[27-28]中的方法来获得任务信息。任务卸载计算或本地执行均会带来时延和能量消耗, 接下来, 将具体讨论在不同任务处理方式下的时延和能耗计算模型。

1) 本地计算

车辆 v_i 的计算能力为 f_i^l , 并且不同的车辆拥有不同的计算能力, 则本地执行时延为

$$t_i^{\text{local}} = \frac{S_i}{f_i^l} \quad (3)$$

根据文献[29], 计算能量消耗为

$$e_i^{\text{local}} = \kappa S_i (f_i^l)^2 \quad (4)$$

其中, κ 为能耗参数^[30-31], 设置为 $\kappa=10^{-28}$ 。

结合式(3)和式(4), 将本地执行方案的总开销定义为

$$Z_i^{\text{local}} = \alpha t_i^{\text{local}} + \beta e_i^{\text{local}} \quad (5)$$

其中, α 与 β 分别表示时延和能耗的权重因子, 且满足 $\alpha + \beta = 1, 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$ 。

2) 卸载计算

当车辆 v_i 选择将其计算任务卸载至服务节点 g_j 进行计算时, 将产生传输时延、传输能耗以及计

算时延。基于此, 上行传输时延为

$$t_i^{\text{trans}} = \frac{d_i}{r_{i,j}} \quad (6)$$

传输能耗为

$$e_i^{\text{off}} = p_i t_i^{\text{trans}} \quad (7)$$

卸载完成以后, 服务节点 g_j 将进行任务计算。

计算过程所消耗的时间为

$$t_i^{\text{exe}} = \frac{S_i}{f_j^i} \quad (8)$$

其中, f_j^i 表示服务节点 g_j 分配给任务请求车辆 v_i 的计算资源 (表征为每秒 CPU 周期)。

回程时延^[32]为

$$t_i^{\text{down}} = \frac{d_i^{\text{out}}}{r_{i,j}} \quad (9)$$

因此, 卸载计算方案的总开销为

$$Z_i^{\text{off}} = \alpha t_i^{\text{off}} + \beta e_i^{\text{off}} = \alpha (t_i^{\text{trans}} + t_i^{\text{exe}} + t_i^{\text{down}}) + \beta p_i t_i^{\text{trans}} \quad (10)$$

为了简单起见, 本文省略了服务节点的计算能耗^[33]。

2.4 优化问题制定

对于车辆 v_i , 其计算总开销为

$$Z_i = (1 - a_i) Z_i^{\text{local}} + a_i Z_i^{\text{off}} \quad (11)$$

本文的优化目标为在 SDN 控制器中以系统效益最优为目标, 获得最优卸载决策 \mathcal{A}^* 、传输功率分配 \mathcal{P}^* 、上行信道分配 \mathcal{C}^* 和计算资源分配 \mathcal{F}^* , 使系统开销最小。因此, 将优化问题制定为

$$\min_{\mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{F}} Z(\mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{F}) = \sum_{i=1}^N (1 - a_i) Z_i^{\text{local}} + a_i Z_i^{\text{off}} =$$

$$\sum_{i=1}^N (1 - a_i) \left[\alpha \frac{S_i}{f_i^l} + \beta \kappa S_i (f_i^l)^2 \right] +$$

$$a_i \left[\alpha \left(\frac{d_i}{r_{i,j}} + \frac{S_i}{f_j^i} + \frac{d_i^{\text{out}}}{r_{i,j}} \right) + \beta p_i \frac{d_i}{r_{i,j}} \right]$$

$$\text{s.t. C1: } a_i \in \{0, 1\}, \forall i \in N$$

$$\text{C2: } c_{i,k} \in \{0, 1\}, \forall i \in N, k \in \mathcal{K}$$

$$\text{C3: } c'_{i,k} \in \{0, 1\}, \forall i \in N, k \in \mathcal{K}$$

$$\text{C4: } 0 < p_i < p_{\max}, \forall i \in N$$

$$\text{C5: } 0 \leq f_j^i \leq a_i F_j^{\max}, \forall i \in N$$

$$\text{C6: } \sum_{i \in N} a_i f_j^i \leq F_j^{\max}, \forall i \in N, j \in M$$

$$\begin{aligned}
 \text{C7: } & \sum_{k=1}^K c_{i,k} \in \{0,1\}, \forall i \in N, k \in K \\
 \text{C8: } & \sum_{k=1}^K c'_{i,k} \in \{0,1\}, \forall i \in N, k \in K \\
 \text{C9: } & (1-a_i) \frac{s_i}{f'_i} + a_i \left(\frac{d_i}{r_{ij}} + \frac{s_i}{f'_j} + \frac{d_i^{\text{out}}}{r_{ij}} \right) \leq t_{\max}, \forall i \in N
 \end{aligned} \tag{12}$$

其中, $\mathcal{A}=\{a_1, a_2, a_3, \dots, a_N\}$ 为卸载决策集合, $\mathcal{C}=\{c_1, c_2, c_3, \dots, c_N\}$ 为信道分配集合, $\mathcal{P}=\{p_1, p_2, p_3, \dots, p_N\}$ 为传输功率分配集合, $\mathcal{F}=\{f_1, f_2, f_3, \dots, f_N\}$ 为计算资源分配方案, f'_i 为服务节点 g_j 分配给车辆 v_i 的计算资源。约束条件 C1 表示卸载决策, 意味着车辆可以将其计算任务卸载给服务节点进行计算或在本地执行任务。C2 和 C3 保证信道分配矩阵是二进制变量。C4 表示上行传输功率的变化范围。C5 和 C6 表示分配给卸载车辆的计算资源之和为非负数值, 且不能超过服务节点的最大计算能力。C7 和 C8 表明任务请求车辆只能分配一个上行子信道。C9 表示时延控制, 即无论是本地执行还是卸载到服务节点进行计算, 总时延不能超过任务所能容忍的最大时延。

整数约束 C1~C3 使式(12)问题成为一个混合整数非线性规划。由于该优化问题具有 NP-hard 性质和非凸性, 求解困难且不切实际。因此, 本文提出了一个联合任务卸载和资源分配机制。

3 卸载策略和资源分配方案

3.1 基于 Agglomerative Clustering 层次聚类的通信模式选择

层次聚类作为机器学习中非监督学习的一类方法, 可以灵活控制不同层次的聚类粒度, 具有强聚类能力, 能满足不同的应用场景。Agglomerative Clustering 层次聚类^[34]算法能够保证距离近的对象被聚类到一个簇中, 采用“自底向上”聚类的思路。在本文系统模型下结合车辆任务时延要求进行聚类, 将其时延要求作为特征输入, 算法思路具体到本文模型如下。

输入 车辆任务时延要求数据集 $\text{TD}=\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, 聚类簇距离度量函数, 聚类簇数量 $J=2$

输出 簇划分 $H=\{h_{V2I}, h_{V2V}\}$

1) 将数据集中的每一个对象生成一个簇, 得到簇列表 $H=\{h_1, h_2, \dots, h_N\}$, 其中, 每个簇只包含一

个数据对象, 即 $h_i=\{\mathbf{x}_i\}$ 。

2) 重复以下步骤, 直到得到 J 个数据簇。

① 根据距离度量函数计算其距离, 在簇列表 H 中找出距离最近的 2 个簇 $\min\{\text{TD}(h_i, h_j)\}$ 。

② 合并簇 h_i 与 h_j , 形成新的簇 $h(i+j)$ 。

③ 在簇列表 H 中删除簇 h_i 与 h_j , 并添加簇 $h(i+j)$ 。

针对步骤 2) 中①的距离计算, 采用平均连锁进行计算

$$d_{\text{avg}}(h_i, h_j) = \frac{1}{|h_i||h_j|} \sum_{\bar{\mathbf{x}}_i \in h_i} \sum_{\bar{\mathbf{x}}_j \in h_j} d(\mathbf{x}_i, \mathbf{x}_j) \tag{13}$$

式(13)表示 2 个簇之间任意两点之间距离的平均值, 该方式可以有效地排除噪点的影响。

3) 通过 Agglomerative Clustering 层次聚类, 将道路上的任务请求车辆进行模式分割, 确定其以 V2I 模式还是 V2V 模式进行通信。若为 V2I 模式, 则选择本小区内 RSU 进行通信; 若为 V2V 模式, 则根据距离选择就近车辆为服务节点。通过模式选择, 在初始阶段即为车辆选择较为合适的服务节点匹配。

3.2 卸载博弈建模

对于一个势博弈存在性质, 任何异步更新过程必须是有限的, 并收敛于纳什均衡^[13]。将 JTORA 机制建模为势博弈, 表示为 $G=\{\mathcal{N}, (a_i)_{i \in N}, (u_i)_{i \in N}\}$, 其中, a_i 为 v_i 的卸载决策, u_i 为 v_i 的代价函数。接下来将详细介绍 JTORA 博弈模型。

玩家。每个任务请求车辆均为一个参与者, 存在 N 个参与者竞争有限资源, 每个参与者可以将任务卸载至服务节点进行任务计算或者本地计算。

策略。 $a_i \in \{0,1\}$ 为 v_i 的卸载决策, \mathcal{A} 为所有任务请求车辆的卸载决策集合, $a_i=1$ 表示 v_i 将卸载其计算任务至选定的卸载节点进行计算, $a_i=0$ 表示 v_i 将在本地执行其任务。

代价函数。当 v_i 选择其卸载决策为 a_i 时的代价函数表示为 $u_i(a_i, a_{-i})$, 其中 a_{-i} 代表除 v_i 以外的任务请求车辆的卸载决策集合。每辆车都希望通过适当的卸载策略选择来最小化开销, 如式(14)所示。

$$\min_{a_i \in \{0,1\}} u_i(a_i, a_{-i}) = (1-a_i)Z_i^{\text{local}} + a_iZ_i^{\text{off}} \tag{14}$$

势博弈收敛于纳什均衡, 表示没有任何玩家有偏离当前策略的动机。具体描述介绍如下。

定义 1 在博弈 $G = \{a_1, a_2, \dots, a_N : u_1, u_2, \dots, u_N\}$ 中, 存在一个策略集合 $\mathcal{A}^* = \{a_1^*, a_2^*, \dots, a_N^*\}$ 使所有参与者均没有改变当前决策的动机, 则称 \mathcal{A}^* 为 G 的一个纳什均衡, 即

$$u_i(a_i^*, a_{-i}^*) \leq u_i(a_i, a_{-i}^*) \quad a_i \in \{0, 1\}, i \in N \quad (15)$$

纳什均衡具有显著的自稳定性, 这一特性使处于平衡状态的玩家能够得到一个双方都满意的解, 因此没有玩家有偏离平衡点的动机^[35]。纳什均衡对于非合作问题的解决非常重要。因为每一个任务请求车辆都满足自身理性和利己性, 因此均以最小化自身代价为目标。

卸载策略制定以后, Q-learning 将进行通信和计算资源分配, 以降低系统开销。

3.3 基于 Q-learning 的通信与计算资源分配

Q-learning 是一种典型的强化学习方法, 具有收敛性, 能够使代理在给定的环境下学习最优策略^[36]。将每个请求车辆视作一个代理, 状态、行为、奖励设置如下。

状态。系统状态包括 3 个部分 $\mathbf{s} = (t_c, h_c, f_c)$ 。 t_c 定义为当前系统总开销, 即 $t_c = Z_{\text{all}}$; h_c 定义为当前信道分配状态; f_c 定义为当前服务节点的有效计算资源。

动作。动作包括 3 个部分, 表示为信道、功率和计算资源矩阵 $\mathbf{at} = [\text{ch}, p, \text{cr}]$ 。

回报。在 t 时刻, 代理在执行每一个可能的动作 \mathbf{at} 后, 在一定的状态 \mathbf{s} 下会得到一个奖励 $r_i^t = \frac{\hat{Z}_{i,t}^{\text{off}}}{Z_{i,t}^{\text{off}}}$, 其中, $Z_{i,t}^{\text{off}}$ 为卸载车辆 v_i 在 t 迭代时刻的卸载开销, $\hat{Z}_{i,t}^{\text{off}}$ 为卸载车辆 v_i 在 t 迭代时刻的无干扰卸载开销。

Q-learning 采用 Q 函数作为评价函数, 并将其定义为状态-动作值函数, 利用 Bellman 最优性方程给出最大化总期望收益的最优策略 π^* ^[36-37]。

$$Q^\pi(\mathbf{s}, \mathbf{at}) = \mathbb{E}^\pi \left[r(\mathbf{s}' | \mathbf{s}, \mathbf{at}) + \gamma \max_{\mathbf{at}'} Q(\mathbf{s}', \mathbf{at}') \mid \mathbf{s}_0 = \mathbf{s}, \mathbf{at}_0 = \mathbf{at} \right] \quad (16)$$

其中, $\gamma (0 < \gamma < 1)$ 为折扣因子, 反映了未来收益相对于当前收益的重要性。若 γ 倾向于 0, 则代理倾向于考虑即时回报; 若 γ 倾向于 1, 则代理更关注于未来的收益。

其最优策略表示为

$$\pi^* = \arg \max_{\mathbf{at}} Q^\pi(\mathbf{s}, \mathbf{at}) \quad (17)$$

在迭代过程中, 需要对 Q 值进行更新, 但是无法遍历所有状态-动作对, 因此通过学习速率 μ 对有限样本进行操作。则 Q 值的更新式可以表示为

$$\hat{Q}_{t+1}(\mathbf{s}, \mathbf{at}) \leftarrow (1 - \mu) \hat{Q}_t(\mathbf{s}, \mathbf{at}) + \mu \left(r + \gamma \max_{\mathbf{at}'} \hat{Q}_t(\mathbf{s}', \mathbf{at}') \right) \quad (18)$$

其中, $\mu (0 \leq \mu \leq 1)$ 为学习速率, 代表了当前学习知识对之前学习知识的影响。

在迭代过程中本文采用 ϵ -greedy 算法来选择动作^[36]。其中, 车辆 v_i 的行为表示为信道、功率和计算资源矩阵, 表示为

$$\mathbf{at}_i = [\text{ch}_i, p_i, \text{cr}_i] \quad (19)$$

动作轮廓为

$$\mathbf{at}_i = \begin{cases} \arg \max_{i \in N} \hat{Q}(s_i, \mathbf{at}_i), 1 - \epsilon \\ \text{random}, \epsilon \end{cases} \quad (20)$$

ϵ -greedy 机制中每一个动作都有非零的概率被选择, 随机选择行为 (random) 的概率是 ϵ , 选择与最大 Q 值相关的动作的概率则是 $1 - \epsilon$ 。

Q-learning 算法详细描述如下。

输入 学习速率 μ , 折扣因子 γ , 贪婪系数 ϵ , 最大收敛次数 E , 卸载车辆集合 \hat{N} , 服务节点状态信息

输出 车辆开销 Z_i^{off}

- 1) 初始化 Q 矩阵, 迭代次数 $t = 0$, 开始迭代;
- 2) 根据式(20)选择动作;
- 3) 根据式(18)更新 Q 矩阵;
- 4) 将 \mathbf{s}' 更新为 \mathbf{s} ;
- 5) 更新迭代次数 $t = t + 1$;
- 6) 重复步骤 2)~步骤 5), 直到达到最大收敛次数 E 。

3.4 纳什均衡证明

定义 2 若一个势博弈 $G = \{\mathcal{N}, (a_i)_{i \in N}, (u_i)_{i \in N}\}$ 遵从势函数 $\zeta(\mathcal{A})$, 满足式(21), 则此博弈为完全势博弈。

$$u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) = \zeta_i(a_i, a_{-i}) - \zeta_i(a'_i, a_{-i}) \quad (21)$$

$i \in N, a_i, a'_i \in \mathcal{A}, a_{-i} \in \mathcal{A} \setminus a_i$

定理 1 JTORA 博弈模型是一个完全势博弈, 并且总是存在纳什均衡, 满足有限改进性质。其遵从的势函数 $\zeta(\mathcal{A})$ 为

$$\zeta(\mathcal{A})=(1-a_i) \cdot \left\{ \sum_{i' \neq i}^N \left[\frac{d_{i'}(\alpha + \beta p_{i'}) + \alpha d_{i'}^{\text{out}}}{\text{Blb} \left(1 + \frac{p_{i'} h_{i'}}{N_0 + I_{i'}^C + I_{i'}^D + p_i h_i} \right)} + \alpha t_{i'}^{\text{exe}} \right] + Z_i^{\text{local}} \right\} + a_i \sum_{i=1}^N Z_i^{\text{off}} \quad (22)$$

证明 根据式(22), 可以得到

$$\zeta(0, a_{-i}) = \sum_{i' \neq i}^N \left[\frac{d_{i'}(\alpha + \beta p_{i'}) + \alpha d_{i'}^{\text{out}}}{\text{Blb} \left(1 + \frac{p_{i'} h_{i'}}{N_0 + I_{i'}^C + I_{i'}^D + p_i h_i} \right)} + \alpha t_{i'}^{\text{exe}} \right] + Z_i^{\text{local}} \quad (23)$$

$$\zeta(1, a_{-i}) = \sum_{i=1}^N Z_i^{\text{off}} = Z_i^{\text{off}} + \sum_{i' \neq i}^N Z_{i'}^{\text{off}} = Z_i^{\text{off}} + \sum_{i' \neq i}^N \left[\frac{d_{i'}(\alpha + \beta p_{i'}) + \alpha d_{i'}^{\text{out}}}{\text{Blb} \left(1 + \frac{p_{i'} h_{i'}}{N_0 + I_{i'}^C + I_{i'}^D + p_i h_i} \right)} + \alpha t_{i'}^{\text{exe}} \right] \quad (24)$$

由式(14)可得

$$u(0, a_{-i}) = Z_i^{\text{local}} \quad (25)$$

$$u(1, a_{-i}) = Z_i^{\text{off}} \quad (26)$$

因此, 联系式(25)和式(26)可得

$$u(0, a_{-i}) - u(1, a_{-i}) = Z_i^{\text{local}} - Z_i^{\text{off}} \quad (27)$$

组合式(23)与式(24)可知

$$\zeta(0, a_{-i}) - \zeta(1, a_{-i}) = \left(\sum_{i' \neq i}^N \left[\frac{d_{i'}(\alpha + \beta p_{i'}) + \alpha d_{i'}^{\text{out}}}{\text{Blb} \left(1 + \frac{p_{i'} h_{i'}}{N_0 + I_{i'}^C + I_{i'}^D + p_i h_i} \right)} + \alpha t_{i'}^{\text{exe}} \right] + Z_i^{\text{local}} \right) - \left(Z_i^{\text{off}} + \sum_{i' \neq i}^N \left[\frac{d_{i'}(\alpha + \beta p_{i'}) + \alpha d_{i'}^{\text{out}}}{\text{Blb} \left(1 + \frac{p_{i'} h_{i'}}{N_0 + I_{i'}^C + I_{i'}^D + p_i h_i} \right)} + \alpha t_{i'}^{\text{exe}} \right] \right) = Z_i^{\text{local}} - Z_i^{\text{off}} \quad (28)$$

因此, 根据式(27)和式(28)可得

$$u(0, a_{-i}) - u(1, a_{-i}) = \zeta(0, a_{-i}) - \zeta(1, a_{-i}) \quad (29)$$

$$u(1, a_{-i}) - u(0, a_{-i}) = \zeta(1, a_{-i}) - \zeta(0, a_{-i}) \quad (30)$$

基于上述信息, JTORA 博弈模型是一个完全势博弈, 满足式(22)所示势函数, 因此至少存在一个纯策略纳什均衡并具有有限改进性质。这意味着车辆卸载决策可以达到一个有效的平衡, 即任何车辆单方面改变现有卸载决策不会使自身开销进一步降低。

3.5 联合优化机制总结

JTORA 机制将初始卸载策略设置为所有车辆将其计算任务卸载到每个请求车辆使用 Aggregative Clustering 选择的服务节点。计算本地计算代价, 通过 Q-learning 得到相应的卸载代价。比较本地计算成本和卸载成本, 修改卸载策略 \mathcal{A} 。然后进入循环阶段, 逐个检查卸载策略中的零元素, 重新分配最优资源, 直到所有的车辆没有改变现有决策的动机。由于车辆满足自身理性和利己主义, 只希望降低自身任务计算的成本, 因此每辆车通过比较自身的本地计算成本和卸载成本来修改卸载决策。具体算法描述如下。

输入 任务请求车辆数量 N , 服务节点数量 M , 初始卸载决策集合 \mathcal{A}_0 及 $\mathcal{T}_i, L, b, \kappa, \alpha, \beta, \mu, \gamma$

输出 $\mathcal{A}^*, \mathcal{C}^*, \mathcal{P}^*, \mathcal{F}^*, \mathcal{Z}^*$

- 1) 初始化: 全卸载; $L=1$;
 - 2) 针对每个车辆进行 Aggregative Clustering 聚类, 以确定通信模式, 计算 z_i^{local} , 以及通过 Q-learning 获得 z_i^{off} ;
 - 3) 对比 z_i^{local} 与 z_i^{off} , 更新车辆卸载决策 \mathcal{A} ;
 - 4) 判断 \mathcal{A} 是否与 \mathcal{A}_0 相同, 若不相同, 则执行以下步骤; 若相同, 则算法结束;
 - 5) 更新卸载决策 $\mathcal{A}_L \leftarrow \mathcal{A}_0$;
 - 6) 针对每个车辆, 使其卸载决策 $a_i = 1$, 然后更新 \mathcal{A}_L , 以及通过 Q-learning 获得 z_i^{off} ;
 - 7) 对比 z_i^{local} 与 z_i^{off} , 更新车辆卸载决策 \mathcal{A}_L , $L=L+1$;
 - 8) 判断 \mathcal{A}_L 是否与 \mathcal{A}_{L-1} 相同, 若不相同, 则重复步骤 5)~步骤 7); 若相同, 则算法结束;
- JTORA 机制将优化问题转化为 3 个子算法的相互迭代进行有效求解。在步骤 2)中, Aggregative Clustering 聚类需要计算数据点之间的相互距离, 时间复杂度为 $O(N^2)$, 之后根据距离度量函数更新迭代, 因此聚类

所需时间复杂度为 $O(N^2 \log N)$ 。本地计算与卸载计算均需 N 次计算, 因此步骤 2)和步骤 3)的时间复杂度为 $O(2N + EN)$; 步骤 5)~步骤 8)的时间复杂度为 $O(L(2N + E\hat{N}))$ 。因此, JTORA 机制的总体时间复杂度为 $O(2N(L+1) + E(L\hat{N}+N) + N^2 \log N)$ 。

4 仿真分析

本节通过 Matlab 仿真平台对 SDN 和 MEC 环境下的 V2X 卸载与资源分配算法进行验证, 给出了仿真结果, 并评估了所提出机制的性能。本节在 IEEE 802.11p 车辆网络场景标准和 MEC 白皮书的背景下展开, 采用 3GPP 标准化中提出的信道增益模型^[33]。考虑路边有 2 个小区, 每个小区都配置了 RSU 和 MEC 服务器。详细仿真参数如表 1 所示。

表 1 仿真参数

参数	数值
任务计算大小 d_i /KB	400 ~ 500
计算任务所需 CPU 周期 s_i /Megacycle	1 200 ~ 2 000
上行传输信道数量 K	10
单小区车辆数目	18
信道总带宽/MHz	20
高斯白噪声功率 N_0 /dBm	-60
权重因子设置 $\alpha=\beta$	0.5
车辆计算能力 F_i /GHz	0.7 ~ 1
MEC 服务器计算能力 f_j /GHz	4
车辆最大发射功率 p_{max} /dBm	20
RSU 的覆盖半径/m	250

为了评估所提机制的性能, 将 JTORA 与其他几种机制进行比较, 例如所有本地计算机制(ALCM, all local computing mechanism)、全卸载计算机制(AOCM, all offloading computing mechanism)、无聚类的 Q 学习资源分配(QRNC, Q-learning resource allocation without clustering)机制、文献[33]中联合任务计算卸载与干扰管理(JCOIM, joint computation offloading and interference management)机制、初始节点筛选但随机资源分配 (ARRA, agglomerative clustering with random resource allocation)机制、文献[18]中的多用户计算卸载 (MCO, multi-user computation offloading) 机制。

图 2 为基于 Aggregative Clustering 的通信模式选择, 将车辆进行层次聚类, 以确定其为 V2I 模式

通信还是 V2V 模式通信。由图 2 可知, 距离 RSU 较远的车辆由于其时延需求更倾向于进行 V2V 模式通信, 而距离 RSU 距离较近的用户, 根据其计算能力的需求选择 V2I 通信方式, 部分车辆由于其对时延要求高于计算能力需求, 因此选择 V2V 通信方式, 如图 2(b)中虚线圈出区域所示。

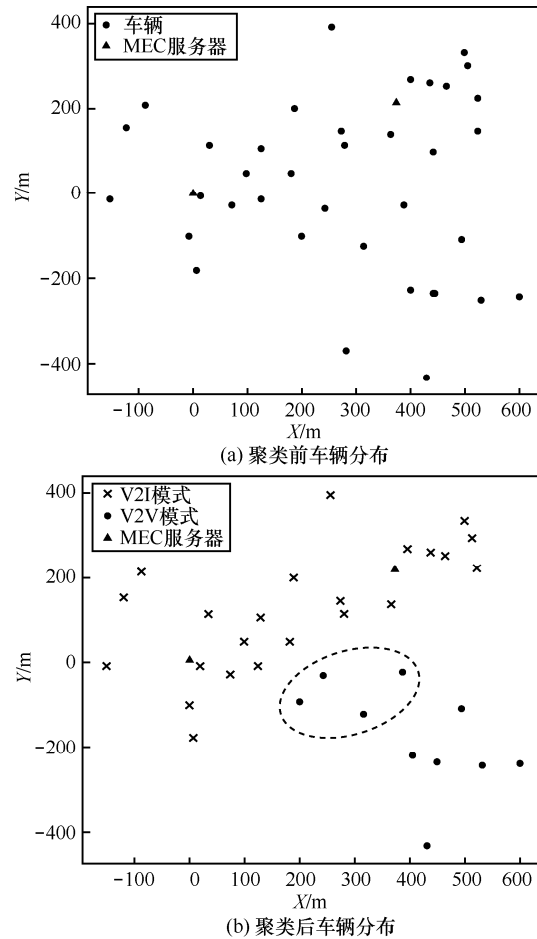


图 2 基于 Aggregative Clustering 的通信模式选择

图 3 和图 4 分别为任务大小和车辆数目对于总开销的影响。由图 3 和图 4 可知, 数据大小和车辆数目的增加, 使系统总开销相应增长。本文所采用的 JTORA 机制在降低系统开销方面明显优于其他策略, 随着数据量和车辆数目的增加, 这些机制之间的性能差异逐渐明显。导致以上仿真结果的主要原因分析如下。传统的 ALCM 由于车辆本身计算能力有限而带来较大时延开销; AOCM 随着车辆数目增加导致信道干扰逐渐严重; ARRA 由于未对资源进行有效管理, 当任务大小增加时, 传输时延增长, 因此性能较差; MCO 方案只关注卸载策略而不涉及动态资源分配, 因此随着车辆数目增加, 信道干

扰逐渐严重,使总开销快速增长;JCOIM 应用于本文场景,由于其忽略功率分配对系统开销的影响,因此效果稍有不理想。本文基于 SDN 辅助的 JTORA 机制由于有效的卸载与资源管理,使系统开销降低,同时还可以看出,卸载任务的数据量越大,人们从卸载计算中获得的时延和能耗收益就越大。

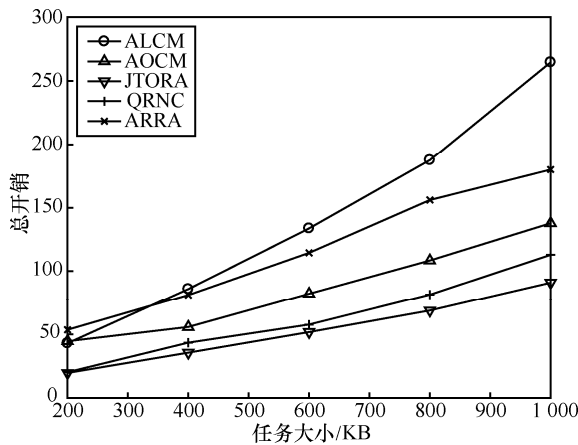


图 3 任务大小与总开销关系

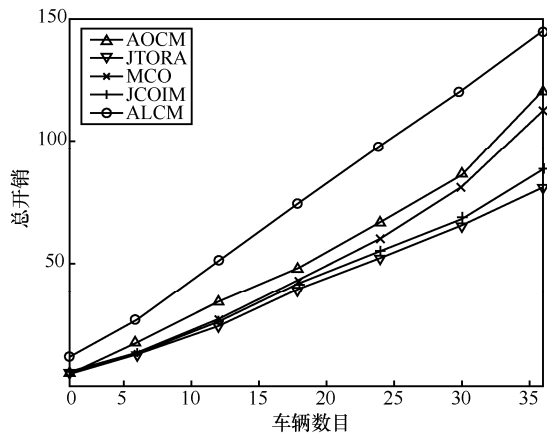


图 4 车辆数目与总开销关系

图 5 显示了算法的收敛性。随着迭代次数的增加,系统开销逐渐趋于稳定,算法收敛。AOCM 和本文提出的 JTORA 机制在经过约 600 次迭代后趋于收敛,而 QRNC 在约 800 次迭代后收敛。这是因为在选择初始卸载节点时,通过 Aggregative Clustering 可以选择一个相对满意的服务节点,从而使服务请求者在初始阶段获得一定的满意度,因此可以加快收敛。

图 6 显示了信道数量和系统总开销之间的关系。随着信道数的增加,干扰逐渐减小,带来传输时延的降低,从而使系统开销逐渐减小。由于

AOCM 受信道数量的影响较大,对于信道干扰更加敏感,因此 AOCM 曲线相比于 JTORA 曲线下降速度更快。MCO 机制由于其仅考虑卸载决策的优化,却未曾考虑资源分配,因此随着信道数量增加,干扰减小,车辆更倾向于卸载计算,因此在信道数量大于 19 以后,MCO 机制开销略大于 AOCM。本文基于 SDN 辅助的 JTORA 机制能够更加灵活地管理资源,控制卸载流量,保证资源有效利用,降低系统开销。

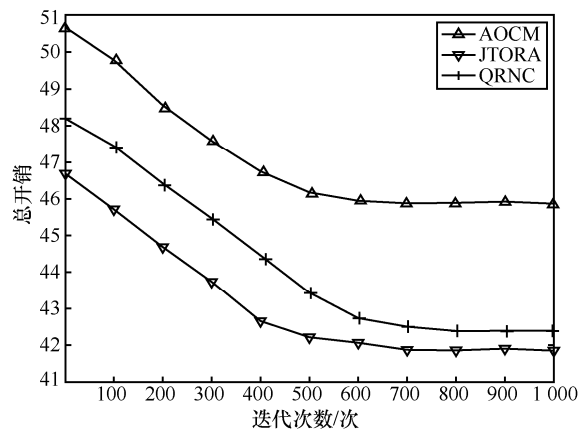


图 5 算法收敛性

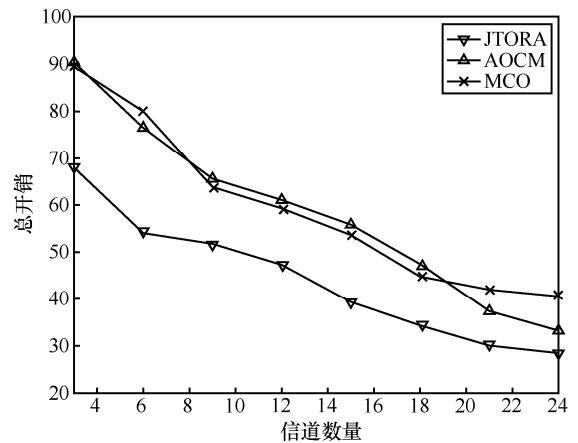


图 6 信道数量与总开销关系

5 结束语

本文通过考虑车载网络复杂的网络环境,构建了一种基于 SDN 辅助的 MEC 车载网络架构,通过引入 SDN 控制器,可灵活管理网络资源,易于收集车辆状态信息,应用程序可以选择合适的无线接口进行数据传输。在此基础上,本文提出了一种基于 MEC 的 V2X 任务卸载和资源分配机制,通过初始卸载节点的选择、计算卸载决策、

无线资源和计算资源的合理分配来最小化系统开销。仿真结果表明,该机制在不同参数变化下均能取得较好效果。

参考文献:

- [1] XU L D, HE W, LI S. Internet of things in industries: a survey[J]. *IEEE Transactions on Industrial Informatics*, 2014, 10(4): 2233-2243.
- [2] 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.
- [3] YU R, HUANG X, KANG J, et al. Cooperative resource management in cloud-enabled vehicular networks[J]. *IEEE Transactions on Industrial Electronics*, 2015, 62(12): 7938-7951.
- [4] 彭鑫, 邓清勇, 田淑娟, 等. 多信道车联网 V2R/V2V 数据传输调度算法[J]. *通信学报*, 2019, 40(3): 92-101.
PENG X, DENG Q Y, TIAN S J, et al. Data dissemination scheduling algorithm for V2R/V2V in multi-channel VANET[J]. *Journal on Communications*, 2019, 40(3): 92-101.
- [5] ZHANG K, MAO Y, LENG S, et al. Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks[C]//International Workshop on Resilient Networks Design & Modeling. IEEE, 2016: 288-294.
- [6] MUSTAFA A M, ABUBAKR O M, AHMADIEN O, et al. Mobility prediction for efficient resources management in vehicular cloud computing[C]//International Conference on Mobile Cloud Computing. IEEE, 2017: 53-59.
- [7] 田辉, 范绍帅, 吕昕晨, 等. 面向 5G 需求的移动边缘计算[J]. *北京邮电大学学报*, 2017, 40(2): 1-10.
TIAN H, FAN S S, LYU X C, et al. Mobile edge computing for 5G demand[J]. *Journal of Beijing University of Posts and Telecommunications*, 2017, 40(2): 1-10.
- [8] 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.
- [9] MACHARDY Z, KHAN A, OBANA K, et al. V2X access technologies: regulation, research, and remaining challenges[J]. *IEEE Communications Surveys & Tutorials*, 2018, 20(3): 1858-1877.
- [10] LIU J, WAN J, ZENG B, et al. A scalable and quick-response software defined vehicular network assisted by mobile edge computing[J]. *IEEE Communications Magazine*, 2017, 55(7): 94-100.
- [11] 邵雯娟, 沈庆国. 软件定义的 D2D 和 V2X 通信研究综述[J]. *通信学报*, 2019, 40(4): 179-194.
SHAO W J, SHEN Q G. Survey of software defined D2D and V2X communication[J]. *Journal on Communications*, 2019, 40(4): 179-194.
- [12] 陈兴蜀, 滑强, 王毅桐, 等. 云环境下 SDN 网络低速率 DDoS 攻击的研究[J]. *通信学报*, 2019, 40(6): 210-222.
CHEN X S, HUA Q, WANG Y T, et al. Research on low-rate DDoS attack of SDN network in cloud environment[J]. *Journal on Communications*, 2019, 40(6): 210-222.
- [13] HU F, HAO Q, BAO K. A survey on software-defined network and OpenFlow: from concept to implementation[J]. *IEEE Communications Surveys & Tutorials*, 2014, 16(4): 2181-2206.
- [14] ZHANG K, MAO Y, LENG S, et al. Mobile-edge computing for vehicular networks: a promising network paradigm with predictive offloading[J]. *IEEE Vehicular Technology Magazine*, 2017, 12(2): 36-44.
- [15] LI T Z, WU M Q, ZHAO M, et al. An overhead-optimizing task scheduling strategy for ad-hoc based mobile edge computing[J]. *IEEE Access*, 2017, 5: 5609-5622.
- [16] KAN T Y, CHIANG Y, WEI H Y. Task offloading and resource allocation in mobile-edge computing system[C]//Wireless and Optical Communications Conference. IEEE, 2018: 1-4.
- [17] YOU C, HUANG K. Multiuser resource allocation for mobile-edge computation offloading[C]// IEEE Global Communications Conference. IEEE, 2016: 1-6.
- [18] DENG M, TIAN H, LYU X. Adaptive sequential offloading game for multi-cell mobile edge computing[C]// International Conference on Telecommunications. IEEE, 2016:1-5.
- [19] ZHANG H, GUO F, JI H, et al. Combinational auction based service provider selection in mobile edge computing networks[J]. *IEEE Access*, 2017, 5: 13455-13464.
- [20] ZHENG J, CAI Y, WU Y, et al. Dynamic computation offloading for mobile cloud computing: a stochastic game-theoretic approach[J]. *IEEE Transactions on Mobile Computing*, 2019, 18(4): 771-786.
- [21] MAO Y Y, ZHANG J, LETAIEF K B. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems[C]//Wireless Communications and Networking Conference. IEEE, 2017: 1-6.
- [22] WAN J, TANG S, SHU Z, et al. Software-defined industrial Internet of things in the context of industry 4.0[J]. *IEEE Sensors Journal*, 2016, 16(20): 7373-7380.
- [23] HUANG X, RONG Y, KANG J, et al. Exploring mobile edge computing for 5G-enabled software defined vehicular networks[J]. *IEEE Wireless Communications*, 2018, 24(6): 55-63.
- [24] HUANG C M, CHIANG M S, DAO D T, et al. V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture[J]. *IEEE Access*, 2018, 6: 17741-17755.
- [25] 李萌, 司鹏搏, 孙恩昌, 等. 基于车联网和移动边缘计算的时延可容忍数据传输[J]. *北京工业大学学报*, 2018(4): 529-537.
LI M, SI P B, SUN E C, et al. Delay-tolerant data traffic based on connected vehicle network and mobile edge computing[J]. *Journal of Beijing university of technology*, 2018(4): 529-537.
- [26] 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.
- [27] YANG L, CAO J, YUAN Y, et al. A framework for partitioning and execution of data stream applications in mobile cloud computing[J].

- ACM Sigmetrics Performance Evaluation Review, 2013, 40(4): 23-32.
- [28] LI T Z, WU M Q, ZHAO M. Consumption considered optimal scheme for task offloading in mobile edge computing[C]//International Conference on Telecommunications. IEEE, 2016: 1-6.
- [29] 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.
- [30] PHAM Q V, LEANH T, TRAN N H, et al. Decentralized computation offloading and resource allocation in heterogeneous networks with mobile edge computing[J]. IEEE Access, 2018, 6: 75868-75885.
- [31] WEN Y, ZHANG W, LUO H. Energy-optimal mobile application execution: taming resource-poor mobile devices with cloud clones[C]//INFOCOM. IEEE, 2012: 2716-2720.
- [32] WU S, XIA W, CUI W, et al. An efficient offloading algorithm based on support vector machine for mobile edge computing in vehicular networks[C]//International Conference on Wireless Communications and Signal Processing. IEEE, 2018: 1-6.
- [33] 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.
- [34] LIU A A, SU Y T, NIE W Z, et al. Hierarchical clustering multi-task learning for joint human action grouping and recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(1): 102-114.
- [35] ZHANG J, XIA W, YAN F, et al. Joint computation offloading and resource allocation optimization in heterogeneous networks with mo-

bile edge computing[J]. IEEE Access, 2018(6): 19324-19337.

- [36] 林晓升. 基于强化学习的缓存策略研究[D]. 广州: 广州大学, 2019.
- LIN X S. Research on cache strategy based on reinforcement learning[D]. Guangzhou: Guangzhou University, 2019.
- [37] YANG T Y, HU Y L, GURSOY M C, et al. Deep reinforcement learning based resource allocation in low latency edge computing networks[C]//International Symposium on Wireless Communication Systems. IEEE, 2018: 1-5.

[作者简介]



张海波 (1979-)，男，重庆人，博士，重庆邮电大学副教授、硕士生导师，主要研究方向为车联网、移动边缘计算等。



王子心 (1995-)，女，重庆人，重庆邮电大学硕士生，主要研究方向为车联网、移动边缘计算。

贺晓帆 (1985-)，男，河北保定人，博士，武汉大学教授，主要研究方向为无线资源优化。