

车联网场景下的边缘计算卸载论文综述

王琦

2019 年 1 月 22 日

摘要

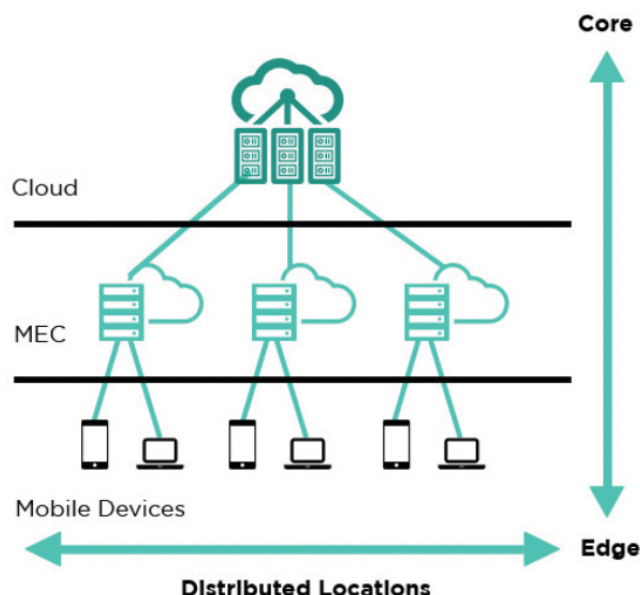
新兴的车载应用有很多计算密集型和延迟敏感的任务，这通常需要车辆有更强的计算能力，但又受限于升级设备的成本，而移动边缘计算专注于将计算资源转移到网络的边缘，为移动设备提供高性能、低延迟的服务，因此可作为处理计算密集型和延迟敏感的任务的一种有效方法。由于城市地区拥有大量车辆，利用这些未充分使用的车辆资源可提供巨大的机会和价值，因此，将车辆作为计算的平台是十分有意义的，本文称其为车辆边缘计算(Vehicular Edge Computing, VEC)。本文介绍了VEC的产生背景、研究情况、几种常见的车辆环境下任务卸载的结构以及关键的研究点。

1 介绍

云计算[1]自2005年诞生以来，已经极大地改变了我们的生活、工作和学习方式，如Software as a Service(SaaS)服务被广泛的应用到阿里、腾讯、百度等IT巨头的数据中心。而随着物联网和4G/5G无线网络的快速发展，网络边缘的设备(如手机、平板电脑、传感设备等)的数量迅速增加，以云计算模型为核心的集中式大数据处理模式已经不能高效的处理边缘设备所产生的数据。同时一些新兴应用程序的发展，如视频分析、在线游戏、模式识别等，由于具有复杂度高、数据密集、延迟敏感等特点，受到了设备能源消耗和计算能力的限制。将计算任务卸载到传统云是一种解决方案，但这又有新的问题——延时增加，包括将卸载的任务交付到云、云上计算时间和结果返回时间，这样的处理方案不能满足延时敏感型的应用。因此，边缘计算(Edge Computing)[2]应运而生。

在计算模式的领域内，边缘计算(也称雾计算)已经开始变得至关重要，尤其是移动网络网络中的移动边缘计算(Mobile Edge Computing, MEC)。移动边缘计算[3][4]在移动网络的边缘，无线接入网络(Radio Access Network, RAN)内以及移动用户附近提供IT服务环境和云计算能力。这里将“边缘”定义为数据源和云数据中心之间路径上的任何计算和网络资源，如基站、无线接入点等。移动边缘计算模型将原有云计算中心的部分或者全部计算任务迁移到数据源附近执行，弥补了边缘移动设备计算能力不足以及电池能源有限的缺点，提高了数据的传输性能，保证了处理的实时性，同时又降低了云计算中心的计算负载。图1是一种典型的移动边缘计算框架。

随着智能车辆的发展，越来越多的计算机应用被部署在车辆环境中，例如实时态势感知(Real-time Situational Awareness)、增强现实(AR)、自动驾驶等数据密集和延迟敏感的计算任务，这对车辆的计算能力提出了新的要求。升级车载计算机是一种选择，但是升级硬件成本很高；卸载任务到云进行计算，使用远程服务器来满足计算要求是另一种选择，但是由于传统云计算高延迟和不稳定性，无法满足车载应用延迟敏感的要求。边缘计算为车载应用提供了选择，可以将车辆任务卸载到其他计算资源空闲的车辆[5][6]、基站等固定基础设施或者移动边缘设备上执行，以增强计算能力、提高服务质量。在车联网的环境下，车载网络技术已经比较成熟，车辆之间的通信可以基于专用短程通信(Dedicated Short-Range Communications, DSRC)[7]来实现，DSRC是基于IEEE802.11p设计的，在300米覆盖内DSRC的数据速率可以达到27Mb/s左右。车辆与固定基础设施之间的通信则可以使用Long-Term Evolution(LTE)[8]，与DSRC相比，LTE的覆盖范围更广，服务质量(QoS)更具有保障，LTE还支持350公里/小时的高移动性



的用户设备，可以很好的适应高速移动的车辆。DSRC和LTE解决了车联网下的边缘计算卸载的通信问题，使任务的卸载成为可能。

2 研究现状

[9][10]等工作对云计算资源进行分层,分为RC(Remote Cloud)和VC(Vehicle Cloud),其中RC为远程云,通常认为计算能力是无限的,但是传输延迟比较大,VC为车辆计算资源构成的虚拟资源池,计算资源有限,但传输延时比较小。卸载时会优先考虑将任务卸载到VC上执行,当VC的资源不足时才会考虑卸载到RC,并使用SMDP(Semi-Markov Decision Process)来最小化车辆计算任务的能量和时间消耗,其中[10]在[9]的基础上将VC扩展为车辆和路边单元(Roadside Units,RSUs)的异构环境。

[12]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,将车辆应用程序依据其重要程度划分为关键应用程序、高优先级应用程序和低优先级应用程序,其中关键应用程序时和安全相关的核心应用程序,不考虑在系统内。车辆之间通过广播的方式来相互交换信息,由车辆依据速度、计算资源的空闲状态来决定任务卸载的策略,而不依靠外界基础设施。该文章

介绍了车辆边缘自主组织任务卸载的工作流程,提出了有效的工作缓存,并使用群智能算法Ant Colony Optimization(ACO)来优化任务的处理延时。

[13]则提出将边缘云上的计算任务卸载到VC的车辆上(任务可能由车辆产生,也可能从其他的边缘设备产生),其计算任务具有相互依赖性,最终通过Modified Genetic Algorithm(MGA)算法来最小化在不同车辆中执行总体响应时间。[13]将相互依赖的任务分为三种类型,分别为线形、树形和网形,并用矩阵的形式来表示其依赖性,为求解依赖性任务的卸载提供了很好的解决思路。

[14]将移动用户产生的任务分为高、低优先级两类,分别服从泊松分布。然后将任务卸载到车辆计算资源构成的云上,当资源不足时再将其卸载到远程云,最后使用博弈论的方法最小化系统的延时以及消耗的能源。

[15]提出一种具有三层结构的车辆计算框架,由云计算层,雾计算层和车辆计算层组成。雾计算层位于网络边缘,它可以提供更短延迟和更大带宽的本地服务。云计算层将所需资源预先调度到雾计算层,车辆计算层与雾计算层接触。[15]清楚地表明了雾计算层和车载计算层之间带宽分配的相关性,旨在通过为四种服务分配可用带宽来减少服务时间。

[16]则提出一种Parked Vehicle Edge Computing(PVEC),它将静止停放的车辆空闲的计算资源利用起来,将其看作是基础设施的一部分(但有区别与固定的基础设施),并通过使用Stackelberg博弈方法来衡量资源的可用性并制定资源调度优化问题。

车联网下的移动边缘计算,通过将车辆、边缘设备和云资源的协调管理,可以对资源进行合理的分配,来解决包括延迟问题、能耗问题、成本问题等许多任务调度问题。即便是同样的优化目标,因为会考虑不同的计算资源、任务之间依赖性、单用户或多用户等一些其他因素的限制,所设计的系统模型各有相同,采用的解决问题方法也不一样。本文的第三部分总结车联网下边缘计算卸载的一些关键研究点,包括卸载模型的选择、任务卸载的优化目标以及车辆的移动性。

3 主要研究内容

在车联网下的移动边缘计算任务卸载的调度研究工作重点关注以下研究点:1)车辆环境下卸载模型的选择;2)确定任务卸载的优化目标;3)车辆的移动性分析。

3.1 车辆环境下卸载模型的选择

依据对上述论文中车联网下任务卸载模型的分析,可以将车联网环境中拥有计算能力的设备分为三种,下图2为车联网下移动边缘计算的计算资源分类图。第一种为传统云计算中的云资源,这里称为远程云(Remote Cloud,RC),通常认为RC的计算能力是无限的(计算延时很小或是没有),但是由于云的位置比较远,它所带来的传输开销不能满足数据密集型和延迟敏感型任务的需求,因此不会优先考虑将任务卸载到RC。

第二种为由边缘拥有计算能力的设备组成,称为边缘云(Edge Cloud,EC),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。在车辆环境中,EC由如路边单元(Roadside Units)、基站(Base Station)、无线接入点(Access Point)以及移动设备等其他边缘设备组成。虽然EC计算能力不如RC,但是由于设备处在网络的边缘,对于数据密集型和延迟敏感型任务来说,大大缩短了传输延时,因此也能够减少整体的延迟。

第三种是拥有计算能力的车辆形成的车辆云(Vehicular Cloud,VC),随着车联网的发展,使得车辆之间、车辆与固定基础设施之间的通信不再是难题,因此可以充分利用车辆上空闲的计算资源,在保证车辆安全行驶的前提下,提高用户的驾驶体验。车辆可以接受来自其他车辆的任务请求,也可以充当移动边缘计算设备,接受来自其他边缘设备的任务请求。

在进行研究时,可以依据具体的场景和需求进行从三种类型的计算资源中选择。因为我们研究的内容是在车辆环境下的计算卸载,因此第三种一般是需要的,前两种可选可不选。例如在[12]中,作者将车

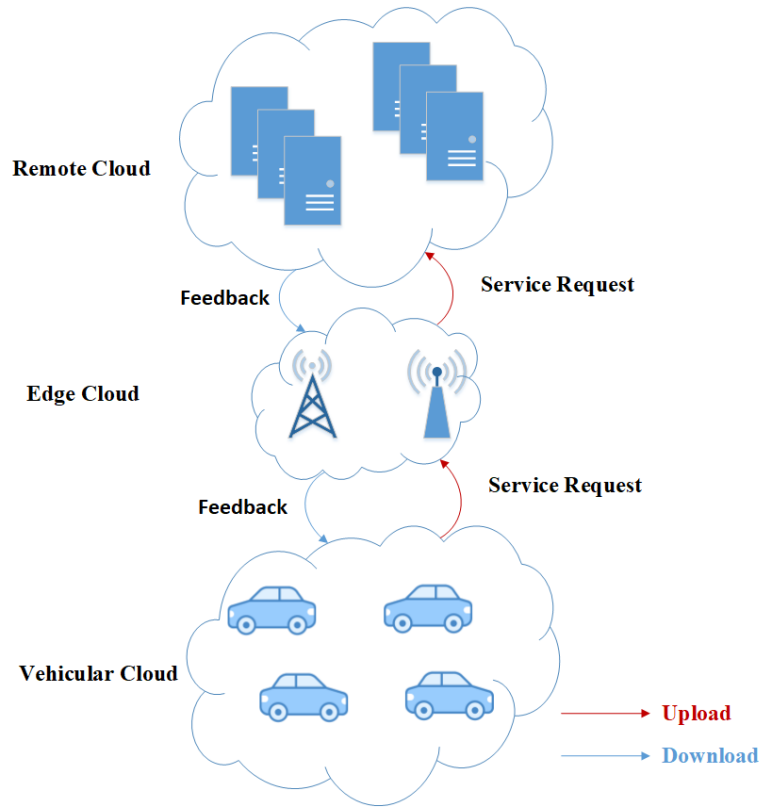


图 2: 车联网下移动边缘计算的计算资源分类图

辆看成自组网，车辆产生的任务全部在本地或者其他车辆执行，不会卸载到远程云或者边缘云。[9][10]将计算资源分为RC和VC，卸载时会优先考虑在VC上执行，当VC的资源不足时才会考虑卸载到RC。[11]利用基站进行一片区域内的车辆的任务进行协同调度，但基站自身并不会计算任务，只负责调度。[13]则将VC也看作是边缘计算移动资源，利用基站将EC的任务卸载到VC执行。[15]则完整的利用了RC、EC和VC这三种计算资源，优先考虑在更靠下的地方执行任务。

3.2 任务卸载的优化目标

在车联网环境下任务卸载常见的优化目标有如下几点：

1)最小化执行延迟。由于目前车辆中有很多延迟敏感型的任务，但是由于本地的计算能力有限，升级硬件成本又太高，如果计算延迟太高，会影响用户的体验。因此车辆应用程序中的任务卸载很多情况下是注重计算的延迟，即通过成熟的网络将计算任务卸载到其它计算平台执行，再将结果返回。

2)最小化能量消耗。移动设备的能耗通常是移动边缘计算中经常考虑的问题。在车辆边缘计算中，多数文献中，都不考虑车辆的能耗，因为他们认为和车辆运行消耗的能量相比用于计算的能量很少，可以忽略不计。但是也有一些文献将能量消耗考虑在内，将其看作最终优化目标的一部分。

3)优化服务质量。有些应用对服务质量有比较高的要求，比如视频中的物体识别，如果所计算的图片像素质量太差，那么结果也不会很好，从而导致服务质量很差，因此服务质量也是一个通常会考虑在内的优化目标。

在建模时，可以根据不同的卸载模型，考虑上述几种优化目标，制定一种恰当的优化目标。例如，[12][13][15]只将执行延迟作为优化的目标，不考虑车辆的能耗。[9][10][14]同时考虑了执行延迟和车辆的能量消耗，在延迟和能耗之间权衡。[11]则将执行延迟和服务质量联合作为共同优化的目标。

3.3 车辆的移动性分析

由于车辆具有移动性，因此由车辆形成的车辆云VC相对于固定基础设施和远程云RC来说是非常不稳定的，在考虑车辆计算任务卸载时，需要将车辆的移动性包括在内。车辆在行驶时对于固定基础设施而言的速度差会比较大，对于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些。不同的模型车辆移动性的要求也不同。

在一定区域内车辆形成云并由某固定基础设施进行统一管理时，车辆在进入或者离开这个区域时要向管理者进行汇报，如果离开的车辆仍然有计算任务在执行，那么任务会执行失败，这时需要对系统进行惩罚；如果车辆离开时没有任务正在执行，则可以顺利离开不会对系统造成损失。当车辆自组织不形成车辆云时，则需要考虑车辆之间的相对速度，速度差越小，通信的链路越稳定、越持久，因此在卸载时要选择速度差比较小并且有充分计算资源的车辆。

例如，[12]形成的自组织网络只会将任务卸载到和自己速度差较小的移动的车辆上，而不会卸载对向车辆或是固定的边缘设备上。[9][10][11]则将一定范围内的车辆划分成车辆云，车辆进入或者离开都会对系统的计算资源产生影响，因此在建模时需要认真考虑车辆移动所带来的影响。[16]则是利用静止的车辆，由于车辆没有移动性，因此车辆云VC就相当于边缘的固定设备。

4 重要方法与工具

4.1 马尔可夫决策过程(Markov Decision Processes,MDP)

马尔可夫决策过程是指决策者周期地或连续地观察具有马尔可夫性的随机动态系统，序贯地作出决策。即根据每个时刻观察到的状态，从可用的行动集中选用一个行动作出决策，系统下一步的状态是随机的，并且其状态转移概率具有马尔可夫性。决策者根据新观察到的状态，再做新的决策，依此反复地进行。[9][10]使用了半马尔可夫来最小化车辆计算任务的能量和时间消耗。[17]考虑了资源密集型和延迟敏感型移动应用的云资源环境的多资源分配问题，将资源分配模型表示为半马尔可夫决策过程，并使用线性规划技术解决优化问题。

MDP模型是一个四元组 (S, A, T, R) ，其中：

S 是一个有限集，其中每个元素 $s \in S$ 代表一个状态

A 是一个有限集，其中每个元素 $a \in A$ 代表一个行动

$T : S \times A \rightarrow \Pi(S)$ 称为状态转移函数，将每一对“状态——行动”映射为 S 上的一个概率分布，用记号 $T(s, a, s')$ 表示在状态 s 上执行 a 达到 s' 的概率

$R : S \times A \rightarrow \mathcal{R}$ 称为回报(Reward)函数， $R(s, a)$ 表示在 s 上执行行动 a 所得到的即时回报(直接回报)

MDP的核心问题是为决策者找到一个策略 π 计算预期(长期)回报，用值函数(value function)度量预期回报。

4.2 博弈论

博弈论作为一种有效的分布式资源分配方法，在移动边缘计算中得到了广泛的应用。博弈论主要研究多个个体或团队之间在特定条件制约下的对局中利用相关方的策略，而实施对应策略。它考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。表面上不同的相互作用可能表现出相似的激励机构。具有竞争或对抗性质的行为称为博弈行为。在这类行为中，参加斗争或竞争的各方各自具有不同的目标或利益。为了达到各自的目标和利益，各方必须考虑对手的各种可能的行动方案，并力图选取对自己最为有利或最为合理的方案。比如日常生活中的下棋，打牌等。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。[14][16]等文章使用博弈论来解决资源分配的问题并取得了很好的效果。

4.3 群体智能优化算法

群体智能优化算法主要模拟了昆虫、兽群、鸟群和鱼群的群集行为，这些群体按照一种合作的方式寻找食物，群体中的每个成员通过学习它自身的经验和其他成员的经验来不断地改变搜索的方向。它突出特点就是利用了种群的群体智慧进行协同搜索，从而在解空间内找到最优解。常见的智能优化算法如以下几类：蚁群算法(Ant Colony Optimization,ACO)、粒子群优化算法(Particle Swarm Optimization,PSO)、人工蜂群算法(Artificial Bee Colony Algorithm,ABC)等。[11]中采用基于二进制的粒子群优化算法(BPSO)对NP-hard问题进行求解，[12]则使用蚁群算法进行求解。

4.4 仿真工具

1)SUMO

SUMO[18](Simulation of Urban MObility)是一种开源、高度可移植、微观和连续的交通仿真软件包，旨在处理大型道路网络。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、使用各种各样的输入源的需求生成和路由工具、高性能模拟用于单路口以及整个城市，包括TraCI接口，以在线调整模拟以及大量其他工具和脚本。

2)OMNET++

OMNET++[19]是一个开源的，基于C++的离散事件模拟器，用于通信网络、多处理器和其他分布式或并行系统的建模。开发OMNeT++的动机是生成一个强大的开源离散事件仿真工具，可由学术、教育或面向研究的商业机构用于计算机网络和分布式或并行系统的仿真。OMNeT++是完全可编程和模块化的，它从头开始设计，以支持由可重用模型组件构建的非常大的网络建模。

3)Veins

Veins[20](Vehicles in Network Simulation)框架包括一套全面的模型，使车辆网络模拟尽可能现实，而不降低仿真的速度，由网络模拟器OMNeT++和道路交通模拟器SUMO组成。道路交通仿真是由SUMO进行的，网络仿真由OMNeT++和物理层建模工具包MiXiM共同完成，使得对无线电干扰以及静止和移动障碍物的遮挡使用精确的模型成为可能。

4)VeinsLTE

VeinsLTE[21]是Veins的延伸，是一个开源的车辆间通信模拟器。由于Veins目前不支持异构网络，特别是Long Term Evolution(LTE)，因此提出了VeinsLTE。VeinsLTE通过交通控制接口(TraCI)将道路交通模拟器SUMO与OMNET++连接起来。在VeinsLTE中，仿真中的车辆可以通过DSRC相互交换数据，也可以通过LTE连接基站。

以上的车联网下的仿真工具使得移动车辆的计算任务卸载的仿真极具说服力，尤其是Veins和VeinsLTE，它们将SUMO与OMNET++完美连接，可以更加真实的模拟现实车辆任务卸载。[12]利用Veins仿真工具，将车辆自组织，进行任务的调度。[11]利用VeinsLTE，允许车辆先于基站进行任务分配请求，然后根据分配结果将任务直接传输到对应的车辆。

5 总结

本文介绍了车辆边缘计算的研究的现实价值和意义，它可以充分利用车辆中空闲的计算资源，减少云资源的压力，并能够较少服务延迟。根据前面的研究工作，总结了车辆边缘计算的几种常见的模型，按照组成可以大概分为远程云、边缘云和车辆云，主要的优化目标则是计算延迟、能量消耗和服务质量，然后可以采用不同的方法对优化目标进行求解，以达到最终目标。但是随着5G网络的变革，车辆边缘计算将越来越有发展和利用的价值，其中的问题也值得进一步的深思。

参考文献

- [1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. A view of cloud computing. *International Journal of Computers and Technology*, 4(2b1):50–58, 2013.
- [2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016.
- [3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, Feb 2018.
- [4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys Tutorials*, 19(4):2322–2358, Fourthquarter 2017.
- [5] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, June 2016.
- [6] Y. Xiao and Chao Zhu. Vehicular fog computing: Vision and challenges. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 6–9, March 2017.
- [7] Y. Ren, F. Liu, Z. Liu, C. Wang, and Y. Ji. Power control in d2d-based vehicular communication networks. *IEEE Transactions on Vehicular Technology*, 64(12):5547–5562, Dec 2015.
- [8] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro. Lte for vehicular networking: a survey. *IEEE Communications Magazine*, 51(5):148–157, May 2013.
- [9] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen. An smdp-based resource allocation in vehicular cloud computing systems. *IEEE Transactions on Industrial Electronics*, 62(12):7920–7928, Dec 2015.
- [10] C. Lin, D. Deng, and C. Yao. Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units. *IEEE Internet of Things Journal*, 5(5):3692–3700, Oct 2018.
- [11] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, and A. Ylä-Jääski. Folo: Latency and quality optimized task allocation in vehicular fog computing. *IEEE Internet of Things Journal*, pages 1–1, 2018.
- [12] J. Feng, Z. Liu, C. Wu, and Y. Ji. Ave: Autonomous vehicular edge computing framework with aco-based scheduling. *IEEE Transactions on Vehicular Technology*, 66(12):10660–10675, Dec 2017.
- [13] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, L. Gui, and X. Shen. Cooperative task scheduling for computation offloading in vehicular cloud. *IEEE Transactions on Vehicular Technology*, 67(11):11049–11061, Nov 2018.
- [14] J. Klaimi, S. Senouci, and M. Messous. Theoretical game approach for mobile users resource management in a vehicular fog computing environment. In *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 452–457, June 2018.
- [15] F. Lin, Y. Zhou, G. Pau, and M. Collotta. Optimization-oriented resource allocation management for vehicular fog computing. *IEEE Access*, 6:69294–69303, 2018.

- [16] X. Huang, R. Yu, J. Liu, and L. Shu. Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications. *IEEE Access*, 6:66649–66663, 2018.
- [17] Y. Liu, M. J. Lee, and Y. Zheng. Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. *IEEE Transactions on Mobile Computing*, 15(10):2398–2410, Oct 2016.
- [18] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo - simulation of urban mobility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.
- [19] András Varga. The omnet++ discrete event simulation system. 2001.
- [20] C. Sommer, R. German, and F. Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, Jan 2011.
- [21] F. Hagenauer, F. Dressler, and C. Sommer. Poster: A simulator for heterogeneous vehicular networks. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 185–186, Dec 2014.