计算机研究与发展 DOI:10.7544/issn1000-1239.2018.20170714
Journal of Computer Research and Development 55(3):537-550,2018

移动边缘计算任务卸载和基站关联协同决策问题研究

于博文[1] 蒲凌君[1,2] 谢玉婷[1] 徐敬东[1] 张建忠[1]
[1](南开大学计算机与控制工程学院 天津 300071)
[2](广东省大数据分析与处理重点实验室(中山大学) 广州 510006)
(bowenyu@mail.nankai.edu.cn)

Joint Task Offloading and Base Station Association in Mobile Edge Computing

Yu Bowen[1], Pu Lingjun[1,2], Xie Yuting[1], Xu Jingdong[1], and Zhang Jianzhong[1]
[1](*College of Computer and Control Engineering, Nankai University, Tianjin* 300071)
[2](*Guangdong Key Laboratory of Big Data Analysis and Processing* (*Sun Yat-Sen University*), *Guangzhou* 510006)

Abstract In order to narrow the gap between the requirements of IoT applications and the restricted resources of IoT devices and achieve devices energy efficiency, in this paper we design COMED, a novel mobile edge computing framework in ultra-dense mobile network. In this context, we propose an online optimization problem by jointly taking task offloading, base station (BS) sleeping and device-BS association into account, which aims to minimize the total energy consumption of both devicesand BSs, and meanwhile satisfies applications' QoS. To tackle this problem, we devise an online Lyapunov-based algorithm JOSA by exploiting the system information in the current time slot only. As the core component of this algorithm, we resort to the loose-duality framework and propose an optimal joint task offloading, BS sleeping and device-BS association policy for each time slot. Extensive simulation results corroborate that the COMED framework is of great performance: 1) more than 30% energy saving compared with local computing, and on average 10%-50% energy saving compared with the state-of-the-art algorithm DualControl (i.e., energy-efficiency); 2) the algorithm running time is approximately linear proportion to the number of devices (i.e., scalability).

Key words mobile edge computing; task offloading; base station (BS) sleeping; device-BS association; NP-hard problem

摘 要 为了缩小IoT应用的服务质量要求与IoT设备有限的计算资源之间的差距,提高设备与基站能源利用率,设计了基于超密集网络的移动边缘计算框架COMED,提出了一个结合任务卸载、设备-基站关联以及基站睡眠调度的在线优化问题,旨在最小化设备和基站的整体能量消耗,同时满足IoT应用的服务质量要求.针对这一在线优化问题,提出了一个基于李雅普诺夫优化理论的任务调度算法JOSA,该算法只使用当前时间片的系统信息进行调度.仿真实验证明了COMED框架具有良好的性能:1)与设备本地处理相比,系统整体节能30%以上,与DualControl算法相比平均节能10%~50%;2)算法的执行时间与IoT设备数量呈近似线性的关系.

收稿日期:2017-09-25;**修回日期**:2017-12-07
基金项目:国家自然科学基金项目(61702287,61702288);天津市自然科学基金项目(16JCQNJC00700);南开大学基础科研业务项目(070-63171112)
This work was supported by the National Natural Science Foundation of China (61702287, 61702288), the Natural Science Foundation of Tianjin (16JCQNJC00700), and the Basic Research Project of Nankai University (070-63171112).
通信作者:蒲凌君(pulingjun@nankai.edu.cn)

关键词 移动边缘计算;任务卸载;基站睡眠;设备-基站关联;NP 难问题

中图法分类号 TP393.1

随着无线通信技术的发展以及传感器种类的丰富,诸如智能汽车、手机等 IoT 设备可以通过蜂窝网络、低功耗广域网等方式接入互联网,并能使用装备的传感器感知周围环境的状态.在这一背景下,众多 IoT 应用应运而生,例如群智感知、智能监控、车联网应用等.在这类应用中,IoT 设备会周期性地采集指定的数据,本地处理或上传到云端服务器进行分析,产生相应知识或模式,为人们的生活和工作提供便利.随着 IoT 应用的类型不断丰富、功能日益强大,它们对计算能力和响应延时提出了更高的要求.例如在车联网和增强现实等领域,这些应用要求实时处理采集到的视频信息并将结果反馈给用户.大量的密集型计算任务势必会加快 IoT 设备的能源消耗,缩短其使用寿命.目前,IoT 设备与云端相结合是 IoT 应用的主要模式.然而 IoT 设备和远程云平台间长距离通信存在网络传输延时不稳定的问题,这将会导致 IoT 应用的延时过长,无法满足对延时有明确要求的应用.

作为一种很有前景的解决方案,移动边缘计算能够有效地解决传统移动云计算的不足.移动网络运营商和云服务提供商以合作的形式在网络边缘提供丰富的通信和计算资源,IoT 设备可以通过高速无线接入网络在蜂窝网络边缘近距离地获取所需的计算资源和服务[1-2].随着万物互联和大数据时代的到来,IoT 设备的数量和移动数据流量得到了飞速提升.根据 Cisco 的报告[3],移动数据流量将在未来的 5 年中增长 7 倍,到 2021 年将达到每月 49 艾字节(exabyte,EB)(1 EB=10^6 TB),同时全球 IoT 设备数量将从目前的 80 亿增长到 120 亿.这将使其接入网络获取移动边缘服务器的计算资源变得困难.目前的网络架构将很难应对未来大规模设备接入网络的需求.为应对海量的设备连接和数据流量,5G 架构下的多基站协作服务场景如超密集网络(ultra-dense networks, UDN)逐渐成为移动网络运营商广泛接受的模式[4-5].在 UDN 网络中,移动网络运营商会部署大量微基站和宏基站为移动设备提供接入服务,这必然会导致蜂窝网络的能耗显著提升,使运营成本大幅度提高.因此如何降低 UDN 网络的能耗是目前蜂窝网络研究关注的核心问题[6].本文希望通过对 IoT 设备的计算卸载、设备-基站关联以及基站睡眠的合理调度,提高 IoT 设备的计算能力,同时尽可能减少 IoT 设备和蜂窝基站的能源开销.

结合 5G 的移动边缘计算的能耗问题近年来得到了工业界和学术界的广泛关注[7-10].文献[11-13]研究了多设备的计算卸载的最优能耗问题.文献[14-16]对计算卸载与移动边缘服务器资源联合调度展开了研究,解决了移动设备和服务器整体能耗或能源效率最优化的问题.然而这些研究工作通常只考虑单一基站(即宏基站)为 IoT 设备提供接入服务,没有解决未来 5G 环境下的多基站协作服务场景下的 IoT 设备、基站和边缘服务器联合调度的问题.

本文提出了一个基于超密集网络的移动边缘计算框架(computing offloading framework based on MEC and UDN, COMED).与以往研究工作不同的是:COMED 框架实现了超密集网络中高效的计算卸载、基站睡眠以及 IoT 设备-基站关联三者的调度.框架的示意图如图 1 所示,其中包括 IoT 设备、微基站、宏基站以及边缘服务器.IoT 设备通过自身的传感器采集数据.宏基站负责调度 IoT 设备与最合适的基站进行关联并令没有关联 IoT 设备的基站进入睡眠状态以节约能量;同时负责调度 IoT 设备本地处理任务或将任务卸载到边缘服务器上进行处理.服务器根据定制的服务计划为 IoT 设备上传的任务分配计算资源.当 IoT 设备的任务在本地或边缘服务器上处理完毕之后,结果将会被传送到互联网上.例如车联网应用中的危险预警系统需要使用高清摄像头、超声波和激光雷达周期性地采集道路上的视频和数据信息,分析当前道路上是否存在异常状况,进而为人们的生命财产安全提供保障.在 COMED 框架中,车辆可以本地处理采集到的视频和数据,也可以通过相关调度利用移动边缘计算提供的丰富的计算资源处理任务.为了让 COMED 框架在实际中能够对 IoT 应用提供高效的计算卸载支持,它需要满足 3 个要求:

1) COMED 框架需要在保证 IoT 应用的服务质量(如延时)的前提下尽可能做到高效节能.降低 IoT 设备和基站的能源消耗,对延长 IoT 设备的使用寿命、提高设备和基站的能源利用率具有十分重要的现实意义.

2) COMED 框架需要在线运行.由于蜂窝网络状态、IoT 设备移动性以及边缘云可用资源均随时

间动态变化,准确预测未来系统信息比较困难,因此这就要求 COMED 框架能够仅使用当前系统信息进行任务卸载、基站开关和设备-基站关联决策.

3) COMED 框架需要具有可伸缩性. 由于在城市环境中会部署大规模的 IoT 设备,因此要求 COMED 框架设计的任务卸载、基站开关和关联决策具有较低时间复杂度,从而能为更多的设备和应用提供服务.

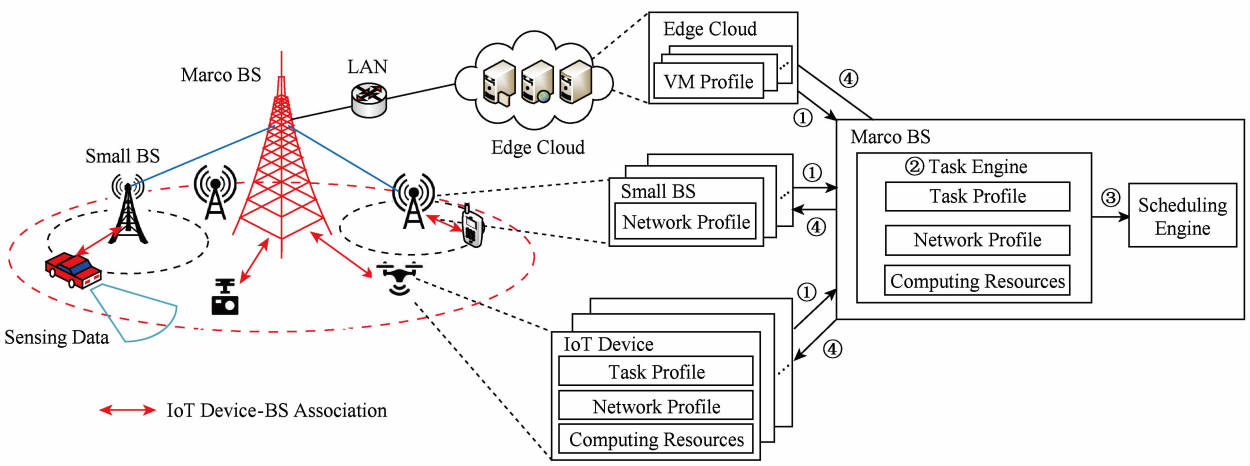


Fig. 1 The framework of task offloading for COMED

图 1 COMED 任务卸载框架示意图

基于上述要求,本文首先对 COMED 框架进行系统建模,包括 IoT 设备资源模型、边缘服务器模型、IoT 应用和负载排队模型、设备和基站能耗模型等. 在此基础上,本文提出并形式化了一个针对全网络的任务负载调度、IoT 设备-基站关联以及基站睡眠调度的联合任务卸载问题,旨在最小化时间平均意义下 COMED 框架的能耗(即 IoT 设备和基站的能耗之和),同时兼顾应用任务的延时要求(第 2 节). 针对优化问题,本文根据李雅普诺夫优化理论提出了联合计算卸载、基站睡眠和用户-基站关联(joint computing offloading, BS sleeping and user-BS assoication, JOSA)的在线任务卸载算法. 作为该算法的核心模块,本文提出了一个基于松弛-对偶理论的 SlotCtrl 算法,该算法仅根据当前时间片内的系统信息进行调度. 通过数学证明可知,本文提出的在线 JOSA 算法与线下理论最优算法的节能效果十分接近(第 3 节). 大量模拟实验结果不但证实了理论分析结果,而且与设备本地处理相比,系统整体节能 30%以上(第 4 节).

1 相关工作

为了节约 IoT 设备的能量,同时有效地利用边缘服务器的计算资源,近年来许多研究者开展了针对移动边缘计算中计算卸载与服务器资源联合调度问题的研究. 其中,文献[14]将移动边缘计算与无线充电技术相结合,研究了可无线充电的 IoT 设备的计算卸载问题,提出了基于李雅普诺夫优化的在线调度策略;文献[15]假设云端服务器为每个 IoT 设备分配一台虚拟机,研究了多设备任务卸载开销最小化的问题,提出了高效的资源分配策略;文献[16]研究了车载云计算场景中最大化数据传输和计算的能源效率问题,在最小的传输率、最大的延时和时延抖动的情况下,满足应用的服务质量要求;文献[17]假设 IoT 设备共享有限的无线带宽和服务器资源,研究了同时满足无线带宽和服务器资源限制条件下最大化计算卸载执行收益的优化问题. 通过对问题进行解耦分析,该文提出了高效的设备无线传输功率设置及服务器资源分配策略. 然而上述这些研究工作只针对单一蜂窝基站场景,难以适用于未来 5G 架构下的 UDN 网络.

针对 UDN 网络的移动边缘计算卸载问题,文献[18]开展了基于 MIMO 的多蜂窝基站场景下多设备任务执行能耗最小化的研究;文献[19]解决了 C-RAN 网络架构中多设备任务执行能耗最小化问题. 然而这些工作只关注设备任务执行能耗最优,并没有涉及 5G 架构下蜂窝基站睡眠、设备与蜂窝基站关联[20]等关键问题.

在这些研究的基础上,本文重点研究了 5G 超密集网络中移动边缘计算的计算卸载、IoT 设备-基站关联及基站睡眠调度的联合优化问题,提出了 IoT 设备与基站能源最优的调度方案,同时保证了任务的延迟约束.

2 系统模型与问题形式化

本文提出的 COMED 框架如图 1 所示,其中包括一个基站集合 $M=\{1,2,\cdots,m\}$ 和一个 IoT 设备集合 $N=\{1,2,\cdots,n\}$. 在基站集合 M 中存在一个宏基站以及 $m-1$ 个微基站. IoT 设备可以通过无线网络接入基站,获得处于边缘云服务器的资源. IoT 设备的任务可以通过调度,本地执行或卸载到边缘云服务器上执行. 本文考虑框架内的每个 IoT 设备用户在边缘云服务提供商处购买计算资源,用来处理 IoT 设备卸载到边缘云的任务.

类似于 UDN 以宏基站作为小型基站和微基站的控制器,本文将宏基站作为 COMED 框架的中央控制器对系统中的任务卸载、基站睡眠以及设备-基站关联进行调度,在宏基站中执行一个任务引擎进程. 任务引擎负责收集 IoT 设备本地及边缘服务器端待完成的任务信息、计算资源信息以及 IoT 设备和基站的网络状况信息. 这些信息可以用于估计当前任务在每个 IoT 设备和边缘服务器上的执行时间和能耗,对于框架的调度至关重要. COMED 需要访问这些配置文件,以便更好地对任务卸载、基站睡眠以及设备-基站关联进行调度.

本文考虑 COMED 框架以时间片为单位运行. 单位时间片 $t\in\{1,2,3,\cdots\}$ 的长度由 IoT 应用的服务提供商决定. 对每个时间片,IoT 设备和边缘服务器需要向宏基站提供当前未完成的任务量和可用的计算资源,IoT 设备和基站需要提供当前的网络状况(如图 1 步骤①). 宏基站收到当前任务量和网络状况后,由任务引擎构建完整的任务信息、网络信息以及计算资源信息(如图 1 步骤②). 这些信息将用于决定任务卸载、基站睡眠以及设备-基站关联的调度. 接下来,任务引擎将启动一个负责当前时间片任务卸载、基站睡眠以及设备-基站关联调度的调度器(如图 1 步骤③). 调度器利用当前时间片的任务信息、网络信息以及计算资源信息来估计如何更好地对任务卸载、基站睡眠以及设备-基站关联进行调度. 为此,调度引擎需要确定调度策略. 目前 COMED 使用在满足任务平均延时的前提下尽可能降低执行任务能耗的策略. 在调度器计算得到该时间片的调度结果之后,调度结果会通过网络传给 IoT 设备、基站和边缘服务器(如图 1 步骤④). 任务引擎会通知 IoT 设备关联到哪个基站、本地执行和卸载到边缘服务器的任务量;通知基站是否开启以及关联哪些 IoT 设备;通知边缘服务器为相应 IoT 设备的虚拟机提供服务.

2.1 IoT 设备资源模型

本文假设 IoT 设备搭载单核 CPU,按照 FIFO 顺序处理 IoT 应用的任务. 目前主流处理器有许多不同的工作模式,比如根据工作负载通过动态电压频率调节 CPU 频率(DVFS)的按需模式、以最高 CPU 频率运行的性能模式以及以最低 CPU 频率运行的节能模式等. 令 $s_i(t)$ 为单位时间片 IoT 设备 i 的 CPU 周期数,其值可通过稳定 CPU 工作频率得到[21]. 令 $p_i(s_i(t))$ 为 CPU 单位时间片周期数所对应的功率.

定义 0-1 控制变量 $a_{ij}(t)$ 为 IoT 设备-基站关联变量. 当 $a_{ij}(t)=1$ 时表示 IoT 设备 i 与基站 j 关联;否则 $a_{ij}(t)=0$. 对于 IoT 设备 i,本文定义其上传功率为 p_i. 当 IoT 设备 i 与基站 j 关联时,IoT 设备 i 与基站 j 的上行链路 SINR $\Gamma_{ij}(t)$ 可以通过 IoT 设备 i 以及 i 周围的其他 IoT 设备的上行传输链路的准静态衰落信道增益预估得到. 本文假设 IoT 设备只有在当前上行链路 SINR $\Gamma_{ij}(t)$ 大于目标 SINR γ_{ij} 时才能与相应的基站建立关联. 对于给定的 IoT 设备 i 的上行链路 SINR $\Gamma_{ij}(t)$,如果不等式 $\Gamma_{ij}(t)\geqslant\gamma_{ij}$ 成立,本文将 IoT 设备 i 与基站 j 的上传链路的速度定义为[19,22]

$$r_{ij}=B\log(1+\gamma_{ij}),$$

其中,B 为链路带宽.

对于 IoT 设备的传输模型,有约束条件:

$$\sum_{j\in M}a_{ij}(t)\leqslant 1, \tag{1}$$

$$\sum_{i\in N}a_{ij}(t)\leqslant h_j, \tag{2}$$

$$a_{ij}(t)\leqslant\left\lfloor\Gamma_{ij}(t)/\gamma_{ij}\right\rfloor. \tag{3}$$

式(1)表示每个 IoT 设备在同一时间片至多与 1 个基站进行关联;式(2)表示对于任意基站 j,同一时刻最多可关联 h_j 个 IoT 设备;式(3)表示 IoT 设备 i 只有在上行链路 SINR $\Gamma_{ij}(t)$ 大于目标 SINR γ_{ij} 时才能与基站 j 建立关联. 此外为了方便起见,本文定义 IoT 设备 i 在时间片 t 的上传速度为

$$r_i(t)=\sum_{j\in M}a_{ij}(t)r_{ij}.$$

出于对 IoT 设备的安全和隐私考虑,本文不考虑宏基站对 IoT 设备的 CPU 频率进行控制. 同时为了不增加现有蜂窝网络调度的开销,本文不考虑控制 IoT 设备的上传功率. 为方便读者阅读,本文只对问题的约束条件以及重要的公式加以编号.

2.2 边缘服务器模型

本文考虑每个 IoT 设备的用户分别在边缘云服务提供商处购买一台特定的虚拟机.虚拟机每个时间片的计算能力为 v_i.令 0-1 控制变量 $c_i(t)$表示在时间片 t 边缘云是否为虚拟机 i 分配相应的计算资源.如果 $c_i(t)=1$,表示边缘云在时间片 t 为虚拟机 i 分配计算资源;否则 $c_i(t)=0$.边缘云每个时间片分配的计算资源不能超过服务器的物理资源 $v_{\max}(t)$,即存在约束条件:

$$\sum_{i\in N} c_i(t)v_i \leqslant v_{\max}(t). \tag{4}$$

2.3 任务模型

本文考虑如下类型的 IoT 设备应用:应用按照一定频率周期性使用 IoT 设备的传感器采集信息,并进行处理分析.通常,此类应用的任务被要求在一定延时内处理完毕,如车载计算中绿灯最优速度建议应用、蜂窝 IoT 应用以及通过 IoT 设备收集数据的 Crowdsourcing 应用等.与许多现有工作相同[13,16,23],COMED 框架关注于对 IoT 应用本地执行与卸载执行的任务工作量调度.

1) 任务模型.在每个时间片 t,移动节点 i 会生成 $k_i(t)$比特的任务.本文假设任务的平均到来量为 λ,其值可以通过离线统计得到.CPU 处理任务需要消耗一定的 CPU 计算资源.定义任务的处理密度为 ρ,即处理 1 b 任务需要 ρ 个 CPU 运行周期.例如识别高清视频一帧中的某一种物体(如车辆、障碍物或行人等)大约需要 9.6×10^4 个 CPU 周期,其任务处理密度大约为 50CPU cycles/b[24].

2) 任务队列模型.本文引入任务队列来描述当前 IoT 设备和边缘服务器中尚未处理的任务量.定义 $Q_i(t)$为 IoT 设备 i 的任务队列,其值为 IoT 设备 i 在时间片 t 时的本地任务残余量.$Q_i(t)$在下一个时间片的更新规则:

$$Q_i(t+1)=Q_i(t)+k_i(t)-x_i(t)-y_i(t),$$

其中,控制变量 $x_i(t)$为时间片 t 时的本地任务执行量,控制变量 $y_i(t)$为时间片 t 时的卸载任务量.这里包含约束条件:

$$x_i(t)\leqslant s_i(t)/\rho, \tag{5}$$

$$y_i(t)\leqslant r_i(t), \tag{6}$$

$$x_i(t)+y_i(t)\leqslant Q_i(t)+k_i(t). \tag{7}$$

式(5)表示 IoT 设备每个时间片本地处理的任务量不能超过其计算能力;式(6)表示 IoT 设备每个时间片卸载的任务量不能超过其传输能力;式(7)表示 IoT 设备每个时间片本地处理的任务量和卸载的任务量之和不能超过其任务队列中的任务残余量.

定义 $L_i(t)$为 IoT 设备 i 在边缘云虚拟机的任务队列,其值为虚拟机 i 在时间片 t 时的任务残余量.$L_i(t)$在下个时间片的更新规则为

$$L_i(t+1)=L_i(t)+y_i(t)-c_i(t)\ \min\ (v_i/\rho,L_i(t)),$$

其中,$c_i(t)\ \min\ (v_i/\rho,L_i(t))$表示虚拟机 i 在一个时间片内处理的任务量.

2.4 能耗模型

在 COMED 框架中,本文考虑的能耗包括 IoT 设备和基站的能耗①.在 IoT 设备的能耗模型中,IoT 设备能耗由本地计算的能耗和卸载任务的能耗组成.本文假设 IoT 设备本地计算与卸载任务产生的能耗分别与其计算和传输的时间成正比.因此 IoT 设备的能耗可以表示 $e_i(t)$为

$$e_i(t)=p_i(s_i(t))\frac{\rho x_i(t)}{s_i(t)}+p_i\frac{y_i(t)}{r_i(t)},$$

等式右侧前一部分为 IoT 设备 i 本地处理 $x_i(t)$比特任务的能耗,后一部分为卸载 $y_i(t)$比特任务的能耗.

基站的功耗可由一个基础功耗加一个动态功耗表示[25].为了方便起见,本文将基站的功耗 P_j 定义为其基础功耗的 β 倍,β 可通过对基站能耗的统计得到.在 UDN 中基站可以通过睡眠调度节省能量.本文定义 0-1 控制变量 $b_j(t)$为基站睡眠调度变量.当 $b_j(t)=1$ 时,表示基站 j 处于工作模式;否则 $b_j(t)=0$.框架内作为调度服务器的宏基站需要一直处于工作状态.基站 j 的能耗可表示为

$$E_j(t)=b_j(t)P_j.$$

0-1 控制变量 $b_j(t)$有约束条件:

$$a_{ij}(t)\leqslant b_j(t), \tag{8}$$

即 IoT 设备 i 只有在基站 j 处于工作模式时,才能与 j 进行关联.

2.5 延 时

IoT 设备在运行应用时,通过传感器采集到的数据应在有限的延时 $d_{\max}$ 内被执行.为了满足应用的延时要求,同时考虑未来 5G 网络环境下大规模的 IoT 设备,根据利特尔法则[26],本文定义在全局意义上的任务平均延时为

$$\overline{d}=\frac{\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i\in N}[Q_i(t)+L_i(t)]}{n\lambda}\leqslant d_{\max}. \tag{9}$$

① 本节只考虑了 IoT 设备和基站的能耗,对于边缘服务器能耗在 3.2 节中讨论.

为了方便起见,本文引入辅助变量 $U(t)$,令 $U(t)=\sum_{i\in N}[Q_i(t)+L_i(t)]$,$U(t)$用来描述 IoT 设备 i 在时间片 t 时本地及边缘云服务器端待完成任务量的总和.约束条件式(9)可以改写为

$$\frac{1}{T}\sum_{t=0}^{T-1}U(t)\leqslant n\lambda d_{\max}.$$

根据任务队列模型,通过迭代可以得到:

$$U(t)=\sum_{i\in N}[k_i(t)-x_i(t)-c_i(t)\min(v_i/\rho,L_i(t))].$$

需要说明的是,全局意义上的任务平均延时 $\bar{d}$ 既包括了任务的执行延时,也包括了任务的传输延时,这是因为 $\bar{d}$ 表示的是任务在系统中的平均停留时间.

2.6 问题形式化

本文通过对 IoT 设备任务、基站资源以及边缘服务器资源的分配,旨在最小化系统整体的能量开销,同时从长远角度满足应用的平均延时.本文制定出优化问题:

$$P:\min_{\substack{\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{c}(t),\\ \boldsymbol{x}(t),\boldsymbol{y}(t)}}\ \frac{1}{T}\sum_{t=0}^{T-1}\Big(\sum_{i\in N}e_i(t)+\sum_{j\in M}E_j(t)\Big),$$

s.t. 式(1)~(9)成立.

3 算法设计

李雅普诺夫优化[27]对于解决具有时间平均意义下的目标函数和约束条件的优化问题是非常有效的.通过调用李雅普诺夫 drift-plus-penalty 方法,本文设计了一个利用每个时间片当前系统信息的任务卸载、基站睡眠和 IoT 设备-基站关联调度的在线算法 JOSA 对问题 P 进行求解.

3.1 问题转化

为了满足应用延时的约束条件,本文引入虚队列 B:

$$B(t+1)=[B(t)-n\lambda d_{\max}]^+ +U(t),$$

其中,$[x]^+=\max(0,x)$,$U(t)$为虚队列 B 每个时间片的任务到来量,$n\lambda d_{\max}$为相应的离开量.虚队列 B 用来描述当前系统整体待完成任务的堆积情况.本文规定虚队列 B 的初始值为 0.

本文采用李雅普诺夫优化框架对目标函数进行优化,同时保证虚队列和实队列的稳定性.定义如下的二次李雅普诺夫方程:

$$Y(\boldsymbol{\Theta}(t))=\frac{1}{2}\Big(\sum_{i\in N}[Q_i(t)^2+L_i(t)^2]+B(t)^2\Big),$$

其中,向量 $\boldsymbol{\Theta}(t)=[Q_1(t),Q_2(t),\cdots,Q_n(t),L_1(t),L_2(t),\cdots,L_n(t),B(t)]^{\mathrm{T}}$ 为框架中所有队列的剩余量.本文定义每个时间片的 drift-plus-penalty 方程为

$$D(\boldsymbol{\Theta}(t))=\Delta Y(\boldsymbol{\Theta}(t))+VE\Big(\sum_{i\in N}e_i(t)+[\sum_{j\in M}E_j(t)]\mid\boldsymbol{\Theta}(t)\Big),$$

其中,$\Delta Y(\boldsymbol{\Theta}(t))=E[Y(\Theta(t+1))-Y(\boldsymbol{\Theta}(t))]$,$V$ 是目标函数最优性和队列稳定性之间的权衡参数.根据李雅普诺夫 drift-plus-penalty 方程,本文将问题 P 转化为

$$P:\min_{\boldsymbol{\Theta}(t)}\ D(\boldsymbol{\Theta}(t)),$$

s.t. 式(1)~式(8)成立.

因为新的目标函数包含难以解决的二次项,所以本文将在最小化引理 1 中给出目标函数的上限(即下面引理 1 中式(10)的右边).

引理 1. 对于任意 $\boldsymbol{\Theta}(t)$,李雅普诺夫 drift-plus-penalty 方程 $D(\boldsymbol{\Theta}(t))$满足:

$$D(\boldsymbol{\Theta}(t))\leqslant E[F^*+f(t)+\sum_{i\in N}\mu_i(t)x_i(t)+\sum_{i\in N}\delta_i(t,\boldsymbol{a}(t))y_i(t)+\sum_{i\in N}\vartheta_i(t)c_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t)\mid\boldsymbol{\Theta}(t)],\tag{10}$$

其中:

$$\mu_i(t)=-Q_i(t)-B(t)+V\rho\frac{p_i(s_i(t))}{s_i(t)},$$

$$\delta_i(t,\boldsymbol{a}(t))=-Q_i(t)+L_i(t)+V\frac{p_i}{\sum_{j\in M}a_{ij}(t)r_{ij}},$$

$$\vartheta_i(t)=-(Q_i(t)+B(t))\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big),$$

$$\zeta_i(t)=VP_j(t),$$

F^* 是常量,$f(t)$是不包含控制变量的部分.

证明. 根据 Q_i 和 $L_i(t)$的更新规则,可以得到

$$Q_i(t+1)^2=Q_i(t)^2+(k_i(t)-x_i(t)-y_i(t))^2+2Q_i(t)[k_i(t)-x_i(t)-y_i(t)],$$

$$L_i(t+1)^2=L_i(t)^2+\Big[y_i(t)-c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)\Big]^2+2L_i(t)\Big[y_i(t)-c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)\Big].$$

根据 $B(t)$的更新规则以及事实$([a-b]^+ +c)^2\leqslant a^2+b^2+c^2+2a(c-b)$可以得到:

$$B(t+1)^2\leqslant B(t)^2+(N\lambda d_{\max})^2+U(t)^2+2B(t)[U(t)-N\lambda d_{\max}].$$

将上述等式和不等式代入李雅普诺夫 drift-plus-penalty 方程 $D(\boldsymbol{\Theta}(t))$可以得到:

$$D(\boldsymbol{\Theta}(t)) \leqslant E\Big\{\frac{1}{2}\sum_{i\in N}\Big[(k_i(t)-x_i(t)-y_i(t))^2+\Big[y_i(t)-c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)\Big]^2+(n\lambda d_{\max})^2+U(t)^2\Big]+\sum_{i\in N}\Big[Q_i(t)(k_i(t)-x_i(t)-y_i(t))+L_i(t)\Big[y_i(t)-c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)\Big]+B(t)(U(t)-n\lambda d_{\max})\Big]\Big\}+VE\Big(\sum_{i\in N}e_i(t)+\Big[\sum_{j\in M}E_j(t)\Big]\mid\boldsymbol{\Theta}(t)\Big).$$

整理得到:

$$D(\boldsymbol{\Theta}(t)) \leqslant E\Big[\sum_{i\in N}\mu_i(t)x_i(t)+\sum_{i\in N}\delta_i(t,\boldsymbol{a}(t))y_i(t)+\sum_{i\in N}\vartheta_i(t)c_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t)+\sum_{i\in N}\Big(k_i^2(t)+x_i^2(t)+y_i^2(t)+\Big[c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)\Big]^2\Big)+(n\lambda d_{\max})^2+\sum_{i\in N}[B(t)k_i(t)-B(t)n\lambda d_{\max}]\mid\boldsymbol{\Theta}(t)\Big],$$

其中:

$$\mu_i(t)=-Q_i(t)-B(t)+V\rho\frac{p_i(s_i(t))}{s_i(t)},$$

$$\delta_i(t,\boldsymbol{a}(t))=-Q_i(t)+L_i(t)+V\frac{p_i}{\sum_{j\in M}a_{ij}(t)r_{ij}},$$

$$\vartheta_i(t)=-(Q_i(t)+B(t))\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big),$$

$$\zeta_i(t)=VP_j(t).$$

我们假设 $k_i(t)\leqslant k_{\max}$, $x_i(t)\leqslant x_{\max}$, $y_i(t)\leqslant y_{\max}$, $\min(v_i/\rho,L_i(t))\leqslant z_{\max}$,整理得到:

$$D(\boldsymbol{\Theta}(t)) \leqslant E[F^*+f(t)+\sum_{i\in N}\mu_i(t)x_i(t)+\sum_{i\in N}\delta_i(t,\boldsymbol{a}(t))y_i(t)+\sum_{i\in N}\vartheta_i(t)c_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t)\mid\boldsymbol{\Theta}(t)],$$

其中:

$$F^*=nk_{\max}^2+nx_{\max}^2+ny_{\max}^2+nz_{\max}^2+(n\lambda d_{\max})^2,$$

$$f(t)=\sum_{i\in N}B(t)k_i(t)-B(t)n\lambda d_{\max}.\quad \text{证毕.}$$

由于 F^* 是常量,$f(t)$中不包含控制变量,因此本文在求解 drift-plus-penalty 方程 $D(\boldsymbol{\Theta}(t))$上界的过程中不考虑 F^* 和 $f(t)$. 那么最小化 drift-plus-penalty 方程 $D(\boldsymbol{\Theta}(t))$上界的问题可以表示为

$$P_1:\min_{\substack{\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{c}(t),\\ \boldsymbol{x}(t),\boldsymbol{y}(t)}}\sum_{i\in N}\mu_i(t)x_i(t)+\sum_{i\in N}\delta_i(t,\boldsymbol{a}(t))y_i(t)+\sum_{i\in N}\vartheta_i(t)c_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t),$$

s.t. 式(1)~(8)成立.

在此情况下,最小化 drift-plus-penalty 方程 $D(\boldsymbol{\Theta}(t))$的上限能使原问题 P 有良好的表现是可以被证明的(见 3.5 节定理 1).

3.2 JOSA 算法

本文提出一个在线算法 JOSA 用来最小化 drift-plus-penalty 方程 $D(\boldsymbol{\Theta}(t))$的上限.

1) 找到每个 IoT 设备中的任务队列的最佳工作负载分配、IoT 设备-基站关联以及基站睡眠调度:

$$P_2:\min_{\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{x}(t),\boldsymbol{y}(t)}\sum_{i\in N}\mu_i(t)x_i(t)+\sum_{i\in N}\delta_i(t,\boldsymbol{a}(t))y_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t),$$

s.t. 式(1)~(3),式(5)~(8)成立.

2) 找到边缘服务器资源的最佳分配:

$$P_3:\min_{\boldsymbol{c}(t)}\sum_{i\in N}\vartheta_i(t)c_i(t),$$

s.t. 式(4)成立.

3) 根据步骤 1 和步骤 2 的结果,更新框架中的实队列和虚队列.

在系统的能耗模型中(2.4 节),本文只考虑了 IoT 设备和基站的能耗,没有考虑边缘服务器的能耗. 这里讨论如何加入边缘服务器能耗对问题的影响. JOSA 算法将系统的整体调度分为 2 个独立的部分:1)IoT 应用的执行调度、IoT 设备-基站关联以及基站睡眠调度;2)服务器资源调度. 如果考虑加入边缘服务器的能耗,那么它并不会影响算法的第 1 部分. 第 2 部分则会变成求解形如 $\min_{\boldsymbol{c}(t)}\sum_{i\in N}(\vartheta_i(t)+\omega_i(t))c_i(t)$ 的问题,其中 $\omega_i(t)$为 IoT 设备 i 的虚拟机能耗.

3.3 求解 P_2 的 SlotCtrl 算法实现

问题 P_2 是在线优化算法 JOSA 需要求解的核心问题. P_2 的控制变量中包含 2 个 0-1 整数变量和 2 个连续变量,因此 P_2 是典型的混合整数规划问题,通常是 NP 难问题. 为了能够高效地对 P_2 进行求解,本文提出了一个基于松弛-对偶理论的 SlotCtrl 算法作为在线优化算法 JOSA 的核心模块. 其核心思路如图 2 所示. 首先算法将 P_2 中的 0-1 整数控制变量松弛至$[0,1]$,使问题转化为 P_4. 其次通过给定控制变量 $\boldsymbol{b}(t)$,$\boldsymbol{x}(t)$和 $\boldsymbol{y}(t)$,将 P_4 转化为 P_5. 问题 P_5 是一个较难求解的分数规划问题,因此算法引入与 P_5 具有相同最优解的线性规划(LP)问题 P_6. 通

过引入拉格朗日乘子得到 P_6 的对偶问题 Dual－P_6，并使用次梯度法获得对偶问题的优化方案. 在给定引入拉格朗日乘子的前提下，求解 LP 问题 P_7 得到 $\boldsymbol{a}(t)$. 在此基础上，SlotCtrl 算法使用次梯度法对控制变量 $\boldsymbol{b}(t)$ 和 $\boldsymbol{y}(t)$ 进行优化，即 P_8. 求解 LP 问题 P_9 得到 $\boldsymbol{x}(t)$. 最后算法通过迭代（P_5 至 P_9），直至控制变量 $\boldsymbol{a}(t)$，$\boldsymbol{b}(t)$，$\boldsymbol{x}(t)$ 和 $\boldsymbol{y}(t)$ 收敛，得到问题 P_4 的最优解. 通过分析，可以证明经由上述步骤得到的问题 P_4 的解是整数解，即 P_2 与 P_4 具有相同的最优解.

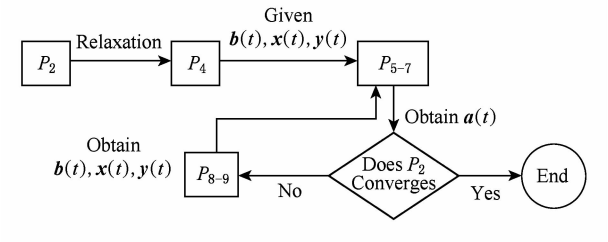


Fig. 2 Schematic diagram of SlotCtrl algorithm

图 2 SlotCtrl 算法流程示意图

1) 松弛 0-1 整数控制变量 $\boldsymbol{a}(t)$ 和 $\boldsymbol{b}(t)$.

通过将变量 $a_{ij}(t)$ 和 $b_j(t)$ 松弛至 $[0,1]$，P_2 转化为 P_4.

$$P_4:\min_{\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{x}(t),\boldsymbol{y}(t)}\ \sum_{i\in N}\mu_i(t)x_i(t)+\sum_{i\in N}\delta_i(t,\boldsymbol{a}(t))y_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t),$$

s. t. 式(1)～(3)、式(5)～(8)成立，且 $0\leqslant a_{ij}(t),b_j(t)\leqslant 1$.

在 P_4 目标函数的第 2 项中，包含控制变量 $a_{ij}(t)$ 的分数与控制变量 $y_i(t)$ 相乘的部分. 同时在 P_4 的约束条件式(6)中，$a_{ij}(t)$ 和 $y_i(t)$ 也耦合在一起. 这使问题 P_4 很难直接求解. 因此本文将 P_4 分解为 2 个子问题：首先，对于给定的控制变量 $\boldsymbol{b}(t)$，$\boldsymbol{x}(t)$ 和 $\boldsymbol{y}(t)$，对 P_4 进行求解（问题 P_5～P_9）；然后根据之前的结果，使用次梯度法对 $\boldsymbol{b}(t)$ 和 $\boldsymbol{y}(t)$ 进行优化（问题 P_8），最后求解 LP 问题 P_9 得到 $\boldsymbol{x}(t)$.

2) 对于给定的 $\boldsymbol{b}(t)$，$\boldsymbol{x}(t)$ 和 $\boldsymbol{y}(t)$，求解 P_4.

通过给定 $\boldsymbol{b}(t)$，$\boldsymbol{x}(t)$ 和 $\boldsymbol{y}(t)$，P_4 转化为问题 P_5.

$$P_5:\min_{\boldsymbol{a}(t)}\sum_{i\in N}\left(-Q_i(t)+L_i(t)+V\frac{p_i}{\sum_{j\in M}a_{ij}(t)r_{ij}}\right)y_i(t)+\sum_{i\in N}\mu_i(t)x_i(t)+\sum_{j\in M}\zeta_j(t)b_j(t),$$

s. t. 式(1)～(3)、式(6)、式(8)成立，且 $0\leqslant a_{ij}(t)\leqslant 1$.

P_5 是关于控制变量 $\boldsymbol{a}(t)$ 的分数规划. 由于其目标函数中除了 $a_{ij}(t)$ 之外的参数都是定值且 $a_{ij}(t)$ 只存在于目标函数第 1 项的分母部分，因此可以通过求解与 P_5 具有相同最优解的 LP 问题 P_6 获得 P_5 的最优解.

$$P_6:\max_{\boldsymbol{a}(t)}\sum_{i\in N}\left\{\frac{1}{Vp_iy_i(t)}\sum_{j\in M}a_{ij}(t)r_{ij}\right\},$$

s. t. 式(1)～(3)、式(6)、式(8)成立，且 $0\leqslant a_{ij}(t)\leqslant 1$.

通过引入拉格朗日乘子 $\boldsymbol{\theta}$ 和 $\boldsymbol{\varphi}$，可以得到 P_6 的对偶问题 Dual－P_6.

$$\text{Dual}-P_6:\min_{\boldsymbol{\theta},\boldsymbol{\varphi}}\ g(\boldsymbol{\theta},\boldsymbol{\varphi}),$$

其中：

$$g(\boldsymbol{\theta},\boldsymbol{\varphi})=\max_{\boldsymbol{a}(t)}\sum_{i\in N}\left\{\frac{1}{Vp_iy_i(t)}\sum_{j\in M}a_{ij}(t)r_{ij}\right\}+\{\sum_{i\in N}\theta_i[(\sum_{j\in M}a_{ij}(t)r_{ij})-y_i(t))]+\sum_{i\in N}\sum_{j\in M}\varphi_{ij}(b_j(t)-a_{ij}(t))\}. \tag{11}$$

Dual－P_6 的优化方案可以通过次梯度法获得[28]：

$$\theta_i^{iter+1}=[\theta_i^{iter}+\tau(y_i^{iter}(t)-\sum_{j\in M}a_{ij}^{iter}(t)r_{ij})]^+,$$

$$\varphi_{ij}^{iter+1}=[\varphi_{ij}^{iter}+\tau(a_{ij}^{iter}(t)-b_j^{iter}(t))]^+, \tag{12}$$

其中，τ 为每次迭代的步长，$iter$ 为迭代次数.

通过给定 $\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{b}(t)$ 和 $\boldsymbol{y}(t)$，最大化式(11)是一个标准的 LP 问题，可以表示为

$$P_7:g(\boldsymbol{\theta},\boldsymbol{\varphi})$$

s. t. 式(1)～(3)成立，且 $0\leqslant a_{ij}(t)\leqslant 1$.

P_7 可以通过单纯形法等算法有效地进行求解. 虽然 $a_{ij}(t)$ 被松弛为 0～1 的小数，但通过 P_7 求得的 $a_{ij}(t)$ 中的最优解仍然是整数. 这部分的证明详见本节引理 2 的证明.

3) 优化 $\boldsymbol{b}(t)$，$\boldsymbol{x}(t)$ 和 $\boldsymbol{y}(t)$.

由于问题 P_6 的目标函数和约束条件都是线性的，因此 P_6 是一个凸优化问题. 同时，由于 P_6 是具有线性约束条件的 LP 问题，因此 P_6 满足 Slater 条件. 根据 Slater 定理，P_6 具有强对偶性，P_6 和其对偶问题 Dual－P_6 的对偶间隙为 0.

令 $f(\boldsymbol{a}(t))$ 为在给定 $\boldsymbol{a}(t)$ 的前提下，问题 P_6 的目标函数. $\boldsymbol{b}(t)$ 和 $\boldsymbol{y}(t)$ 的优化可以通过求解问题 P_8 得到：

$$P_8:\min_{\boldsymbol{b}(t),\boldsymbol{y}(t)}\ f(\boldsymbol{a}(t)),$$

P_8 可以通过次梯度法求解：

$$b_j^{iter+1}(t)=b_j^{iter}(t)+\tau\sum_{i\in N}\varphi_{ij}^{iter},$$

$$y_i^{iter+1}(t)=y_i^{iter}(t)+\tau\theta_i^{iter}. \tag{13}$$

令 $f'(\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{y}(t))$ 为在给定 $\boldsymbol{a}(t)$，$\boldsymbol{b}(t)$，$\boldsymbol{y}(t)$ 的前提下，问题 P_5 的目标函数. $\boldsymbol{x}(t)$ 的优化可

以通过求解问题 P_9 得到:

$$P_9:\min_{\boldsymbol{x}(t)} f(\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{y}(t)),$$

s.t. 式(5)、式(7)成立.

问题 P_9 是一个标准的 LP 问题,可以通过单纯形法等算法有效地进行求解.

算法 1. SlotCtrl 算法.

输入:框架中所有队列的剩余量 $\boldsymbol{\Theta}(t)$;

输出:$\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{x}(t)$和 $\boldsymbol{y}(t)$.

① 初始化 $\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{a}(t),\boldsymbol{b}(t),\boldsymbol{x}(t)$和 $\boldsymbol{y}(t)$;

② do{

③ do{

④ 用 LP solver 求解 P_7 得到 $\boldsymbol{a}(t)$;

⑤ 根据式(12)更新 $\boldsymbol{\theta}$ 和 $\boldsymbol{\varphi}$;}

⑥ while(P_7 的目标函数最大值未收敛);

⑦ 根据式(13)更新 $\boldsymbol{b}(t)$和 $\boldsymbol{y}(t)$;

⑧ 用 LP solver 求解 P_9 得到 $\boldsymbol{x}(t)$}

⑨ while (P_5 的目标函数最小值未收敛);

4) SlotCtrl 算法分析

性质 1. 令 $\boldsymbol{A}$ 为完全单模矩阵,$\boldsymbol{b}$ 为整数向量,则多面体 $P:=\{x|\boldsymbol{A}\boldsymbol{x}\leqslant\boldsymbol{b}\}$的顶点为整数[29].

性质 2. 如果一个 LP 问题具有最优解,则至少有一个最优解在由约束条件定义的多面体的顶点处[30].

引理 2. P_7 的最优解是整数解.

证明. 为了分析 P_7 的性质,本文定义一个新向量 $\boldsymbol{a}'(t)$,$\boldsymbol{a}'(t)$将 $\boldsymbol{a}(t)$中所有的列整合在一起:

$$\begin{aligned}\boldsymbol{a}'(t)=[&a_{1,1}(t),a_{1,2}(t),\cdots,a_{1,m}(t),a_{2,1}(t),\\&a_{2,2}(t),\cdots,a_{2,m}(t),\cdots,a_{n,1}(t),\\&a_{n,2}(t),\cdots,a_{n,m}(t)]^{\mathrm{T}}.\end{aligned}$$

然后将 P_7 改写为标准形式:

$$P_7:\max_{\boldsymbol{a}'(t)} \boldsymbol{c}\boldsymbol{a}'(t),$$

s.t. $\boldsymbol{A}\boldsymbol{a}'(t)\leqslant\boldsymbol{b}$,

其中,约束矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为

$$\boldsymbol{A}\doteq\begin{bmatrix}1&1&\cdots&1&0&0&\cdots&0&\cdots&0&0&\cdots&0\\0&0&\cdots&0&1&1&\cdots&1&\cdots&1&0&\cdots&0\\\vdots&\vdots&&\vdots&\vdots&\vdots&&\vdots&&\vdots&\vdots&&\vdots\\0&0&\cdots&0&0&0&\cdots&0&\cdots&1&1&\cdots&1\\1&0&\cdots&0&1&0&\cdots&0&\cdots&1&0&\cdots&0\\0&1&\cdots&0&0&1&\cdots&0&\cdots&0&1&\cdots&0\\\vdots&\vdots&&\vdots&\vdots&\vdots&&\vdots&&\vdots&\vdots&&\vdots\\0&0&\cdots&1&0&0&\cdots&1&\cdots&0&0&\cdots&1\end{bmatrix},$$

$$\boldsymbol{b}\doteq[h_1,h_2,\cdots,h_m,1,1,\cdots,1]^{\mathrm{T}}.$$

显然向量 $\boldsymbol{b}$ 是一个整数向量. 本文需要证明约束矩阵 $\boldsymbol{A}$ 为完全单模矩阵. 将矩阵 $\boldsymbol{A}$ 分块:

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{W}_1&\boldsymbol{W}_2&\cdots&\boldsymbol{W}_m\\\boldsymbol{I}_1&\boldsymbol{I}_2&\cdots&\boldsymbol{I}_m\end{pmatrix},$$

其中,$\boldsymbol{W}_{j\in M}$是 $m\times n$ 的矩阵,其第 j 行元素全为 1,其余元素全为 0;$\boldsymbol{I}_{j\in M}$是 $n\times n$ 的单位矩阵. 令 $\boldsymbol{G}_k$ 为约束矩阵 $\boldsymbol{A}$ 的任意 $k\times k$ 的子方阵,$det(\boldsymbol{G}_k)$为 $\boldsymbol{G}_k$ 的行列式的值. 当 $k=1$ 时,$det(\boldsymbol{G}_k)=\{0,1\}$. 当 $k\geqslant 2$ 时,会有 2 种情况:

情况 1. $\boldsymbol{G}_k$ 是 $\boldsymbol{W}_j$,$\boldsymbol{I}_j$ 或 0 的子方阵. 如果 $\boldsymbol{G}_k$ 是 $\boldsymbol{W}_j$ 的子方阵,由于 $\boldsymbol{G}_k$ 至少有一行元素全为 0,因此 $det(\boldsymbol{G}_k)=0$. 如果 $\boldsymbol{G}_k$ 是 $\boldsymbol{I}_j$ 的子方阵,由于 $\boldsymbol{I}_j$ 是单位矩阵,因此 $det(\boldsymbol{G}_k)=\{0,1\}$.

情况 2. $\boldsymbol{G}_k$ 的行或列是由多于一个 $\boldsymbol{W}_j$ 或 $\boldsymbol{I}_j$ 组成. 当 $k=2$ 时,由于 $\boldsymbol{G}_k$ 的 4 个元素只能是 0 或 1,且至少有一个元素为 0,因此 $det(\boldsymbol{G}_k)=\{0,\pm1\}$. 当 $k>2$ 时,假设 $det(\boldsymbol{G}_{k-1})=\{0,\pm1\}$,我们需要证明 $det(\boldsymbol{G}_k)=\{0,1,-1\}$. 令 $\boldsymbol{G}_k(u,v)$为 $\boldsymbol{G}_k$ 第 u 行第 v 列的元素. 令 $v^*=\arg\min_v\{\sum_u \boldsymbol{G}_k(u,v)\}$, 则第 v^* 列为 $\boldsymbol{G}_k$ 中元素 1 的个数最少的一列. 令 Δv^* 为第 v^* 列元素 1 的个数,则 $\Delta v^*=\{0,1,2\}$.

如果 $\Delta v^*=0$,则第 v^* 列的元素全为 0,因此 $det(\boldsymbol{G}_k)=0$.

如果 $\Delta v^*=1$,则 $det(\boldsymbol{G}_k)=det(\boldsymbol{G}_{k-1})=\{0,\pm1\}$.

如果 $\Delta v^*=2$,则 $\boldsymbol{G}_k$ 的每列都正好有 2 个元素为 1,其中一个来自 $\boldsymbol{W}_j$,另一个来自 $\boldsymbol{I}_j$. 由于 $\boldsymbol{W}_j$ 和 $\boldsymbol{I}_j$ 中元素 1 的个数相等,因此可以通过变化使 $\boldsymbol{G}_k$ 的一行全为 0. 因此 $det(\boldsymbol{G}_k)=0$.

所以,约束矩阵 $\boldsymbol{A}$ 的任意子方阵的行列式值为 0 或 1. 根据完全单模矩阵的定义,约束矩阵 $\boldsymbol{A}$ 为完全单模矩阵. 最后,通过性质 1 和性质 2 可以得到 P_7 的最优解一定是整数解. 证毕.

引理 3. 算法 1 的结果是 P_2 的最优解.

证明. 根据引理 2 可以得到对于任意可行的 $\boldsymbol{b}(t),\boldsymbol{x}(t),\boldsymbol{y}(t),\boldsymbol{\theta}$ 和 $\boldsymbol{\varphi}$,P_4 的最优解中控制变量 $a_{ij}(t)$是取值为 0 或 1 的整数. 因此通过算法 1 得到的最优解中,$a_{ij}(t)$也是取值为 0 或 1 的整数.

对于 $\boldsymbol{b}(t)$,根据约束式(8),即 $a_{ij}(t)\leqslant b_j(t)$,如果存在任意 i 使得 $a_{ij}(t)=1$,则 $b_j(t)=1$. 而如果 $\forall i\in N,a_{ij}(t)=0$,则相应的 $b_j(t)=0$. 这是因为 P_4 的目标函数中 $b_j(t)$的系数 $\sigma_j(t)$是非负的.

综上所述,根据算法 1 求得的最优解中,$a_{ij}(t)$

和 $b_j(t)$都是取值为 0 或 1 的整数,可以确定算法 1 的最优解即为 P_2 的最优解. 证毕.

3.4 求解 P_3 的算法实现

P_3 是一个 0-1 整数规划问题,可以通过与 P_2 类似的方法进行求解. 本文将 P_3 中的 0-1 控制变量 $\boldsymbol{c}(t)$松弛至[0,1],则 P_3 转化为 P_{10}.

$$P_{10}:\min_{\boldsymbol{c}(t)} \sum_{i\in N}\vartheta_i(t)c_i(t),$$

s.t. 式(4)成立,且 $0\leqslant c_i(t)\leqslant 1$.

通过引入拉格朗日乘子 η,可以得到 P_{10}的对偶问题 Dual$-P_{10}$.

$$\text{Dual}-P_{10}:\max_{\eta} g'(\eta),$$

其中:

$$g'(\eta)=\min_{\boldsymbol{c}(t)}\Big[\sum_{i\in N}\vartheta_i(t)c_i(t)\Big]+\eta\Big[v_{\max}(t)-\sum_{i\in N}c_i(t)v_i\Big]. \tag{14}$$

Dual$-P_{10}$的优化方案可以通过次梯度法获得:

$$\eta^{iter+1}=\{\eta^{iter}+\tau[(\sum_{i\in N}c_i^{iter}(t)v_i)-v_{\max}(t)]\}^+,$$

其中,τ 为每次迭代的步长,$iter$ 为迭代次数. 通过给定 η,最大化式(14)是一个标准的 LP 问题,可以表示为

$$P_{11}:g'(\boldsymbol{\eta}),$$

s.t. 式(4)成立,且 $0\leqslant c_i(t)\leqslant 1$.

证明 $\boldsymbol{c}(t)$的最优解为整数的过程与引理 2 相同.

3.5 性能分析

定理 1 表明框架在时间平均意义下的最小能耗期望值的范围,以及框架内实队列和虚队列剩余量在时间平均意义下的积压范围.

定理 1. 对于任意 IoT 设备 i,假设平均工作负载到来量 λ_i 严格在系统处理能力范围内(例如 $\lambda_i+\varepsilon\leqslant\Omega$),那么可以通过推导得到:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}E\{[\sum_{i\in N}(Q_i(t)+L_i(t))]+B(t)\}\leqslant\frac{1}{\varepsilon}(F^*+V\sum_{i\in N}J^*(\lambda_i+\varepsilon)),$$

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}E[J(t)]\leqslant[\sum_{i\in N}J^*(\lambda_i+\varepsilon)]+\frac{F^*}{V},$$

其中,$J(t)$是 JOSA 算法调度的全网络能耗,$\sum_{i\in N}J^*(\lambda_i+\varepsilon)$是使用离线算法计算得到的全网络最小能耗.

证明. 设通过离线优化得到的最佳调度策略为 $\boldsymbol{\Phi}^*(t)=\{\boldsymbol{x}^*(t),\boldsymbol{y}^*(t),\boldsymbol{a}^*(t),\boldsymbol{b}^*(t),\boldsymbol{c}^*(t)\}$,它满足:

$$E[\lambda_i]=E\Big[x_i(t)+c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)\Big]-\varepsilon_i^1,$$

$$E[y_i(t)]=E[c_i(t)\ \min\Big(\frac{v_i(t)}{\rho},L_i(t)\Big)]-\varepsilon_i^2,$$

$$E[J(\boldsymbol{\Phi}^*(t))]=\sum_{i\in N}J^*(\lambda_i+\varepsilon).$$

首先证明平均队列的上限. 由于 JOSA 算法可以得到每个时间片的最优解,因此对于李雅普诺夫 drift-plus-penalty 方程有:

$$\Delta Y(\boldsymbol{\Theta}(t))+VE[J(t)]\leqslant F^*-\{\varepsilon\sum_{i\in N}[Q_i(t)+L_i(t)]\}-\varepsilon B(t)+VE[J(\boldsymbol{\Phi}^*(t))],$$

通过对不等式两边取期望并对 $t=0,1,2,\cdots,T-1$ 进行累加,可以得到:

$$E[Y(T)-Y(0)]+V\sum_{t=0}^{T-1}\sum_{i\in N}E(J(t))+\varepsilon\sum_{t=0}^{T-1}\{E\Big[\sum_{i\in N}Q_i(t)+L_i(t)\Big]+B(t)\}\leqslant TF^*+V\sum_{t=0}^{T-1}\sum_{i\in N}E[J(\boldsymbol{\Phi}^*(t))]=TF^*+TV\sum_{i\in N}E[J(\lambda+\varepsilon)]. \tag{15}$$

又因为 $Y(0)=0$,$Y(T)\geqslant 0$,$J(T)\geqslant 0$,式(15)两边分别除以 εT 可以得到:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}E\{[\sum_{i\in N}(Q_i(t)+L_i(t))]+B(t)\}\leqslant\frac{1}{\varepsilon}(F^*+V\sum_{i\in N}J^*(\lambda+\varepsilon)).$$

接下来证明平均能量的上限. 对于式(15),由于:

$\varepsilon\sum_{t=0}^{T-1}\{E\Big[\sum_{i\in N}Q_i(t)+L_i(t)\Big]+B(t)\}\geqslant 0$ 以及 $Y(T)\geqslant 0$,因此有:

$$V\sum_{t=0}^{T-1}\sum_{i\in N}E(J(t))\leqslant TF^*+TV\sum_{i\in N}E[J^*(\lambda+\varepsilon)].$$

不等式两边同时除以 TV,并令 $T\to\infty$得到:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}E[J(t)]\leqslant\Big[\sum_{i\in N}J^*(\lambda+\varepsilon)\Big]+\frac{F^*}{V}.$$

综上所述,系统的任务队列的堆积量和能量的上限得证. 证毕.

定理 1 表明,JOSA 算法可以在 V 增加时近似实现离线最优化. 同时,框架中的所有实队列和虚队列在时间平均意义下都是稳定的.

4 仿真评估

本文采用芬兰阿尔托大学和德国慕尼黑大学开

发的机会网络模拟器 ONE[31],结合模拟器自带的虚拟城市区域场景对提出的 JOSA 算法进行了仿真评估.本文将宏基站设置在场景的中心,30 台微基站随机分布在场景内,宏基站和微基站分别可同时关联 50 个和 20 个 IoT 设备.本文将微基站的功耗设置为 190 W,由于宏基站需要一直开启,因此在仿真评估中不计算宏基站的能耗.对于 IoT 设备每个时间片的本地计算资源,本文考虑 IoT 设备 CPU 的工作频率为 1.0 GHz.IoT 设备的 CPU 和传输功率分别设置为 60W 和 3W.IoT 设备与基站之间的传输带宽为 10 MHz,并采用文献[32]中的 pass-loss 和 SINR 模型.对于每个 IoT 设备在边缘服务器中的虚拟机资源,本文将每个设备的虚拟机 CPU 设置为单核 1.5 GHz,所有用户共享 20 台 16 核物理 CPU 的服务器.IoT 应用的平均到来量为 1.5×10^6 bps,任务处理密度为 1 CPU cycles/b,延时为 1 个单位时间片长度.在仿真模拟执行过程中,IoT 设备在场景内沿着道路按照 WorkingDay 移动模型移动.仿真模拟实验执行 3 600 个时间片.

4.1 仿真结果

1) IoT 设备数量对系统的影响.IoT 设备的数量对系统整体的影响如图 3 所示.我们发现随着框架中 IoT 设备的增多,系统整体能耗与本地执行任务相比,有稳定在 30%左右的整体节能率.随着 IoT 设备数量的逐渐增加,系统会开启更多的基站为设备提供服务.本地执行任务的比率相应上升,而卸载执行的任务比例下降.这是因为这组仿真中我们设置了只有 1 个时间片的严格的任务延时约束,即任务最多在系统中停留 1 个时间片.而当 IoT 设备增多时,服务器为每个用户提供服务的平均时间会减少,因此在严格的任务延时设置的条件下,IoT 设备不得不选择本地处理未完成的任务,以满足应用的 QoS 要求.

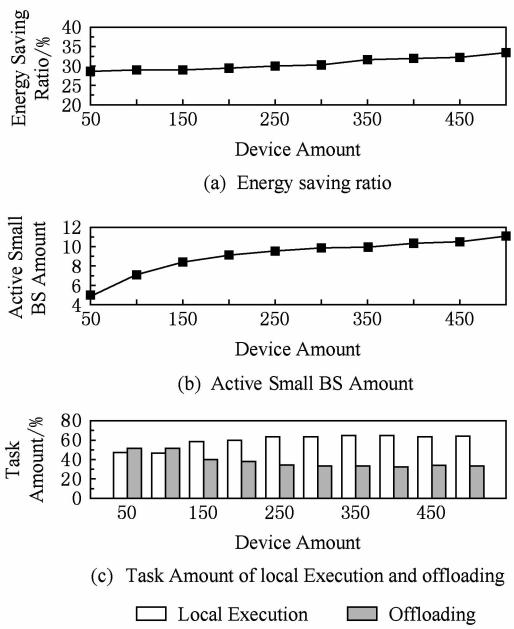


Fig. 3 The impact of IoT devices number on the COMED system ($V=1$)

图 3 IoT 设备的数量对系统的影响($V=1$)

2) V 值对系统的影响.图 4 展示了不同的 V 值对于系统性能的影响.在李雅普诺夫优化理论中,V 是权衡目标函数最优性和队列稳定性之间的参数.在 COMED 框架中,V 用来权衡系统能耗与队列的稳定性.V 值越大,代表系统更多关注于能量的节约,而更少关注队列的堆积状况.本文通过设置不同的 V 值,评估其对系统的能耗和队列堆积的影响.实验结果与理论相符,随着 V 的增大,系统的能量开销变少,更加节能.这是因为,当 V 较小时,系统会倾向于即时处理到来的任务,这种方式可能不是最节能的任务执行方式;而当 V 较大时,在延时允许的情况下,任务在系统中会不断堆积,等待最佳的执行任务的时机,例如 IoT 设备在本地积累一定的任务量,然后在某一个时间片将所积累的任务卸载到服务器上执行.

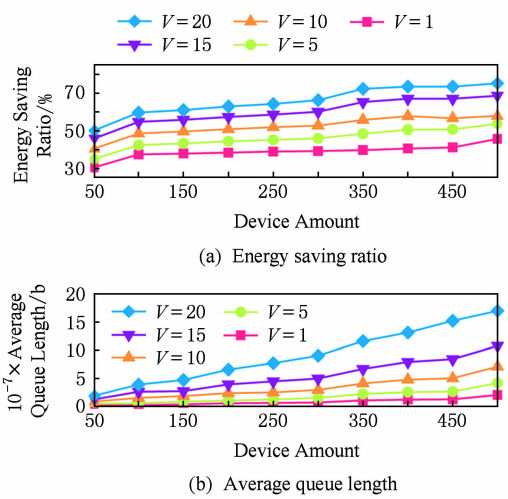
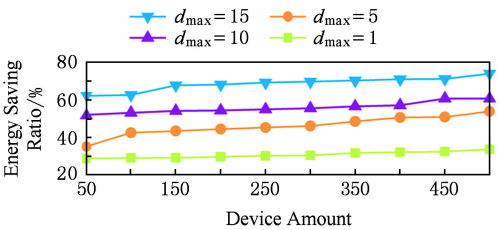


Fig. 4 The impact of V on the COMED system ($d_{\max}=5$)

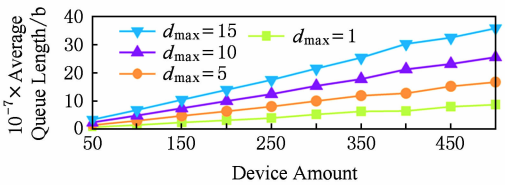
图 4 V 对系统的影响($d_{\max}=5$)

3) 任务延时对系统的影响.不同的任务延时 $d_{\max}$ 设置对系统的影响结果如图 5 所示.$d_{\max}$ 主要影响系统中队列堆积的上限.$d_{\max}$ 设置的越长,系统中队列所能堆积的任务数量越多.与 V 对系统的影响类似,系统会为队列中堆积的任务寻找最佳的执行

时机和方式,因此较长的任务延时设置将会节省更多的能量.



(a) Energy saving ratio



(b) Average queue length

Fig. 5 The impact of the task delay on the COMED system ($V=1$)

图 5 任务延时设置对系统的影响($V=1$)

4) 性能比较. 由于目前很少有研究涉及任务卸载、基站睡眠与车辆-基站关联协同调度的研究,因此本文考虑使用目前已有的任务卸载策略与不同的基站调度策略相结合的方式与 COMED 框架进行比较. 本文与 2 种方法进行了比较:①DualControl+微基站以 50% 的概率随机开启. DualControl[15] 是一种用户-运营商协作任务调度策略,在假设云端服务器为每个移动设备分配一台虚拟机的前提下,通过用户调整 CPU 速度以及运营商将云资源分配给用户虚拟机最大化两者的整体效用;②DualControl+微基站一直开启. 对于这 2 种策略,本文考虑其在调度过程中根据需求使用网络资源,但不考虑基站能耗对于调度策略的影响. 图 6 展示了 COMED 框架

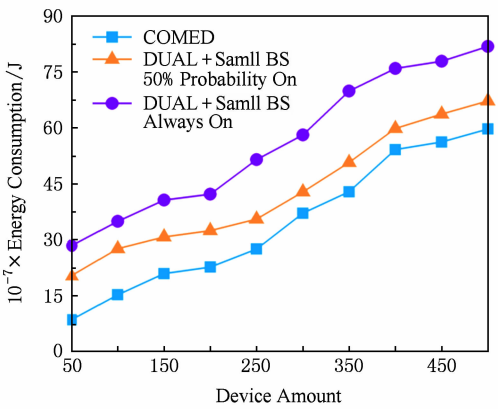
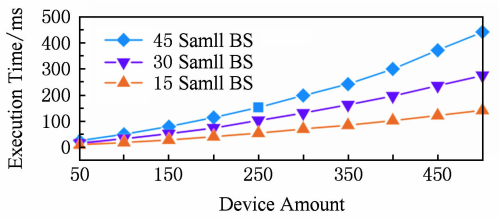


Fig. 6 System energy consumption versus DualControl algorithm

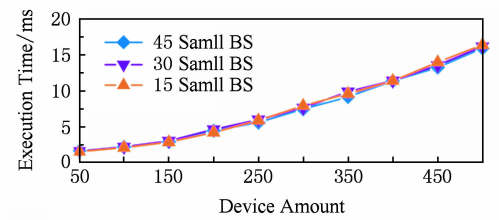
图 6 与 DualControl 算法的比较

与这 2 种方法的能量开销的比较结果. 结果说明,任务卸载、设备-基站关联以及基站睡眠调度对框架的整体能耗起着至关重要的作用. 换句话说,COMED 框架将三者结合起来,对于系统整体的节能是有意义的.

5) 算法执行时间. 本文在不同微基站数量的情况下,评估了 IoT 设备节点的数量对 JOSA 算法执行时间的影响. 本文使用了一台处理器为 Intel Core i5-2400 Processor@3.1 GHz 的台式机对算法的执行时间进行了评估,结果如图 7 所示. JOSA 算法有 2 个主要的步骤,步骤 1 用来调度 IoT 设备的任务执行和卸载量、设备基站-关联与基站睡眠,步骤 2 用来求解移动边缘服务器的任务调度. 可以看到随着 IoT 设备的数量的增加,算法的 2 个步骤执行时间的增长都近似呈线性. 在步骤 1 中,微基站数量和用户数量共同影响算法的执行时间. 而在步骤 2 中,微基站的数量对算法的执行时间几乎没有影响.



(a) Step 1



(b) Step 2

Fig. 7 The execution time of the JOSA algorithm

图 7 JOSA 算法执行时间

5 结束语

本文提出了一个基于超密集网络的移动边缘计算框架 COMED. IoT 设备通过与最适合的基站进行关联可以将任务卸载到移动边缘云服务器上执行,服务器根据定制的服务计划为每个 IoT 设备分配计算资源,在处理完设备卸载的任务后将结果传到互联网上. 为了实现 COMED 框架,本文制定了一个任务卸载问题,通过对任务负载调度、设备-基站关联以及基站睡眠调度,旨在最小化 IoT 设备和基站的能耗,同时满足任务的延时限制. 针对这一优化问题,本文提出了 JOSA 在线任务卸载算法,并通

过理论分析和仿真实验证实了 COMED 框架的性能适用于 IoT 应用.

参 考 文 献

[1] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading [J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656

[2] Zhang Ke, Mao Yuming, Leng S, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks [J]. IEEE Access, 2016, 4: 5896-5907

[3] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2016—2021 [OL]. [2017-09-23]. http://www. cisco. com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862. html

[4] Wu Jun, Zhang Zhifeng, Hong Yu, et al. Cloud radio access network (C-RAN): A primer [J]. IEEE Network, 2015, 29(1): 35-41

[5] Ge Xiaohu, Tu Song, Mao Guoqiang, et al. 5G ultra-dense cellular networks [J]. IEEE Wireless Communications, 2016, 23(1): 72-79

[6] Wu Jingjin, Zhang Yujing, Zukerman M, et al. Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey [J]. IEEE Communications Surveys & Tutorials, 2015, 17(2): 803-826

[7] Hu Yunchao, Patel M, Sabella D, et al. Mobile edge computing—A key technology towards 5G [OL]. [2017-09-23]. http://www. etsi. org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g. pdf

[8] Brown G. Mobile edge computing use cases & deployment options [OL]. [2017-09-18]. https://www. juniper. net/assets/us/en/local/pdf/whitepapers/2000642-en. pdf

[9] Atreyam S. Mobile edge computing—A gateway to 5G era [OL]. [2017-09-18] http://carrier. huawei. com/~/media/CNBG/Downloads/track/mec-whitepaper. pdf

[10] Intel® Builders. Real-world impact of mobile edge computing (MEC) [OL]. [2017-09-18]. https://builders. intel. com/docs/networkbuilders/Real-world-impact-of-mobile-edge-computing-MEC. pdf

[11] Chen Xu. Decentralized computation offloading game for mobile cloud computing [J]. IEEE Trans on Parallel and Distributed Systems, 2015, 26(4): 974-983

[12] Chen Xu, Jiao Lie, Li Wenzhong, et al. Efficient multi-user computation offloading for mobile-edge cloud computing [J]. IEEE/ACM Trans on Networking, 2016, 24(5): 2795-2808

[13] You Changsheng, Huang Kaibin, et al. Energy-efficient resource allocation for mobile-edge computation offloading [J]. IEEE Trans on Wireless Communications, 2017, 16(3): 1397-1411

[14] Mao Yuyi, Zhang Jun, Letaief K B. Dynamic computation offloading for mobile-edge computing with energy harvesting devices [J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3590-3605

[15] Kim Y, Kwak J, Chong S. Dual-side dynamic controls for cost minimization in mobile cloud computing systems [C] // Proc of the 13th IEEE Int Symp on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. Piscataway, NJ: IEEE, 2015: 443-450

[16] Shojafar M, Cordeschi N, Baccarelli E. Energy-efficient adaptive resource management for real-time vehicular cloud services [OL]. [2017-09-01]. http://ieeexplore. ieee. org/document/7448886/

[17] Lyu X, Tian H, Sengul C, et al. Multiuser joint task offloading and resource optimization in proximate clouds [J]. IEEE Trans on Vehicular Technology, 2017, 66(4): 3435-3447

[18] Sardellitti S, Scutari G, Barbarossa S. Joint optimization of radio and computational resources for multicell mobile-edge computing [J]. IEEE Trans on Signal and Information Processing over Networks, 2015, 1(2): 89-103

[19] Cheng Jinkun, Shi Yuanming, Bai Bo, et al. Computation offloading in cloud-RAN based mobile cloud computing system [C] //Proc of the 17th IEEE Int Conf on Communications. Piscataway, NJ: IEEE, 2016

[20] Liu Dantong, Wang Lifeng, Chen Yue, et al. User association in 5G networks: A survey and an outlook [J]. IEEE Communications Surveys & Tutorials, 2016, 18(2): 1018-1044

[21] Kim J M, Kim Y G, Chung S W. Stabilizing CPU frequency and voltage for temperature-aware DVFS in mobile devices [J]. IEEE Trans on Computers, 2015, 64(1): 286-292

[22] Lin Yicheng, Bao Wei, Yu Wei, et al. Optimizing user association and spectrum allocation in HetNets: A utility perspective [J]. IEEE Journal on Selected Areas in Communications, 2015, 33(6): 1025-1039

[23] Kwak J, Kim Y, Lee J, et al. DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems [J]. IEEE Journal on Selected Areas in Communications, 2015, 33(12): 2510-2523

[24] Han Siyang, Wang Xiao, Xu Linhai, et al. Frontal object perception for Intelligent Vehicles based on radar and camera fusion [C] //Proc of the 35th IEEE Chinese Control Conf. Piscataway, NJ: IEEE, 2016: 4003-4008

[25] Yan Ming, Chan C A, Li Wenwen, et al. Network energy consumption assessment of conventional mobile services and over-the-top instant messaging applications [J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3168-3180

[26] Little J D C, Graves S C. Little's law [G] //Building Intuition. Berlin: Springer, 2008: 81-100

[27] Neely M J. Stochastic network optimization with application to communication and queueing systems [J]. Synthesis Lectures on Communication Networks, 2010, 3(1): 1-211

[28] Fisher M L. The Lagrangian relaxation method for solving integer programming problems [J]. Management Science, 1981, 27(1): 1861-1871

[29] Schrijver A. Theory of Linear and Integer Programming [M]. New York: John Wiley & Sons, 1998

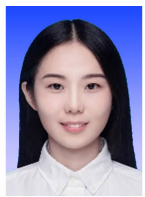
[30] Berenstein C A, Gay R. Complex Variables: An Introduction [M]. Berlin: Springer Science & Business Media, 2012

[31] Keränen A, Ott J, Kärkkäinen T. The ONE simulator for DTN protocol evaluation [C] //Proc of the 2nd Int Conf on Simulation Tools and Techniques. New York: ACM, 2009: Article No. 55

[32] Bethanabhotla D, Bursalioglu O Y, Papadopoulos H C, et al. User association and load balancing for cellular massive MIMO [C] //Proc of the 5th Information Theory and Applications Workshop. Piscataway, NJ: IEEE, 2014: 1-10

Yu Bowen, born in 1987. PhD candidate. His main research interests include traffic classification and mobile edge computing.

Pu Lingjun, born in 1988. PhD. lecturer. His main research interests include mobile edge computing, cellular IoT and SDN.



Xie Yuting, born in 1992. Master candidate. Her main research interests include vehicular cloud computing and mobile edge computing.

Xu Jingdong, born in 1965. PhD, professor and PhD supervisor. Senior member of CCF. Her main research interests include sensor networks, vehicle ad hoc networks, network security and management, and opportunistic network and computing.

Zhang Jianzhong, born in 1964. PhD, professor and PhD supervisor. Member of CCF. His main research interests include big data, network security, mobile computing and SDN.