

异构蜂窝网络中分层任务卸载：建模与优化

赵临东^{1,2}, 庄文芹^{1,2}, 陈建新^{1,2}, 周亮^{1,2}

(1. 南京邮电大学通信与网络技术国家工程研究中心, 江苏 南京 210003;

2. 南京邮电大学宽带无线通信与传感网技术教育部重点实验室, 江苏 南京 210003)

摘要: 为提高计算任务卸载的效率, 提出了一种基于 D2D 通信、移动边缘计算和云计算的分层任务卸载框架, 并引入 D2D 协作中继技术辅助用户接入远端计算资源。针对所提任务卸载框架在多用户场景中可能存在上行通信拥塞、边缘计算资源受限、D2D 复用干扰和云计算回程时延等问题, 设计了一种基于博弈论的卸载调度和负载均衡方案, 充分利用了所提任务卸载框架中各层计算和通信资源。仿真结果表明, 所提方案能够有效降低端到端时延和卸载能耗, 并在资源受限的条件下具有良好的稳定性。

关键词: 分层任务卸载; 移动边缘计算; D2D 通信; 云计算; 博弈论

中图分类号: TN92

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2020081

Hierarchical task offloading in heterogeneous cellular network: modeling and optimization

ZHAO Lindong^{1,2}, ZHUANG Wenqin^{1,2}, CHEN Jianxin^{1,2}, ZHOU Liang^{1,2}

1. National Engineering Research Center of Communication and Network Technology,
Nanjing University of Posts and Telecommunications, Nanjing 210003, China

2. Key Laboratory of Broadband Wireless Communication and Sensor Network Technology (Ministry of Education),
Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: To improve the efficiency of computation offloading, a hierarchical task offloading framework based on device-to-device (D2D) communication, mobile edge computing and cloud computing was proposed, in which cooperative D2D relay technology was introduced to assist users to access remote computing resources. Considering the negative effects of uplink channel congestion, limited edge computing capability, D2D reuse interference and backhaul delay of cloud computing in the multi-user scenario of the proposed framework, a game theory based offloading scheduling and load balancing scheme was developed by fully taking advantage of the computing and communication resources in the proposed framework. The simulation results demonstrate that the proposed scheme can effectively reduce end-to-end delay and offloading energy consumption, and also has good stability even when the network resources are limited.

Key words: hierarchical task offloading, mobile edge computing, device-to-device communication, cloud computing, game theory

1 引言

随着第五代移动通信技术 (5G, the fifth generation) 的应用, 触觉互联网等新型多媒体服务和自

动驾驶等人工智能应用将成为移动用户的热门选择^[1-2]。这些应用无一例外地需要庞大的计算资源来对数据进行实时处理, 且往往对时延有着十分苛刻的要求。然而, 在物理尺寸受限的情况下, 仅具备

收稿日期: 2020-01-06; 修回日期: 2020-03-16

基金项目: 国家自然科学基金资助项目 (No.61571240, No.61671253); 江苏高校优势学科建设工程基金资助项目

Foundation Items: The National Natural Science Foundation of China (No.61571240, No.61671253), The Priority Academic Program Development of Jiangsu Higher Education Institutions

有限计算能力和电池容量的移动设备如何承载这些计算密集型任务是一个不小的挑战^[3]。为了解决计算密集型应用和资源受限的移动设备之间的矛盾,任务卸载作为一种极具前景的解决方案应运而生。通过将任务卸载给计算能力更强大和电量更充足的节点处理,移动用户可以在现有硬件条件下享受更多新兴服务^[4]。

在任务卸载这一研究方向下,移动边缘计算(MEC, mobile edge computing)这一5G新兴卸载模式因进一步减少了卸载时延而得到业界的广泛关注^[5]。与传统云计算将计算资源配置在远离用户的远端云不同,MEC在更加靠近用户的基站侧即可实时处理数据,可以避免核心网传输的额外时延以及高峰期网络拥塞等不利影响^[6]。然而,尽管MEC通过将计算资源下沉获得了时延方面的优势,该技术仍不可避免地受到上行通信带来的额外时延和能耗的制约^[7]。尤其是考虑到任务卸载时上行流量远大于下行流量这一显著特点,如何设计高效的卸载调度方案以提升能量效率^[8-10]或削减端到端时延^[12-14]或实现二者的联合优化^[15-19],已成为MEC研究的一大热点。

为了进一步提高任务卸载的效率,有研究者提出引入基于终端直通(D2D, device-to-device)技术的卸载模式作为MEC的有效补充^[20]。该类方案又可称为D2D计算卸载,其基本原理是利用移动设备间可用计算资源的强度异构性和运行时间异步性,将计算任务通过传输效率更高的D2D链路传送给当前计算资源闲置的邻近设备处理^[21]。得益于设备间计算资源的合作共享和D2D通信的复用增益等优势,MEC和D2D的结合带给移动用户更多符合自身需求的卸载选择,并进一步提高了任务卸载的能量效率和网络资源利用率^[22]。

目前,围绕MEC结合D2D的任务卸载方案已得到了一定研究^[23-26]。文献[23]将任务分割为3份,分别交给边缘节点、D2D合作设备和本机同时处理,并以系统承载人数为目标优化任务分割。文献[24]考虑设备移动性,为车联网设计了D2D-MEC任务卸载系统,并借助增强学习的手段实现卸载时延和能耗的联合优化。文献[25]则针对无线网络的随机性为D2D-MEC任务卸载系统设计了一种基于李雅普诺夫优化的能量效率优化方案。文献[26]提出了一种本地计算、D2D卸载和边缘计算三选一的任务卸载系统,并重点研究了蜂窝上行通信拥塞和D2D复用干扰对卸载决策的影响。

尽管现有的MEC结合D2D计算卸载的方案已取得不俗成效,但其仍在以下三方面存在着不足。第一,设备间通信资源的强度异构性和运行时间异步性没有得到利用。第二,现有方案往往假设边缘服务器给每个用户提供固定的计算资源而不设接入数量上限,忽略了用户对有限边缘计算资源的竞争带给任务卸载性能的影响。第三,现有方案往往忽略云计算在任务卸载系统中的作用,或假设边缘服务器具有和远端云对等的计算能力^[23-26]。在实际边缘计算资源受限的情况下,存在额外往返时延但计算资源更充足的云计算仍然可以起到分流卸载用户和缓解边缘负担的作用。

针对上述问题,本文提出了一种基于D2D通信、MEC和云计算的三层任务卸载框架。在所提框架下,用户自低到高依次具有设备层(本地设备和D2D合作设备)、边缘层(边缘服务器)和云层(云计算中心)3个卸载层级可选。层级越高的卸载模式具有越丰富的计算资源,而将任务卸载到更高层级处理所付出的传输成本也更高。除了构建上述分层卸载理念之外,所提框架还引入了D2D协作中继技术用于辅助用户接入更高卸载层级,以使设备间通信资源的强度异构性和运行时间异步性得到开发。本文通过考虑资源受限、复用干扰和回程时延等实际因素的影响,对所提框架中具有异构设备能力和需求的用户的卸载决策进行合理调度,任务卸载效率和网络资源利用率得到显著提高。

本文主要贡献概括如下。

1) 为了最大化任务卸载可用的计算和通信资源并提高任务卸载的效率,本文提出了一种基于D2D通信、MEC和云计算的三层任务卸载框架,并引入D2D协作中继技术辅助用户接入远端计算资源。

2) 针对所提出的三层任务卸载框架,本文兼顾上行通信拥塞、D2D复用干扰、边缘计算资源受限和云计算回程时延等实际因素的影响,为多用户场景设计了一种基于博弈论的卸载调度和负载均衡方案。

3) 通过设置仿真实验与其他计算卸载方案对比,所提方案可以有效降低端到端时延和卸载能耗,并在面对边缘计算资源减少和接入用户增多的不利情况下也保持良好的稳定性能。

2 系统模型

本节将给出结合D2D通信、MEC和云计算所设计的三层任务卸载框架的系统模型,并借助多用

户任务卸载示意（如图 1 所示）和单用户各卸载模式示意（如图 2 所示），详细说明所提框架的工作原理和各环节技术细节。

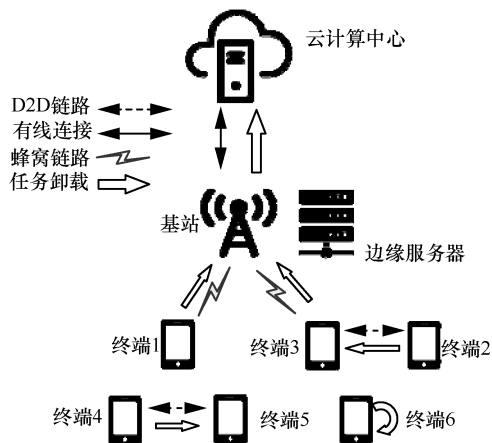


图 1 三层框架下多用户任务卸载示意

假设在一个基站覆盖范围下有若干移动设备同时配有蜂窝接口和 D2D 接口，其中具有计算密集型任务亟待处理的移动设备是所提框架的目标用户(如图 1 中的终端 1、终端 2、终端 4 和终端 6)，而 CPU 负载较低或信道质量较好的设备（如图 1 中的终端 3 和终端 5）则是目标用户潜在的 D2D 合作设备。假设每个目标用户都有一个不重复的备选合作设备，能够为其提供中继和计算卸载服务。基站侧配有计算资源有限的边缘服务器，可以处理移动设备通过蜂窝上行传来的任务，同时可以通过核

心网将用户任务卸载到计算资源充足的远端云。

每个用户可自由选择任务卸载的模式，共有 6 种模式，分别介绍如下。模式 1 为本地计算，即用本机 CPU 处理任务（如图 1 终端 6）；模式 2 为边缘计算，即通过蜂窝上行将任务卸载给基站侧边缘服务器（如图 1 终端 1）；模式 3 为云计算，即通过蜂窝上行和核心网将任务卸载给远端云（如图 1 终端 1）；模式 4 为 D2D 卸载，即通过 D2D 链路将任务交由合作设备处理（如图 1 终端 4）；模式 5 为 D2D 辅助边缘计算，即通过合作设备的 D2D 中继接入基站，再将任务卸载给基站侧边缘服务器（如图 1 终端 2）；模式 6 为 D2D 辅助云计算，即通过合作设备的 D2D 中继接入基站，再将任务卸载给远端云（如图 1 终端 2）。定义 $\mathcal{N} = \{1, 2, \dots, N\}$ 为 N 个目标用户的集合， $i \in \mathcal{N}$ 表示集合中某一用户， d_i 为与用户 i 配对的 D2D 合作设备，集合 $S = \{1, 2, 3, 4, 5, 6\}$ 为所有用户备选的策略集， $a_i \in S$ 为用户 i 所选决策， a_i 的值对应用户 i 选择的卸载模式编号。用户 i 选择不同卸载模式对应的任务卸载路径、时延来源和能耗来源等，如图 2 所示。接下来将给出具体的参数设定，并结合图 2 详细阐述这 6 种卸载模式的技术细节。

2.1 本地计算模式

当 $a_i = 1$ 时，用户 i 在本地独立处理计算任务。从图 2 可以看出，模式 1 对应的时延为本地 CPU 计算耗时，能耗为本地 CPU 计算耗电。用户 i 的计

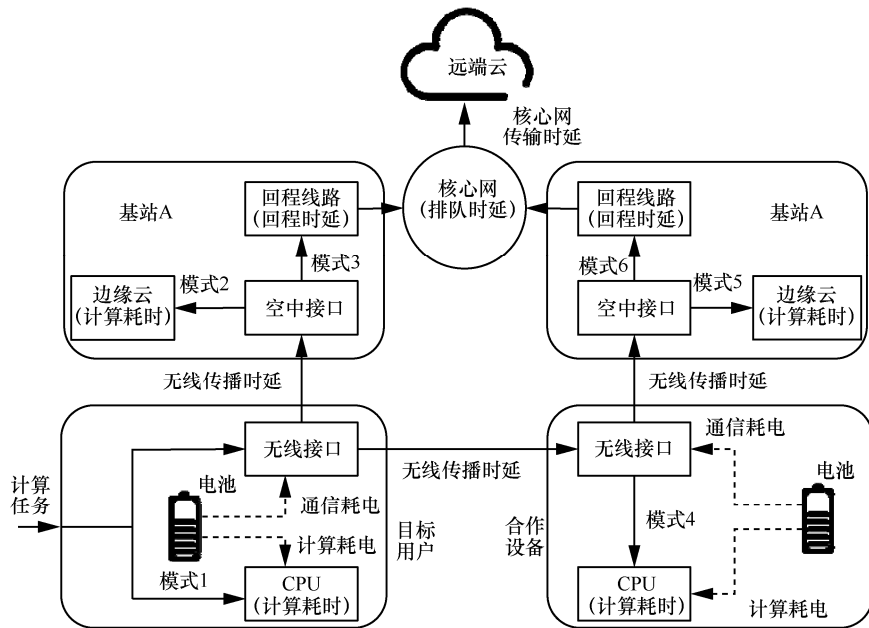


图 2 三层框架下单用户各卸载模式示意

算任务可由三元组 $T_i \triangleq \langle \lambda_i, \mu_i, \psi_i \rangle$ 描述，其中， λ_i 表示任务卸载时本地设备需向外传输的数据量， μ_i 表示任务处理完成后由外部节点传回的数据量， ψ_i 表示处理该任务所需的机器语言指令数， f_i^U 表示用户 i 本机 CPU 的计算速度， ω_i 表示用户 i 当前可用计算资源的总占比，则用户 i 在本地计算模式下的总时延可以表示为

$$t_i^L = \frac{\psi_i}{\omega_i f_i^U} \quad (1)$$

用户 i 在本地计算模式下的总能耗表示为

$$e_i^L = \gamma_i^U \psi_i \quad (2)$$

其中， γ_i^U 表示用户 i 的 CPU 执行每一机器语言指令所消耗的能量，这一参数可以通过实测得到。

考虑到时延和能耗都是任务卸载所重点关注的指标，并且不同用户对二者的重视程度和要求会有所不同，本文定义任一卸载模式下任务处理时延和能耗的加权和为用户选择该模式的开销。用户 i 本地计算模式下的开销为

$$K_i^L = \eta_i^t t_i^L + \eta_i^e e_i^L \quad (3)$$

其中， $\eta_i^t, \eta_i^e \in [0, 1]$ 分别表示用户 i 在决策时对时延和能耗的敏感程度， $\eta_i^t + \eta_i^e = 1$ 。极端情况下，如果用户所运行应用为时延敏感型且自身设备电量充足，用户可仅考虑时延 ($\eta_i^t = 1, \eta_i^e = 0$) 进行决策；如果用户设备电量紧张，可仅考虑能耗 ($\eta_i^e = 1, \eta_i^t = 0$) 进行决策。参考文献[27]，在实际应用中，借助多目标决策理论中的多属性效用方法，可以确定用户对计算能耗和时延评价的适当权重。

2.2 边缘计算模式

当 $a_i = 2$ 时，用户 i 经蜂窝上行将任务卸载给基站侧边缘服务器。在时延方面，参照文献[23]的假设，本文考虑所有用户处在准静态信道状态下，即所有蜂窝链路和 D2D 链路的信道增益在一个计算任务周期内保持不变，仅在进入下一个周期时发生变化；假设蜂窝通信和 D2D 通信使用不同的频段，因此不考虑彼此干扰对传播时延的影响；由于任务处理后的计算结果相对输入数据规模较小而蜂窝下行速率较高，因此忽略计算结果传回的时延；由于边缘服务器就近部署在基站侧并与基站以有线方式连接，因此忽略任务从基站接入边缘服务器的耗时；但考虑到边缘服务器的计算资源有限，其计算耗时不可忽略。在能耗方面，由于蜂窝下行时延

可忽略且移动设备接收功率较小，因此忽略用户接收计算结果的能耗。

从图 2 可以看出，模式 2 对应的时延为蜂窝上行的无线传播时延和边缘服务器计算耗时，能耗为用户设备无线接口发射耗电。用户 i 的蜂窝上行传输速率 $R_{i,\text{cell}}(\mathbf{a})$ 可以表示为

$$R_{i,\text{cell}}(\mathbf{a}) = \frac{B_1}{Y_1(\mathbf{a})(1 + \beta_1 Y_1(\mathbf{a}))} \text{lb} \left(1 + \frac{P_{i,\text{cell}} H_{i,\text{cell}}}{\omega_0} \right) \quad (4)$$

$$Y_1(\mathbf{a}) = \sum_{j \in \mathcal{N}} Y(a_j \in \{2, 3\})$$

其中， $\mathbf{a} = (a_1, a_2, \dots, a_N)$ 为集合 \mathcal{N} 中每个用户所选决策组成的决策向量； $Y(A)$ 为一指示函数，当 A 为真时函数值为 1，反之为 0； ω_0 为传输背景噪声功率； B_1 为蜂窝上行带宽； $H_{i,\text{cell}}$ 为用户 i 与基站间蜂窝上行信道增益； $P_{i,\text{cell}}$ 为用户 i 的蜂窝上行发射功率。需要说明的是，本文主要研究资源受限条件下的多用户接入竞争问题，并着重考虑过量用户无序竞争带来的网络拥塞效应对卸载性能的影响。因此参考文献[28]的假设，在频谱分配处额外引入参与用户数的非线性项，以反映通信拥塞的边际效应递增这一特性， β_1 为控制蜂窝上行网络拥塞效应强弱的系数。该模型可用于对基于时分多址 (TDMA, time division multiple access) 或正交频分多址 (OFDMA, orthogonal frequency division multiple access) 的带宽共享机制进行建模。

考虑到边缘服务器的计算能力有限，根据文献[11]，定义多用户参与下边缘服务器的计算耗时为

$$t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) = \frac{\psi_i \sum_{j \in \mathcal{N}} Y(a_j \in \{2, 5\})}{F} \quad (5)$$

其中， F 表示边缘服务器可用 CPU 转速。需要说明的是，描述边缘服务器计算能力受限的主流模型共有 2 种，一种是硬性设置边缘云的任务负载上限；另一种是假设计算资源均分给用户，边缘服务器计算时延将随过量接入增大^[11]。本文为反映多用户对计算资源的竞争而使用了后一种假设。

综上，用户 i 在边缘计算模式下的总时延表示为

$$t_i^M(\mathbf{a}) = \frac{\lambda_i}{R_{i,\text{cell}}(\mathbf{a})} + t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) \quad (6)$$

用户 i 在边缘计算模式下的总能耗表示为

$$e_i^M(\mathbf{a}) = P_{i,\text{cell}} \frac{\lambda_i}{R_{i,\text{cell}}(\mathbf{a})} \quad (7)$$

用户 i 边缘计算模式下的开销为

$$K_i^M(\mathbf{a}) = \eta_i^t t_i^M(\mathbf{a}) + \eta_i^e e_i^M(\mathbf{a}) \quad (8)$$

2.3 云计算模式

当 $a_i = 3$ 时, 用户 i 经蜂窝上行将计算任务传给基站, 再经核心网卸载给远端云。如图 2 所示, 模式 3 的时延除了上行无线传播时延外, 主要为回程时延、排队时延、核心网传输时延等核心网相关参数。为了简化分析, 本文假设每个用户的计算任务和计算结果通过核心网带来的时延总和均为定值 t^{core} , 这一参数可以由当前时段历史平均时延表示或通过分析近期网络时延动态预测得到。考虑到远端云的计算资源较充足, 故忽略其计算耗时。

由此, 用户 i 在云计算模式下的总时延可以表示为

$$t_i^C(\mathbf{a}) = \frac{\lambda_i}{R_{i,\text{cell}}(\mathbf{a})} + t^{\text{core}} \quad (9)$$

用户 i 在云计算模式下的总能耗与边缘计算模式能耗相同, 即 $e_i^C(\mathbf{a}) = e_i^M(\mathbf{a})$, 其在云计算模式下的开销为

$$K_i^C(\mathbf{a}) = \eta_i^t t_i^C(\mathbf{a}) + \eta_i^e e_i^C(\mathbf{a}) \quad (10)$$

2.4 D2D 卸载模式

当 $a_i = 4$ 时, 用户 i 将任务卸载给 D2D 合作设备。如图 2 所示, 模式 4 对应的时延为 D2D 链路无线传播时延以及合作设备 CPU 计算耗时。由于 D2D 回传链路也受拥塞影响, 传输速率较蜂窝下行更低, 因此尽管计算结果数据量较小, 仍不可忽略其回传时延。

由此, 用户 i 与其合作设备的 D2D 链路传输速率为

$$R_{i,d_i}(\mathbf{a}) = \frac{B_2}{Y_2(\mathbf{a})(1 + \beta_2 Y_2(\mathbf{a}))} \text{lb} \left(1 + \frac{P_{i,d_i} H_{i,d_i}}{\varpi_0} \right) \quad (11)$$

$$Y_2(\mathbf{a}) = \sum_{j \in \mathcal{N}} Y(a_j = 4)$$

其中, B_2 是模式 4 下 D2D 链路的可用带宽, H_{i,d_i} 是用户 i 与其合作设备间 D2D 信道增益, P_{i,d_i} 是用户 i 进行 D2D 通信的功率。为了简化分析, 这里假设用户 i 的合作设备 d_i 的 D2D 发射功率也为 P_{i,d_i} , 故 $R_{i,d_i}(\mathbf{a})$ 既是用户 i 进行 D2D 通信的发送速率, 也是接收速率。本文出于公平性的考虑, 假设模式 4 中 D2D 链路与模式 5 和模式 6 中 D2D 协作通信使用

不同的频段。这是由于 D2D 协作通信时用户及其合作设备均处于发射状态, 较模式 4 带来的网络拥塞效应更强。此外, 为研究过量用户无序竞争带来的网络拥塞效应对卸载性能的影响, 本文同样参考文献[28]在频谱分配时额外加入参与用户数的非线性项, β_2 为反映 D2D 通信网络拥塞效应强弱的系数。

用户 i 的合作设备 d_i 的 CPU 计算速度用 f_i^D 表示, 则用户 i 在 D2D 卸载模式下的总时延可以表示为

$$t_i^D(\mathbf{a}) = \frac{\psi_i}{f_i^D} + \frac{\lambda_i + \mu_i}{R_{i,d_i}(\mathbf{a})} \quad (12)$$

合作设备 d_i 的 CPU 每执行一机器语言指令所消耗能量用 γ_i^D 表示, 用户 i 在 D2D 卸载模式下的总能耗为

$$e_i^D(\mathbf{a}) = \gamma_i^D \psi_i + P_{i,d_i} \frac{\lambda_i + \mu_i}{R_{i,d_i}(\mathbf{a})} \quad (13)$$

此处合作设备的能耗也被计入用户在模式 4 下的总能耗。这是因为合作设备通常也是能量受限的, 其能耗往往以报酬形式在激励机制中有所体现, 使用户在请求 D2D 协作时不得不计入这一成本。为了简化分析, 本文不再展开相关议价细则, 而是以一比一的转化关系将合作设备能耗计入用户总能耗。

综上, 用户 i 在 D2D 卸载模式下的总开销为

$$K_i^D(\mathbf{a}) = \eta_i^t t_i^D(\mathbf{a}) + \eta_i^e e_i^D(\mathbf{a}) \quad (14)$$

2.5 D2D 辅助边缘计算模式

当 $a_i = 5$ 时, 用户 i 经合作设备中继与基站连接, 再将任务卸载给基站侧边缘服务器。如图 2 所示, 模式 5 的时延包括中继链路时延和边缘服务器计算耗时, 能耗为用户及其合作设备无线接口通信耗电。为了简化分析, 本文考虑模式 5 和模式 6 的 D2D 中继采用译码重传 (DF, decode-and-forward) 模式。参考文献[29], 协作通信可实现的传输速率 $R_i^{\text{DF}}(\mathbf{a})$ 表示为

$$R_i^{\text{DF}}(\mathbf{a}) = \frac{B_3 r_i}{Y_3(\mathbf{a})(1 + \beta_3 Y_3(\mathbf{a}))}$$

$$r_i = \min \{ \text{lb}(1 + \xi_{i,d_i}), \text{lb}(1 + \xi_{i,\text{cell}} + \xi_{d_i,\text{cell}}) \}$$

$$\xi_{i,d_i} = \frac{P_i^{\text{DF}} H_{i,d_i}}{\varpi_0}, \xi_{i,\text{cell}} = \frac{P_i^{\text{DF}} H_{i,\text{cell}}}{\varpi_0}, \xi_{d_i,\text{cell}} = \frac{P_{d_i}^{\text{DF}} H_{d_i,\text{cell}}}{\varpi_0}$$

$$Y_3(\mathbf{a}) = \sum_{j \in \mathcal{N}} Y(a_j \in \{5, 6\}) \quad (15)$$

其中, B_3 表示 DF 中继的可用带宽, ξ_{i,d_i} 表示合作

设备 d_i 在 DF 中继模式下收到用户 i 信号的信干噪比, $\xi_{i,\text{cell}}$ 和 $\xi_{d_i,\text{cell}}$ 分别表示基站在 DF 中继模式下收到用户 i 及其合作设备信号的信干噪比, P_i^{DF} 和 $P_{d_i}^{\text{DF}}$ 分别表示用户 i 及其合作设备在 DF 中继模式下的发射功率, β_3 为 D2D 中继网络拥塞效应强弱的系数。

由此, 用户 i 在 D2D 辅助边缘计算模式下的总时延表示为

$$t_i^{\text{DM}}(\mathbf{a}) = \frac{\lambda_i}{R_i^{\text{DF}}(\mathbf{a})} + t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) \quad (16)$$

用户 i 在 D2D 辅助边缘计算模式下的总能耗表示为

$$e_i^{\text{DM}}(\mathbf{a}) = (P_i^{\text{DF}} + P_{d_i}^{\text{DF}}) \frac{\lambda_i}{R_i^{\text{DF}}(\mathbf{a})} \quad (17)$$

用户 i 在 D2D 辅助边缘计算模式下的总开销为

$$K_i^{\text{DM}}(\mathbf{a}) = \eta_i^t t_i^{\text{DM}}(\mathbf{a}) + \eta_i^e e_i^{\text{DM}}(\mathbf{a}) \quad (18)$$

2.6 D2D 辅助云计算模式

当 $a_i = 6$ 时, 用户 i 经合作设备中继与基站连接, 再经核心网将任务卸载给远端云。如图 2 所示, 模式 6 的时延包括 D2D 协作中继链路时延和核心网传输相关时延, 能耗为用户及其合作设备无线接口通信耗电。

用户 i 在 D2D 辅助云计算模式下的总时延表示为

$$t_i^{\text{DC}}(\mathbf{a}) = \frac{\lambda_i}{R_i^{\text{DF}}(\mathbf{a})} + t^{\text{core}} \quad (19)$$

用户 i 在 D2D 辅助边缘计算模式下的总能耗与模式 5 能耗相同, 即 $e_i^{\text{DC}}(\mathbf{a}) = e_i^{\text{DM}}(\mathbf{a})$, 其在 D2D 辅助云计算模式下的开销为

$$K_i^{\text{DC}}(\mathbf{a}) = \eta_i^t t_i^{\text{DC}}(\mathbf{a}) + \eta_i^e e_i^{\text{DC}}(\mathbf{a}) \quad (20)$$

由上述分析可以看出, 本文框架下不同用户的决策之间存在耦合关系, 过多用户选择相同的卸载路径不利于提高任务卸载性能。因此, 网络运营者有必要为多用户参与的情况设计卸载调度方案, 以起到优化系统效率和实现负载均衡的效果。

3 三层框架下多用户任务卸载方案

本节基于上述系统模型为所提框架设计了一个基于博弈论的多用户任务卸载方案。考虑到所提框架下所有用户均希望最小化自身开销, 而不同用

户设备性能、所运行应用和对时延能耗的敏感程度不同, 基于最优化的方案得到的全局最优或次优调度结果往往建立在牺牲部分用户的基础上。本节旨在寻求一个兼具公平性和效率的帕累托最优调度结果, 因此使用博弈论设计卸载方案。

这里首先给出博弈论中最优决策的定义。假设 $\mathbf{a}_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$ 为用户 i 以外的其他用户所选决策组成的向量。约定俗成地, 给定向量 \mathbf{a}_{-i} , 用户 i 的最优决策将最大化自身收益。考虑到所提框架下的理性决策人以最小化自身开销为目标, 故将用户的收益函数定义为开销值的相反数, 进而可将最优决策表示为

$$a_i^* = \arg \max_{a_i \in S} U_i(a_i, \mathbf{a}_{-i}) \quad (21)$$

其中, $U_i(a_i, \mathbf{a}_{-i})$ 是用户 i 的收益函数。根据式(3)、式(8)、式(10)、式(14)、式(18)、式(20), 可得 $U_i(a_i, \mathbf{a}_{-i})$ 表达式为

$$U_i(a_i, \mathbf{a}_{-i}) = \begin{cases} -K_i^L, & a_i = 1 \\ -K_i^M(\mathbf{a}), & a_i = 2 \\ -K_i^C(\mathbf{a}), & a_i = 3 \\ -K_i^D(\mathbf{a}), & a_i = 4 \\ -K_i^{\text{DM}}(\mathbf{a}), & a_i = 5 \\ -K_i^{\text{DC}}(\mathbf{a}), & a_i = 6 \end{cases} \quad (22)$$

用户能否取得最优策略对于本文所提计算卸载框架具有重要意义。从用户角度来看, 最优策略保证了决策的个人理性, 即用户不会主动选择有损计算卸载效果的模式; 从网络运营者角度来看, 最优策略保证了空闲网络资源能够得到充分使用, 提高了用户购买云/边缘计算服务的积极性。因此, 有别于传统的计算卸载或无线接入方案(如文献[11, 23]), 所提方案需要保证用户选择最优决策, 从而不会拒绝调度结果(如拒绝使用云/边缘计算服务, 而选择本地计算)。

基于上述分析, 三层框架下各用户有关计算卸载模式的决策问题可以建模为博弈 $\Omega = \{\mathcal{N}, \Phi_{\mathcal{N}}, U_i\}$, 相关要素描述如下。

1) 集合 \mathcal{N} 为博弈 Ω 的玩家集合, 即三层框架下的用户集合。

2) $\Phi_{\mathcal{N}} = \{(a_i)_{i \in \mathcal{N}} : a_i \in S, \forall i \in \mathcal{N}\}$ 为博弈 Ω 中每个玩家可选的策略空间。

3) 博弈 Ω 中玩家 i 的收益函数 U_i 如式(22)所示。

4) 博弈 Ω 中玩家 i 的最优反应策略如式(21)所示。

首先, 假设博弈过程在基站控制下进行, 而基站可以通过反馈的方式了解到各用户及其合作设备的信道状况、CPU 性能、任务属性、当前决策等相关信息。因此, 可以认为在博弈 Ω 中的玩家将依次做出决策 (即每一步有且仅有一个玩家行动), 且任一玩家可以知道之前每一步何玩家做出了何种决策, 即博弈 Ω 为一完美信息序贯博弈。本节将说明博弈 Ω 不在基站集中求解时, 如何保证其为完美信息序贯博弈。

下面给出博弈论中纳什均衡的定义, 通过求解博弈 Ω 的纳什均衡, 可以得到符合上述要求的调度结果。

定义 1 博弈 Ω 的纳什均衡是一个决策向量 \mathbf{a}^{NE} , 当博弈中的各玩家决策按此分布时, 没有玩家可以通过单方面改变自身决策来提高自身收益。或者说, \mathbf{a}^{NE} 满足对每一玩家 i , a_i^{NE} 是 i 针对其他 $N-1$ 个玩家所选策略 \mathbf{a}_{-i} 的最优反应策略, 即

$$a_i^{\text{NE}} = \arg \max_{a_i \in \mathcal{N}} U_i(a_i, \mathbf{a}_{-i}^{\text{NE}}), \forall i \in \mathcal{N} \quad (23)$$

考虑到策略空间离散且有限的完美信息序贯博弈必然存在纳什均衡, 因此博弈 Ω 的纳什均衡存在且能够由本地搜索算法迭代有限步逼近得到^[26,30]。接下来具体分析当单个用户的 6 种卸载模式之一分别作为最优决策时需要满足的条件。

首先, 将 6 种模式对应的开销简化表示为

$$\begin{aligned} K_i^{\text{L}} &= c_1 \\ K_i^{\text{M}}(\mathbf{a}) &= \frac{x}{R_{i,\text{cell}}(\mathbf{a})} + \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) \\ K_i^{\text{C}}(\mathbf{a}) &= \frac{x}{R_{i,\text{cell}}(\mathbf{a})} + c_2 \\ K_i^{\text{D}}(\mathbf{a}) &= \frac{y}{R_{i,d_i}(\mathbf{a})} + c_3 \\ K_i^{\text{DM}}(\mathbf{a}) &= \frac{z}{R_i^{\text{DF}}(\mathbf{a})} + \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) \\ K_i^{\text{DC}}(\mathbf{a}) &= \frac{z}{R_i^{\text{DF}}(\mathbf{a})} + c_2 \end{aligned} \quad (24)$$

其中, 定常数 c_1 由式(3)得到, 其余定常数表示如下:

$$x = (\eta_i^{\text{t}} + \eta_i^{\text{e}} P_{i,\text{cell}}) \lambda_i, \quad y = (\eta_i^{\text{t}} + \eta_i^{\text{e}} P_{i,d_i}) (\lambda_i + \mu_i), \quad z = [\eta_i^{\text{t}} + \eta_i^{\text{e}} (P_i^{\text{DF}} + P_{d_i}^{\text{DF}})] \lambda_i, \quad c_2 = \eta_i^{\text{t}} t^{\text{core}}, \quad c_3 = \eta_i^{\text{t}} \frac{\psi_i}{f_i^{\text{D}}} + \eta_i^{\text{e}} \gamma_i^{\text{D}} \psi_i \circ$$

由于博弈 Ω 为一策略空间有限且离散的完美信息序贯博弈, 在确定某玩家的最优决策时, 其他玩家的决策可被视为定常值 (完备的全局信息已知)。比较该玩家不同决策对应的收益函数值大小, 便可通过有限次排序得到结果。由此整理式(21)、式(22)、式(24), 可得到如下结论。

1) $a_i = 6$ 是玩家 i 当前最优决策的条件表示为

$$\begin{aligned} \frac{z}{R_i^{\text{DF}}(\mathbf{a})} &\leq c_1 - c_2 \\ \frac{z}{R_i^{\text{DF}}(\mathbf{a})} - \frac{x}{R_{i,\text{cell}}(\mathbf{a})} &\leq 0 \\ \frac{z}{R_i^{\text{DF}}(\mathbf{a})} - \frac{y}{R_{i,d_i}(\mathbf{a})} &\leq c_3 - c_2 \\ \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) &\geq c_2 \end{aligned} \quad (25)$$

2) $a_i = 5$ 是玩家 i 当前最佳决策的条件表示为

$$\begin{aligned} \frac{z}{R_i^{\text{DF}}(\mathbf{a})} + \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) &\leq c_1 \\ \frac{z}{R_i^{\text{DF}}(\mathbf{a})} - \frac{x}{R_{i,\text{cell}}(\mathbf{a})} &\leq 0 \\ \frac{z}{R_i^{\text{DF}}(\mathbf{a})} + \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) - \frac{y}{R_{i,d_i}(\mathbf{a})} &\leq c_3 \\ \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) &\leq c_2 \end{aligned} \quad (26)$$

3) $a_i = 4$ 是玩家 i 当前最佳决策的条件表示为

$$\begin{aligned} \frac{y}{R_{i,d_i}(\mathbf{a})} &\leq c_1 - c_3 \\ \frac{x}{R_{i,\text{cell}}(\mathbf{a})} + \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) - \frac{y}{R_{i,d_i}(\mathbf{a})} &\geq c_3 \\ \frac{y}{R_{i,d_i}(\mathbf{a})} - \frac{x}{R_{i,\text{cell}}(\mathbf{a})} &\leq c_2 - c_3 \\ \frac{z}{R_i^{\text{DF}}(\mathbf{a})} + \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) - \frac{y}{R_{i,d_i}(\mathbf{a})} &\geq c_3 \\ \frac{y}{R_{i,d_i}(\mathbf{a})} - \frac{z}{R_i^{\text{DF}}(\mathbf{a})} &\leq c_2 - c_3 \end{aligned} \quad (27)$$

4) $a_i = 3$ 是玩家 i 当前最佳决策的条件表示为

$$\begin{aligned} \frac{x}{R_{i,\text{cell}}(\mathbf{a})} &\leq c_1 - c_2 \\ \eta_i^{\text{t}} t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) &\geq c_2 \\ \frac{x}{R_{i,\text{cell}}(\mathbf{a})} - \frac{y}{R_{i,d_i}(\mathbf{a})} &\leq c_3 - c_2 \\ \frac{x}{R_{i,\text{cell}}(\mathbf{a})} - \frac{z}{R_i^{\text{DF}}(\mathbf{a})} &\leq 0 \end{aligned} \quad (28)$$

5) $a_i = 2$ 是玩家 i 当前最佳决策的条件表示为

$$\begin{aligned} \frac{x}{R_{i,\text{cell}}(\mathbf{a})} + \eta_i^t t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) &\leq c_1 \\ \eta_i^t t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) &\leq c_2 \\ \frac{x}{R_{i,\text{cell}}(\mathbf{a})} + \eta_i^t t_{i,\text{comp}}^{\text{edge}}(\mathbf{a}) - \frac{y}{R_{i,d_i}(\mathbf{a})} &\leq c_3 \\ \frac{x}{R_{i,\text{cell}}(\mathbf{a})} - \frac{z}{R_i^{\text{DF}}(\mathbf{a})} &\leq 0 \end{aligned} \quad (29)$$

由此, 可以改写博弈 Ω 中玩家的最优决策式(21)为

$$a_i^* = \begin{cases} 6, & \text{式(25)成立} \\ 5, & \text{式(26)成立} \\ 4, & \text{式(27)成立} \\ 3, & \text{式(28)成立} \\ 2, & \text{式(29)成立} \\ 1, & \text{其他} \end{cases} \quad (30)$$

通过上述分析可以看出, 单个玩家在任一次决策更新时用到的定常数只在系统初始化时计算一次并存储即可。该玩家只需知道当前公开的决策向量便可由式(30)更新得到该步最优策略。通过使用最优决策响应求解式(23)所示的纳什均衡, 为本文框架设计了一种多用户任务卸载方案, 如算法1所示。

算法1 三层框架下多用户任务卸载方案

输入 背景噪声 σ_0 , 边缘计算能力 F , 核心网时延 t^{core} , 各用户计算任务 T_i , 权重 η_i^t, η_i^e ; 本地设备计算相关参数 $f_i^U, \omega_i, \gamma_i^U$; D2D 合作设备计算相关参数 f_i^D, γ_i^D ; 蜂窝通信相关参数 $B_1, H_{i,\text{cell}}, P_{i,\text{cell}}$; D2D 通信相关参数 $B_2, P_{i,d_i}, H_{i,d_i}$; 协作中继相关参数 $B_3, P_i^{\text{DF}}, P_{d_i}^{\text{DF}}$

输出 达到纳什均衡的决策向量

- 1) set 初始决策向量 $\mathbf{a}^1 = (0, \dots, 0)$, 迭代指数 $\text{index} = 1$ 。
- 2) repeat
- 3) set $\text{flag} = 0$;
- 4) for each $i \in \mathcal{N}$ do
- 5) 估计通信干扰, 计算传输速率 $R_{i,\text{cell}}, R_{i,d_i}, R_i^{\text{DF}}$ 和边缘计算耗时 $t_{i,\text{comp}}^{\text{edge}}$
- 6) 根据式(30)选出玩家 i 当前最优决策 a_i^* ;
- 7) if $a_i^{\text{index}} \neq a_i^*$ then
- 8) 玩家 i 更新当前决策, $a_i^{\text{index}} = a_i^*$,

$\text{flag} = 1$;

9) end if

10) end for

11) 系统更新决策向量, $\mathbf{a}^{\text{index}+1} = \mathbf{a}^{\text{index}}$,

$\text{index} = \text{index} + 1$

12) until $\text{flag} = 0$

13) return $\mathbf{a}^{\text{index}}$

算法1可分为系统初始化和纳什均衡拟合2个阶段。拟合阶段中, 系统决策向量通过迭代更新, 该过程又可分为两步, 第一步为玩家根据当前公开的决策向量计算传输速率和计算耗时, 第二步则根据第一步计算结果判断是否需要更新卸载模式。当没有玩家更新自身决策时, 系统达到稳态, 即取得纳什均衡。

4 仿真结果

本节通过仿真分析评估所提出的任务卸载框架和多用户卸载方案的性能。具体仿真环境设置如下。

假设有 N 个用户随机分布在边长为 200 m 的正方形小区中, 基站部署在小区的中心位置, 用户的合作设备在距其 1~50 m 随机均匀分布。用户计算任务输入数据大小 λ_i 在 50~500 KB 随机均匀分布, 输出数据规模 μ_i 为输入的 $\frac{1}{10}$,

语言指令数 $\frac{\Psi_i}{\lambda_i}$ 在 [50, 1 000] 随机均匀分布, 用户设备

计算能力 f_i^U 在 {1.0, 1.1, ..., 2.0} 等概率取值, 单位为 GIPS (giga instruction per second)。用户及其合作设备 CPU 每指令耗电 γ_i^U, γ_i^D 在 $[0, 20] \times 10^{-11}$ J/指令随机均匀分布^[11]。用户可用计算资源占比 ω_i 在 [0.75, 1] 随机均匀分布, 用户决策时对时延的敏感程度 η_i^t 在 {0, 0.5, 1} 等概率取值。合作设备计算能力 f_i^D 为 2 GIPS, 默认边缘服务器可用计算能力 F 为 40 GIPS, 核心网往返时延为 500 ms。设置蜂窝链路和 D2D 链路的信道增益为 $H_{i,j} = l_{i,j}^{-\alpha}$, 其中 $l_{i,j}$ 为节点 i 到节点 j 间距离, $\alpha = 4$ 为路径损耗因子, 取拥塞效应系数 $\beta_1 = \beta_3 = 0.01, \beta_2 = 0.001$, 其他相关参数如表1所示^[19]。

为了更好地评估所提框架和所提方案的性能, 本文选择在同等网络资源条件下, 所有用户分别采取本地计算、边缘计算、D2D 卸载、云-边缘计算的

方案进行对比,其中云-边缘计算表示用户同时具有云计算和边缘计算 2 种模式可供任务卸载。不失一般性地,用户总数 $N = \{5, 10, \dots, 50\}$ 情况下的各仿真结果均为各仿真实验重复 1 000 次并取平均的结果。

表 1 蜂窝链路和 D2D 链路的相关参数

信道参数	参数值
蜂窝带宽/MHz	10
上行发射功率/mW	200
D2D 带宽/MHz	10
D2D 发射功率/mW	100
协作通信带宽/MHz	10
DF 模式发射功率/mW	200
噪声功率/mW	10^{-10}

不同用户总数条件下参与所提方案中本地计算模式、边缘计算模式(不限接入方式)、云计算模式(不限接入方式)和 D2D 卸载模式参与用户数如图 3 所示。可以看出,随着用户总数的增加,选择边缘计算模式的参与用户数呈增长放缓趋势,选择云计算模式的参与用户数近似呈线性增长,而选择本地计算和 D2D 卸载模式的参与用户数呈增长加快趋势。这是因为,随着用户总数的增加,受限的上行通信资源和边缘计算资源渐渐不能满足用户需求,带给用户更大的开销,进而更多用户分流到计算资源稳定、通信质量更好的本地计算模式和 D2D 卸载模式。

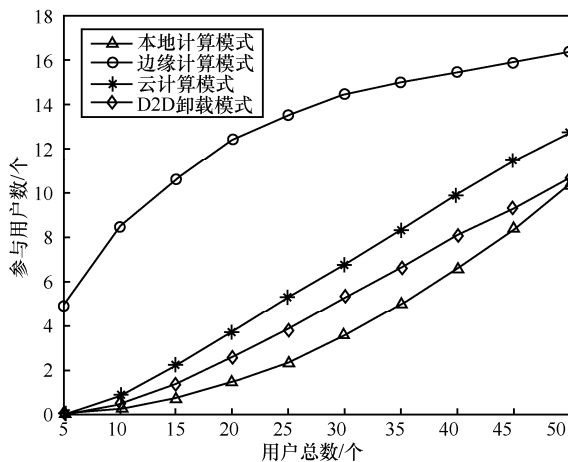


图 3 不同用户总数条件下所提方案中各卸载模式参与用户数

不同用户总数条件下所提方案和对照方案的平均计算卸载开销对比如图 4 所示。随着用户总数的增加,边缘计算方案开销很快超过其他方案,本地计算方案开销近似不变, D2D 卸载方案开销呈

近似线性增长趋势并最后超过本地计算方案开销,云-边缘计算方案开销则稳定保持增长,所提方案开销增速较缓慢。这表明边缘计算在竞争较缓和时表现占优,而 D2D 卸载方案和云-边缘计算方案得益于稳定供给的计算资源不会随系统用户数增加而迅速增长,但同样会受通信资源制约。所提方案因充分利用了空闲网络资源,在不同用户数条件下性能均明显占优,开销相比云-边缘计算方案平均减少了 61%。

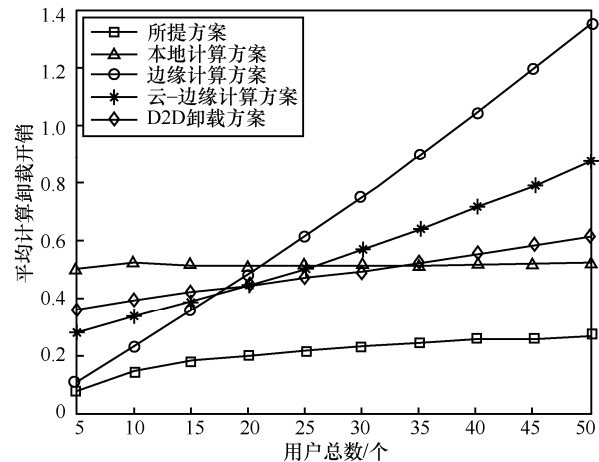


图 4 不同用户总数条件下所提方案和对照方案的平均计算卸载开销

本节分别设定用户仅考虑能耗或时延进行决策的情况,以分析调度方案对这 2 个指标的影响,平均卸载能耗和时延性能曲线分别如图 5 和图 6 所示。可以看出,随着用户总数逐渐增多,所提方案较云-边缘计算方案的能耗差逐渐增大。这是由于所提方案具备更丰富的接入手段,在上行带宽占用不断饱和时,部分用户可智能分流到负载较小的其他模式,从而避免加重网络拥塞效应。所提方案的平均时延相较云-边缘计算方案平均减少了近 68%。

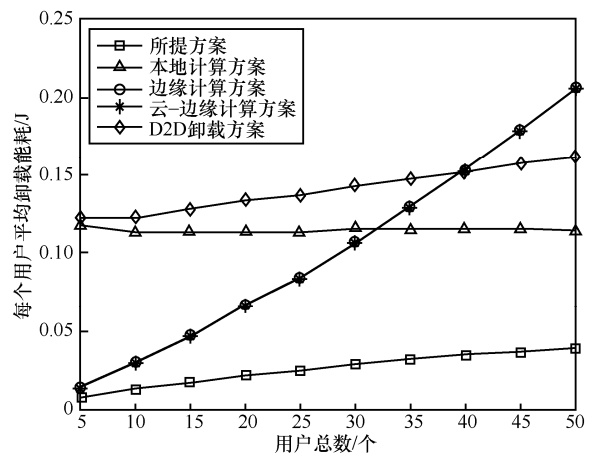


图 5 不同用户总数条件下平均卸载能耗

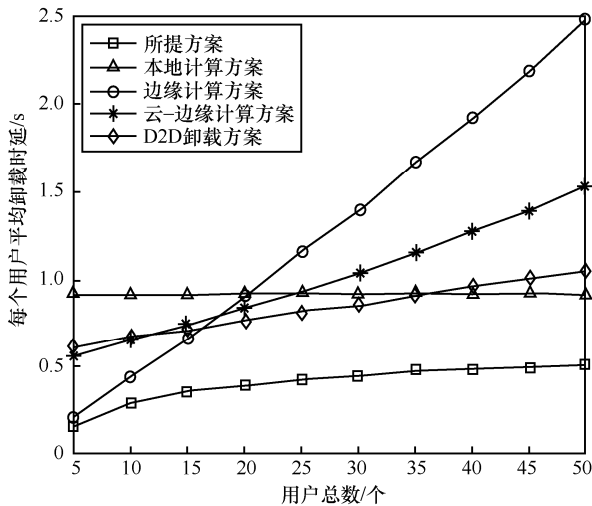


图 6 不同用户总数条件下平均卸载时延

本节在默认系统参与用户数为 30 个的条件下，针对边缘节点计算能力不同的情况对所提方案性能进行了仿真。选择所有用户均采用不受边缘节点影响的本地计算模式和均采用直接受边缘计算资源影响的边缘计算模式进行对比。不失一般性地，边缘计算资源 $F = \{20, 25, \dots, 60\}$ GIPS 情况下各仿真结果均为仿真实验重复 1 000 次并取平均的结果。不同边缘计算资源条件下所提方案中各卸载模式参与用户数如图 7 所示，不同边缘计算资源条件下所提方案和对照方案的平均计算卸载开销如图 8 所示。从图 7 可以看出，随着边缘计算资源增多，本地计算模式参与用户数总体保持不变，用户主要由计算资源供给稳定的云计算和 D2D 卸载模式向边缘计算模式迁移。图 8 则展示了所提方案的稳健性和稳定性，即通过整合网络中所有可用的计算和通信资源，在边缘计算资源较少时依然保持性能平稳占优。

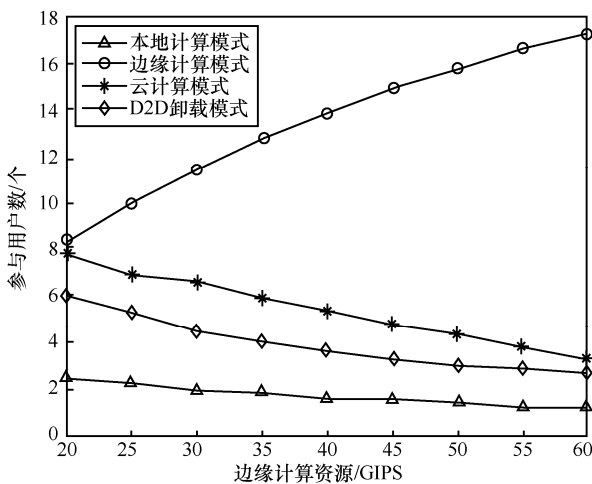


图 7 不同边缘计算资源条件下所提方案中各卸载模式参与用户数

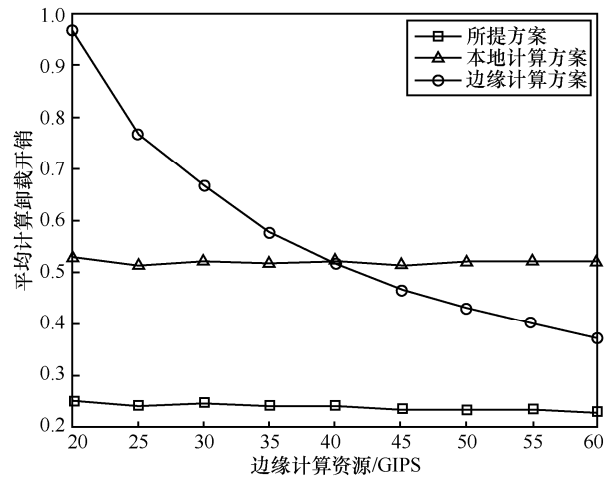


图 8 不同边缘计算资源条件下所提方案和对照方案的平均计算卸载开销

5 结束语

本文提出了一种基于 D2D 通信、移动边缘计算和云计算的分层任务卸载框架。所提框架通过自低到高提供设备层、边缘层和云层 3 种卸载层级，并引入 D2D 协作中继技术辅助用户接入更高层级的卸载模式，实现了对设备间计算和通信资源的能力异构性和运行时间异步性的充分开发，发挥了云计算分流卸载用户和减轻边缘计算负担的作用。借助博弈论的设计理念，本文针对所提框架设计的多用户任务卸载调度方案，实现了兼具公平性的效率提升和负载均衡效果，从而提升了空闲网络资源的利用率和用户购买云-边缘计算服务的积极性。仿真结果证实了所提方案在减少端到端时延与卸载能耗方面和面对边缘计算资源受限等不利条件时的优越性。未来研究将着眼于所提分层任务卸载框架中，不同业务属性带来的固定时延限制或安全性要求对卸载决策和性能的影响。

参考文献：

- [1] ZHOU L, WU D, WEI X, et al. Seeing isn't believing: QoE evaluation for privacy-aware users[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(4):1656-1665.
- [2] ZHOU L, WU D, CHEN J, et al. When computation hugs intelligence: content-aware data processing for industrial IoT[J]. IEEE Internet of Things Journal, 2018, 5(3):1657-1666.
- [3] IMTIAZ P, ALI R, ISMAIL G, et al. A survey on low latency towards 5G: RAN, core network and caching solutions[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4):3098-3130.
- [4] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3):1628-1656.
- [5] YANG K, OU S, CHEN H H. On effective offloading services for resource-constrained mobile devices running heavier mobile Internet

- applications[J]. IEEE Communications Magazine, 2008, 46(1):56-63.
- [6] TALEB T, SAMDANIS K, MADA B, et al. On multi-access edge computing: a survey of the emerging 5G network edge architecture & orchestration[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1657-1681.
- [7] MAO Y, YOU C, ZHANG J, et al. A survey on mobile edge computing: the communication perspective [J]. IEEE Communications Surveys & Tutorials, 2017, 19(4):2322-2358.
- [8] ZHANG W, WEN Y, GUAN K, et al. Energy-optimal mobile cloud computing under stochastic wireless channel[J]. IEEE Transactions on Wireless Communications, 2013, 12(9):4569-4581.
- [9] YOU C, HUANG K, CHAE H. Energy efficient mobile cloud computing powered by wireless energy transfer[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5):1757-1771.
- [10] WANG Y, SHENG M, WANG X, et al. Mobile-edge computing: partial computation offloading using dynamic voltage scaling[J]. IEEE Transactions on Communications, 2016, 64(10):4268-4282.
- [11] YOU C, HUANG K, CHAE H, et al. Energy-efficient resource allocation for mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2017, 16(3):1397-1411.
- [12] REN J, YU G, CAI Y, et al. Latency optimization for resource allocation in mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(8):5506-5519.
- [13] KAO Y H, KRISHNAMACHARI B, RA M R, et al. Hermes: latency optimal task assignment for resource-constrained mobile computing[J]. IEEE Transactions on Mobile Computing, 2017, 16(11):3056-3069.
- [14] WU Y, SHI J, NI K, et al. Secrecy-based delay-aware computation offloading via mobile edge computing for internet of things[J]. IEEE Internet of Things Journal, 2019, 6(3):4201-4213.
- [15] ZHANG G, ZHANG W, CAO Y, et al. Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices[J]. IEEE Transactions on Industrial Informatics, 2018, 14(10):4642-4655.
- [16] KIM Y, KWAK J, CHONG S. Dual-side optimization for cost-delay tradeoff in mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2017, 67(2):1765-1781.
- [17] ZHANG J, HU X, NING Z, et al. Energy-latency trade-off for energy-aware offloading in mobile edge computing networks[J]. IEEE Internet of Things Journal, 2017, 5(4):2633-2645.
- [18] CUI L, XU C, YANG S, et al. Joint optimization of energy consumption and latency in mobile edge computing for Internet of things[J]. IEEE Internet of Things Journal, 2019, 6(3):4791-4803.
- [19] CHEN X, JIAO L, LI W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2016, 24(5):2795-2808.
- [20] CHEN X, PU L, GAO L, et al. Exploiting massive D2D collaboration for energy-efficient mobile edge computing[J]. IEEE Wireless Communications, 2017, 24(4):64-71.
- [21] CHEN X, ZHOU Z, WU W, et al. Socially-motivated cooperative mobile edge computing[J]. IEEE Network, 2018, 32(6):177-183.
- [22] WEN J, REN C, ARUN K S. Energy-efficient device-to-device edge computing network: an approach offloading both traffic and computation[J]. IEEE Communications Magazine, 2018, 56(9):96-102.
- [23] HE Y, REN J, YU G, et al. D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks[J]. IEEE Transactions on Wireless Communications, 2019, 18(3): 1750-1763.
- [24] WANG Y, WANG K, HUANG H, et al. Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications[J]. IEEE Transactions on Industrial Informatics, 2019, 15(2):976-986.
- [25] JIA Q, XIE R, TANG Q, et al. Energy-efficient computation offloading in 5G cellular networks with edge computing and D2D communications[J]. IET Communications, 2019, 13(8):1122-1130.
- [26] HU G, JIA Y, CHEN Z. Multi-user computation offloading with D2D for mobile edge computing[C]// 2018 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2018:1-6.
- [27] WALLENIUS J, DYER J S, FISHBURN P C, et al. Multiple criteria decision making, multi-attribute utility theory: recent accomplishments and what lies ahead[J]. Management Science, 2008, 54(7):1336-1349.
- [28] JOSILO S, DAN G. Decentralized scheduling for offloading of periodic tasks in mobile edge computing [C]//2018 IFIP Networking Conference and Workshops. Geneva: IFIP Newsletter, 2018:1-9.
- [29] HOST-MADSEN A, ZHANG J. Capacity bounds and power allocation for wireless relay channels[J]. IEEE Transactions on Information Theory, 2005, 51(6):2020-2040.
- [30] OSBORNE M J, RUBINSTEIN A. A course in game theory[M]. Cambridge: MIT Press, 1994.

[作者简介]



赵临东 (1996-)，男，山东临沂人，南京邮电大学博士生，主要研究方向为无线资源分配、多媒体通信。



庄文芹 (1986-)，女，安徽阜阳人，南京邮电大学博士生、实验师，主要研究方向为多媒体通信。



陈建新 (1973-)，男，江苏南通人，博士，南京邮电大学副教授、硕士生导师，主要研究方向为人体运动追踪、触觉网络。



周亮 (1981-)，男，安徽芜湖人，博士，南京邮电大学教授、博士生导师，主要研究方向为多媒体通信。