

DOI: 10.13382/j.jemi.B2003083

车联网中基于任务紧急性的联合卸载方案*

张建军 代帅康 张本宏

(合肥工业大学 计算机与信息学院 合肥 230601)

摘要:在车联网中,任务卸载可以有效地解决车辆的存储资源和计算资源不足的问题,单个的 MEC(mobile edge computing)服务器通常无法满足车辆密集区域的任务卸载需求。针对上述不足,设计了一种多 MEC 服务器的联合卸载方案(joint offloading method based on task urgency, JOMTU)。车辆递交任务卸载请求给本地 MEC 服务器时,后者在负载严重的情况下,会根据任务的紧急性和服务器负载情况等因素,将任务发送给附近 MEC 服务器处理以满足任务的截止日期要求。仿真实验结果表明,与传统的方案相比,所提出的方案将系统的整体任务失败率降低 17%,并且优化了整个网络的服务器负载情况、增加了网络的可靠性。

关键词:移动边缘计算;车联网;MEC 服务器;任务紧急性;任务卸载

中图分类号: TP393.0;TN01 **文献标识码:** A **国家标准学科分类代码:** 520

Joint offloading method based on task urgency in the VANETs

Zhang Jianjun Dai Shuaikang Zhang Benhong

(School of Computer and Information, Hefei University of Technology, Hefei 230601, China)

Abstract:In the VANETs, task offloading can effectively solve the problem of insufficient resource storage and computing resources of vehicles, but a stand-alone mobile edge computing (MEC) server is usually unable to meet the task offloading requirements in vehicle-intensive areas. Aiming at the above shortcoming, a joint offloading method based on task urgency is proposed. When the vehicle offloads the task to the local MEC server, the local MEC server will send the task to the nearby MEC server to meet the task deadline according to the task urgency, deadline and the load of servers. The simulation results show that compared with the traditional scheme, the proposed scheme reduces the overall task failure rate of the whole system by 17%, optimizes the server load of the whole network and increases the reliability of the network.

Keywords:mobile edge computing; VANETs; MEC server; task urgency; task offloading

0 引言

车联网技术的快速发展,车辆在支持视频辅助实时导航、自动驾驶和交互式游戏等应用方面变得更加智能^[1-3]。这些智能应用^[4]通常具有计算密集、延迟敏感和带宽要求高等特征,对车载终端的处理能力造成很大的压力,也给无线接入网络带来很大的负担^[4-5]。

随着 5G 到来,它提供的超低时延和高容量的带宽^[6],可以为车联网中车辆高速移动导致的拓扑的结构

变化和通信的连续性提供了强有力的支撑。边缘计算作为 5G 的关键技术^[7],将计算资源推送到无线接入网络,在网络边界提供分布式计算和存储功能。任务卸载作为边缘计算的关键技术之一^[8],可以将车载设备的部分或全部的计算任务交给边缘计算服务器处理,解决车载设备在资源存储和计算能力等方面存在的不足^[9],减少通信和计算延迟,以实现实时操作和更高的响应能力^[10]。在移动边缘计算车联网环境下,面对车辆行驶产生的繁多的数据和请求,如何卸载决策和执行边缘资源分配已经成为实现可靠通信和资源合理利用的关键问题。

收稿日期:2020-04-19 Received Date: 2020-04-19

* 基金项目:安徽省科技重大专项(201903a05020049)、国家自然科学基金(61701005)资助项目

文献[11]阐述了基于马尔可夫决策过程的多资源分配策略,该策略确定了用于实现最优系统效益的确切无限带宽资源和计算资源的管理。文献[12]基于马尔可夫决策过程,提出了一种单用户 MEC (mobile edge computing) 系统的时延最优随机任务调度策略,考虑了计算资源分配和传输功率分配,联合优化用户关联和计算卸载。文献[13]对于具有严格执行期限的任务,通过采用动态电压和频率缩放技术将本地执行能耗降至最低,并通过数据传输调度对计算卸载的能耗进行了优化。文献[14]提出了一种基于网络辅助的端到端通信任务卸载框架,该框架能够实现移动用户之间的资源共享。文献[15]提出了一种协作下载方案,通过 VANETs 来卸载来自蜂窝网络的流量,从蜂窝网络中获取适当的数据,并以近似最优的方式将数据分发给车辆,设计了用于规划数据传输的存储时间聚合图。文献[16]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。上述文献考虑到带宽限制,将任务在本地服务器计算的结果在基站间通过 V2I (vehicle-to-infrastructure) 的方式发送给车辆,可以有效的降低任务的失败率,但是由于本地服

务器计算资源的有限,这种方式容易造成服务器过载的情况,会造成任务卸载失败的增加和资源利用不充分的情况。本文在上述文献的基础上,考虑到 5G 网络低时延和高带宽的特点,提出一种基于任务紧急性的联合任务卸载方案 (joint offloading method based on task urgency, JOMTU),方案中考虑任务的紧急性、截止日期和服务器负载等因素,通过联合多 MEC 服务器的方式,将本地任务卸载请求发送给附近服务器,减轻本地服务器的负载压力,避免单个 MEC 服务器出现资源不足的情况。论文最后通过仿真实验对 JOMTU 方法进行分析和验证。

1 系统模型

本文考虑将 VANET 中的 RSU (road side unit) 和基站作为 MEC 服务器的载体。RSU 是具有存储、计算和短距离无线传输功能的设备,边缘设备部署在 RSU 或基站上,为 RSU 和基站提供有限的存储和计算能力,以分担 RSU 和基站处理数据和任务请求等能力不足的压力。MEC 服务器可利用无线接入网络为就近用户提供所需服务和远端计算功能。车辆与 RSU 之间的通信是基于 V2I 方式进行。

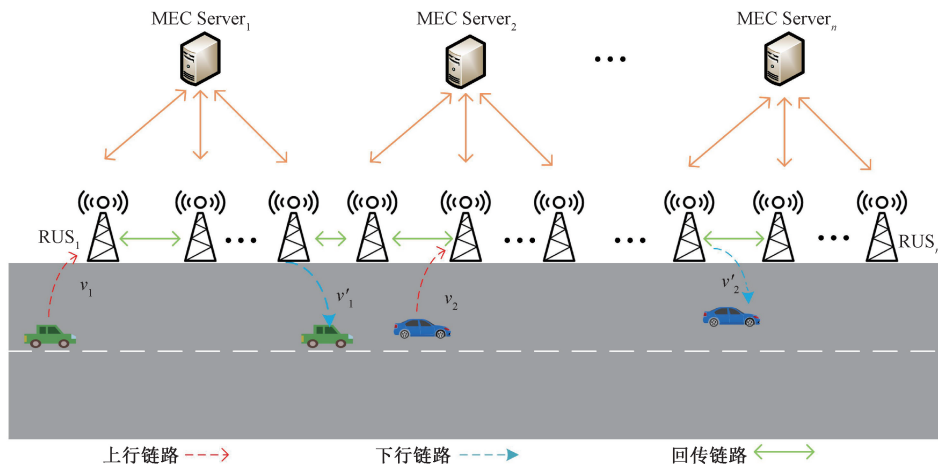


图 1 联合卸载任务模型

Fig. 1 Joint unloading task model

图 1 所示为联合卸载任务模型。图 1 中,所有的 RSU 均通过高容量带宽的回传链路连接,且每个 RSU 均可通过基带单元获取整个系统的负载信息。在得知负载信息后,可以根据递交的请求,分配给可以处理的 MEC 服务器进行处理。如车辆 v_1 和 v_2 向其覆盖区域的 RSU 发送任务卸载请求,此时本地 RSU 通过回传链路获取各个 MEC 服务器的状态信息,根据任务的截至日期和车辆的速度,可判定任务是否可以在本地边缘服务器处理。车辆 v_2 所在区域车辆密集,车辆上传过多的任务卸载请求,本地 MEC Server₂ 无法满足其任务所需的计算能力要求或者服务器的存储能力不足,无法及时响应相应的请

求,则根据前方 MEC 服务器的负载状态、车辆的行驶时间和任务截止时间等因素,向车辆前进方向的 RSU_i 发送卸载请求,转发本地任务请求至前方的边缘服务器进行处理。等到车辆进入其任务处理的覆盖范围内,车辆会立即得到相应的结果。如车辆 v_2 行驶一段距离后,来到了邻近的边缘服务器,则在车辆 v_2 在行驶的时间内,MEC 服务器处理并完成其任务卸载请求,等到车辆 v'_2 进入其覆盖范围,返回其请求结果。当然,如果车辆 v_2 所处区域的车辆稀疏,本地 MEC 服务器未达到过载状态,可直接在本地服务器上进行处理。

2 问题描述

本文将各个车辆发送至 RSU 的任务的集合表示为 $H = \{h_1, h_2, h_3, \dots, h_m\}$, 任务 h_i 定义为一个七元组 $\langle c_i, d_i, t_{\max}^i, x_i, y_i, t_i, k_i \rangle$, 其中 c_i 为满足任务 h_i 所需分配的计算资源, d_i 为任务 h_i 上传的数据包大小, t_{\max}^i 为完成任务的最大容忍时间, 即任务的截止日期。 t_i 表示任务 h_i 的生成时间。 k_i 表示任务 h_i 的类型。 x_i, y_i 表示任务的生成位置, 根据其位置和车辆的速度, 得出车辆是否可以在离开本地 RSU 覆盖范围前, 获得请求结果。 车辆经过各个 RSU 的时间 $t_s(v_i)$ 如下:

$$t_s(v_i) = \frac{dis_i(t) + (n-1) \times l}{s_i(t)} \quad (1)$$

式中: $dis_i(t)$ 为车辆 v_i 离开本地 RSU 覆盖区域的直线距离; l 为 RSU 覆盖范围下的道路长度; $s_i(t)$ 为车辆 v_i 的行驶速度。 当任务的紧急程度高时, 需要在车辆离开本地 RSU 范围前接收到结果, 则可认为车辆仅会经过一个 RSU。 此时 $t_s(v_i)$ 也表示车辆离开本地 RSU 覆盖区域的时间。 同时应满足 $t_s(v_i) \leq t_{\max}^i$, 此约束保证了车辆可以按时到达对应的 RSU 覆盖范围。

任务 h_i 的优先级表示为 $P(h_i)$, 该值由任务的截止时间、任务种类、车辆停留时间来确定大小, 如式(2)所示。

$$P(h_i) = \begin{cases} \frac{\chi}{dis_i(t)/s_i(t)} \frac{1}{t_{\max}^i}, & \text{紧急任务} \\ \frac{1}{dis_i(t)/s_i(t)} \frac{1}{t_{\max}^i}, & \text{非紧急任务} \end{cases} \quad (2)$$

式中: χ 是个常量, 目的是为了保证紧急任务的优先级始终大于非紧急任务。 当车辆停留时间为 0 时, 车辆处于两个 RSU 的交界处, 这时可根据车辆行驶的方向, 将发送的任务算入下一个 RSU 的覆盖范围内, 重新计算车辆发送的任务紧急程度。

车辆传输任务给 RSU。 车辆的传输功率记为 P_v , 信道增益记为 G_v , N_0 是高斯白噪声的单边功率谱密度, B_v 是信道的带宽。 任务的传输率 r_d 如下:

$$r_d = B_v \log_2 \left(1 + \frac{P_v G_v}{N_0} \right) \quad (3)$$

通信延迟分为 3 个部分, 上行链路延迟 D_{up}^i 、传输延迟 D_{tr}^i 和下行链路延迟 D_{down}^i , 计算方式如式(4)~(6)所示, 任务卸载过程的总通信延迟 D_{h_i} 如式(7)所示。

$$D_{\text{up}}^i = \frac{d_i}{r_d} \quad (4)$$

$$D_{\text{down}}^i = \frac{d'_i}{r_d} \quad (5)$$

$$D_{\text{tr}}^i = \frac{d_i}{B(s, j)} \quad (6)$$

$$D_{h_i} = D_{\text{up}}^i + D_{\text{tr}}^i + D_{\text{down}}^i \quad (7)$$

式中: d'_i 为 MEC 服务器处理任务后返回结果的任务大小; $B(s, j)$ 表示 rsu_s 和 rsu_j 之间回传线路的带宽。

假设每个区域车辆发送的任务是服从泊松分布, 则 MEC 服务器可看作是一个 $M/M/1$ 模型^[17]。 令任务到达 MEC 服务器的到达率为 λ_i , 平均服务率为 τ_i , 服务密集度记为 $\rho_i = \lambda_i / \tau_i$ 。 根据排队论可知, 输入任务的平均等待时间如下:

$$E(t_{\text{que}}^i) = \frac{\rho_i}{1 - \rho_i} \frac{1}{\tau_i} \quad (8)$$

任务 h_i 卸载请求发送后, 会为其分配一个独立的时间 t_d^i , 如下:

$$t_d^i = E(t_c^i) + E(D_i) + E(t_{\text{que}}^i) \quad (9)$$

式中: t_d^i 包含了产生的各种通信延迟时间、任务计算时间和等待时间; $E(t_c^i)$ 表示任务 h_i 在 MEC 服务器中的处理平均完成所消耗的时间; $E(D_i)$ 则代表平均通信延迟。 并且 $t_d^i \leq t_{\max}^i$, 表示分配给任务的独立期限 t_d^i 应小于任务的最大容忍时间 t_{\max}^i 。

3 JOMTU 卸载方法

RSU 可以通过回传链路获取各个边缘服务器的负载状态, 同时负责任务的转发。 在任务卸载执行前, RSU 会持续地更新和维护两个矩阵, 任务完成的时间矩阵 \mathbf{A} 和估计任务传输时间矩阵 \mathbf{B} 。

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}_{n \times m} \quad (10)$$

任务完成时间矩阵 \mathbf{A} 的行对应不同区域的 rsu_i , 列为任务的紧急度, 矩阵中的各项表示不同紧急度的任务卸载在不同区域 MEC 服务器的估计任务完成时间, 其任务完成时间被 a_{ij} 认为是服从正态分布 $X \sim N(\mu, \sigma^2)$, 因此存在平均值(μ)和方差(σ^2)两个值, 其数值大小是基于每种类型任务的历史执行时间。

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n1} & \cdots & b_{nn} \end{bmatrix}_{n \times n} \quad (11)$$

估计任务传输时间矩阵 \mathbf{B} 中的行列均代表各个区域的 rsu_i , 矩阵的各项 b_{ij} 为任务卸载到 MEC 服务器的估计延迟时间, 延迟时间 b_{ij} 也被认为是服从正态分布 $Y \sim N(\mu, \sigma^2)$, 其中 μ 和 σ^2 是根据不同种类的任务传输到

附近边缘服务器的历史传输时间所确定的。

通过任务完成的时间矩阵 \mathbf{A} 和估计任务传输的时间矩阵 \mathbf{B} , 本地 RSU 可根据接收到的任务信息计算出各个 MEC 服务器满足任务截止日期的概率。假设 rsu_j 收到任务 h_i , 对于接收任务的 rsu_j , 满足其截止日期 t_{\max}^i 的概率如下:

$$P_i(\omega(h_i, rsu_j) = P_i(Z < z) \tag{12}$$

其中 $\omega(h_i, rsu_j)$ 表示任务预期的完成时间。将式 (12) 根据标准正态分布 $\mu = 0$ 和 $\sigma^2 = 1$ 将其标准化, 可知 z 的值如下:

$$z = \frac{t_{\max}^i - \mu_i^j}{\sigma_i^j} \tag{13}$$

为了获得任务卸载在不同服务器的全部概率, 本文将任务完成时间矩阵 \mathbf{A} 和估计任务传输时间矩阵 \mathbf{B} 相对应项的分布进行卷积, 以找到任务卸载到所有服务器的延迟分布, 如下:

$$W \sim N(\mu, \sigma^2) = X \sim N(\mu, \sigma^2) \otimes Y \sim N(\mu, \sigma^2) \tag{14}$$

当 rsu_j 接收到任务卸载请求后, 便会将 \mathbf{A} 矩阵中的相应条目与 \mathbf{B} 矩阵中的相应条目进行卷积。通过卷积的方式获得了延迟的分布, 再根据其分布可以计算得出在各个 MEC 服务器完成任务的概率, 当可接收的各个服务器均被计算了概率, 就将任务请求通过 RSU 发送给概率最大的 MEC 服务器处理。当任务完成的概率为零时, 则该任务将被丢弃。基于任务紧急度的联合卸载方法步骤如表 1 所示。

表 1 基于任务紧急度的联合卸载方法描述

Table 1 Description of joint unloading method based on task urgency

算法 JOMTU 分配算法	
输入: 任务 h_i ; \mathbf{A} 和 \mathbf{B} ; $List$ (可接收任务的 RSU 集合)	
输出: 任务 h_i 将分配给 rsu_j	
1 获取 $p_s(h_i)$; $\leftarrow p_s(h_i)$ 表示 rsu_s 接收任务 h_i 的概率	
2 将任务 h_i 临时分配给 rsu_s ;	
3 for each $rsu_j \in List$ do	
4 获取 $p_j(h_i)$; $\leftarrow p_j(h_i)$ 表示分配给附近 rsu_j 的概率	
5 if $p_j(h_i) > p_s(h_i)$ then	
6 分配任务 h_i 给 rsu_j ;	
7 break;	
8 else if $p_j(h_i) = p_s(h_i)$ then	
9 if $\sigma_j < \sigma_s$ then	
10 分配任务 h_i 给 rsu_j ;	
11 break;	
12 end if	
13 end if	
14 end for	
15 if $p_s(h_i) = 0 \& p_j(h_i) = 0$ then	
16 丢弃任务 h_i ;	
17 end if	

4 仿真分析

为了对本文所提出的系统模型和算法进行评估, 首先模拟实验的场景, MEC 服务器将被看作小型处理机的来模拟实验, 每个 MEC 服务器在 X - Y 平面里都有一个指定的位置, 车辆可以根据发送任务请求信息找到离其最近的 MEC 服务器。仿真实验工具是 MATLAB R2016a, 仿真本文实验的部分参数设置如表 2 所列。

表 2 仿真参数

Table 2 Simulation parameters

参数名称	数值(单位)
通信延迟/s	2
MEC 服务器数量	10
紧急任务/MIPS	2 000~3 000
非紧急任务/MIPS	10 000~15 000
实验重复次数/次	20
网络带宽/Mbps	200

将任务具体分配为危险警报、变道警告、车载娱乐和燃料数据^[18] 4 种任务。前两种为紧急任务, 后两种为非紧急任务。

将 JOMTU 与最小期望完成时间 (minimum expected completion time, MECT)^[19] 和本地执行任务卸载 (LOC)^[20] 进行对比, 评估本文所提的系统模型和算法性能。其中 MECT 为最小期望完成时间方法, 对于特定类型的任务, 该启发式算法利用根据任务完成时间计算所有基站的平均期望完成时间, 并选择期望完成时间最小的基站处理请求。LOC 则是以不转发的方式, 即将任务转移到邻近基站, 无论任务何时到达, 只要车辆进入 RSU 的覆盖范围内发送任务请求, 任务只会在本地执行卸载。

如图 2 所示, 横轴表示系统中车辆的数量, 纵轴表示错过整体的任务平均任务失败率占比。错过截止日期的任务数量会随着车辆数目的增加而增加。对比与其他两种算法, 可以看出 LOC 的效果最不理想, 因为它单一服务器计算资源有限。而本文所述的 JOMTU 算法在各种车辆数目的环境下, 对比 MECT 和 LOC 算法, 最大可降低约 11% 和 17% 的任务失败率。

为了评估模型对紧急类型任务数量的增加的处理情况, 将在车辆数目为 4 000 辆时, 增加紧急任务类型数目的占比, 从 10% 增加到 90%。如图 3 所示, 任务的平均失败率会随着紧急任务占比的增加而升高, 因为紧急任务具有较高的时间敏感性。从对比来看, JOMTU 具有较好的鲁棒性, 即面对紧急情况的能力。当系统中的紧急任务类型百分比达到 70% 以上时, JOMTU 算法比 MECT 算法的任务失败率少 5%~9%, 比 LOC 方式任务失败率最高相差约 20%。

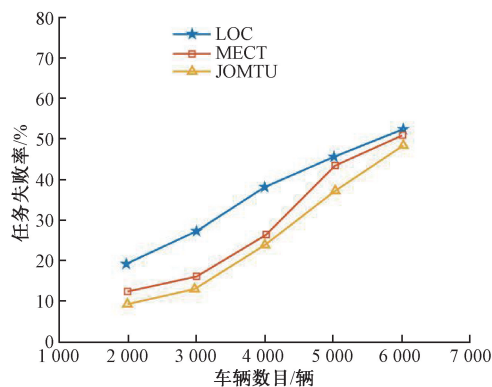


图 2 车辆数量增加的任务失败率

Fig. 2 The failure rate with increased number of the vehicle

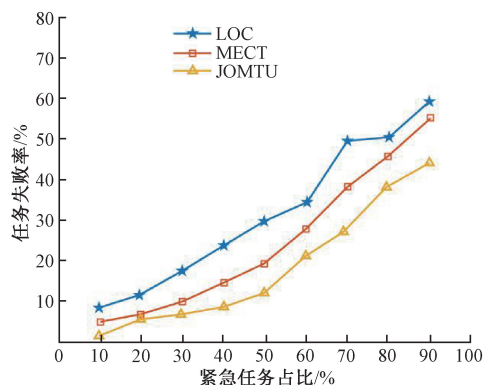


图 3 紧急任务数量增加的任务失败率

Fig. 3 The failure rate with increased number of urgent tasks

图 4 所示为单个 MEC 服务器的任务平均失败率,与图 2 的考虑总体任务失败率不同的是,这里只考虑某个服务器上的任务失败率。当车辆数目在 2 000~3 000 时,可以看出 JOMTU 算法的效果优于其他两种,与 MECT 算法相比最高可降低 9% 的任务失败率,但随着车辆数目增加时,可以看出单个 MEC 服务器的局限性导致了任务的失败率急剧上升。

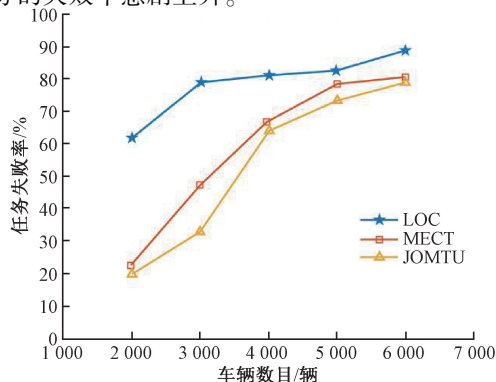


图 4 单个 MEC 服务器情况下的任务失败率

Fig. 4 The failure rate of the task in a single MEC server

5 结 论

本文研究了一种通过联合多 MEC 服务器的方式,来解决单一边缘服务器有资源不足的问题。提出的一种基于任务紧急性的 JOMTU。仿真实验表明, JOMTU 可以合理的利用计算资源,降低任务卸载的失败率和减少边缘服务器的负载压力。下一步工作将考虑将移动边缘计算和云服务器结合,进一步增加车辆任务卸载所需的计算资源,同时减少通信延迟和计算延迟所带来的影响,以满足日益增长的智能车辆的服务需求。

参考文献

- [1] 苏金亚,张伟伟,吴训成,等. 基于多结构特征提取和轨迹跟踪的车辆远光识别[J]. 电子测量与仪器学报, 2018, 32(10): 103-110.
SU J Y, ZHANG W W, WU X CH, et al. Recognition of vehicle high beams based on multi-structure feature extraction and path tracking [J]. Journal of Electronic Measurement and Instrumentation, 2018 32 (10): 103-110.
- [2] 范宜标, 卢玮, 傅智河. 自动驾驶场景下的鲁棒车辆检测[J]. 电子测量与仪器学报, 2018, 32(12): 60-65.
FAN Y B, LU W, FU ZH B. Robust vehicle detection under autonomous driving scenarios [J]. Journal of Electronic Measurement and Instrumentation, 2018, 32(12): 60-65.
- [3] ZHANG K, MAO Y, LENG S, et al. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading[J]. IEEE Vehicular Technology Magazine, 2017, 12(2): 36-44.
- [4] 张艳辉, 徐坤, 郑春花, 等. 智能电动汽车信息感知技术研究进展[J]. 仪器仪表学报, 2017, 38(4): 794-805.
ZHANG Y H, XU K, ZHENG CH H, et al. Research progress of intelligent electric vehicle information perception technology [J]. Journal of Instrumentation and Instrumentation, 2017, 38 (4): 794-805.
- [5] SOOKHAK M, YU F R, HE Y, et al. Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing [J]. IEEE Vehicular Technology Magazine, 2017, 12(3): 55-64.
- [6] AGIWAL M, ROY A, SAXENA N. Next generation 5G wireless networks: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2016, 18(3): 1617-1655.
- [7] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J].

- 计算机研究与发展, 2019, 56(1): 73-93.
- SHI W S, ZHANG X F, WANG Y F, et al. Edge computing: Status and prospects[J]. Computer Research and Development, 2019, 56(1): 73-93.
- [8] CHEN M, HAO Y. Task offloading for mobile edge computing in software defined ultra-dense network[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(3): 587-597.
- [9] 谢人超, 廉晓飞, 贾庆民, 等. 移动边缘计算卸载技术综述[J]. 通信学报, 2018, 39(11): 138-155.
- XIE R CH, LIAN X F, JIA Q M, et al. Overview of unloading techniques for mobile edge computing [J]. Journal of Communications, 2018, 39 (11): 138-155.
- [10] 罗峰, 徐金鹏. 基于车载环境的时间同步协议优化[J]. 仪器仪表学报, 2019, 40(2): 96-104.
- LUO F, XU J P. Time synchronization protocol optimization for automotive environment [J]. Journal of Electronic Measurement and Instrumentation, 2019, 40(2): 96-104.
- [11] WANG X, NING Z, WANG L. Offloading in internet of vehicles: A fog-enabled real-time traffic management system[J]. IEEE Transactions on Industrial Informatics, 2018, 14(10): 4568-4578.
- [12] 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.
- [13] ZHAO Y, LEUNG V C M, GAO H, et al. Uplink resource allocation in mobile edge computing-based heterogeneous networks with multi-band RF energy harvesting[C]. 2018 IEEE International Conference on Communications (ICC), 2018: 1-6.
- [14] LIU M, YU F R, TENG Y, et al. Computation offloading and content caching in wireless blockchain networks with mobile edge computing [J]. IEEE Transactions on Vehicular Technology, 2018, 67(11): 11008-11021.
- [15] SUN Y, XU L, TANG Y, et al. Traffic offloading for online video service in vehicular networks: A cooperative approach [J]. IEEE Transactions on Vehicular Technology, 2018, 67(8): 7630-7642.
- [16] ZHANG K, MAO Y, LENG S, et al. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading[J]. IEEE Vehicular Technology Magazine, 2017, 12(2): 36-44.
- [17] NAZAROV A, SZTRIK J, KVACH A, et al. Asymptotic analysis of finite-source M/M/1 retrial queueing system with collisions and server subject to breakdowns and repairs[J]. Annals of Operations Research, 2019, 277(2): 213-229.
- [18] XIAO Y, REN Z, ZHANG H, et al. A novel task allocation for maximizing reliability considering fault-tolerant in VANET real time systems [C]. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017: 1-7.
- [19] NASR A A, EL-BAHNASAWY N A, AYMAN E L S. A new duplication task scheduling algorithm in heterogeneous distributed computing systems [J]. Bulletin of Electrical Engineering and Informatics, 2016, 5(3): 373-382.
- [20] CHOO S, KIM J, PACK S. Optimal task offloading and resource allocation in software-defined vehicular edge computing [C]. 2018 International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 2018: 251-256.

作者简介



张建军, 2005 年于合肥工业大学获得博士学位, 现任合肥工业大学教授, 主要研究方向为计算机集成系统、汽车电子控制系统与嵌入式软件、车载信息与网络传输技术。

E-mail: zjj@ialab. hfut. edu. cn

Zhang Jianjun received his Ph. D. degree in 2005 from Hefei University of Technology. Now he is a professor at Hefei University of Technology. His main research interests include computer integration system, automotive electronic control system and embedded software, and vehicle information and network transmission technology.



代帅康, 2015 年于中国计量学院获得学士学位, 现为合肥工业大学硕士研究生, 主要研究方向为车联网、移动边缘计等。

E-mail: 907557396@qq. com

Dai Shuaikang received his B. Sc. degree from China Jiliang University in 2015. His is now a M. Sc. candidate at Hefei University of Technology. His main research interests include VANETs and mobile edge computing.



张本宏, 2010 年于合肥工业大学获得博士学位, 现任合肥工业大学副教授, 主要研究方向为无线传感器网络、汽车电子、边缘计算。

E-mail: zhangbh@ hfut. edu. cn

Zhang Benhong received his Ph. D. degree from Hefei University of Technology in 2010. Now he is an associate professor at Hefei University of Technology. His main research interests include wireless sensor networks, automotive electronics and edge computing.