

# 面向云网融合的细粒度多接入边缘计算架构

王璐<sup>1</sup> 张健浩<sup>1</sup> 王廷<sup>2</sup> 伍楷舜<sup>1</sup>

<sup>1</sup>(深圳大学计算机与软件学院 广东深圳 518060)

<sup>2</sup>(上海市高可信计算重点实验室(华东师范大学) 上海 200062)

(wanglu@szu.edu.cn)

## A Fine-Grained Multi-Access Edge Computing Architecture for Cloud-Network Integration

Wang Lu<sup>1</sup>, Zhang Jianhao<sup>1</sup>, Wang Ting<sup>2</sup>, and Wu Kaishun<sup>1</sup>

<sup>1</sup>(College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060)

<sup>2</sup>(Shanghai Key Laboratory of Trustworthy Computing (East China Normal University), Shanghai 200062)

**Abstract** Nowadays, a paradigm shift in mobile computing has been introduced by the ever-increasing heterogenous terminal devices, from the centralized mobile cloud towards the mobile edge. Multi-access edge computing (MEC) emerges as a promising ecosystem to support multi-service and multi-tenancy. It takes advantage of both mobile computing and wireless communication technologies for cloud-network integration. However, the physical hardware constraints of the terminal devices, along with the limited connection capacity of the wireless channel pose numerous challenges for cloud-network integration. The incapability of control over all the possible resources (e.g., computation, communication, cache) becomes the main hurdle of realizing delay-sensitive and real time services. To break this stalemate, this article investigates a software-defined fine-grained multi-access architecture, which takes full control of the computation and communication resources. We further investigate a Q-Learning based two-stage resource allocation strategy to better cater the heterogenous radio environments and various user requirements. We discuss the feasibility of the proposed architecture and demonstrate its effectiveness through extensive simulations.

**Key words** cloud-network integration; multi-access edge computing (MEC); fine-grained access network; software-defined network; deep reinforcement learning

**摘要** 随着智能终端设备的爆发式增长,多接入边缘计算(multi-access edge computing, MEC)成为支持多服务、多租户生态系统的关键技术之一.多接入边缘计算通过结合云端的移动计算技术和接入网

收稿日期:2020-12-26;修回日期:2021-03-22

基金项目:国家自然科学基金项目(61872246, U2001207, 61872248);广东特支计划;广东省自然科学基金项目(2017A030312008);深圳市基础研究项目(ZDSYS20190902092853047);广东省高等学校科技创新项目(2019KCXTD005);广东省“珠江人才计划”(2019ZT08X603)

This work was supported by the National Natural Science Foundation of China (61872246, U2001207, 61872248), Guangdong Special Support Program, the Natural Science Foundation of Guangdong Province of China (2017A030312008), the Basic Research Program of Shenzhen (ZDSYS20190902092853047), the Science and Technology Innovative Program of Higher Education of Guangdong Province (2019KCXTD005), and the “Pearl River Talent Recruitment Program” of Guangdong Province (2019ZT08X603).

通信作者:伍楷舜(wu@szu.edu.cn)

的无线通信技术,实现了云端和网络的高效融合.然而,目前的边缘计算技术对于所有可能的资源(例如计算、通信、缓存)并没有细粒度的控制能力,因此并不能对延迟敏感的实时服务提供很好的支持.为了解决这个关键问题,设计了一种基于软件定义(software defined)的细粒度多接入边缘计算架构,可以对网络资源和计算资源进行细粒度的控制并进行协同管理,并设计了一种基于深度强化学习 Q-Learning 的两级资源分配策略,从而提供更有效的计算卸载和服务增强.大量的仿真实验证明了该架构的有效性.

**关键词** 云网融合;多接入边缘计算;细粒度接入网;软件定义网络;深度强化学习

**中图法分类号** TP393

近年来,平板电脑、智能手机、大规模传感器以及各式各样的异构物联网设备变得越来越流行,已经成为我们日常生活中的主要计算资源<sup>[1]</sup>.据保守估计,到 2022 年,将有 500 亿台终端设备进行互联<sup>[2]</sup>.随着终端设备的爆炸式增长,为终端设备设计的应用程序也大量涌现,如交互式游戏、自然语言处理、面部识别、增强现实等.这种类型的应用程序往往需要大量的资源,包括密集的计算资源和高速的传输资源.随着新颖的交互式移动应用程序的日益丰富和终端设备功能的日益强大,我们正处在移动计算的重大变革之中.

近期的研究进展见证了移动计算的模式转变.在终端设备不断产生的海量数据驱动下,集中式的移动云计算正在向移动边缘计算进行迁移.计算、存储和网络资源均在基站(base station, BS)端集成<sup>[3-4]</sup>.网络边缘大量空闲的计算资源和存储空间可以被充分利用,以完成计算密集型和延迟关键型的计算任务<sup>[5-6]</sup>.随着各种计算和存储资源越来越贴近终端用户,移动边缘计算有望为资源消耗庞大的应用程序提供超低延迟和超低网络拥塞的服务.

终端设备的爆发式增长使得无线连接成为发掘移动边缘计算潜力的关键技术之一<sup>[7]</sup>.因此,移动边缘计算的适用范围已经扩展到无线接入网络(radio access network, RAN)内,以提供边缘计算的能力.这也称为多接入边缘计算(multi-access edge computing, MEC).在多接入计算架构中,边缘计算的资源可以部署在 LTE 基站(eNodeB)、3G 无线电网络控制器(radio network controller, RNC)或多天线聚合基站中.多接入边缘计算将移动计算和无线通信 2 个学科的理论和技术进行了深度融合,是云网融合的典型技术之一,因此受到学术界和工业界研究人员的一致倡导.可以预见,将新型的无线网络技术与面向服务的边缘-云体系结构进行结合,可以显著降低网络拥塞和用户延时,提高用户的服务质

量(quality of service, QoS)和服务体验(quality of experience, QoE)<sup>[8-9]</sup>,为终端应用程序、内容提供商和第三方运营商提供更好的服务.

尽管研究人员在多接入边缘计算方向进行了不断的尝试和大量的努力,然而,由于终端设备的物理硬件限制以及无线信道的连接能力限制,多接入边缘计算依然面临着诸多挑战<sup>[10]</sup>.目前的已有工作大多考虑粗粒度的资源分配策略,包括传输、计算和存储资源,这些分配策略缺乏对所有可能资源的细粒度控制,这成为实现延迟敏感服务的主要障碍.目前亟需新的多接入边缘计算架构,可以对所有的资源进行细粒度的灵活分配.具体来说,对于新的多接入边缘计算架构,我们考虑 3 个问题:

- 1) 是否存在一种细粒度的多接入边缘计算架构,可以对网络资源和计算资源进行细粒度的协同优化,从而更好地支持终端多样化的云网融合?
- 2) 如果存在这样的架构,那么,应该如何设计接入网的信道接入策略,从而使终端用户可以并发地使用细粒度的网络资源?
- 3) 基于细粒度的网络接入策略,如何灵活地进行计算任务卸载,并对边缘计算资源进行分配,从而更好地支持多服务、多租户生态系统?

针对这 3 个问题,本文将系统地研究一种面向云网融合的细粒度多接入边缘计算架构.本文从网络底层入手,通过研究媒体接入控制层(media access control layer, MAC)和物理层(physical layer, PHY)的特性,讨论细粒度接入策略简单高效的解决方案、优化策略及系统结构,并结合边缘计算的计算资源分配需求,对计算卸载服务进行协同优化,从而较大程度地减轻网络传输负担,提高计算卸载效率.具体来说,本文提出了一种基于软件定义的细粒度多接入边缘计算架构,通过对网络资源和计算资源的灵活控制,可以更好地满足异构物联网服务的服务质量.具体来说,首先,由于无线接入网建立在多接入

策略之上为异构物联网提供服务,因此本文解决了如何利用物理层/MAC层的切片来支持多用户简单高效的并发传输.进一步地,本文设计了一种基于深度强化学习的资源分配策略,可以对资源分配进行自适应的学习,同时,通过设计一种基于软件定义的多接入边缘计算架构,将控制平面与数据平面解耦,可以对所有可能的资源,包括网络资源和计算资源,进行统一的细粒度管理.最后,本文通过大规模的仿真实验,验证了该架构的有效性和可靠性.总的来说,本文的主要贡献有3个方面:

1) 提出了一种基于软件定义的多接入边缘计算架构,通过对物理层/MAC进行简单高效的切片设计,可以对通信资源和计算资源进行细粒度的控制,从而更好地支持复杂的底层环境,满足多样的用户需求.

2) 基于所提出的多接入边缘计算架构,我们设计了一种基于深度强化学习的资源分配策略,通过对网络资源和计算资源的自适应训练和学习,实现了低延时的计算卸载服务.

3) 利用网络仿真软件NS3进行了大规模的仿真实验,并通过实验验证了所提架构的有效性.实验证明,相比于传统的MEC架构,该架构性能提高30%以上.

## 1 相关工作

由于物联网中计算密集型应用程序的大量涌现,以及工业物联网、车联网等新型业务的爆发式增长,网络业务需求日益复杂.大量的研究人员将精力投入到云计算与网络架构的融合设计中去,从而协同应对计算与传输的双重挑战.

Cloudlet是首个将云计算放在网络边缘的计算模式<sup>[11]</sup>,其主要目的是为了支持资源匮乏的移动用户运行资源密集型和交互式应用程序.事实上,Cloudlet将计算资源接近移动用户的这种思想和WiFi的概念很相似,WiFi的设计目的就是为了让移动用户可以便捷地访问互联网资源<sup>[12]</sup>.随后,Cisco提出了雾计算,将云计算从网络核心扩展到网络边缘,减少了需要向中央云系统所传输的数据量<sup>[13]</sup>.因此,大部分的密集计算,以及终端用户收集到的数据,都可以由雾计算中网络边缘的雾节点进行处理和分析,从而大大减少了计算延迟和网络拥塞.然而,Cloudlet和雾计算并没有将计算资源整合到移动网络的体系结构中去.因此,Cloudlet和雾计

算的节点常常由私有企业进行部署,很难为移动用户提供有QoS和QoE保证的计算服务<sup>[14-15]</sup>.

云无线接入网络(cloud radio access network, C-RAN)首次将云计算的概念引入无线接入网中.在C-RAN架构中,传统BS被分布式远程无线电头(remote radio heads, RRH)和集中式基带单元(baseband units, BBU)所取代.而传统由BS负责的基带信号迁移到中央BBU中进行<sup>[16-17]</sup>.通过这种去耦的方式,RRH只需要负责基本的射频功能,因此可以大容量地网络接入,而中央BBU则提供大规模的信号处理,例如集中编码和解码,以及联合波束成形和资源分配.

2014年末,欧洲电信标准研究所(European Telecommunications Standards Institute, ETSI)制定了移动边缘计算的行业规范,并首次提出了MEC的概念.作为C-RAN架构的补充,MEC旨在对接入网和云服务进行融合,将传统的云计算功能融入无线接入网中,从而更接近移动用户端<sup>[17]</sup>.因此,MEC可以支持多种终端用户应用,例如无人驾驶、虚拟现实(virtual reality, VR)/增强现实(augmented reality, AR)、沉浸式媒体等.为了通过异构接入网,如5G, WiFi, LoRa来发掘更多MEC的潜力,ETSI在2017年将移动边缘计算正式更名为多接入边缘计算<sup>[18]</sup>.经过这次修订,MEC服务器可以由网络运营商在接入网的任何位置进行部署,如BS(4G中的eNodeB、5G中的gNodeB)、光纤网络单元、WiFi接入点等.这次转型将计算智能迁移到网络边缘,可以更好地对网络资源和计算资源进行融合.

在MEC中,网络和计算的资源分配一直是关系到MEC性能的关键问题.文献<sup>[19]</sup>提出了一种集中式的多用户资源分配策略<sup>[19]</sup>.通过对计算、存储和通信资源的协同优化,实现了较高性能的计算卸载策略.集中式的分配策略可以对资源进行协同优化,却具有较高的计算复杂度和传输代价,因此并不适用于分布式的多接入边缘计算架构.文献<sup>[20]</sup>提出了一种分布式的多用户MEC资源分配和计算卸载策略<sup>[20]</sup>.通过交替方向乘子法(alternating direction method of multipliers, ADMM),文献<sup>[20]</sup>实现了较为高效的分布式资源分配策略.

近几年,人工智能与网络的结合取得了巨大的进步.由于MEC资源优化问题的复杂性,很多学者将深度强化学习(deep reinforcement learning, DRL)引入MEC之中.通过和网络环境不断地进行交互,

从而自动学习不同情况下的最优分配策略.文献[21]将 MEC 的计算卸载问题描述成一个 Markov 决策过程(Markov decision process, MDP),并将用户和 Cloudlet 节点间的信道质量引入状态决策之中<sup>[21]</sup>.文献[22]将 Q-Learning 和深度学习(deep learning)进行结合,从而自适应地学习最优卸载决策和卸载速率<sup>[22]</sup>.

网络切片技术是 5G 新无线电(new radio, NR)的关键技术之一.切片技术允许运营商共享相同的网络基础架构资源,如频谱、基站等<sup>[23]</sup>.通过切片技术,运营商可以基于单一物理设施为用户提供多虚拟网络运行服务,满足个性化的业务需求.第三代合作伙伴计划(3rd generation partnership project, 3GPP)将网络切片定义为一项使运营商能够创建不同的网络来满足和优化不同市场需求的技术<sup>[24]</sup>.而 ITU-T 将网络切片定义为逻辑隔离网络分区(logical isolated network partitions, LNP).隔离分区由多种虚拟资源组成,这些资源相互隔离并配有可编程控制和数据平面<sup>[25]</sup>.大量的现有研究已解决了网络中不同级别的切片问题<sup>[26-30]</sup>.在文献[26]中,作者提出了一种基于部分频域复用的资源共享协议政策,通过比较数据间的相关性来减少丢包.在文献[27]

中,作者设计了一种切片资源的分配方案.通过调节切片资源的利用率来提高网络的整体性能.在文献[28]中,作者通过减少片间干扰来解决相邻基站之间的干扰问题.然而,目前很少有工作从接入网的物理层切片技术入手,讨论无线网络的资源优化问题.文献[29]第 1 次着重阐述了物理层切片对于无线网络优化的重要性,并提出了自适应的物理层切片分配算法.在文献[30]中,作者进一步提出,物理层网络切片要和边缘计算进行联合,并设计了相应的优化方案 SI-EDGE,对 MEC 资源进行统一优化.

然而,当前的 MEC 架构没有将网络特性和计算特性进行很好地融合.而且,由于资源的分配过于粗粒度,并不能很好地适应底层网络复杂多变的情况和上层用户动态多样的资源需求.如图 1 所示,本文旨在提出一种细粒度的多接入边缘计算架构,将接入网的信道传输特性和基于软件定义的计算资源进行深度融合,通过深度强化学习,对所有可能的资源进行协同优化,以满足资源的高效利用.

为了实现以上目标,我们要从网络底层入手,重新思考整个网络的架构.因此,当前的设计面临着 3 个挑战:

1) 为了保证接入网的信道资源可以细粒度地

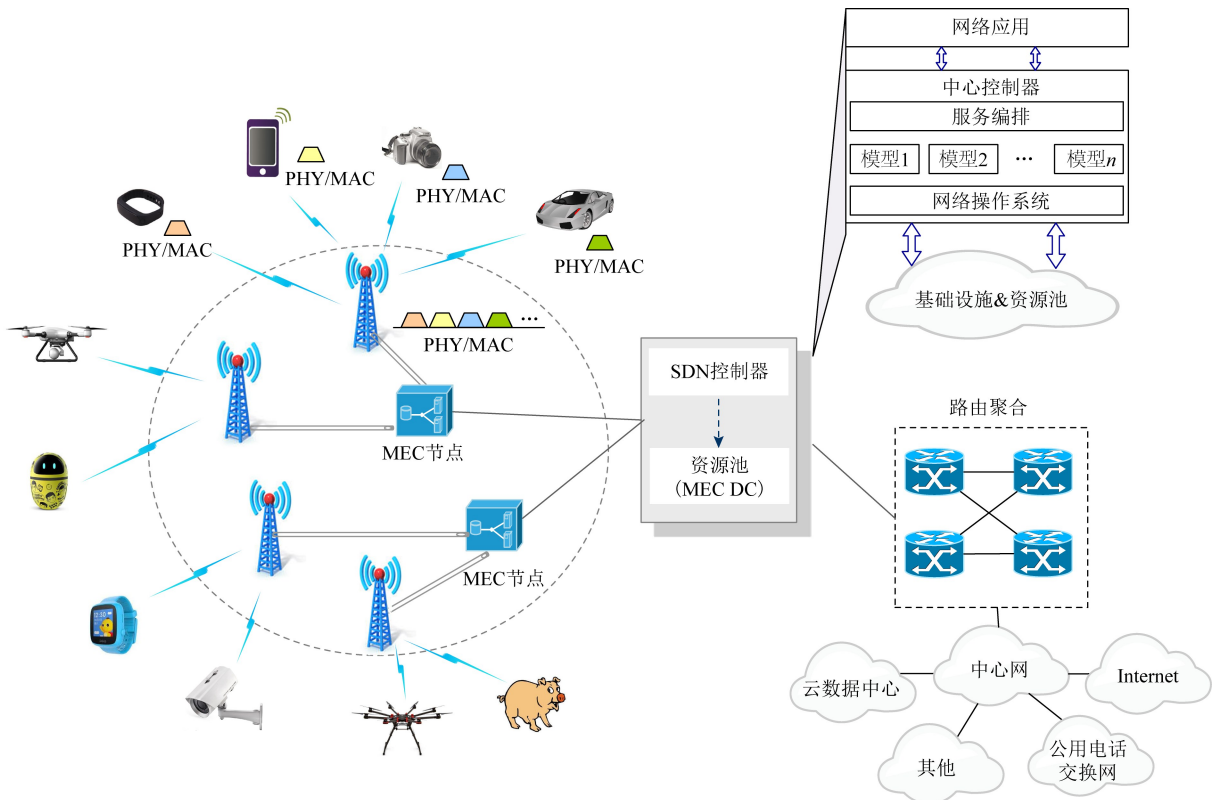


Fig. 1 Fine-grained multi-access edge computing architecture for cloud-network Integration

图 1 面向云网融合的细粒度多接入边缘计算架构

访问,接入网的物理层要实现切片,并对传输信号进行有效分离.这样,终端用户才可以在传统的接入网信道中细粒度地使用信道资源,实现按需的混合MAC并发传输.然而,如何在不改变原有调制编码的情况下,有效地将不同用户的数据区分开,进行MAC协议协同合作,是一个严峻的挑战.

2) 传统的 MEC 架构基于固定的网络部署,这往往会增加资源调用的代价.基于软件定义的网络控制器应该如何部署,才能结合接入网的网络特性,尽可能减少控制信息传输代价,进行有效的资源卸载,也是一个继续思考的问题.

3) 在保证细粒度的网络资源接入和灵活的计算资源部署之后,如何利用深度强化学习相关技术,更好地结合云网各自的优势来实现资源的有效利用,是本文需要解决的重要问题.

## 2 面向云网融合的细粒度物理层/MAC层切片

本节我们主要介绍面向云网融合的细粒度物理层/MAC层切片.该架构从网络底层入手,通过物理层/MAC层切片,可以实现接入网细粒度的混合MAC并发传输.

### 2.1 物理层/MAC层切片概述

由于智能终端用户的传输需求动态多样,加之无线接入网底层的环境复杂多变,传统单一且粗粒度的MAC接入方式已经不能应对如此巨大的挑战.目前急需细粒度且灵活的MAC接入方式,来满足不同用户的传输需求.因此,本文从接入网的底层入手,首先提出了一种基于物理层和MAC层的切片设计.图2描述了基于物理层/MAC层切片设计

的协议栈.原始的物理信道被切分成若干个细粒度的单元(例如,由若干子载波组成的子信道),每个子信道可以支持一种MAC接入方式.这样,不同的终端设备就可以根据自己的信道质量和传输需求,在同一个信道通过不同的MAC协议进行接入,从而最大化地对信道资源进行利用.具体来说,物理层/MAC层切片建立在正交频分复用技术(orthogonal frequency-division multiplexing, OFDM)之上.通过对物理层子载波之间进行解耦和,从而实现细粒度的MAC访问.物理层/MAC层切片为终端用户接入网络提供了灵活、自适应的传输承诺.多种MAC访问在频域上的并发传输,可以更好地迎合终端用户动态多样的传输需求.在这种背景下,协议栈配有一个软件定义网络(software defined network, SDN)控制器,运行在物理层/MAC层切片之上,负责物理层子信道和MAC层协议的分配.通过及时调整物理层的子信道资源和MAC层的接入协议,SDN控制器旨在利用底层信道的多样性,对物理层和MAC层资源进行协同优化,最大化地提高资源利用率.

### 2.2 物理层/MAC层切片设计

为了使各式各样的智能终端设备可以利用物理层/MAC层切片设计进行细粒度的信道接入,本文设计中使用的基本访问粒度是元信道(meta channel).每个元信道由一组OFDM子载波组成.以无线局域网(wireless local area network, WLAN)为例,对于一个具有64个快速傅里叶变换(fast Fourier transform, FFT)点的20 MHz信道,通常每16个子载波(带宽为1.33 MHz)组成一个元信道,而若干元信道组成一个子信道.子信道可以是连续的,也可以在频域上相对分离.子信道的大小由SDN控制器统一进行安排,一般遵循2个标准:1)每个子信道的分配应足够宽,以确保满足终端用户的传输需求;2)子信道的分配也应该足够窄,从而充分利用频率选择效应进行适合的MAC传输.当一种MAC协议在一组子信道上进行接入时,剩余的子信道资源由SDN控制器进行分配,分配主要依据2个原则:终端用户的信道质量和传输计算需求.其中,子信道的质量由终端用户进行评估,并反馈给SDN控制器;而传输计算需求则在竞争轮询时将请求报告给SDN控制器.由于传统的时间竞争策略大大增加了传输代价,因此,本文设计了一种频域竞争策略.与传统的时域竞争相比,频域竞争策略可以充分利用频域资源.在频域竞争策略中,终端用户可以利用不同的频域同时进行资源

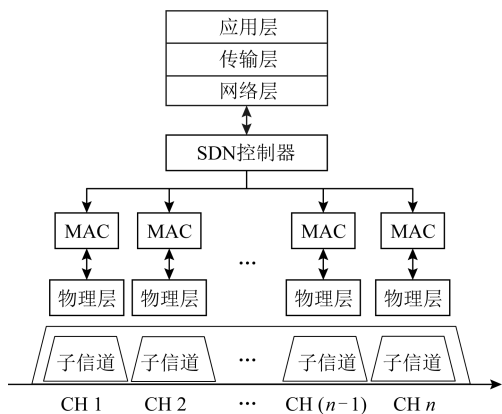


Fig. 2 Design of edge computing protocol stack for cloud-network integration

图2 面向云网融合的边缘计算协议栈设计

需求的请求,因此可以避免时域竞争的传输冲突,大大降低了竞争的开销.

频域竞争策略的想法虽然听上去简单直接,然而,为了可以应用到细粒度的边缘计算架构中,需要设计两级 F-RTS/F-CTS 结构.图 3 给出了一个频域竞争策略的示例,整个信道分为 5 个子信道.当网络初始化之后,终端用户先等待 DCF 帧间间隔(DCF interframe space, DIFS),以完成同步.随后,在轮询

时段,终端用户通过频域竞争策略进行传输/计算需求的请求.当 SDN 控制器获得所有用户的需求之后,将运行资源分配策略,根据终端用户的信道质量和传输/计算需求,对所有可能的资源(如传输资源、计算资源和存储资源)进行分配,并在 F-CTS 中反馈分配结果.在等待 PCF 帧间间隔(PCF interframe space, PIFS)之后,终端用户可以根据通知结果,去接入自己分配的信道资源.

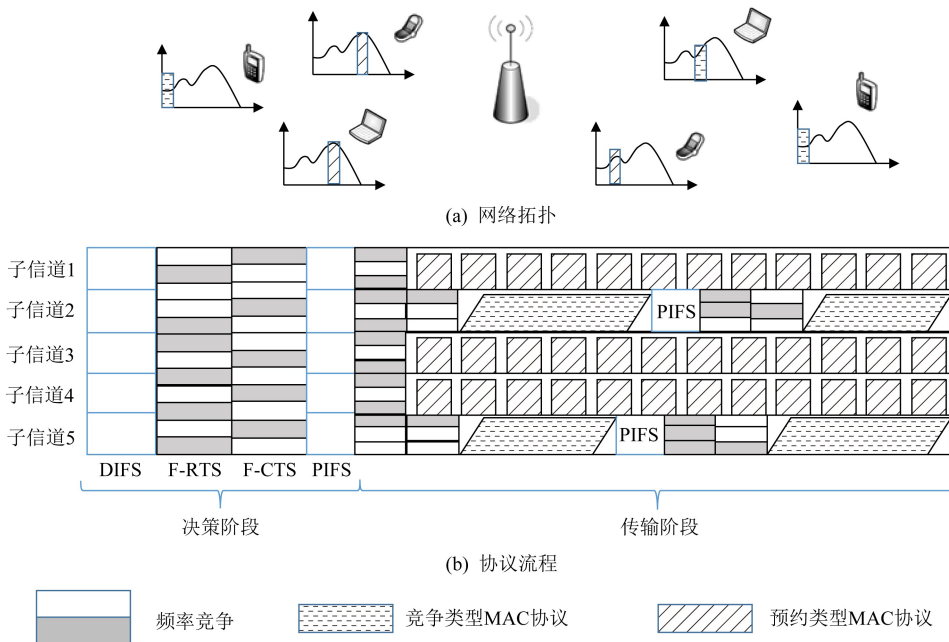


Fig. 3 Physical layer/MAC layer slice example

图 3 物理层/MAC 层切片示例

具体来说,两级 F-RTS/F-CTS 结构实现了两级的竞争轮询,分别是竞争/通知轮询阶段和传输轮询阶段.第 1 级的竞争/轮询阶段由 F-RTS/F-CTS 负责,用于信道和计算资源的分配.在第 2 级的传输轮询阶段,F-RTS/F-CTS 有另外的用途.如果终端用户被分配的是竞争类型的 MAC 协议,如载波监听多址接入(carrier sense multiple access, CSMA),即如图 3 中的子信道 2 和子信道 5,那么这 2 个子信道的 F-RTS/F-CTS 用于组织终端用户进行的接入网的访问.如果终端用户被分配的是预约类型的 MAC 协议,如图 3 中的子信道 1,3,4,那么这 3 个子信道的 F-RTS/F-CTS 用于安排传输时间表.因此,F-RTS/F-CTS 在这 2 层中具有不同的帧格式.举例说明,在第 1 级,F-RTS/F-CTS 用于资源的竞争,因此在频域中,F-RTS/CTS 包括 2 个部分:1)作为标识;2)作为竞争/通知频段.标识部分用来说明当前的帧是 F-RTS 还是 F-CTS.这部分标识放置在

帧的开头.以 64 点的 FFT 为例,通常来说,为了保证 BAM 的抗干扰能力,16 个子载波比较合适.然后,在竞争/通知频段,为了保证抗干扰性能,以每 4 个子载波为一个基本单元进行传输需求的请求,而每 12 个子载波为一个基本单元进行计算资源的请求.相应地,在通知阶段,子载波也分成 2 个部分:1)用来确认 MAC 协议的分配;2)用来确认子信道的分配.在第 2 级中,F-RTS/F-CTS 用于细粒度的接入网资源访问,由于终端设备的传输需求动态多样,第 2 级的传输可以用于不同类型的 MAC 协议进行并发传输.这部分的具体分配也由 SDN 控制器进行规划.例如,预约类型的 MAC 协议,如时分复用的多址接入(time division multiple access, TDMA)可以通过 F-CTS 进行传输调度安排.而竞争类型的 MAC 协议,如 CSMA 可以利用 F-RTS/F-CTS 进行细粒度子信道访问竞争.在第 2 级的 CSMA 竞争中,用户可以随机选择一个子载波作为其标识,并使

用该子载波在 F-RTS 期间发送 BAM 符号.如果该终端用户的访问被授予,SDN 控制器将在 F-CTS 期间在对应的子载波进行通知.通过这种方式,可以实现细粒度的接入网资源分配,最大化地提高接入网的频谱利用率和系统效能.我们进一步设计了基于 Q-Learning 的两级自适应资源分配学习策略,将在第 3 节中详细介绍.

### 3 基于软件定义的多接入边缘计算架构

本节我们将详细介绍基于软件的细粒度多接入边缘计算架构.传统的边缘计算架构无法充分地利用接入网和边缘-云节点的资源.在本文中,基于细粒度的物理层/MAC 层切片,边缘计算可以更好地利用接入网特性和 MEC 节点的资源进行协同优化.

#### 3.1 基于软件定义的多接入边缘计算架构设计

如图 1 所描述,具有存储和计算资源的 MEC 节点(例如服务器)部署在接入网之中,从而为终端用户提供具有弹性的网络服务,如计算卸载和服务缓存.这些 MEC 节点可以部署在 eNodeB、BS、宏站甚至是小型基站之中.此外,MEC 节点也可以部署在住宅区域,并通过边缘交换机或集成交换机进行接入访问.除了接入网络中的 MEC 节点之外,我们也在接入网和汇聚网之间根据需要部署具有适当规模的 MEC 数据中心(data center, DC),通常,汇聚网的聚合节点,例如公用电话交换网(public switched

telephone network, PSTN)、中央或移动电话交换局(mobile telephone switching office, MTSO),是部署 MEC 数据中心的理想场所,因为所有流量在接入到 Internet 之前,都会经过这些节点.并且,MEC DC 也是基于软件定义的数据中心,其中 MEC 节点(即包含计算、存储能力的资源池)根据需要由一个或多个 SDN 控制器进行控制.

图 4 给出了基于软件定义的多接入边缘计算分布式节点架构设计.这些 MEC 节点有几种不同的功能和角色,包括公共节点(common node, CNode)、区域代理节点(regional node, RANode)、超级节点(super node, SNode)和证书授权节点(certification authority, CA).这些 MEC 节点的角色描述为:

1) CNode.CNode 是最常见的 MEC 节点,分布在接入网中.CNode 用于为终端用户提供计算卸载服务,并为远程 Internet/云服务的服务缓存提供存储资源.CNode 是高度虚拟化的,并且虚拟机(virtual machine, VM)可以在 SNode 的控制下远程安装/迁移到 CNodes 上.

2) RANode. RANode 代表位于接入网中的区域代理 MEC 节点. RANode 由 SNode 进行选择.SNode 从接入网范围内的 CNode 中确定合适的 RANode. RANode 负责资源发现、管理/监视其区域内所有 CNode 的状态.尽管具有区域代理的作用,但 RANode 本身还是 CNode,可以执行计算卸载.

3) SNode.SNode 代表位于接入网和汇聚网之间 MEC 数据中心中的超级智能节点.每个 SNode

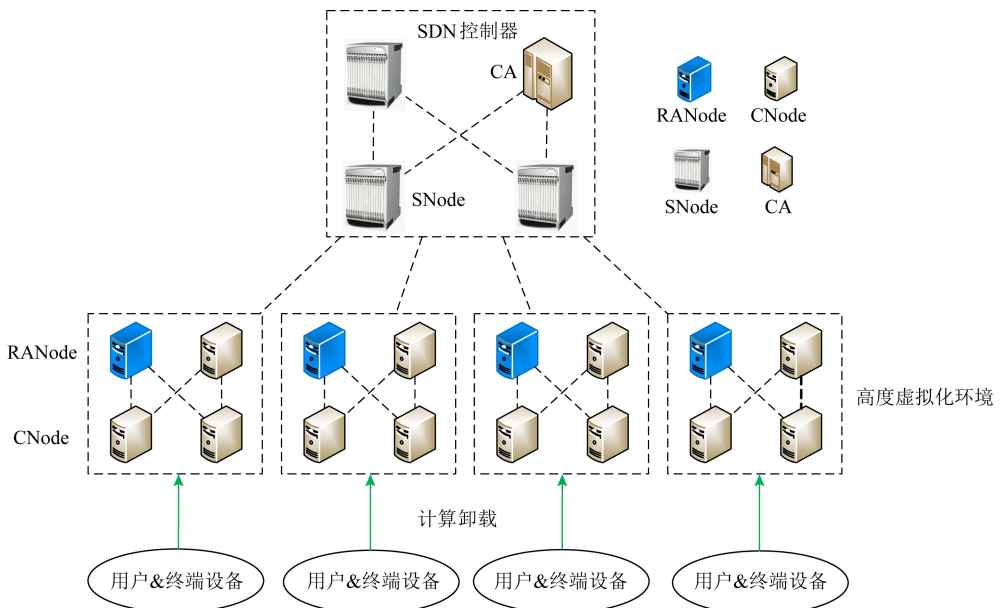


Fig. 4 Software-defined based multi-access edge computing architecture design

图 4 基于软件定义的多接入边缘计算架构设计

负责管理 SDN 控制器分配给它的一定数量的 CNode 和 RANode. SNode 的任务包括管理 CNodes/RANode 上的 VM 远程安装、节点的加入/离开(无缝扩展)、节点配置、用户管理等. SNode 由 SDN 控制器控制, 并且可以与其他 SNode 通信. SNode 还可以缓存从远程数据中心卸载的 Internet 服务和云服务. 进一步地, SNode 可以将部分缓存服务卸载到更接近终端用户的 CNode 或 RANode 中, 从而极大地提高计算卸载效率.

4) CA. 证书授权节点 CA 是位于 MEC 数据中心内的节点, 负责用户的证书生成和管理, 提供签名、授权和证书的功能, 并保留证书存储库内所有授权用户的信息.

### 3.2 云网资源分配问题建模

我们将面向云网融合的细粒度边缘架构抽象成图 5 所示的系统模型. 该模型设定为由  $m$  个用户  $client_i, i \in \{1, 2, \dots, m\}$ ,  $n$  个边缘计算节点  $CNode_j, j \in \{1, 2, \dots, n\}$ , SDN 控制器 SNode 和中心云所组成的多层次网络. 基于该系统模型, 我们将资源分配算法进行建模, 具体为:

1) 用户任务.  $ClientTask_i (i \in \{1, 2, \dots, m\})$ , 表示用户  $i$  所提交的请求.  $ClientTask_i$  又可细分为  $ClientTask_i = \{Mem_i, f_i\}$ . 其中,  $Mem_i$  表示任务所需内存大小, 代表该任务所需的存储能力;  $f_i$  则表示处理 1 b 数据所需的时钟周期, 代表该任务所需的计算能力.

2) 服务节点资源.  $ServerSource_j (j \in \{1, 2, \dots, n\})$ , 表示服务节点  $j$  拥有的资源量.  $ServerSource_j$  同样可细分为  $ServerSource_j = \{Mem_j, f_j\}$ .

3) MAC 协议.  $MACProtocol = \{TDMA, CSMA\}$ .

4) 信道质量 CSI.  $CSI = \{CSI_{c_1}, CSI_{c_2}, \dots, CSI_{c_m}\}$ .

该模型应符合约束条件, 即用户任务应小于服务节点资源  $ClientTask_i = \{Mem_i, f_i\} \leq \forall ServerSource_j = \{Mem_j, f_j\}, i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$ . 而用户所选择的 MAC 协议与信道质量 CSI 之间也存在关系:  $MACProtocol = \lambda \times CSI$ , 其中,  $\lambda$  是选择某一种 MAC 协议的概率.

给定了用户任务  $ClientTask_i$  和信道质量  $CSI_i$ , 我们的目标是优化整个系统的网络效用和计算效用, 即

$$Utility = \max \{ \Phi( ServerSource, ClientTask, MACProtocol, CSI) \} =$$

$$\max \sum_{i=1}^m \Phi_{ServerSource_i}^{ClientTask_i} \Phi_{CSI_i}^{MACProtocol_i}$$

优化问题复杂度: 这个优化问题可以转化成算法布林可满足性问题 (Boolean satisfiability problem, SAT)<sup>[31]</sup>. 由于给定变量, 包括  $Server, Task, CSI, MACProtocol$ , 都对应对应 SAT 问题中的一个析取范式  $C_i$ , 而 SAT 问题求解的实质是要确保每一个  $C_i$  中都能够存在 1, 从而保证子句  $C = 1$ . 对应于我们提出的问题, 其实质就是对应给定的一个  $Task$ , 能够正确地选择出合适的  $MACProtocol, CSI$  以及  $Server$  作为一组结果, 也即 SAT 问题的解. 因此, 我们将 SAT 问题规约到我们的算法中继续解决, 从而证明本算法属于 NP-Complete 问题.

### 3.3 基于 Q-Learning 的细粒度自适应资源分配算法

由于该问题是 NP-Complete 问题, 本节提出了一种基于深度强化学习 Deep Q-Learning (DQN)<sup>[32]</sup> 的细粒度自适应资源分配方法. 深度强化学习可以通过感知周围环境, 不断进行学习, 并根据环境的反馈进行策略调整, 最终学习到最优的策略. 在本文中, 我们设计了一种融合网络和云端特性的协同学习优化策略. 系统首先通过终端用户的接入网络特性对传输资源的分配进行细粒度的学习, 根据传输资源的学习结果, 再结合边缘计算节点的计算性能进行计算卸载的策略学习, 从而最大化地利用网络和计算资源.

具体来说, SNode 首先根据终端用户的信道质量  $CSI_i, i \in \{1, 2, \dots, m\}$ , 对于每一个用户的物理层/

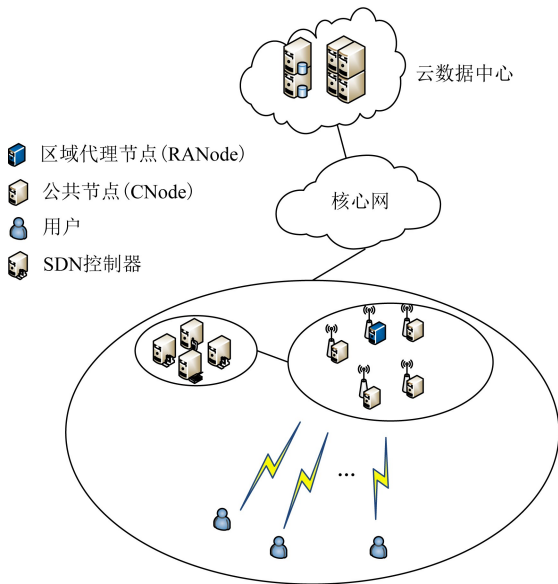


Fig. 5 Resource allocation model for cloud-network integration

图 5 面向云网融合的资源分配模型



MAC层策略进行自适应学习.假设当前一共有  $S$  个子信道  $channel_j, j \in \{1, 2, \dots, s\}$ , 用户  $client_i$  所需要卸载的计算任务可表示为  $Task_i = \{s_i, g_i\}, i \in \{1, 2, \dots, Q\}$ , 其中  $Q$  表示用户  $i$  卸载的任务总量,  $s_i$  表示任务数据输入大小,  $g_i$  则表示请求的服务器计算资源大小.则 SNode 需要为用户  $client_i$  确定使用的 MAC 策略  $MACProtocol_i$ .

1) 状态(State).control 拥有所有的终端用户信道质量,因此 control 表示的系统整体状态为

$$State = \{Subch_1, Subch_2, \dots, Subch_n, MAC_i\},$$

其中,Subch 表示信道资源,MACProtocol<sub>*i*</sub> 表示第  $i$  个用户分配的子信道 MAC 协议.

2) 动作(Action).动作表示为 control 为用户选择的 MAC 协议,即 Action 集合为

$$Action = \{MACProtocol_1, MACProtocol_2, \dots, MACProtocol_n\}.$$

3) 奖励(reward).奖励部分由执行动作前后的环境信息状态进行确定.如基于时延指标的环境信息,若  $State_t = 100$  s 以及  $State_{t+1} = 120$  s, 则易知存在  $-0.2$  的增长.因此,得到指标增长比:

$$IncreaseRatio = \frac{|State_t - State_{t+1}|}{State_t}. \quad (1)$$

式(1)等号右侧分子部分采用绝对值形式,是因为基于不同的指标形式,指标增长比所带来的效果是不一致的.对于时延指标来说,指标值增加带来负收益,而对于吞吐量等指标来说,指标值增长则是正收益.因此,需要依照具体指标形式,设定指标增长比计算公式.

依照上述分析,可获取指标值的增长比,但是为了防止出现过拟合现象以及 Q 表更新合理性分析,设定为

$$reward = \begin{cases} 0, & IncreaseRatio > 0 \wedge \text{MAC 无变化}, \\ \lambda \times IncreaseRatio, & \text{其他}. \end{cases} \quad (2)$$

式(2)中,在已经获取优势 MAC 协议后,设定  $reward = 0$ , 以防止 Q 表值过大, Q-Learning 收敛过快;  $\lambda$  则为衰减因子,防止 IncreaseRatio 过大,从而导致 reward 过大、过快收敛.算法 1 描述了基于 Q-Learning 的自适应 MAC 分配协议.

**算法 1.** 基于 Q-Learning 的 MAC 协议选择算法.

输入:网络指标类型;

输出:网络性能指标.

① 初始化 Q 表、初始状态;

重复:

② 由当前状态以及网络指标类型确定当前网络状况;

③ 利用 Q 表或者  $\epsilon$ -贪婪策略,基于当前网络状况选择 MAC 协议;

④ 用户改变 MAC 协议,并报告新状态;

⑤ 由新状态以及网络指标类型确定新网络状况;

⑥ 通过前后网络状况对比,计算动作奖励;

⑦ 依照获取的奖励更新 Q 表,并更新当前状态;

⑧ 记录本次网络性能指标;

结束条件:达到终止状态或者达到训练次数;

Return:网络性能指标.

当确定终端用户的物理层/MAC 层策略之后, SNode 进一步对终端用户进行计算卸载的策略学习.假设服务器  $Server_j$  资源可表示为  $source_j = \{f_j\}, j \in \{1, 2, \dots, n\}$ , 其中  $f_j$  表示对服务器资源的综合评价.而通信方面分配的资源用 MAC 协议进行描述,即  $c_k = \{\text{TDMA}, \text{CSMA}\}, k \in \{1, 2, \dots, n\}$ .

1) 状态(State).control 拥有所有的边缘服务器信息以及用户所提交的任务信息,因此 control 表示的系统整体状态为

$$State = \{EdgeServerSource_1, EdgeServerSource_2, \dots, EdgeServerSource_n, Task_u\},$$

其中,EdgeServerSource<sub>*n*</sub> 表示第  $n$  个边缘服务器所拥有的资源,Task<sub>*u*</sub> 表示第  $u$  个用户的任务请求信息.

2) 动作(Action).动作表示为 control 为用户选择卸载服务器,即 Action 集合为

$$Action = \{EdgeServer_1, EdgeServer_2, \dots, EdgeServer_n, cloud\},$$

其中,EdgeServer<sub>*n*</sub> 表示第  $n$  个边缘服务器,cloud 表示云服务器.

3) 奖励(reward).由计算卸载的目标准则可知,针对用户任务卸载请求,完成任务所需要消耗的时延越短越好.因此易知,当用户任务卸载至本区域内的边缘服务器时,其所消耗时延将小于卸载至其他区域内的边缘服务器,远小于卸载至云服务器的时延.同理,卸载至其他区域边缘服务器所消耗的代价略大于本区域的边缘服务器,小于云服务器的代价.因此,设定不同的优先级标识不同层次的服务器级别.

$$reward = \begin{cases} \text{高优先级, 本地服务器,} \\ \text{中优先级, 其他区域服务器,} \\ \text{低优先级, 云服务器.} \end{cases}$$

依照不同的优先级,将获得不同收益;我们期望

高优先级能够获取更好的收益,而中优先级所获取收益相对来说较少,而对于云服务器来说,由于其长距离导致的高时延问题,因此将获得负收益.其  $reward$  计算为

$$reward = \begin{cases} \lambda \times 1, & \text{高优先级,} \\ \lambda \times 0.5, & \text{中优先级,} \\ \lambda \times (-1), & \text{低优先级,} \end{cases}$$

其中,  $\lambda$  表示衰减因子,防止网络收敛过快.算法 2 描述了基于 DQN 的计算卸载自适应学习过程.

**算法 2.** 基于 DQN 的计算卸载算法.

输入:预测神经网络、目标神经网络、经验池;

输出:网络性能指标.

- ① 初始化 经验池、神经网络参数;  
重复:
  - ② 由边缘服务器信息以及任务请求确定当前状态;
  - ③ 利用预测神经网络或者  $\epsilon$ -贪婪策略,基于当前状态选择动作(卸载节点)  $Action$ ;
  - ④ 用户卸载至边缘服务器 [ $Action$ ] 或者云服务器;
  - ⑤ 获取新状态,并计算动作奖励;
  - ⑥ 将四元组(当前状态,动作,奖励,新状态)存储到经验池;
  - ⑦ 由经验池随机抽样作为预测神经网络的训练数据;
  - ⑧ 基于预测神经网络、目标神经网络,计算损失函数;
  - ⑨ 利用损失函数更新,预测神经网络参数;
  - ⑩ 经过一定次数训练探索后将预测神经网络参数替换目标神经网络参数;
  - ⑪ 记录本次网络性能指标;
- 结束条件:达到终止状态或者达到训练次数;  
Return:网络性能指标.  
算法复杂度:

我们定义函数  $regret$  来衡量采用的  $\epsilon$ -greedy 策略对探索和利用的平衡效果.具体而言,  $regret$  表示每一步平均的可能机会损失,其复杂度为  $\Omega(\min\{T, A^{H/2}\})$ ,其中  $T$  表示 Q-Learning 总训练步数,  $H$  表示每个 episode 的训练步数,  $A$  表示动作数量.因此,时间复杂度为一次 Q-Learning 算法训练时间,主要在于读取 Q 表,而 Q 表读取时间为  $S \times A$ .因此总时间为:总训练次数  $\times$  读取 Q 表时间,即  $O(T \times A \times S)$ ,  $T = KH$ ,其中  $T$  表示 Q-Learning 总训练步数,  $K$  表示总的 episode,  $H$  表示每个

episode 的训练步数,  $S$  表示服务器数量,  $A$  表示动作数量.而空间复杂度包含信道以及服务器资源存储空间和 Q 表存储空间,即  $O((C + S \times |R|) \times A \times H)$ ,其中  $C$  表示信道资源量,  $S$  表示服务器数量,由多元组表示服务器资源,  $A$  表示动作数量,  $H$  表示每个 episode 的训练步数.

具体来说,SDN 在本文设计的多接入边缘计算架构中的工作机制为:

首先,Internet 服务提供商主动将与他们自己相关联的服务卸载到 MEC 节点.以图 6 为例,假设移动用户正在通过常规数据路径访问基于 Internet 的游戏前端服务器.该游戏服务提供商通过注册来使用 MEC 节点,从而主动卸载其服务并进行缓存,并将其存储在适当的 SNode 中.进一步地, SNode 通过将服务复制到 CNode 来将服务推向最终用户.同时, CNode 在固定时间间隔内搜集覆盖范围内的移动用户的信道信息 CSI,并上报给 SNode.

在计算服务被分流之后,当一个移动用户想要访问游戏服务器时,该用户向 SNode 提出自己的请求. SNode 首先根据用户的信道质量,运行物理层/MAC 层分配算法.下一步,根据终端用户分配的 MAC 协议及访问需求, SNode 为该用户选择最佳的 CNode 进行卸载,若不存在合适的 CNode,则将用户任务卸载至中心云.这样,相关的流量从网络核心和 Internet 上成功进行了卸载,从而大大减轻了流量负担.

## 4 实验与结果

本节我们使用网络模拟器 NS3 对本文提出的多接入边缘计算架构进行了仿真.我们首先验证了物理层/MAC 层切片的可行性.进一步地,我们验证了基于 Q-Learning 的自适应资源分配算法的有效性.最后,我们对比了 MEC 架构中的随机卸载算法,以及目前最好的工作 SI-EDGE 卸载算法<sup>[30]</sup>.通过大量的仿真实验证明,我们设计的细粒度资源分配算法相比经典的卸载算法,延时降低了 30%,而对比 SI-EDGE 卸载算法,延时降低了 10%.

### 4.1 实验设置

实验拓扑图设置如图 5 所示,我们考虑一个典型的 MEC 计算卸载场景.终端用户的信道模拟为频率选择性衰落信道(frequency selective fading channel),因此不同子信道之间的质量差异较大.信道基本带宽为 10 MHz.终端用户随机分布在边缘

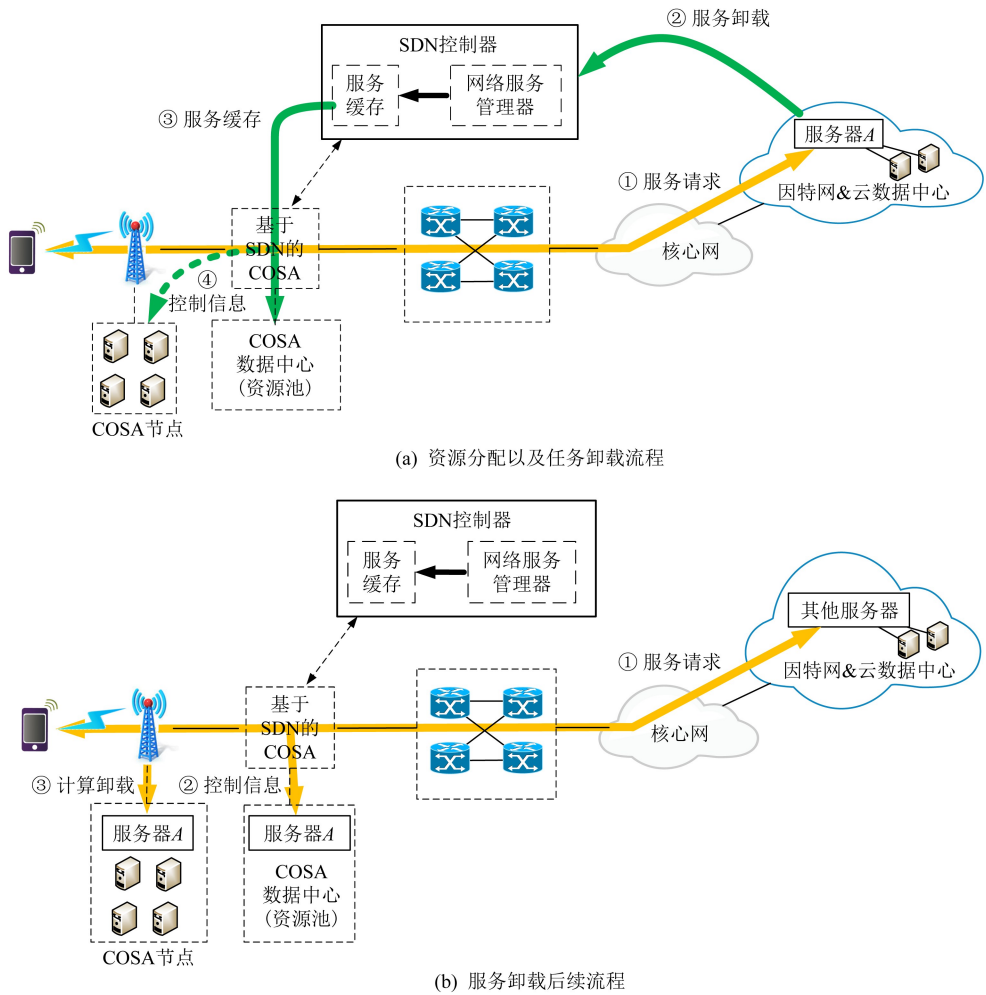


Fig. 6 Example of SDN-based resource allocation and task offloading process

图 6 基于 SDN 的资源分配及任务卸载流程示例

计算节点 CNode 周围,MEC 服务器的计算能力为  $f_{MEC} = 5 \text{ GHz/s}$ ,每个终端用户的计算能力为  $f_{UE} = 1 \text{ GHz/s}$ .具体的实验参数设置如表 1 所示:

Table 1 Experimental Setup

表 1 实验设置

| 参数                   | 值    | 参数                  | 值  |
|----------------------|------|---------------------|----|
| SIFS/ $\mu\text{s}$  | 10   | DIFS/ $\mu\text{s}$ | 28 |
| PIFS/ $\mu\text{s}$  | 19   | 符号时间/ $\mu\text{s}$ | 32 |
| 块长度/B                | 1460 | 基础数据速率/Mbps         | 6  |
| $f_{MEC}/\text{GHz}$ | 5    | $f_{UE}/\text{GHz}$ | 1  |
| 傅里叶变换点数              | 64   | 带宽/MHz              | 20 |

#### 4.2 物理层/MAC 层切片性能

在物理层/MAC 切片中,频域竞争对于确保有效的无线资源访问至关重要.为了验证其有效性,我们通过单个冲突域拓扑进行模拟频域竞争策略.其中,SDN 控制器作为集中控制部分,负责自适应策

略的学习和最终分配解决方案的决定.因此在传输分配时不会产生冲突.然而,在传输竞争中,很可能几个终端用户随机选择到同一段子载波作为它们的竞争子载波.竞争子载波上的这种冲突将大大降低子信道的访问效率,冲突的终端用户无法将发送任何东西.在这里,我们评估 2 个或多个客户端选择相同的竞争子载波的可能性,从而评估频域竞争策略的可行性.

图 7 描述了频域竞争策略下不同数量的竞争用户碰撞的概率  $P$ .其中横坐标是用户的个数,纵坐标是选择同一段竞争子载波的概率.从图 7 中我们可以看出,碰撞概率  $P$  随着终端用户数量的增加而增加.因此随着用户数量的增加,用户选择同样的子载波的概率变得更高.为了应对这个问题,我们增加了子信道带宽来减少多用户之间的碰撞概率.当子信道中子载波数量  $n=8$  时,终端用户之间的竞争空间是  $2^8 - 1 = 255$ .此时,这个竞争空间产生的最大碰撞

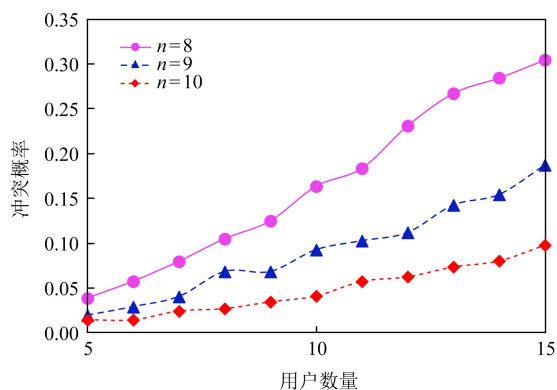


Fig. 7 The conflict probability of the frequency domain competition strategy under different numbers of users

图7 频域竞争策略在不同用户数量下的冲突概率

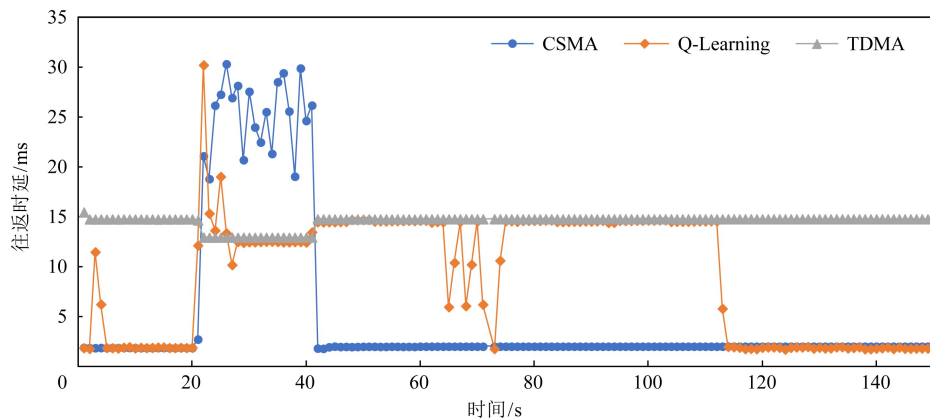
概率为30%。如果我们继续对子信道进行扩充,例如,当 $n=10$ 时,随着竞争空间提升至 $2^{10}-1=1\ 023$ ,则碰撞概率会迅速下降至10%。足够大的竞争空间,

可以为用户提供更多的选择机会,去选择不同的竞争子载波,从而提供更好的接入网频域选择性能。

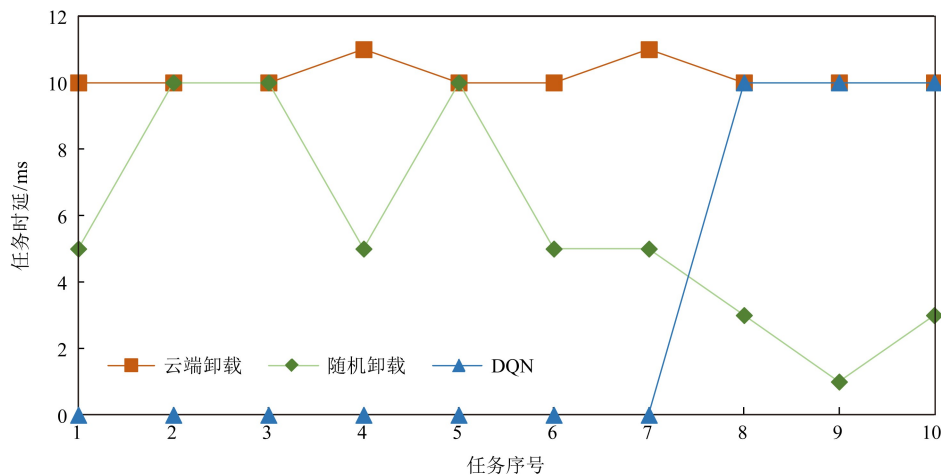
### 4.3 基于Q-Learning的自适应资源分配方法

本节我们对基于Q-Learning的自适应资源分配法进行仿真验证。首先,我们验证基于Q-Learning的MAC协议选择算法。我们设定一个动态网络环境,节点可任意接入,造成信道质量的波动。节点根据信道质量进行学习,从而选择CSMA协议或者TDMA协议。网络状况设置为:网络初始20s保持较为空闲状态,在20~40s网络转换为繁忙状态,在40s后网络又切换为空闲状态。在这种网络动态变化的情况下,我们验证根据不同信道学习对MAC协议进行学习的性能。

图8(a)显示了用户对MAC协议的学习和切换过程。在网络初始的0~20s之内,除了几次基于随机探索外,基于Q-Learning的MAC选择算法都能够正确地选择时延较低的CSMA协议。在20~40s,



(a) 基于Q-Learning的MAC选择测试



(b) 基于DQN的计算机卸载分配算法测试

Fig. 8 Adaptive learning algorithm test

图8 自适应学习算法测试

由于信道质量突变,导致学习过程中产生巨大的负向 *reward*,因此算法能够及时地切换 MAC 协议.可以发现,在该时间段,算法能够正确地选择时延较低的 TDMA 协议.然而在 40~60 s 时,我们可以发现网络状态已经切换为空闲状态了,但是算法不能及时反应时延较低的 CSMA 协议.这是由于从 TDMA 切换到 CSMA 协议所获得正向 *reward* 相对于前面突发网络所造成的负向 *reward* 较小,需要进行多次随机探索才能够很好地消除巨大负向 *reward* 造成的影响.在 113 s 时,算法经过多次随机探索后,正确地切换为时延较短的 CSMA 协议.

下一步,我们验证基于 DQN 的计算卸载算法.在训练初始阶段,根据一定概率,随机为用户选择节点卸载.因此,在训练初始阶段,部分的卸载选择了其他区域或者云服务器进行卸载.而当经过一段实验训练后,DQN 做出的卸载决策主要集中于卸载至本地边缘服务器上.图 8(b)描述了基于 DQN 的训练卸载结果.其中,当任务超过一定数量之后,如 7 个,超过了本地服务器的计算能力.由于任务量过大,DQN 会选择卸载至云端,因此时延会有所增加.

#### 4.4 基于软件定义的细粒度边缘计算架构的性能

本节我们对基于软件定义的细粒度边缘计算架构的性能进行了评估.对于传统物理层/MAC 层未切片的边缘计算架构,我们设计的边缘计算架构可以使终端用户细粒度的接入无线网络进行计算卸载.我们选择了云端卸载、随机卸载以及 SI-EDGE 三种方法作为对比.为了保证对比实验的公平性,我们为这 3 种的计算架构添加了频域竞争策略,从而

降低了竞争的传输代价.与本文设计的频域竞争策略一致,这 3 种边缘计算架构中终端用户利用 F-RTS 宣布其计算需求,并且通过 SDN 控制器,利用 F-CTS 进行分配决定的通知.在云端卸载架构和随机卸载架构中,并没有使用切片化的物理层/MAC 层资源,用户每次分别选择云端进行计算卸载(云端),根据一定的概率选择卸载的节点(随机).而在 SI-EDGE 中,网络使用切片化资源,并利用 Edge Slicing Optimization Problem(ESP)算法进行网络和计算资源的联合优化.

图 9 描述了在不同任务数量的对比下 4 种计算架构的服务时间.首先,云端卸载和随机卸载,由于并没有利用切片化的资源,因此和本文提出的 MEC 架构表现出很大的性能差异.在云端卸载和随机架构中,服务时间随着用户数量的增加而迅速增加.由于传统的架构每次只能服务一类的 MAC 协议,并不能同时满足多种多样的计算要求.相比之下,本文设计的细粒度 MEC 架构在频率上可以对多种 MAC 进行并发传输.由于多个 MAC 协议可以同时运行,并且每个子信道可以根据终端用户的信道质量,最大化地进行计算卸载的协同优化,因此,能够适应动态多变的终端用户的需求.

相比而言,由于 SI-EDGE 的架构也采用了切片技术,因此 SI-EDGE 相较于云端卸载和随机卸载也表现出良好的性能,然而由于 SI-EDGE 并没有采取 MAC 的学习策略,因此其性能相比本文提出的细粒度卸载策略还是有所差距.可以发现,在前面任务数量较少时,SI-EDGE 策略和本文设计的策略之间

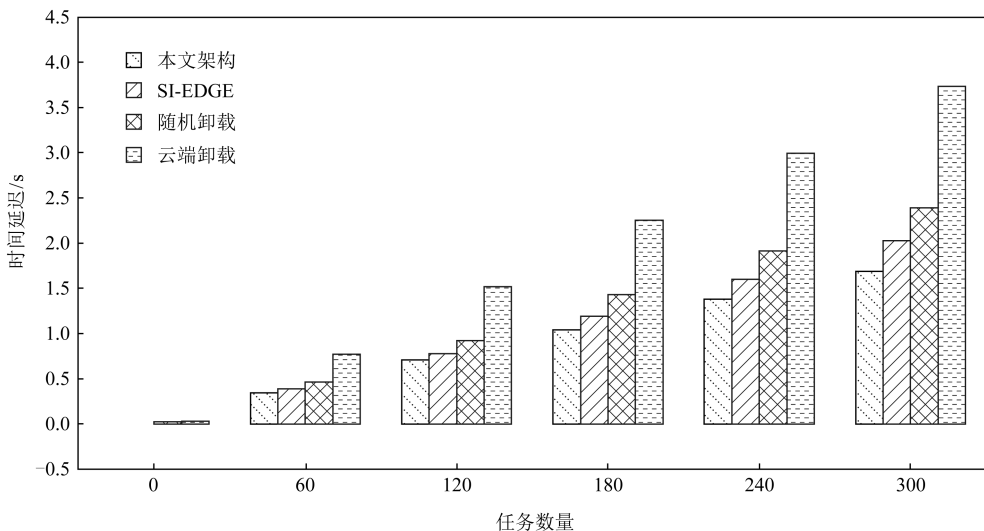


Fig. 9 Performance comparison of MEC

图 9 MEC 的性能比较

差异较小,而随着任务数量的增多,两者之间的差异也越来越大.原因主要是因为任务数量增大,从而使得本文涉及的策略增加更多的训练学习机会,因此能够更好地决定卸载决策.

最后,我们测试了本文设计架构的可靠性.我们分别测试了在链路故障和节点故障下,计算卸载的性能分析.我们随机选择一条链路使其故障,此时, SNode 会直接从路由表中删除当前链路,并启用备用的链路进行传输.图 10(a)描述了链路故障下计算卸载时延的变化.其中,加粗线条表示的是节点切换路由表项的时间,其所需时间较短.由于训练后,节点选择的链路会比备用链路更好,因此链路故障之

后,最初数据包的往返时延会随着更换链路而增大.但是仍然可以继续计算卸载并在线学习.随后,我们进行了节点故障的测试.我们在计算节点中随机选择一个节点使其故障.此时,用户会向 RNode 提出切换节点请求,而 RNode 会返回相应新节点并将业务迁移到新的节点上.因此,节点故障后,并不能够很快发送和接收数据包,而需要等待一段迁移时间才能够获取到达新节点的往返时延.图 10(b)描述了节点故障下计算卸载的实验变化.其中,结果中所示的节点进行业务的迁移时间,由于要进行业务迁移,时延增加较多,然而,在该架构下计算卸载仍可以顺利进行,并可以继续在线学习.

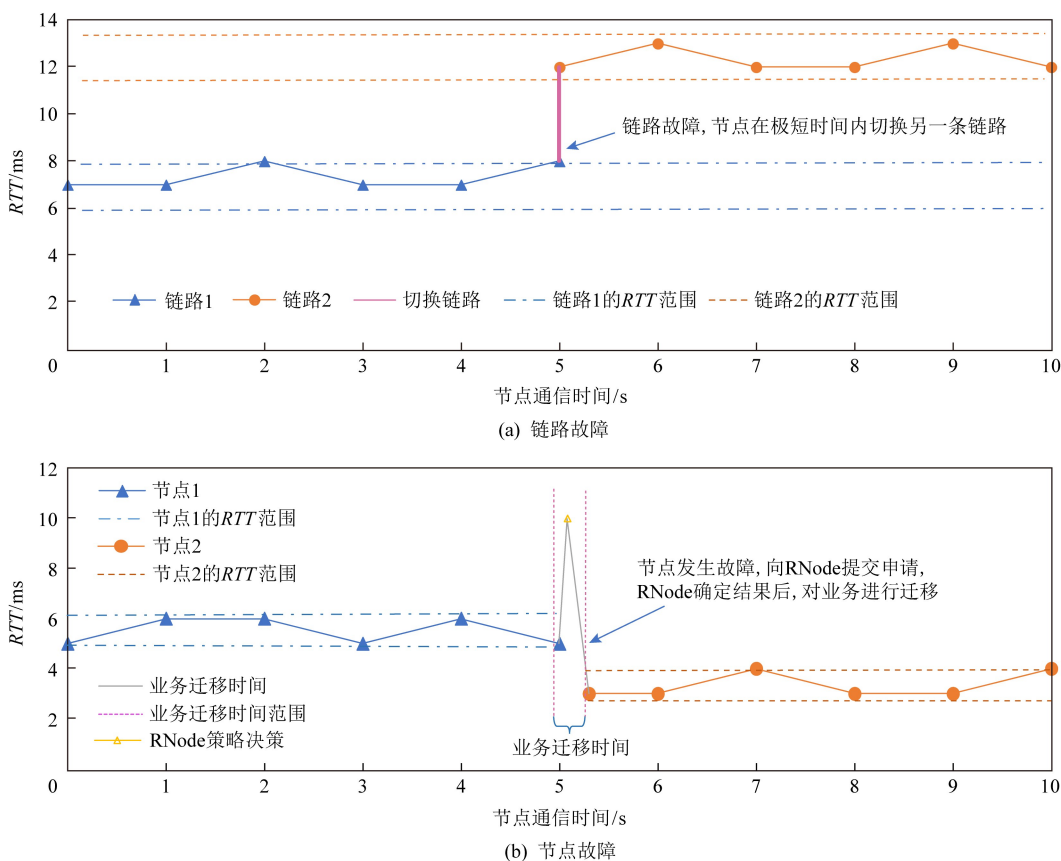


Fig. 10 Reliability analysis of fine-grained multi-access edge computing architecture

图 10 细粒度多接入边缘计算架构的可靠性分析

## 5 总 结

针对目前多边缘计算架构中资源优化粒度粗、掌控不充分等诸多问题,本文设计了一种基于 SDN 的细粒度多接入边缘计算架构,可以对网络资源和计算资源进行细粒度的控制并进行协同管理,从而提高多接入边缘计算的服务质量.本文提出了接入

网简单高效的物理层/MAC 切片方法,可以根据设备的计算要求来支持混合的按需并发传输.本文进一步设计了一种基于深度强化学习 Q-Learning 的两级资源分配策略,对网络资源和计算资源进行自适应的细粒度学习,并通过软件定义的体系结构将控制平面与数据平面分离,增加了边缘计算架构的灵活性和可靠性.与传统固定的边缘计算架构相比,本文提出的细粒度架构可以对所有的资源进行逻辑

集中控制,因此控制平台能够提供更有利的计算分流和服务扩展.我们讨论了所提架构的可行性,并通过仿真证明了其有效性.

为了能够充分发挥多接入边缘计算的软件定义计算卸载的潜力,仍需解决多项研究问题.由于能耗始终是物联网设备的主要问题,因此应在计算分流和能耗之间取得平衡.此外,安全性和隐私问题在物联网计算卸载中也至关重要.在下一阶段,我们建议在实时测试平台(例如软件定义的无线电和软件定义的网络平台)上验证本文提出的多接入边缘计算框架,并将更多功能集成到该框架中,以使更多物联网应用受益.

### 参 考 文 献

- [1] Pan Jianli, McElhannon J. Future edge cloud and edge computing for Internet of things applications [J]. *IEEE Internet of Things Journal*, 2018, 5(1): 439-449
- [2] Zhang Qi, Cheng Lu, Boutaba R. Cloud computing: State-of-the-art and research challenges [J]. *Journal of Internet Services and Applications*, 2010, 1(1): 7-18
- [3] Abbas N, Zhang Yan, Taherkordi A, et al. Mobile edge computing: A survey [J]. *IEEE Internet of Things Journal*, 2017, 5(1): 450-465
- [4] Shi Weisong, Zhang Xingzhou, Wang Yifan, et al. Edge computing: State-of-the-art and future directions [J]. *Journal of Computer Research and Development*, 2019, 56(1): 69-89 (in Chinese)  
(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): 69-89)
- [5] Cao Jiannong, Castiglione A, Motta G, et al. Human-driven edge computing and communication: Part 1 [J]. *IEEE Communications Magazine*, 2017, 55(11): 70-71
- [6] Cao Jiannong, Castiglione A, Motta G, et al. Human-driven edge computing and communication: Part 2 [J]. *IEEE Communications Magazine*, 2018, 56(2): 134-135
- [7] Ren Ju, Zhang Yaoxue, Zhang Ning, et al. Dynamic channel access to improve energy efficiency in cognitive radio sensor networks [J]. *IEEE Transactions on Wireless Communications*, 2016, 15(5): 3143-3156
- [8] Hassan H, El-Desouky A I, Ibrahim A, et al. Enhanced QoS-based model for trust assessment in cloud computing environment [J]. *IEEE Access*, 2020, 8: 43752-43763
- [9] Lu Haodong, He Xiaoming, Du Miao, et al. Edge QoE: Computation offloading with deep reinforcement learning for Internet of things [J]. *IEEE Internet of Things Journal*, 2020, 7(10): 9255-9265
- [10] Yang Xiaowei, Wang Xiaoxiao, Wu Yuan, et al. Small-cell assisted secure traffic offloading for narrow-band Internet of things (NB-IoT) systems [J]. *IEEE Internet of Things Journal*, 2017, 5(3): 1516-1526
- [11] Satyanarayanan M, Bahl P, Caceres R, et al. The case for VM-based cloudlets in mobile computing [J]. *IEEE Pervasive Computing*, 2009, 8(4): 14-23
- [12] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading [J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1628-1656
- [13] Chiang M, Zhang Tao. Fog and IoT: An overview of research opportunities [J]. *IEEE Internet of Things Journal*, 2016, 3(6): 854-864
- [14] Mukherjee M, Shu Lei, Wang Di. Survey of fog computing: Fundamental, network applications, and research challenges [J]. *IEEE Communications Surveys Tutorials*, 2018, 20(3): 1826-1857
- [15] Pham Q V, Hwang W J. Network utility maximization-based congestion control over wireless networks: A survey and potential directives [J]. *IEEE Communications Surveys Tutorials*, 2017, 19(2): 1173-1200
- [16] Peng Mugen, Sun Yaohua, Li Xuelong, et al. Recent advances in cloud radio access networks: System architectures, key techniques, and open issues [J]. *IEEE Communications Surveys Tutorials*, 2016, 18(3): 2282-2308
- [17] Patel M, Naughton B, Chan C, et al. Mobile-edge computing introductory technical white paper [S]. Sophia-Antipolis: European Telecommunications Standards Institute (ETSI), 2014
- [18] Kekki S, Featherstone W, Fang Yonggang, et al. MEC in 5G networks [S]. Sophia-Antipolis: European Telecommunications Standards Institute(ETSI), 2018
- [19] Lyu X, Tian Hui, Sengul C, et al. Multiuser joint task offloading and resource optimization in proximate clouds [J]. *IEEE Transactions on Vehicular Technology*, 2017, 66(4): 3435-3447
- [20] Jia Fan, Zhang Heli, Ji Hong, et al. Distributed resource allocation and computation offloading scheme for cognitive mobile edge computing networks with NOMA [C]//Proc of IEEE/CIC Int Conf on Communications in China (ICCC). Piscataway, NJ: IEEE, 2018: 553-557
- [21] Le D V, Tham C K. A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds [C]//Proc of IEEE Conf on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway, NJ: IEEE, 2018: 760-765
- [22] Min Minghui, Xiao Liang, Chen Ye, et al. Learning-based computation offloading for IoT devices with energy harvesting [J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(2): 1930-1941
- [23] D'Oro S, Restuccia F, Talamonti A, et al. The slice is served: Enforcing radio access network slicing in virtualized 5G systems [C] //Proc of IEEE Int Conf on Computer Communications. Piscataway, NJ: IEEE, 2019. DOI: 10.1109/INFOCOM.2019.8737481
- [24] Sun Tao, Devaki C. Study on architecture for next generation system, release 14 [S]. Sophia-Antipolis, France: 3GPP, 2016

- [25] ITU-T Study Group 13. Framework of network virtualization for future networks, next generation network—future networks [S]. Geneva, Switzerland: International Telecommunication Union (ITU-T), 2012
- [26] Gebremariam A A, Chowdhury M, Usman M, et al. SoftSLICE: Policy-based dynamic spectrum slicing in 5G cellular networks [C] //Proc of IEEE Int Conf Communications (ICC). Piscataway, NJ: IEEE, 2018
- [27] Shrivastava R, Samdanis K, Bakry A. On policy based RAN slicing for emerging 5G TDD networks [C] //Proc of IEEE Global Communication Conf (GLOBECOM). Piscataway, NJ: IEEE, 2018: 1-6
- [28] Zambianco M, Giacomo V. Interference minimization in 5G physical-layer network slicing [J]. IEEE Transactions on Communications, 2020, 68(7): 4554-4564
- [29] D'Oro S, Restuccia F, Talamonti A, et al. The slice is served: Enforcing radio access network slicing in virtualized 5G systems [C] //Proc of IEEE Int Conf on Computer Communications (IEEE INFOCOM). Piscataway, NJ: IEEE, 2019: 442-450
- [30] D'Oro S, Bonati L, Restuccia F, et al. SI-EDGE: Network slicing at the edge [C/OL] //Proc of the 21st Int Symp on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (ACM MobiHoc). New York: ACM, 2020 [2020-11-15]. <https://doi.org/10.1145/3323679.3326503>
- [31] Moshe V. Boolean satisfiability: Theory and engineering [J]. Communications of the ACM, 2014, 57(3): 5-5
- [32] Fakhfakh E, Hamouda S. Optimised Q-learning for WiFi offloading in dense cellular networks [J]. IET Communications, 2017, 11(15): 2380-2385



**Wang Lu**, born in 1986. PhD, associate professor. Her main research interests include wireless communications and mobile computing.  
王璐, 1986年生.博士,副教授.主要研究方向为无线通信和移动计算.



**Zhang Jianhao**, born in 1998. Master candidate. His main research interests include edge computing and network optimization.  
张健浩, 1998年生.硕士研究生.主要研究方向为边缘计算和网络优化.



**Wang Ting**, born in 1985. PhD, associate professor. His main research interests include SDN/NFV, data center networking, machine learning, intelligent networking, and cloud/edge computing.  
王廷, 1985年生.博士,副教授.主要研究方向为软件定义网络、网络功能虚拟化、数据中心网络、机器学习、智能网络、云计算和边缘计算.



**Wu Kaishun**, born in 1985. Distinguished professor, IET Fellow. His main research interests include wireless networking, mobile computing, and Internet of things.  
伍楷舜, 1985年生.特聘教授,英国工程技术学会会士.主要研究方向为无线网络、移动计算和物联网.