

社会属性感知的边缘计算任务调度策略

王汝言 聂轩 吴大鹏* 李红霞

(重庆邮电大学通信与信息工程学院 重庆 400065)

(重庆高校市级光通信与网络重点实验室 重庆 400065)

(泛在感知与互联重庆市重点实验室 重庆 400065)

摘要: 边缘计算服务器的负载不均衡将严重影响服务能力, 该文提出一种适用于边缘计算场景的任务调度策略(RQ-AIP)。首先, 根据服务器的负载分布情况衡量整个网络的负载均衡度, 结合强化学习方法为任务匹配合适的边缘服务器, 以满足传感器节点任务的资源差异化需求; 进而, 构造任务时延和终端发射功率的映射关系来满足物理域的约束, 结合终端用户社会属性, 为任务不断地选择合适的中继终端, 通过终端辅助调度的方式实现网络的负载均衡。仿真结果表明, 所提出的策略与其他负载均衡策略相比能有效地缓解边缘服务器之间的负载和核心网的流量, 降低任务处理时延。

关键词: 计算机网络; 边缘计算; 社会属性; 负载均衡

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2020)01-0271-08

DOI: 10.11999/JEIT190301

Social Attribute Aware Task Scheduling Strategy in Edge Computing

WANG Ruyan NIE Xuan WU Dapeng LI Hongxia

(School of Telecommunication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

(Chongqing Key Laboratory of Optical Communication and Network, Chongqing 400065, China)

(Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing 400065, China)

Abstract: Unbalanced load on the edge computing server will seriously affect service capabilities, a task scheduling strategy Reinforced Q-learning-Automatic Intent Picking (RQ-AIP) for edge computing scenarios is proposed. Firstly, the load balance of the entire network is measured based on the load distribution of the server. By combining the reinforcement learning method, the appropriate edge server is matched for the task to meet the resource differentiation needs of sensor node tasks. Then, a mapping relationship between task delay and terminal transmit power is constructed to satisfy the constraints of the physical domain. Combining the social attributes of terminal, the appropriate relay terminal is continuously selected for the task to achieve the load balancing of network by terminal-assisted scheduling. Simulation results show that compared with other load balancing strategies, the proposed strategy can effectively alleviate the load between the edge servers and the traffic of the core network, reduce task processing latency.

Key words: Computer network; Edge computing; Social attribute; Load balancing

1 引言

移动边缘计算(Mobile Edge Computing,

MEC)通过在接近用户的无线接入网范围内提供便捷的计算和处理能力, 已经被广泛应用到物联网、移动大数据分析等多种场景^[1]。物联网(Internet of Things, IoT)通过将广泛区域内的大量智能设备互联, 已经成为智能城市和互联社区高级应用的一部分, 但传感器设备计算资源有限导致的边缘服务器负载不均衡仍然是物联网的主要瓶颈^[2]。移动边缘计算通过调度海量空闲边缘终端, 利用设备到设备(Device-to-Device, D2D)通信辅助调度的方式支持物联网应用, 减少了任务的延迟^[3], 为互联网设备

收稿日期: 2019-04-27; 改回日期: 2019-10-30; 网络出版: 2019-11-13

*通信作者: 吴大鹏 wudp@cqupt.edu.cn

基金项目: 国家自然科学基金(61771082, 61871062), 重庆市高校创新团队建设计划(CXTDX201601020)

Foundation Items: The National Natural Science Foundation of China (61771082, 61871062), Chongqing Funded Project of Chongqing University Innovation Team Construction (CXTDX201601020)

提供了更大的带宽^[4],避免了将任务卸载到远程计算系统造成的网络拥塞^[5]。

国内外研究人员针对MEC的任务调度问题提出了多种解决方案。文献[6]提出一种将数据和任务在边缘设备之间共享的协作边缘计算模式来实现任务的调度,通过对任务数据输入和网络流量联合建模来最小化任务完成的总体时间。文献[7]提出了一种基于Ad-hoc的移动边缘计算系统开销优化多设备任务调度策略。该策略考虑了机会消耗,时间延迟,能量消耗和货币成本,通过将问题公式化为分布式多参与者博弈问题来最小化每个移动设备的开销。文献[8]提出了一种边缘和云协同混合调度的方式,相应的感知调度程序会提取任务的特征,然后决定任务是在云端还是边缘处执行。上述研究都是单独从任务终端或者服务器的角度出发,将问题建模为时延或能耗最小化问题,没有考虑到任务调度对整个网络负载均衡的影响。文献[9]提出了在网络边缘进行分布式计算的负载均衡方案,将边缘节点之间的负载均衡问题建模为优化问题,进而使用梯度下降求解。文献[10]研究了基站之间的对等卸载问题,通过集中式的Lyapunov优化算法和分布式的非协作博弈来解决存在的负载不均衡问题。文献[11]提出了基于移动代理和蚁群优化技术的负载均衡框架,该模型中调度程序代理根据正确的决策将所接收任务分发给工作代理。文献[12]提出的HETS算法通过分析任务在不同地方的延迟和能耗制定了延迟最小化问题并提出1维搜索算法以找到最优任务调度策略。这些方法在实现负载均衡时,只考虑了任务在服务器端的调度策略,缺乏对任务传输过程的分析。虽然文献[13]在车联网背景下提出了基于社会属性的车辆间数据转发机制,通过分布式学习的方式学习车辆的社交特征并选择中心性更高的车辆作为转发车辆。但是该文没有考虑传输过程中频谱复用干扰对终端选择的影响。

据思科视觉网络指数预测,相比于2015年,2020年全球联网设备数量将达到200多亿,物联网数据将惊人地增长到4.4 ZB^[14]。传统的蜂窝通信往往会消耗大量的网络资源,物联网应用的需求难以被满足。D2D通信通过让数据直接在设备之间传输相对于通过基站传输可以极大地降低时延,减轻网络的拥塞,为任务的需求提供保障。但是,这种方案的性能很大程度上取决于D2D中继的选择以及潜在资源重用干扰的管理。因此,如何根据移动终端的社会属性选择合理的终端协助MEC服务器实现任务的实时调度是本文研究的一个重点问题。

针对上述问题,本文提出了一种带有终端社会

属性感知的RQ-AIP任务调度策略:首先,根据服务器负载程度和任务的差异化需求,使用强化Q学习方法做出任务卸载决策,然后,在充分考虑频谱复用干扰和用户发射功率的情况下,结合终端的社会属性,选取当前任务位置最佳的转发终端完成任务传输,重复执行该步骤,直到某一转发终端在目标服务器的覆盖范围之内,进而对任务进行调度,以最小化网络的负载均衡度,从而提高网络的能效。

2 网络架构与计算模型

2.1 网络架构

针对思科视觉指数指出的移动数据流量爆炸式增长,如何有效地承载各类物联网业务是未来移动通信系统发展的重要方向。为了满足数据传输过程在快速链接、实时业务等方面的需求,需要充分利用传感器节点、移动终端、基础设施组成的边缘计算网络,在靠近数据源头的网络边缘处理物联网相关应用所请求的内容,达到卸载流量、优化连接的目的。

本文的系统框架如图1所示。从各类网元所执行的基本功能来看,网络边缘的设备可以分为3层:最底层是传感器节点层,主要由物联网场景下密集部署的传感器节点组成,它们主要负责对给定事物状态参数的采集,并产生一些待处理的任务;第2层是移动终端层,主要包括用户携带一些智能设备,负责各类业务数据的接收及转发;第3层是基础设施层,包括宏基站和部署了边缘服务器的毫微微蜂窝基站,边缘服务器一般拥有较强的存储计算能力,负责对收集到的数据任务进行计算和处理。

2.2 网络模型

本文的网络模型如图2所示。小区中心部署宏基站负责管理用户之间的通信以及监测边缘服务器的负载状态,边缘服务器部署在相应的毫微微蜂窝小区(Femtocell)负责接收传感器节点的任务并提供计算和存储服务,图2同时显示了用户间的干扰情

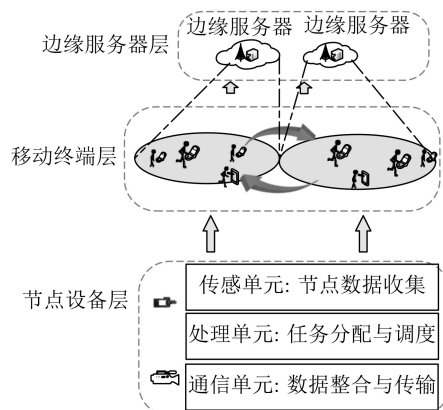


图1 系统框架图

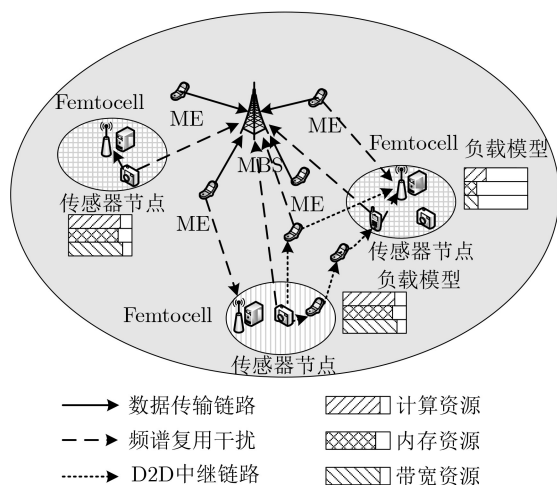


图2 网络模型图

况。毫微微蜂窝小区用户和D2D用户彼此占用正交的频谱资源，且复用宏蜂窝小区的频谱资源，因此需要考虑D2D通信和毫微微蜂窝用户通信对蜂窝用户通信造成的干扰。定义传感器设备，移动用户终端，和边缘服务器的集合分别为： $M = \{1, 2, \dots, m\}$, $N = \{1, 2, \dots, n\}$, $K = \{1, 2, \dots, k\}$ ，其中ME代表直接与宏基站通信用户，FE代表与毫微微基站通信的用户，DE代表D2D用户。

2.3 计算模型

在边缘计算网络中，任务会根据自身的属性以及网络的负载均衡状态，选择在传感器节点处或者在边缘服务器上进行计算。由于本文研究的一个重点是传感器节点如何选择合适的中继转发终端把任务传输到相应的服务器上去，所以主要研究了对D2D辅助任务卸载情形下的计算模型研究。

当边缘服务器处于过载时，需要将任务调度到其它相对空闲的服务器上，以实现边缘服务器之间的负载均衡。本文提出将任务通过D2D通信交付给合适的中继，再通过中继辅助卸载到目的边缘服务器上的方案。然而由于网络的复杂性，D2D通信往往面临多次中继选择的问题，这容易导致任务传输的时延超过任务本身的时延要求。因此，本文在中继选择方面加入了功率因素对时延约束的影响。

由于D2D传输需要复用蜂窝用户的上行链路资源，因此根据文献[15]蜂窝用户对D2D通信的信号干扰噪声比(Signal to Interference plus Noise Ratio, SINR)为

$$\gamma_{i,r_1^1} = \frac{P_i |h_{i,r_1^1}|^2}{P_{ME_c} |h_{ME_c,r_1^1}|^2 + \sigma^2} \quad (1)$$

P_i , P_{ME_c} 分别表示传感器节点和复用的蜂窝用户的发射功率， h_{i,r_1^1} 表示源节点*i*与第1个中继节点

之间的信道增益。 h_{ME_c,r_1^1} 表示干扰蜂窝用户与中继之间的信道增益， σ^2 表示加性高斯白噪声的功率。

本文考虑工作在全双工模式下D2D辅助通信，设一共需要*z*个中继终端，因此，可实现的端到端数据速率可以推导为

$$R_i^{D2D} = B \log_2 [1 + \min(\gamma_{i,r_1^1}, \gamma_{r_1^1,r_2^2}, \dots, \gamma_{r_z^z,j})] \quad (2)$$

其中， B 表示为D2D通信链路分配的带宽， γ_{i,r_1^1} , $\gamma_{r_1^1,r_2^2}$ 分别表示节点*i*与第1个中继节点，第1个中继节点与第2个中继节点的干扰噪声比。

3 目标服务器选择

3.1 算法描述与负载模型

为了有效地实现网络的负载均衡，首先要确定任务和服务器匹配关系。该问题可以通过寻找最佳的调度策略向量来解决。本文采用强化Q学习的方法来寻找最优调度策略。在任务分配过程中，需要考虑服务器的实时负载，本文设服务器的工作负载状态可由CPU利用率，内存利用率和带宽利用率这3个向量来表示，即 $\tilde{\lambda} = (\text{CPU}, \text{mem}, \text{band})$ ， $\tilde{\lambda}$ 称之为状态向量。

3.2 基于强化Q学习的服务器选择算法

本节提出服务器选择的智能算法，该方法根据不同调度决策下的网络负载均衡度进行学习。小区中心的宏基站负责收集不同服务器的负载状态以及传感器节点的实时任务分布，并将学习的结果保存下来用以进行后面的终端选择过程。接下来本文将分别介绍状态、行为和奖励函数。

(1) 状态：学习过程中的状态对应任务调度到某一服务器后的网络的负载状况。设系统中有*b*个延迟容忍型任务，每个任务都可能调度到任何一个服务器上去执行，因此整个系统中一共有 k^b 种状态。本文用负载均衡度来衡量整个网络的负载均衡情况。

首先根据所有服务器的负载情况计算出整个网络不同资源的负载中心值，然后根据中心值求出负载均衡度。负载中心值的计算如式(3)

$$l_q = \frac{1}{k} \sum_{j=1}^k \text{ava_re}_j^q \quad (3)$$

其中， ava_re_j^q 表示服务器*j*上可用的*q*资源的数量， k 表示服务器的数目。则负载均衡度求解如式(4)

$$\text{lb} = \frac{1}{k} \sum_{q=1}^3 \sum_{j=1}^k (l_q^j - l_q)^2 \quad (4)$$

定义一个负载均衡因子 ε_{ij} 来衡量任务*i*分配到服务器*j*上后服务器的实时负载情况，具体计算公式如式(5)

$$\varepsilon_{ij} = \sum_{q=1}^3 w_{ij}^q (1 - \mu_{ij}) \quad (5)$$

其中, $q=1$ 表示CPU资源; $q=2$ 表示内存资源; $q=3$ 表示带宽资源。 w_{ij}^q 表示任务*i*分配到服务器*j*后每种资源在该服务器3种资源中所占的权重, μ_{ij}^q 表示任务*i*分配到服务器*j*上后*q*资源的使用率, 如式(6)所示

$$\mu_{ij}^q = \frac{\text{req_re}_i^q}{\text{ava_re}_j^q} \quad (6)$$

req_re_i^q 表示任务*i*请求*q*资源的数量。 ava_re_j^q 的计算如式(7)所示

$$\text{ava_re}_j^q = \text{tot_re}_j^q - \sum_{i=1}^d \text{req_re}_i^q \quad (7)$$

其中*d*表示已经分配到服务器*j*上的任务的个数, tot_re_j^q 表示*j*服务器上*q*资源的总量。

(2) 动作: 动作定义为调度决策变量 $A\{a_1, a_2, \dots, a_m\}$, $a_i = \{e_i^1, e_i^2, \dots, e_i^k\}$, 其中 $\sum_{j=1}^k e_i^k = 1$ 表示一个任务只能调度到一个边缘服务器上去执行。

(3) 奖励: 这里的优化目标是使整个网络的负载均衡度最小, 而强化学习的目标是获得最大的奖励, 奖励与目标成负相关。因为最优情况下的负载均衡度为0, 因此强化学习的奖励函数定义为 $0 - \text{lb}(s, a) = -\text{lb}(s, a)$, 其中 s, a 是当前的状态和动作, $\text{lb}(s, a)$ 是在该状态和动作下的负载均衡度。如果在某个服务器上执行时服务器处于过载或满载情况, 则奖励值为-1。

在强化学习过程中, 本文使用了Q-learning算法, 它记录了每个状态的Q值。对于每一步, 计算 $Q(s, a)$ 并存储在Q表中, 这个值可以被认为是一个长期的奖励。 $Q(s, a)$ 可以表示为

$$Q(s, a) = \text{Re}(s, a) + \zeta \cdot \max Q(s', a') \quad (8)$$

其中 $\text{Re}(s, a)$ 表示初始状态下的奖励, s', a' 是下一个状态和动作。定义 ζ 为学习参数, 取值范围 $0 \leq \zeta \leq 1$ 。如果 ζ 趋近于0意味着主要考虑直接奖励, 如果 ζ 趋近于1意味着非常关注未来奖励。对于每一步, $Q(s, a)$ 的值被迭代, 直到最后得到最优结果。

4 社会属性感知的任务调度策略

在为每个任务选择了合适的服务器后, 就需要通过用户终端辅助任务进行卸载。本文将移动终端网络分为两个域: 物理域和社交域并考虑来自物理和社交网络的两种约束。具体的分析如下。

4.1 社会关系评估

网络中的终端主要由人携带, 往往具有较强的

社会属性, 因此本文中把社会属性作为判断连接可靠性的的重要依据可以提升任务卸载的成功率。本文主要从终端活跃度和与服务器的亲密度衡量用户的社会属性。

(1) 终端活跃度: 网络中的一些移动终端对网络资源的利用率较高, 并且其余终端对其依赖程度很高, 这类终端活跃度较高, 它们频繁在网络中移动且极大概率与其他终端有交互行为, 因此, 如何量化终端的活跃度对于传感器任务中继的选择有着重要的意义。本文采用给定时间内相遇终端的数量与边缘服务器范围内终端总数量的比值来衡量活跃度, 具体如式(9)所示, 其中 act_x 表示终端*x*的活跃度, $\delta(x, y)$ 表示网络中终端*i*是否与终端*j*相遇, n 表示该区域内的终端数目。

$$\text{act}_x = \frac{\sum_{y=1}^n \delta(x, y)}{n} \quad (9)$$

$$\delta(x, y) = \begin{cases} 1, & x, y \text{ 曾经相遇}, x \neq y \\ 0, & \text{其他} \end{cases} \quad (10)$$

(2) 终端-服务器亲密度: 终端间社会属性的差异还表现在与服务器的交互程度。本文定义终端与服务器的亲密度来衡量它们之间的交互程度。本文把终端与服务器在时间段*T*内的时间划分为交互持续时间和交互间隔时间, 平均交互间隔时间与时间段*T*的比值反映终端与服务器间的交互亲密度, *T*时间段内它们的交互平均间隔 T_{avg} 表示为

$$T_{\text{avg}} = \frac{\sum_{p=1}^{\text{cs}} (\text{STA}_{p+1} - \text{EDT}_p)}{\text{cs}} \quad (11)$$

其中, cs 表示终端与服务器在*T*时间段内的交互次数, STA_{p+1} 表示*p*+1次交互的开始时间, EDT_p 表示*p*次交互的结束时间。因此, 它们之间的亲密度 $\eta_{x,j}$ 如式(12)所示

$$\eta_{x,j} = \frac{\sum_{i=1}^{\text{cs}} \text{STA}_{p+1} - \text{EDT}_p}{T \cdot \text{cs}} \quad (12)$$

如果该时间段内终端与服务器没有交互, 则它们之间的亲密度用上一时间段的亲密度来表示。由于时间越早的信息参考价值越小, 因此对时间段*T*内交互发生的时间采用指数函数衰减的原理进行更新

$$\kappa_{x,j} = \sum_{p=1}^{\text{cs}} \exp\left(-\frac{(\text{NT} - \text{STA}_p)}{T}\right) \cdot \frac{\text{EDT}_p - \text{STA}_p}{T} \quad (13)$$

其中, $\kappa_{x,j}$ 表示更新过后的亲密度, NT 表示当前时

刻。根据上述变化趋势，终端与服务器通信概率 $MP_{x,j}$ 评估方式如式(14)

$$MP_{x,j} = \frac{2}{1 + \exp(-\kappa_{x,j}/\eta_{x,j})} - 1 \quad (14)$$

本文综合考虑活跃度和亲密度量化终端的重要程度，具体如式(15)所示，其中 α 为权重因子。

$$w_k = (1 - \alpha)act_x + \alpha MP_{x,j} \quad (15)$$

显然，权重因子的确定对终端重要程度的感知结果至关重要，权重因子需要能够根据实时情况不断更新以有效地区分不同区域的终端重要程度。本文采用式(16)的方式调整

$$\alpha = \frac{n_j}{n} \left(1 - \frac{\sum_{x=1}^{n_j} \kappa_{x,j}}{n_j} \bigg/ \frac{\sum_{y=1}^n \kappa_{y,j}}{n} \right) \quad (16)$$

4.2 物理域约束

在选择终端的时候，不仅要考虑终端的社会属性，还需要考虑物理域的约束以保证D2D链路的可靠性。

假设传感器节点 i 通过 z 跳D2D通信来访问目标服务器 j ，作为D2D通信的发送器，传感器节点 i 将构建到相应接收器的通信链路。在相应的接收用户处的接收SINR应该大于给定的阈值。此阈值用于确保可靠的多跳通信，其表达为式(17)的限制

$$\gamma_{DE_i, DE_i^1} = \frac{|h_{DE_i, DE_{r_1}}|^2 P_{DE_i}}{|h_{ME_c, DE_{r_1}}|^2 P_{ME_c} + \sigma^2} \geq \gamma_{DE}^{th} \quad (17)$$

其中， $h_{DE_i, DE_{r_1}}$ ， $h_{ME_c, DE_{r_1}}$ 分别表示设备 i 与第1个中继之间的信道增益以及复用的蜂窝用户与第1个中继之间的信道增益， P_{DE_i} ， P_{ME_c} 分别表示传感器节点 i 和复用的蜂窝用户 c 发射功率， γ_{DE_i, DE_i^1} 表示传感器节点 i 与第1个中继之间信道的SINR， γ_{DE}^{th} 表示需要满足的阈值限制。

根据式(2)可知，多跳中继路径的数据传输速率与每一跳的链路的信噪比有关，它取决于路径中较差链路的信噪比。因此，信噪比阈值为满足任务时延的最小阈值，具体如式(18)所示

$$\gamma_{DE}^{th} = 2^{\frac{R_s^{D2D}}{B}} - 1 = 2^{\frac{S_i f_j}{B(h_i f_j - D_i)}} - 1 \quad (18)$$

根据式(17)可以得到传感器节点 i 以及所有中继保持可靠传输的最小所需发射功率约束。

$$P_{DE_i}^{min} = \frac{\gamma_{DE}^{th} \left(|h_{ME_c, DE_{r_1}}|^2 P_{ME_c} + \sigma^2 \right)}{|h_{DE_i, DE_{r_1}}|^2} \quad (19)$$

式(19)给出了确保可靠的多跳传输的功率下限，其

中 DE_{r_1} 表示第1跳中继。另一方面，D2D通信应该遵循的另一个重要原则是新建立的链路不应该干扰现有的ME正在进行的传输。假设蜂窝用户 c 的发送功率 P_{ME_c} 在中继选择和资源分配期间是常数，式(20)表示为采用中继传输后蜂窝用户的SINR。

$$\gamma_{ME_c, MBS} = \frac{|h_{ME_c, MBS}|^2 P_{ME_c}}{\max \left\{ |h_{DE_i, MBS}|^2 P_{DE_i}, \sum_{i \neq c, i \in FE} |h_{FE_i, MBS}|^2 P_{FE_i} \right\} + \sigma^2} \quad (20)$$

同时得出在引入D2D通信之前，蜂窝用户的SINR为

$$\gamma_{ME_c, MBS}^{pre} = \frac{|h_{ME_c, MBS}|^2 P_{ME_c}}{\sum_{i \neq c, i \in FE} |h_{FE_i, MBS}|^2 P_{FE_i} + \sigma^2} \quad (21)$$

与式(21)相比，可以推断出对于ME，除了现有的干扰外，还会有一个新的干扰，标记为 I_{D2D} ，从式(20)，可以得出

$$\sum_{i \neq c, i \in FE} |h_{FE_i, MBS}|^2 P_{FE_i} + I_{D2D} = \max \left\{ \max \left\{ |h_{DE_i, MBS}|^2 P_{DE_i}, \sum_{i \neq c, i \in FE} |h_{FE_i, MBS}|^2 P_{FE_i} \right\} \right\} \quad (22)$$

联立式(19)–式(22)可得

$$\frac{I_{add}}{\sum_{i \neq c, i \in FE} |h_{FE_i, MBS}|^2 P_{FE_i} + \sigma^2} \leq \left(\frac{\gamma_{ME_c, MBS}^{pre} - \gamma_{MBS}^{th}}{\gamma_{MBS}^{th}} \right) \quad (23)$$

将式(20)–式(22)代入式(23)可得

$$\left. \begin{aligned} I_{add} &= |h_{DE_i, MBS}|^2 P_{DE_i} \\ I_{add} &\leq |h_{ME_c, MBS}|^2 P_{ME_c} \left(\frac{1}{\gamma_{MBS}^{th}} - \frac{1}{\gamma_{ME_c, MBS}^{pre}} \right) \end{aligned} \right\} \quad (24)$$

式(24)是采用D2D逐跳传输方式ME所能容忍的最大干扰，将(24)代入式(23)可以获得DE的最大允许发射功率为

$$P_{DE_i}^{max} = \frac{|h_{ME_c, MBS}|^2 P_{ME_c}}{|h_{DE_i, MBS}|} \left(\frac{1}{\gamma_{MBS}^{th}} - \frac{1}{\gamma_{ME_c, MBS}^{pre}} \right) \quad (25)$$

综上所述，可以获得多跳D2D传输的中继发射功率范围

$$P_{DE_i}^{min} \leq P_{DE_i} \leq P_{DE_i}^{max} \quad (26)$$

对于每一个产生任务的传感器节点，在其D2D通信范围内的终端用户中选择满足式(26)的终端作

为候选中继终端，然后在候选中继终端中选择 w 最大的节点作为传输中继终端，判断该节点是否在目标服务器的通信范围之内，如果在，则直接将任务上传到目的服务器上面执行，如果不在，更新式(26)的功率范围，继续执行上述操作，直到找到一个在目标服务器覆盖范围内的终端结束。

5 仿真分析

本文的仿真分析主要分为两部分：首先通过Matlab仿真平台对本文所提算法的前一部分即基于强化Q学习的任务调度算法进行了验证，其对比算法主要包括文献[11]提出的WLB-ACO算法，文献[12]提出HETS算法以及将所有任务卸载到相应本地边缘服务器的Full-Local策略。其次，通过NS2仿真平台对所提出算法后一部分中的任务投递过程进行验证。这里的对比算法包括文献[13]提出的基于社会特征的数据传输方式(SDFM)算法以及文献[16]提出的带有节点间社会等级估计的数据转发策略(Bubble Rap)算法。

本文的主要仿真参数限定如表1所示。

表1 仿真参数设置

参数设定	参数数值
任务到达率(个/s)	[0, 4]
任务所需内存(GB)	[1, 10]
任务所需CPU周期(MHz)	50
任务时延(s)	[200, 1500]
边缘服务器CPU频率(GHz)	3
无线信道带宽(MHz)	5
边缘服务器数量(个)	5
学习因子	0.5
终端发射功率(W)	[0.1, 2]
噪声功率(dBm/Hz)	-170

5.1 不同资源下的负载均衡度

图3表示的是在不同任务分布场景下4种算法负载均衡度的值。由图可知，在任务数量大小以及服务器数量可用资源一定的情况下，本文提出的RQ-AIP负载均衡策略具有最低负载均衡度，这是因为该策略采用Q学习方法通过迭代学习不断寻找最优解，而WLB-ACO会受到参数固定、求解速度慢的影响，Full-Local方案只把任务卸载到本地服务器，在任务所需资源较少时具有良好的性能，HETS算法是以时延为优化目标往往达不到较好的负载均衡度。

服务器可用计算资源不同情况下的负载均衡度变化情况如图4所示。由图可知，当可用资源一般时，Full-Local, WLB-ACO, RQ-AIP算法的负载

均衡度具有相似的下降趋势。但是，随着可用资源的增多，Full-Local算法的负载均衡度值下降的最缓慢。这是由于Full-Local算法将任务卸载到本地服务器；对于WLB-ACO，随着可用资源的增多，服务器一般不会出现过载的情况，WLB-ACO发挥的作用也就逐渐变小，从而使得负载均衡度的值变大；而本文的RQ-AIP策略是通过迭代一直寻找最优解的方法，尽管服务器不会处在过载的边缘，算法仍然会为任务选择使系统负载均衡度最小的调度策略。对于HETS算法的分析同之前一样，由于它是以时延最小为目标的，所以它的负载均衡度往往是这3种算法中最差的。

不同任务数量下的负载均衡度变化情况如图5所示。从图中可以看出，随着任务数量的增加，WLB-ACO, HETS和Full-Local算法的负载均衡度

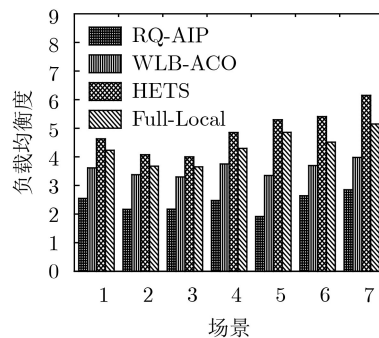


图3 不同场景下的负载均衡度

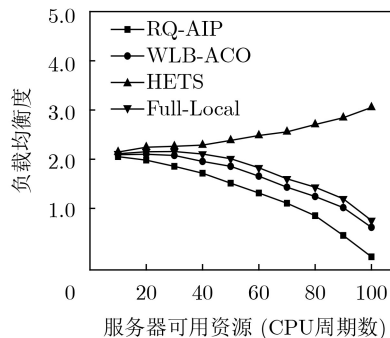


图4 不同服务器计算资源下的负载均衡度

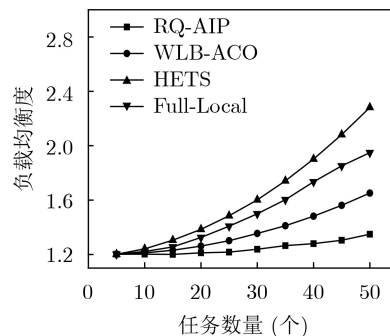


图5 不同任务数量下的负载均衡度

都呈现出较为明显的上升趋势。这是由于随着任务数量的增加，调度问题的规模越来越大，各服务器的压力也越来越大，这3种算法很难在有限的时延要求内完成任务的调度，能够实现的系统负载均衡度是有限的，因此呈现出上升的趋势。本文的RQ-AIP策略中的任务调度算法是一种可以根据历史经验进行学习的算法，所以当实际情况发生变化时，算法不会从最初的状态开始进行迭代，因此本文可以实现低的且较为稳定的负载均衡度。

5.2 不同终端数目下的任务投递率

图6描述了终端数量与任务平均完成时延的关系。可见，随着终端数目的增加，各种算法的任务平均完成时间都呈下降趋势，其原因在于随着终端数目的增加，终端相遇的机会也会增加。在每一次转发的过程中遇到可选中继的概率也会大很多，快速转发到目的服务器的概率也会增大，其中本文提出的算法的传输延迟比SDFM^[13]算法和Bubble Rap^[16]算法平均降低了18%和25%。这是因为本文算法综合考虑了用户的社会属性和物理属性。在社会属性方面考虑了用户的活跃度和与目标服务器亲密密度两者在不同情况下对任务完成时延的影响。物理属性方面在考虑了用户发射功率限制下中继选择对任务完成时延的影响。

图7显示了3种算法的任务投递率与用户发送功率之间的关系。本文中，用户的发送功率作为任务在时延要求范围内成功多跳到目标服务器的物理约

束条件，它考虑了D2D通信对蜂窝用户通信的影响，随着跳数的增加，用户的发射功率也应该增加才能弥补由于跳数增加所导致的时延的增加，因此发射功率越大任务成功交付的概率也就越大。而SDFM和Bubble Rap算法只考虑了用户的社会属性，因此它们具有相似且较低的任务投递速率。

6 结束语

本文在边缘计算场景下提出了一种社会属性感知的任务调度策略(RQ-AIP)。首先，根据边缘服务器的资源分布确定网络的负载均衡度；然后，根据强化Q学习算法确定传感器节点任务的最佳执行服务器；最后传感器节点根据用户的发送功率作为衡量D2D通信干扰的指标，选择合适的中继用户，进而完成任务转发。结果表明，本文所提策略能够在不干扰蜂窝用户通信的情况下完成任务调度，并且能够有效改善网络资源分配不均匀的情况。

参考文献

- [1] KUMAR K, LIU Jibang, LU Y H, *et al.* A survey of computation offloading for mobile systems[J]. *Mobile Networks and Applications*, 2013, 18(1): 129–140. doi: [10.1007/s11036-012-0368-0](https://doi.org/10.1007/s11036-012-0368-0).
- [2] ZENG Deze, GU Lin, GUO Song, *et al.* Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system[J]. *IEEE Transactions on Computers*, 2016, 65(12): 3702–3712. doi: [10.1109/TC.2016.2536019](https://doi.org/10.1109/TC.2016.2536019).
- [3] MAO Yuyi, ZHANG Jun, and 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. doi: [10.1109/JSAC.2016.2611964](https://doi.org/10.1109/JSAC.2016.2611964).
- [4] CHEN Xu, JIAO Lei, LI Wenzhong, *et al.* Efficient multi-user computation offloading for mobile-edge cloud computing[J]. *IEEE/ACM Transactions on Networking*, 2016, 24(5): 2795–2808. doi: [10.1109/TNET.2015.2487344](https://doi.org/10.1109/TNET.2015.2487344).
- [5] MACH P and BECVAR Z. Mobile edge computing: A survey on architecture and computation offloading[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(3): 1628–1656. doi: [10.1109/COMST.2017.2682318](https://doi.org/10.1109/COMST.2017.2682318).
- [6] SAHNI Y, CAO Jiannong, and LEI Yang. Data-aware task allocation for achieving low latency in collaborative edge computing[J]. *IEEE Internet of Things Journal*, 2019, 6(2): 3512–3524. doi: [10.1109/JIOT.2018.2886757](https://doi.org/10.1109/JIOT.2018.2886757).
- [7] LI Tianze, WU Muqing, ZHAO Min, *et al.* An overhead-optimizing task scheduling strategy for ad-hoc based mobile edge computing[J]. *IEEE Access*, 2017, 5: 5609–5622. doi: [10.1109/ACCESS.2017.2678102](https://doi.org/10.1109/ACCESS.2017.2678102).

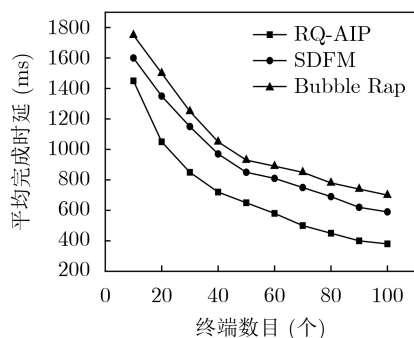


图6 不同终端数目下的平均完成时延

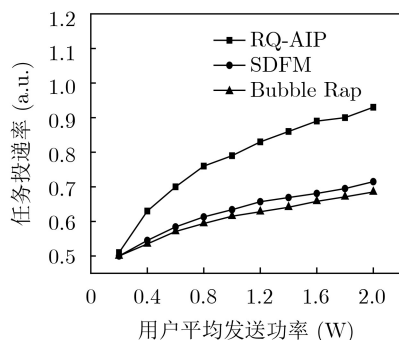


图7 用户不同发送功率下的任务投递率

- [8] SCHÄFER D, EDINGER J, ECKRICH J, *et al.* Hybrid task scheduling for mobile devices in edge and cloud environments[C]. 2018 IEEE International Conference on Pervasive Computing and Communications Workshops, Athens, Greece, 2018: 669–674. doi: [10.1109/PERCOMW.2018.8480201](https://doi.org/10.1109/PERCOMW.2018.8480201).
- [9] THAM C K and CHATTOPADHYAY R. A load balancing scheme for sensing and analytics on a mobile edge computing network[C]. The 18th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, Macau, China, 2017: 1–9. doi: [10.1109/WoWMoM.2017.7974307](https://doi.org/10.1109/WoWMoM.2017.7974307).
- [10] CHEN Lixing, ZHOU Sheng, and XU Jie. Computation peer offloading for energy-constrained mobile edge computing in small-cell networks[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(4): 1619–1632. doi: [10.1109/TNET.2018.2841758](https://doi.org/10.1109/TNET.2018.2841758).
- [11] YOUNES H, BOUATTANE O, YOUSSEFI M, *et al.* New load balancing framework based on mobile AGENT and ant-colony optimization technique[C]. 2017 Intelligent Systems and Computer Vision, Fez, Morocco, 2017: 1–6.
- [12] MASOOD A, MUNIR E U, RAFIQUE M M, *et al.* HETS: Heterogeneous edge and task scheduling algorithm for heterogeneous computing systems[C]. The 17th IEEE International Conference on High Performance Computing and Communications, New York, USA, 2015: 1865–1870. doi: [10.1109/HPCC-CSS-ICESS.2015.295](https://doi.org/10.1109/HPCC-CSS-ICESS.2015.295).
- [13] TIAN Rui, JIAO Zhenzhen, BIAN Guiyun, *et al.* A social-based data forwarding mechanism for V2V communication in VANETs[C]. The 10th International Conference on Communications and Networking in China, Shanghai, China, 2015: 595–599. doi: [10.1109/CHINACOM.2015.7498007](https://doi.org/10.1109/CHINACOM.2015.7498007).
- [14] Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020[EB/OL]. https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf, 2016.
- [15] ZHAO Pengtao, TIAN Hui, QIN Cheng, *et al.* Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing[J]. *IEEE Access*, 2017, 5: 11255–11268. doi: [10.1109/ACCESS.2017.2710056](https://doi.org/10.1109/ACCESS.2017.2710056).
- [16] PAN Hui, CROWCROFT J, and YONEKI E. BUBBLE Rap: Social-based forwarding in delay-tolerant networks[J]. *IEEE Transactions on Mobile Computing*, 2011, 10(11): 1576–1589. doi: [10.1109/TMC.2010.246](https://doi.org/10.1109/TMC.2010.246).
- 王汝言: 男, 1969年生, 教授, 博士, 研究方向为泛在网络、多媒体信息处理等.
- 聂 轩: 男, 1995年生, 硕士生, 研究方向为移动边缘计算.
- 吴大鹏: 男, 1979年生, 教授, 博士, 研究方向为泛在无线网络、无线网络服务质量控制等.
- 李红霞: 女, 1969年生, 高级工程师, 研究方向为光无线融合网络.