

# 边缘计算中基于综合信任评价的任务卸载策略

熊小峰<sup>1</sup>, 黄淳岚<sup>1,2</sup>, 乐光学<sup>1,2</sup>, 戴亚盛<sup>2</sup>, 杨晓慧<sup>2</sup>, 杨忠明<sup>1,2</sup>

(1. 江西理工大学理学院, 江西赣州 341000; 2. 嘉兴学院信息科学与工程学院, 浙江嘉兴 314001)

**摘要:** 针对边缘计算中存在虚假欺骗、搭便车等不良行为导致资源浪费和卸载服务低效的问题, 提出一种基于综合信任评价的任务卸载(Task Offloading scheme based on Comprehensive Trust Evaluation, TOCTE)策略. 建立节点信任遗忘函数, 以任务真实性和节点卸载协作行为特征为约束, 构建节点间卸载请求和协作信任评价模型, 识别并过滤网络中的虚假任务和不良协作节点. 设计搜索发现概率优化粒子群算法, 规避节点重复利用引起的节点过载, 提高边缘计算任务卸载效率和资源有效利用. 仿真实验表明, TOCTE策略模型真实任务执行成功率达到59.22%, 资源损耗率最低达到6.35%, 能有效抑制和过滤虚假任务, 降低资源损耗.

**关键词:** 边缘计算; 可信卸载; 任务过滤; 综合评价; 粒子群算法

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112(2022)09-2134-12

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20210296

## Task Offloading Scheme Based on Comprehensive Trust Evaluation in Edge Computing

XIONG Xiao-feng<sup>1</sup>, HUANG Chun-lan<sup>1,2</sup>, YUE Guang-xue<sup>1,2</sup>, DAI Ya-sheng<sup>2</sup>, YANG Xiao-hui<sup>2</sup>,  
YANG Zhong-ming<sup>1,2</sup>

(1. College of Science, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China;

2. College of Mathematical Information and Engineering, Jiaxing University, Jiaxing, Zhejiang 314001, China)

**Abstract:** In edge computing, waste of resources and inefficient offloading services are key issues caused by bad behaviors such as false deception and free-riding. To address the selfishness key issues, a task offloading scheme based on comprehensive trust evaluation (TOCTE) is proposed. By establishing nodes trust forgetting function, and taking task authenticity and nodes offloading cooperative behavior characteristics as constraints, the offloading request and cooperative trust evaluation models between nodes are constructed to identify and filter false tasks and bad cooperative nodes in the network. Particle swarm algorithm based on search discovery probability is designed to avoid the node overload caused by the repeatedly utilization of nodes, improve task offloading efficiency and resource utilization of edge computing. The simulation results show that the real task execution success rate and the lowest resource consumption rate of TOCTE are 59.22% and 6.35%. TOCTE model can effectively suppress and filter false tasks, and reduce resource consumption.

**Key words:** edge computing; trusted offloading; task filtering; comprehensive evaluation; particle swarm algorithm

### 1 引言

边缘计算将计算、存储等资源下沉到网络边缘侧, 以“就地就近”的服务模式, 实现低延迟、低能耗、隐私保护等, 提高了网络 QoS (Quality of Service) 和综合效用. 任务卸载是边缘计算的关键技术之一, 主要解决边缘计算中任务是否卸载、任务划分和资源分配这三大问题<sup>[1,2]</sup>, 实现网络资源融合与共享.

由于搭便车、自私、理性、恶意等不良节点的普遍存在, 极易引起 On-off 攻击等不良行为<sup>[3]</sup>, 抢占和消耗边缘计算资源, 部分节点因资源被过度消耗“致死”, 导致卸载任务堆积、执行失败, 边缘计算网络 QoS 下降.

为抑制任务卸载过程中节点虚假欺骗、服务不可靠等不良行为, 文献[4]提出了基于社交信任网络的服务器分配策略, 提高游戏视频流服务; 文献[5]以时间可信、行为可信、资源可信为约束, 构建服务质量可信

模型,解决大规模部署边缘节点引起的低效率选择问题;文献[6]基于用户个性化的多维需求,评估盟员卸载优先级,实现协同服务异常时的调度卸载;文献[7]结合用户的心理结构,提出一种面向信任的合作伙伴选择机制;文献[8]构建基于综合信任保障的边缘计算模型,为任务卸载服务请求者和资源提供者进行身份、行为和服务能力定量评价,提出一种移动资源感知调度算法,以计算、存储和通信资源为约束,寻找节点进行协同服务。

依据文献[8],针对任务卸载服务请求者和资源提供者可信问题,提出一种基于综合信任评价的任务卸载策略。基于节点行为的时间关联属性,建立节点信任遗忘函数和节点行为奖惩算子,构建节点信任评价机制;分析节点任务卸载真实特征属性,建立卸载请求行为评价模型;分析节点资源余量及卸载交互行为,建立卸载协作综合评价模型,筛选并构建可信的卸载协作候选节点集,设计基于搜索发现概率的粒子群(Particle Swarm Based On Search Discovery Probability, PS-SDP)算法求解最佳协作节点,实现计算任务的可靠卸载服务。通过仿真实验验证了模型可行性,能够有效抑制边缘计算中虚假任务卸载造成的无效时间和资源开销,提高资源有效利用。

## 2 基于综合信任评价的任务卸载策略

设边缘计算由边缘云服务器、超级节点(Super Node, SN)和边缘节点(Edge Node, EN)构成,以SN为中心形成边缘计算簇,SN规模为 $X^{sn} = \{1, 2, \dots, m\}$ ,EN规模为 $X_i^{en} = \{1, 2, \dots, n_i\}$ ,如图1所示。以SN节点为桥梁实现边缘计算簇间资源聚合与共享,提供任务卸载协同服务;构建卸载请求、协作行为评价模型,过滤虚假卸载任务请求,遴选最佳协作节点,以一对多模式卸载计算任务,获得高效可靠的卸载协同服务,其可信任务卸载协同服务框架如图2所示。

任务卸载是一个具有时间关联性动态决策过程,边缘计算节点信任特征具有时间衰减属性。基于Ebbinghaus<sup>[9]</sup>人类大脑记忆遗忘理论,构建边缘计算节点

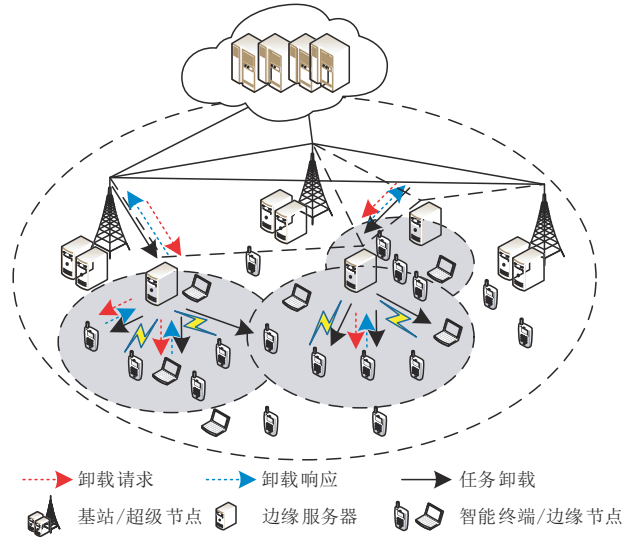


图1 边缘协同服务模型

信任记忆保持量关于时间 $t$ 的衰减函数 $f(t)$ :

$$f(t) = \frac{1}{\alpha \cdot \ln(\beta \cdot t + 1) + 1}, \alpha > 0, \beta > 0 \quad (1)$$

其中,实验中取值 $\alpha = 0.3574, \beta = 531.7$ ,记忆衰减曲线如图3所示。

将每个观察期时长分成 $t_k$ 个时间片,基于式(1),构建边缘计算节点历史信任遗忘函数:

$$\begin{cases} \lambda = f(\sigma \cdot t_k) \\ T(s + \Delta s) = \lambda^{\Delta s} \cdot T(s), \Delta s \geq 0 \end{cases} \quad (2)$$

其中, $T(s)$ 表示第 $s$ 观察期的信任度, $\Delta s$ 表示观察周期间隔数, $\sigma$ 表示遗忘因子, $\lambda$ 表示历史信任的保持率。

### 2.1 卸载请求行为评价模型

边缘计算簇间的卸载请求行为评价模型如图4所示。SN节点根据任务真实性计算卸载源节点的卸载请求信任度,判定合作关系,过滤虚假任务,执行卸载服务。

设第 $s$ 观察期间边缘计算簇 $i$ (以SN节点 $i$ 为中心)中,存在节点 $j$ 卸载的真实任务和虚假任务数分别为 $n_{ij}^{true}(s), n_{ij}^{false}(s)$ 。根据簇内任务执行情况,计算SN节点 $j$

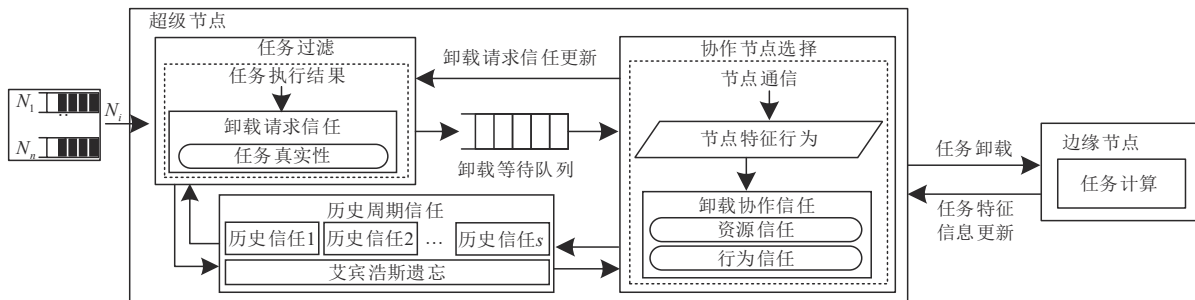


图2 边缘协同服务可信卸载框架

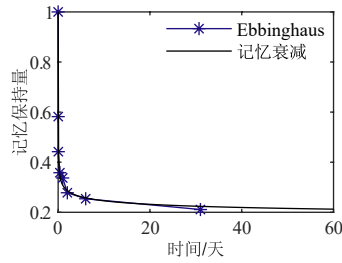


图3 记忆保持衰减曲线示意图

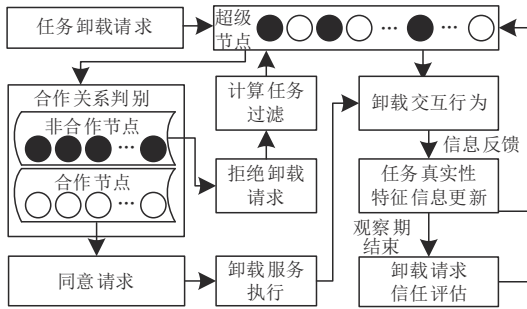


图4 卸载请求行为评价模型

的任务真实率  $p_{ij}^{\text{true}}(s)$  为:

$$p_{ij}^{\text{true}}(s) = \frac{n_{ij}^{\text{true}}(s) + \eta_1}{n_{ij}^{\text{true}}(s) + n_{ij}^{\text{false}}(s) + \eta_2}, \quad (3)$$

$$\text{s.t. } 1 \leq \eta_1 < \eta_2, \eta_2 \geq 2, s \geq 1$$

如式(3),  $\eta_1$  和  $\eta_2$  分别取值为 1 和 2, 当簇内未执行节点  $j$  任务时, 赋予节点中性值 0.5.

为抑制非合作节点制造、卸载虚假任务等不良行为, 提高合作节点信任置信度, 构建节点卸载请求行为奖惩算子  $\varphi_1^{[10]}$ :

$$\varphi_1 = \frac{n_{ij}^{\text{false}}(s)}{(n_{ij}^{\text{true}}(s) + n_{ij}^{\text{false}}(s))^2 + n_{ij}^{\text{true}}(s) + n_{ij}^{\text{false}}(s) + \eta}, \quad \eta \geq 1 \quad (4)$$

其中,  $\eta$  取值为 1.

引入历史信任遗忘函数, 计算经  $s$  个观察周期后, 节点  $j$  的卸载请求信任  $T_{ij}^{\text{req}}(s)^{[11]}$ :

$$\begin{cases} T_{ij}^{\text{true}}(s) = (1 + \varphi_1) \cdot p_{ij}^{\text{true}}(s) - \varphi_1 \cdot (1 - p_{ij}^{\text{true}}(s)) \\ T_{ij}^{\text{req}}(s) = \lambda \cdot T_{ij}^{\text{req}}(s-1) + (1 - \lambda) \cdot T_{ij}^{\text{true}}(s) \end{cases} \quad (5)$$

其中,  $T_{ij}^{\text{true}}(s)$  表示第  $s$  观察期间节点  $i$  对节点  $j$  的真实任务信任度.

## 2.2 卸载协作综合评价模型

依据行为信任, 将 EN 节点分为可靠节点、激励节点和不良节点三类<sup>[7]</sup>: (1) 可靠节点: 卸载协作行为可信, 成功率较高的节点; (2) 激励节点: 初期卸载协作服务能力较差, 通过激励调动服务积极性的节点; (3) 不良节点: 长期以消极服务态度提供协作服务的节点.

以节点资源信任和行为信任, 构建卸载协作综合评价模型, 如图 5 所示. SN 节点结合当前剩余资源和行为信任计算节点卸载协作信任, 判定节点信任关系, 构建可信的卸载协作候选节点集, 甄选最佳协作节点执行卸载服务. 若节点行为信任低于中性值  $\gamma$ , 则生成一组测试任务集由该节点执行, 实施激励, 再次更新其行为信任.

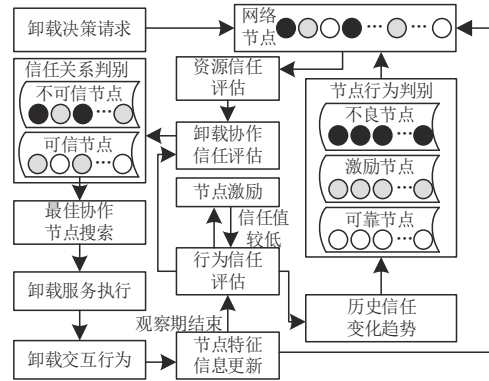


图5 卸载协作综合评价模型

### 2.2.1 资源信任评价模型

边缘计算中资源的合理利用有利于边缘计算任务卸载协同服务的良性运行. 设节点最大负载率为  $\rho$ , 计算节点  $i$  执行任务  $r$  的资源信任度  $T_i^{\text{res}}$ :

$$T_i^{\text{res}} = \frac{(c_i - c_r) \cdot (m_i - m_r) + 1}{C_i \cdot M_i + 1}$$

$$\text{s.t. } \begin{cases} \frac{c_i - c_r}{C_i} > 1 - \rho \\ \frac{m_i - m_r}{M_i} > 1 - \rho \end{cases} \quad (6)$$

其中,  $C_i$  和  $M_i$  表示节点  $i$  可提供的计算和存储资源,  $c_i$  和  $m_i$  表示节点  $i$  的当前剩余计算和存储资源,  $c_r$  和  $m_r$  表示任务  $r$  执行所需的计算和存储资源.  $T_i^{\text{res}}$  值大于  $(1 - \rho)^2$  时, 节点  $i$  可供资源消耗殆尽, 出现过载现象.

为避免拥塞、抖动等影响网络稳定性, 任务的卸载与执行在同一观察期内完成. 设节点  $i$  在线时长为  $t_i^{\text{on}}$ , 计算执行任务  $r$  后的剩余在线时长  $t_i^{\text{rem}}$ :

$$t_i^{\text{rem}} = \frac{(c_i - c_r) \cdot t_i^{\text{on}}}{C_i} \quad (7)$$

模型约束条件为:

$$\text{s.t. } \begin{cases} t_i^{\text{spe}} + t_i^{\text{rem}} + \frac{c_r}{a_i^{\text{exe}}} > t_{\text{cyc}} \\ t_i^{\text{spe}} + \frac{c_r}{a_i^{\text{exe}}} < t_{\text{cyc}} \end{cases} \quad (8)$$

其中,  $t_{\text{cyc}}$  表示观察期时长,  $a_i^{\text{exe}}$  表示节点  $i$  计算能力,  $t_i^{\text{spe}}$

表示观察期内执行任务  $r$  前的已耗时间。

### 2.2.2 行为信任评价模型

边缘计算簇内,以节点的任务执行效率和成功率作为约束,构建节点行为信任模型。

EN 节点  $j$  的任务执行效率  $T_j^{\text{ab}}$  计算公式如下:

$$T_j^{\text{ab}} = e^{-\frac{1}{a_j^{\text{exe}}}} \quad (9)$$

已知第  $s$  观察期内,簇内节点  $j$  接受的任务数为  $n_{ij}^{\text{acc}}(s)$ ,接受且成功执行的任务数为  $n_{ij}^{\text{suc}}(s)$ ,任务执行成功率  $p_{ij}^{\text{suc}}(s)$  计算公式如下:

$$p_{ij}^{\text{suc}}(s) = \frac{n_{ij}^{\text{suc}}(s) + \eta_1}{n_{ij}^{\text{acc}}(s) + \eta_2}, 1 \leq \eta_1 < \eta_2, \eta_2 \geq 2 \quad (10)$$

如式(10),当节点在观察期内未接受任务时,赋予节点中性值 0.5。

为抑制簇内 EN 节点不良行为,激励节点诚信服务,设置节点卸载协作行为奖惩算子  $\varphi_2$ <sup>[10]</sup>:

$$\varphi_2 = \frac{n_{ij}^{\text{acc}}(s) - n_{ij}^{\text{suc}}(s)}{(n_{ij}^{\text{acc}}(s))^2 + n_{ij}^{\text{acc}}(s) + \eta}, \eta \geq 1 \quad (11)$$

计算第  $s$  观察期内节点  $j$  的任务执行信任  $T_{ij}^{\text{suc}}(s)$ <sup>[11]</sup>:

$$T_{ij}^{\text{suc}}(s) = \begin{cases} \omega \cdot \left( (1 + \varphi_2) \cdot p_{ij}^{\text{suc}}(s) - \varphi_2 \cdot (1 - p_{ij}^{\text{suc}}(s)) \right) \\ + (1 - \omega) \cdot \frac{n_{ij}^{\text{suc}}(s)}{n_{ij}^{\text{off}}(s)}, n_{ij}^{\text{off}}(s) > 0 \\ (1 + \varphi_2) \cdot p_{ij}^{\text{suc}}(s) - \varphi_2 \cdot (1 - p_{ij}^{\text{suc}}(s)), \\ n_{ij}^{\text{off}}(s) = 0 \end{cases} \quad (12)$$

其中,  $\omega$  表示任务执行成功率所占权重,  $n_{ij}^{\text{off}}(s)$  表示第  $s$  观察期内节点  $i$  向节点  $j$  卸载的任务数。

结合节点任务执行效率和执行信任,计算第  $s$  观察期内节点  $j$  的行为信任  $T_{ij}^{\text{exe-r}}(s)$ :

$$T_{ij}^{\text{exe-r}}(s) = T_{ij}^{\text{suc}}(s) \cdot T_j^{\text{ab}} \quad (13)$$

引入历史信任遗忘函数,计算经过  $s$  个观察周期后,节点  $i$  对节点  $j$  的行为信任  $T_{ij}^{\text{exe}}(s)$ :

$$T_{ij}^{\text{exe}}(s) = \lambda \cdot T_{ij}^{\text{exe}}(s-1) + (1 - \lambda) \cdot T_{ij}^{\text{exe-r}}(s), s \geq 1 \quad (14)$$

### 2.2.3 综合信任决策模型

综合资源信任和行为信任,构建卸载协作信任  $T_{ij}^{\text{coop}}$ :

$$T_{ij}^{\text{coop}} = T_j^{\text{res}} \cdot T_{ij}^{\text{exe}}(s) \quad (15)$$

由式(15)求得 EN 节点卸载协作信任,筛选得到卸载协作候选节点集  $X_i^{\text{co}} = \{1, 2, \dots, n_i\}$ . 将卸载协作节点择优问题转化为卸载协作信任最大化的目标优化问题,以节点集  $X_i^{\text{co}}$  为解空间,设计 PS-SDP 算法求解。

由式(8)、式(15)得到 SN 节点  $i$  中任务  $r$  的卸载协

作节点寻优目标及约束条件如下:

$$\begin{aligned} & \max \left\{ T_{ix_i}^{\text{coop}} \mid x_i \in X_i^{\text{co}} \right\} \\ & \text{s.t.} \begin{cases} \frac{c_{x_i} - c_r}{C_{x_i}} > 1 - \rho \\ \frac{m_{x_i} - m_r}{M_{x_i}} > 1 - \rho \\ t_{x_i}^{\text{spe}} + t_{x_i}^{\text{rem}} + \frac{c_r}{a_{x_i}^{\text{exe}}} > t_{\text{cyc}} \\ t_{x_i}^{\text{spe}} + \frac{c_r}{a_{x_i}^{\text{exe}}} < t_{\text{cyc}} \\ i \in \{1, 2, \dots, m\}, X_i^{\text{co}} = \{1, 2, \dots, n_i\} \end{cases} \end{aligned} \quad (16)$$

PS-SDP 算法利用粒子群优化(Particle Swarm Optimization, PSO)算法<sup>[12]</sup>求解最佳协作节点。每一个粒子代表解空间中协作节点寻优的一个的潜在解,学习自身和其他成员经验以适应自身搜索模式。飞行速度受个体最佳信任节点  $x_l^{\text{pb}}$  和群体最佳信任节点  $x^{\text{gb}}$  影响,粒子  $l$  在搜索空间的飞行速度  $v_l$  和候选节点  $x_l \in X_i^{\text{co}}$  更新公式如下:

$$\begin{cases} v_l(\tau) = w \cdot v_l(\tau-1) + c_1 \cdot \zeta \cdot (x_l^{\text{pb}}(\tau-1) - x_l(\tau-1)) \\ + c_2 \cdot \zeta \cdot (x^{\text{gb}}(\tau-1) - x_l(\tau-1)) \\ x_l(\tau) = x_l(\tau-1) + v_l(\tau) \\ w = w_{\text{max}} - \frac{(w_{\text{max}} - w_{\text{min}}) \cdot \tau}{G} \end{cases} \quad (17)$$

其中,  $\zeta$  在 (0, 1) 范围内随机取值,  $v_l(\tau)$  和  $x_l(\tau)$  分别表示粒子  $l$  经  $\tau$  次迭代更新后的速度和候选节点,  $x_l^{\text{pb}}(\tau)$  和  $x^{\text{gb}}(\tau)$  分别表示  $\tau$  次迭代后粒子  $l$  和整个粒子群的最佳信任节点,  $c_1$  和  $c_2$  表示粒子自身和群体的学习因子,  $w$  表示惯性因子,  $w_{\text{max}}$  和  $w_{\text{min}}$  分别表示惯性因子最大和最小值,  $G$  为算法最大迭代次数。

为抑制算法寻优陷入局部最优,融合布谷鸟搜索算法<sup>[13]</sup>(Cuckoo Search, CS),引入搜索发现概率  $p \in [0, 1]$ ,实现解域概率取值,以规避局部最优,使求解结果趋于全局最优,获取任务卸载的最佳协作节点,其算法流程如图 6 所示。设置任务最大拒绝次数,避免任务多次卸载被拒产生过多延迟开销,利用 PS-SDP 算法寻找最佳协作节点获得卸载服务。

### 2.3 任务卸载策略建模

任务卸载协同服务工作原理:边缘计算簇中,SN 节点收到任务卸载请求时,启动任务过滤机制,依据任务真实特征信息,计算任务卸载源节点的卸载请求信任,识别节点合作性,过滤非合作节点的虚假任务;启动卸载服务决策机制,依据簇内 EN 节点卸载协作信任,构建可信的卸载协作候选节点集,应用 PS-SDP 算法遴选最佳协作节点,并发起卸载请求,完成卸载协同服务。

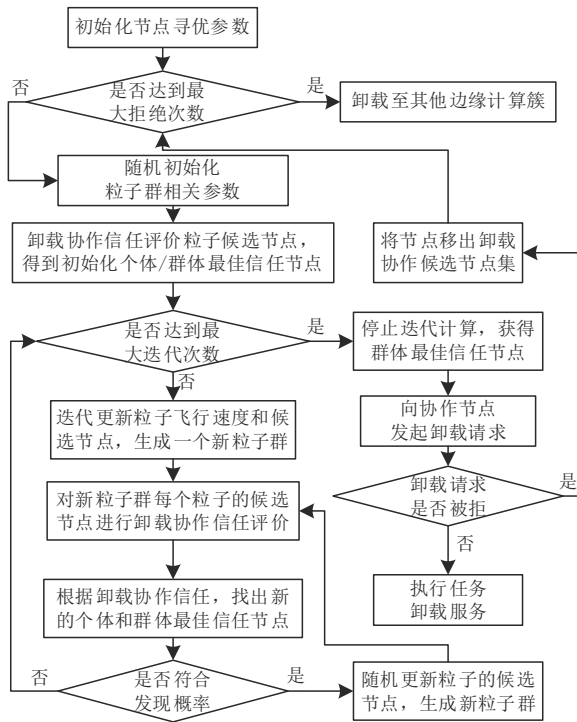


图6 簇内协作节点寻优流程

在基于卸载请求行为评价模型的任务过滤机制和基于卸载协作综合评价模型的卸载服务决策机制中,SN节点的状态变化及服务流程如下。

(1) 根据式(5),将SN节点的任务过滤形式化描述为  $D_{req} = \{Q_1, \Gamma_1, \delta_1, q_{11}, F_1\}$  的状态图,如图7所示,其中,  $Q_1 = \{q_{11}, q_{12}, q_{13}, q_{14}, q_{15}, q_{16}, q_{17}, q_{18}, q_{19}\}$  为有穷状态集,任务状态描述如表1所示。 $\Gamma_1$ 为状态转移条件产生式; $\delta_1$ 为状态转移函数,  $Q_1 \times \Gamma_1 = Q_1$ ;  $q_{11} \in Q_1$ ,为起始状态; $F_1 \subseteq Q_1$ ,为结束状态集,  $F_1 = \{q_{19}\}$ 。

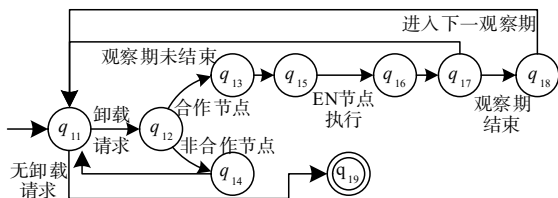


图7 SN节点任务过滤状态图

表1 节点任务过滤状态说明

| 任务状态     | 状态描述:    | 任务状态     | 状态描述:      |
|----------|----------|----------|------------|
| $q_{11}$ | 服务等待状态   | $q_{16}$ | 簇内信息接收状态   |
| $q_{12}$ | 节点关系识别状态 | $q_{17}$ | 任务特征信息更新状态 |
| $q_{13}$ | 卸载请求接受状态 | $q_{18}$ | 卸载请求信任更新状态 |
| $q_{14}$ | 卸载请求拒绝状态 | $q_{19}$ | 结束状态       |
| $q_{15}$ | 簇内任务卸载状态 |          |            |

流程1 任务过滤

Step1: 设置SN节点*i*卸载请求信任阈值  $T_{min}^{req}$ .

Step2: 等待并收到节点*j*的任务卸载请求,判定SN节点*i*与任务卸载源节点*j*的合作关系.若  $T_{ij}^{req} < T_{min}^{req} (i \neq j, i, j \leq m)$ ,则节点*j*为非合作节点,节点*i*拒绝节点*j*的任务卸载请求,转Step4.否则执行任务卸载服务.

Step3: 启动卸载服务决策机制,遴选最佳协作节点.协作节点执行任务并反馈任务真实性,SN节点依据反馈结果更新任务真实特征信息.

Step4: 卸载请求信任更新.若观察期结束,根据任务真实特征信息,结合历史信任关系,更新各SN节点的卸载请求信任.

Step5: 测试是否结束? 是,结束;否则,转Step2.

(2) 根据式(15)、式(16),将SN节点的卸载服务决策形式化描述为  $D_{coop} = \{Q_2, \Gamma_2, \delta_2, q_{21}, F_2\}$  的状态图,如图8所示.其中,  $Q_2 = \{q_{21}, q_{22}, q_{23}, q_{24}, q_{25}, q_{26}, q_{27}, q_{28}, q_{29}\}$  为有穷状态集,任务状态描述如表2所示。 $\Gamma_2$ 为状态转移条件产生式; $\delta_2$ 为状态转移函数,  $Q_2 \times \Gamma_2 = Q_2$ ;  $q_{21} \in Q_2$ ,为起始状态; $F_2 \subseteq Q_2$ ,为结束状态集,  $F_2 = \{q_{29}\}$ 。

表2 节点卸载服务决策状态说明

| 任务状态     | 状态描述:      | 任务状态     | 状态描述:      |
|----------|------------|----------|------------|
| $q_{21}$ | 服务等待状态     | $q_{26}$ | 任务卸载状态     |
| $q_{22}$ | 卸载协作信任更新状态 | $q_{27}$ | 节点特征信息更新状态 |
| $q_{23}$ | 信任关系判别状态   | $q_{28}$ | 行为信任更新状态   |
| $q_{24}$ | 协作节点选择状态   | $q_{29}$ | 结束状态       |
| $q_{25}$ | 卸载协商状态     |          |            |

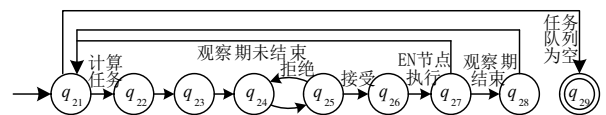


图8 SN节点卸载协作状态图

流程2 卸载服务决策

Step1: 设置SN节点*i*卸载协作信任阈值  $T_{min}^{coop}$  和任务最大拒绝次数  $n_{max}^{rej}$ .

Step2: 根据资源特征信息,更新EN节点资源信任,联合行为信任计算卸载协作信任  $T_{ij}^{coop}(s)$ ;判定节点信任关系,若  $T_{ij}^{coop}(s) \geq T_{min}^{coop} (i \leq m, j \leq n)$ ,则EN节点*j*协同服务可信,划入卸载协作候选节点集.

Step3: 若任务拒绝次数  $n_{ij}^{rej} \leq n_{max}^{rej}$ ,运用PS-SDP算法,在卸载协作候选节点集中寻找最佳协作节点并发起卸载请求;否则,将任务卸载至其他边缘计算簇.

Step4: 与协作节点卸载协商并更新节点行为特征信息,若请求接受,完成任务卸载;否则,转Step3.

Step5:若观察期结束,根据行为特征信息和历史信任关系,更新簇内EN节点行为信任.

Step6:测试是否结束?是,结束;否则,转Step2.

综合任务过滤和卸载服务决策机制,边缘计算簇协同服务过程如图9所示.

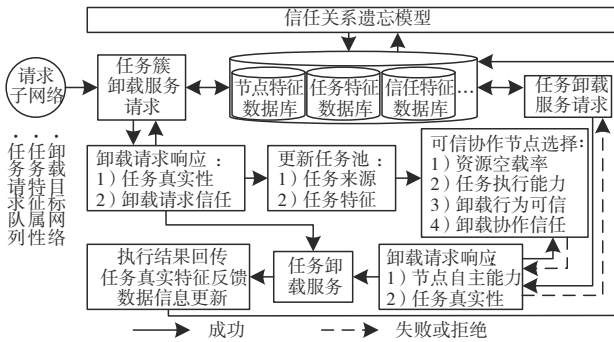


图9 TOCTE卸载协同服务

**流程3 任务卸载协同服务过程**

Step1:SN节点等待并收到任务卸载请求,创建任务特征数据库.

Step2:根据任务真实特征数据,更新任务卸载源节点的合作关系数据,识别、过滤非合作节点的虚假任务.

Step3:根据任务和节点特征数据,计算簇内EN节点资源信任,结合行为信任数据,更新节点卸载协作信任数据库,构建卸载协作候选节点集.

Step4:运用PS-SDP算法,以卸载协作信任最大化为目标,遴选最佳协作节点并发起卸载请求.

Step5:簇内EN节点执行卸载协同服务并返回服务结果,将任务真实特征信息反馈给SN节点,更新任务和节点特征数据库.

Step6:根据任务和节点特征数据,更新SN节点的卸载请求信任、簇内EN节点的行为信任数据.

将任务的一次卸载协同服务过程形式化描述为  $D_{TOCTE} = (Q, \Gamma, \delta, q_1, F)$  状态图,如图10所示,其中,  $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}$  为有穷状态集,任务状态描述如表3所示.  $\Gamma$  为状态转移条件产生式;  $\delta$  为状态转移函数,  $Q \times \Gamma \rightarrow Q; q_1 \in Q$ , 为起始状态;  $F \subseteq Q$ , 为结

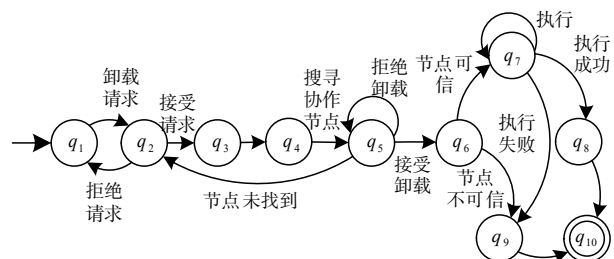


图10 任务卸载执行状态图

束状态集,  $F = \{q_{10}\}$ .

表3 任务状态说明

| 任务状态  | 状态描述:    | 任务状态     | 状态描述:    |
|-------|----------|----------|----------|
| $q_1$ | 簇间卸载抉择状态 | $q_6$    | 簇内卸载执行状态 |
| $q_2$ | 簇间卸载等待状态 | $q_7$    | 任务执行状态   |
| $q_3$ | 簇间卸载执行状态 | $q_8$    | 任务完成状态   |
| $q_4$ | 簇内卸载抉择状态 | $q_9$    | 任务失败状态   |
| $q_5$ | 簇内卸载等待状态 | $q_{10}$ | 结束状态     |

**3 实验分析**

**3.1 实验设置**

以云服务器为根节点,基站或边缘服务器为SN节点,智能终端设备为EN节点,SN节点与根节点构建云边协同骨干网络,SN节点与EN节点构建边缘计算网络.作如下假设:

(1) 在边缘计算网络中,SN节点聚合EN节点构建边缘计算簇,每个簇包含1个SN节点,簇规模一致,仿真实验网络环境设置如表4所示.

表4 网络环境设置

| 测试组编号 | 网络规模  | 云服务器 | SN节点数 | 边缘计算簇数 | EN节点数 | 簇规模 |
|-------|-------|------|-------|--------|-------|-----|
| 1     | 5051  | 1    | 50    | 50     | 100   | 101 |
| 2     | 7071  | 1    | 70    | 70     | 100   | 101 |
| 3     | 10101 | 1    | 100   | 100    | 100   | 101 |
| 4     | 12121 | 1    | 120   | 120    | 100   | 101 |
| 5     | 15151 | 1    | 150   | 150    | 100   | 101 |
| 6     | 20201 | 1    | 200   | 200    | 100   | 101 |

(2) 边缘计算中存在30%的SN节点为非合作节点;单位时间片内卸载请求任务数遵循泊松分布;同一时间片相邻任务间的服务等待和节点卸载协商产生的时间开销忽略不计.

(3) 边缘计算簇内,可靠、激励和不良三类EN节点数目比例为6:2:2;任务可在EN节点堆积,但均在观察期内完成,不滞留至下一观察期.

(4) 模型中随机参数取值服从正态分布,具体参数设置见表5.

**3.2 评价指标**

为验证卸载策略的性能效用,分别对簇间卸载、簇内卸载的信任评价模型以及策略的卸载综合效用的性能指标加以描述,如表6所示.

**3.3 实验分析**

为检验TOCTE策略的卸载协同服务优化效果,分别从卸载请求行为评价模型性能、卸载协作综合评价模型性能以及卸载综合效用,分析TOCTE策略的虚假

表5 参数设置

| 参数                    | 数值         |
|-----------------------|------------|
| 任务计算量 $c_i$           | 1~4 MB     |
| 任务所需内存 $m_i$          | 1~4 MB     |
| EN 工作计算能力 $a_i^{exe}$ | 4~8 MB/s   |
| EN 在线时长 $t_i^{on}$    | 0.5~1 h    |
| EN 可执行任务量 $C_i$       | 300~500 MB |
| EN 可提供内存 $M_i$        | 100~500 MB |
| 观察周期 $t_{cyc}$        | 360 s      |
| 时间片数 $t_k$            | 360        |
| 最大负载率 $\rho$          | 0.8        |
| 任务执行成功率权重 $\omega$    | 0.8        |

任务过滤有效性和卸载服务可靠性.

### 3.3.1 卸载请求行为评价模型

为测试 TOCTE 策略中卸载请求行为评价模型的性能效用,根据非合作节点识别率和识别准确率、卸载请求拒绝率和拒绝准确率、资源有效利用率等指标,在测试组3的网络环境下,将 TOCTE 策略中的任务过滤机制与任务随机接受(random accept)策略<sup>[14]</sup>对比分析.

图11~图13分别描述了 TOCTE 策略的卸载请求行为评价模型的不同信任阈值,对于边缘计算簇的非合作节点识别、虚假任务过滤和资源利用的影响.从图中发现,随着 SN 节点信任阈值的增大,边缘计算中的非合作节点逐渐被识别出来,非合作节点识别率、卸载请求拒绝率和资源有效利用率提高;信任阈值小于0.4时,非合作节点识别和任务卸载请求拒绝准确率均为100%;大于0.4时,信任度较低的合作节点无法满足信任需求,被误判为非合作节点,准确率开始降低;卸载请求信任阈值为0.5时,任务过滤效果最佳,非合作节点识别率达到28%,识别准确率最低为95.83%,卸载请求拒绝率达到32.84%,拒绝准确率最低为88.36%,资源有效利用率最高达到81.52%.

由于 Random Accept 策略在 SN 节点接收到任务卸

载请求后,不考虑任务来源可靠性,随机接受该请求,不具有非合作节点识别能力.图14描述了最佳信任阈值下,TOCTE 策略与 Random Accept 策略对于资源有效利用率的影响.从图中发现,Random Accept 策略的资源有效利用率在水平线上下波动,TOCTE 策略资源有效利用率逐渐提高并最终趋于稳定,平均资源有效利用率为78.60%,较 Random Accept 策略提高9.48%.

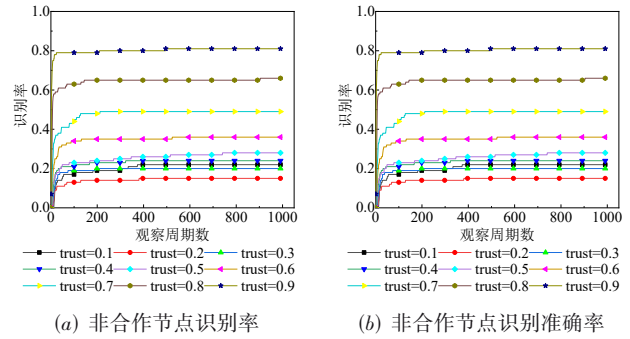


图11 不同信任阈值下非合作节点识别情况

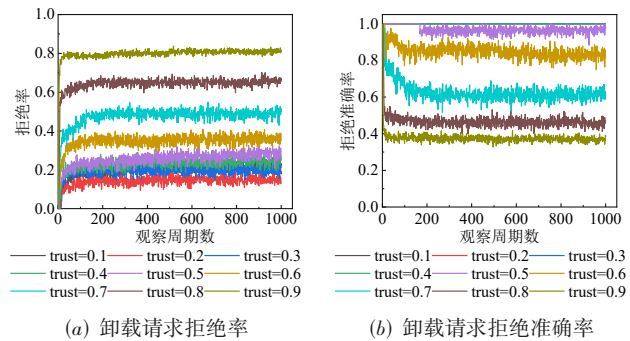


图12 不同信任阈值下虚假任务过滤情况

图15~17分别描述了不同网络规模下,TOCTE 策略的卸载请求行为评价模型对于边缘计算中非合作节点识别、虚假任务过滤和资源利用的影响.从图中发

表6 性能评价指标

| 分析对象       | 评价指标       | 描述  |
|------------|------------|---|
| 卸载请求行为评价模型 | 非合作节点识别率   | 观察周期内,SN 节点识别的非合作节点数与边缘计算中 SN 节点总数的比值;    |
|            | 非合作节点识别准确率 | 观察周期内,SN 节点识别的非合作节点中,身份无误的节点数与节点总数的比值;    |
|            | 卸载请求拒绝率    | 观察周期内,SN 节点拒绝的任务数与接收到的卸载请求任务数的比值;         |
|            | 卸载请求拒绝准确率  | 观察周期内,SN 节点拒绝的任务中,虚假任务数与总任务数的比值;          |
|            | 资源有效利用率    | 观察周期内,SN 节点接受的任务中,来自合作节点的任务数与总任务数的比值;     |
| 卸载协作综合评价模型 | 在线时长       | 观察周期内,簇内最短 EN 节点剩余在线时长;                   |
|            | 任务卸载率      | 观察周期内,边缘计算簇接受的任务中,卸载至其他簇的任务数与总任务数的比值;     |
|            | 任务执行失败率    | 观察周期内,边缘计算簇接受且执行的任务中,执行失败的任务数与总任务数的比值;    |
|            | 任务拒绝次数     | 观察周期内,边缘计算簇中各 EN 节点拒绝任务卸载请求次数的总和;         |
| 卸载综合效用     | 真实任务执行成功率  | 观察周期内,边缘计算簇接受的任务中,真实且被成功执行的任务数与总任务数的比值;   |
|            | 资源损耗率      | 观察周期内,边缘计算中虚假任务执行消耗的资源与各 EN 节点提供的资源总和的比值; |

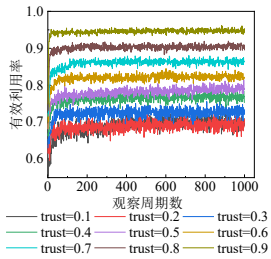


图 13 不同信任阈值下资源有效利用率

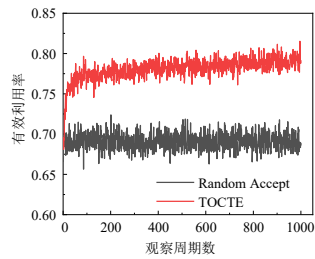


图 14 不同策略下资源有效利用率变化

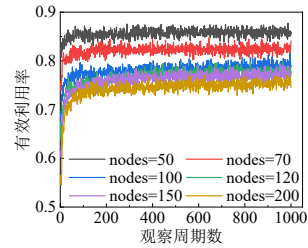


图 17 不同网络规模下资源有效利用率

现,边缘计算中边缘计算簇数目越多,非合作节点识别率和卸载请求拒绝率越高;边缘计算簇数小于 70 时,非合作节点识别准确率和任务卸载请求拒绝准确率为 100%,大于 70 时,随网络规模的增大,边缘计算环境愈加复杂,准确率和资源有效利用率降低.边缘计算簇数为 70 时,策略的任务过滤效果最佳,非合作节点识别率达到 22.86%,卸载请求拒绝率达到 22.93%,非合作节点识别和卸载请求拒绝准确率为 100%,资源有效利用率达到 82.34%.

综上所述,TOCTE 策略不仅能有效识别并拦截非合作节点的不良行为,保证任务来源的可靠性,同时提高了资源的有效利用.

的 EN 节点以信任(以任务执行成功率表示)划分节点集,如表 7 所示.设置每个节点集任务最大卸载次数,在满足任务需求的条件下,将任务优先卸载至优节点集中的节点,若任务卸载次数超过该节点集的最大卸载次数,则考虑良节点集中的节点,以此类推,直至找到协作节点执行卸载服务.若四类节点集均被搜索但仍未找到协作节点,则将任务卸载至其他簇.

表 7 节点集划分

| 节点集  | 优节点集  | 良节点集     | 一般节点集    | 差节点集  |
|------|-------|----------|----------|-------|
| 信任范围 | 0.9~1 | 0.75~0.9 | 0.6~0.75 | 0~0.6 |

根据边缘计算簇在线时长、任务执行失败率和卸载率等指标,将 TOCTE 策略与基于 Random Walk<sup>[14]</sup>、按需路由协议(Ad hoc On-Demand Distance Vector Routing, AODV)<sup>[15]</sup>的随机卸载策略和 TOO 策略对比分析,设置实验对照.

(1) Random Walk 策略:随机选择 EN 节点卸载,并由 EN 节点判断自身资源是否满足任务需求.

(2) AODV 策略:满足任务需求情况下,随机选择 EN 节点卸载.

(3) TOO 策略:满足任务需求情况下,趋优选择 EN 节点卸载.

图 18 描述了 TOCTE 策略的卸载协作综合评价模型的不同信任阈值,对于边缘计算簇在线时长和任务卸载率的影响.从图中发现,随信任阈值的增大,簇内可信协作节点和执行任务数减少,在线时长和任务卸载率逐渐提升;信任阈值为 0.6 时,信任阈值过高导致 SN 节点未找到协作节点,任务均向其他簇卸载,在线时长和卸载率保持 18299.83 s 和 100% 不变,该簇不再具备存在价

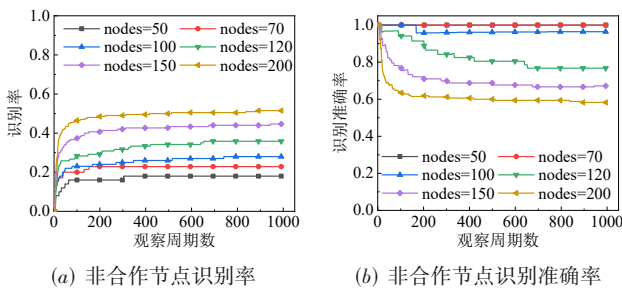


图 15 不同网络规模下非合作节点识别情况

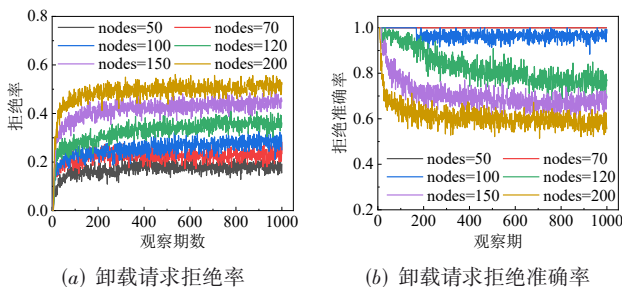


图 16 不同网络规模下虚假任务过滤情况

### 3.3.2 卸载协作综合评价模型

为测试 TOCTE 策略中卸载协作综合评价模型的性能效用,在测试组 3 的网络环境下构建任务趋优卸载(Task Optimal Offloading, TOO)策略,将边缘计算簇下

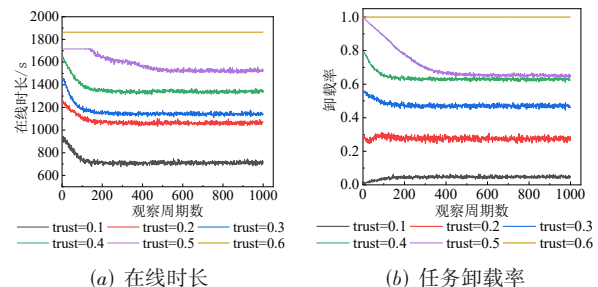


图 18 不同的信任阈值对比分析

值,同时增加了其他簇的计算负载和任务延迟开销。

为了延长边缘计算簇在线时长的同时,提供高效卸载服务,应合理设置卸载协作信任阈值。

图 19 和表 8 呈现了 Random Walk 策略、AODV 策略、TOO 策略和 TOCTE 策略,对于边缘计算簇在线时长、任务卸载率、任务执行失败率、任务拒绝次数和决策时长的影响。从图表中发现,与其他三种策略相比,

TOCTE 策略以决策时长和任务向簇外卸载为代价,有效延长了 147.49%、146.65% 和 156.76% 在线时长,降低了 33.84%、33.84% 和 29.26% 任务执行失败率;节点间卸载协商产生的时间开销由任务拒绝次数衡量,TOCTE 策略将平均拒绝次数减少至 831 次,与其他三种策略相比,分别减少了 35.48%、21.53% 和 52.92%,有效降低了任务多次卸载决策产生的时间开销。

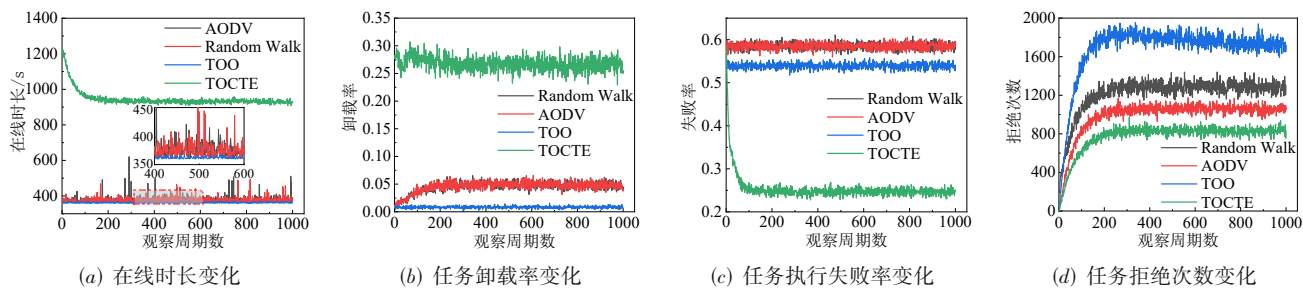


图 19 卸载策略对比

表 8 卸载策略效果对比

| 算法          | 平均决策时长/s | 平均拒绝次数 | 平均在线时长/s | 平均任务卸载率 | 平均任务执行失败率 |
|-------------|----------|--------|----------|---------|-----------|
| Random Walk | 0.5202   | 1288   | 3765.42  | 0.0485  | 0.5852    |
| AODV        | 0.5545   | 1059   | 3778.28  | 0.0487  | 0.5851    |
| TOO         | 0.6007   | 1765   | 3629.58  | 0.0083  | 0.5394    |
| TOCTE       | 1.9344   | 831    | 9319.17  | 0.2659  | 0.2468    |

为检验 TOCTE 策略下,不同协作节点寻优算法对卸载决策的优化效果,将 PS-SDP 算法与 PSO 算法<sup>[12]</sup>、CS 算法<sup>[13]</sup>和基于模拟退火 Metropolis 准则的粒子群 (Particle Swarm based on Simulated Annealing Metropolis Criterion, PS-SA) 算法<sup>[12,16]</sup>进行对比分析。设置实验对照组。

(1) PSO 算法:解空间以粒子群的形式存在,协作和信息共享的方式寻找最佳协作节点。

(2) CS 算法:解空间以寄主巢群的形式存在,Lévy flights 搜索方式寻找最佳协作节点。

(3) PS-SA 算法:解空间以粒子群的形式存在,协作和信息共享的方式寻找最佳协作节点,并利用模拟退火的 Metropolis 准则确定粒子候选节点。

图 20 和表 9 呈现了不同协作节点寻优算法,对于激励和可靠节点行为信任变化、任务执行失败率变化、卸载决策时长的影响。从图表中发现,相较于其他三种寻优算法,PS-SDP 算法在激励节点和可靠节点行为信任变化上收敛速度更快、信任度更高,稳定状态下平均信任度分别达到 0.766 和 0.693;从时间复杂度分析,PS-SDP 算法以搜索发现概率接受新解,增加一定的时间复杂度,但仅次于 PSO 算法;从任务执行失败率分析,PS-SDP 算法平均执行失败率达到 23.98%,与其他算法比,分别降低了 0.99%、0.62%、0.27%,任务卸载服务质量较优。

对于卸载可靠性要求较高的计算任务,以 PS-SDP 算法寻找协作节点,卸载服务更为可靠。

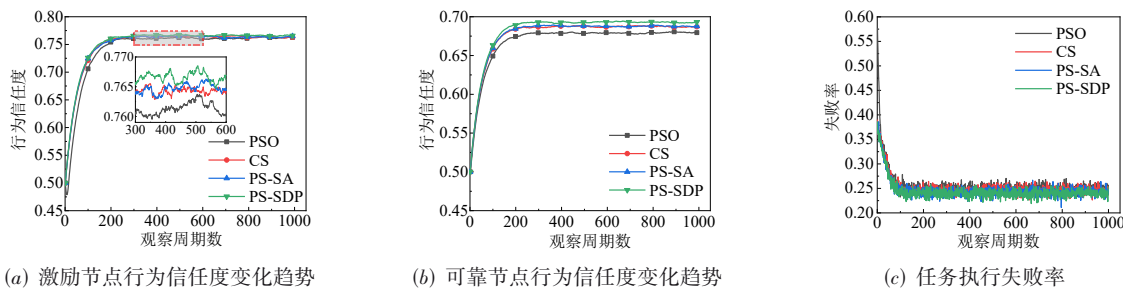


图 20 协作节点寻优算法对比

表 9 协作节点寻优算法决策时长对比

| 搜索算法   | 实验 1   | 实验 2   | 实验 3   | 实验 4   | 实验 5   | 平均时长   | 单位周期时长 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| PSO    | 1568.6 | 1568.7 | 1554.5 | 1552.2 | 1477.6 | 1544.3 | 1.544  |
| CS     | 2915.8 | 2864.4 | 2878.9 | 2858.1 | 2776.9 | 2858.8 | 2.859  |
| PS-SA  | 3852.2 | 3684.7 | 3723.3 | 3770.5 | 3621.3 | 3730.4 | 3.730  |
| PS-SDP | 2737.3 | 2598.5 | 2628.3 | 2643.7 | 2567.9 | 2635.1 | 2.635  |

3.3.3 卸载综合效用

为检验 TOCTE 策略的卸载综合效用,根据边缘计算簇真实任务执行成功率和资源损耗率,在测试组 3 的网络环境下,将 TOCTE 策略与 AODV 策略、TOO 策略、任务可信卸载(Task Credible Offloading, TCO)策略、基于任务可信的任务趋优卸载(Task Optimal Offloading based on Task Credible, TOO-TC)策略和协作节点可信卸载(Cooperative Node Credible Offloading, CNCO)策略对比分析,设置实验对照组。

(1) AODV 随机卸载策略:SN 节点直接接受任务卸载请求,基于 AODV 随机选择 EN 节点执行卸载服务。

(2) TOO 策略:SN 节点直接接受任务卸载请求,趋优选择 EN 节点执行卸载服务。

(3) TCO 策略:SN 节点利用卸载请求行为评价模型评估源节点的卸载请求行为,过滤虚假任务,随机选择 EN 节点执行卸载服务。

(4) TOO-TC 策略:SN 节点利用卸载请求行为评价模型评估任务卸载源节点的卸载请求行为,过滤虚假任务,趋优选择 EN 节点执行卸载服务。

(5) CNCO 策略:SN 节点直接接受任务卸载请求,利用卸载协作综合评价模型评估 EN 节点,遴选最佳协作节点执行卸载服务。

图 21、图 22 和表 10 描述了不同卸载策略对于边缘计算簇真实任务执行成功率和资源损耗率的影响。从图中发现。

(1) 由 AODV 和 TCO 策略、TOO 和 TOO-TC 策略、CNCO 和 TOCTE 策略两两对比,验证了基于卸载请求行为评价的任务过滤机制对系统综合性能优化的有效性,分别提高了 5.38%、5.95% 和 9.72% 真实任务执行成功率,降低了 3.26%、5.13% 和 3.59% 的资源损耗率。

(2) 由 AODV 和 CNCO 策略、TCO 和 TOCTE 策略的两两对比,验证了基于卸载协作综合评价的卸载服务决策机制对系统综合性能优化的有效性,分别提高了 20.64% 和 24.98% 真实任务执行成功率,降低了 1.64% 和 1.97% 资源损耗率。

(3) TOCTE 策略的真实任务执行成功率最高为 59.22%, 平均达到 55.52%, 较 AODV 卸载策略高 30.36%, 较 TOO 策略高 27.55%; 资源损耗率最低为 6.35%, 平均达到 7.53%, 较 AODV 策略低 5.23%, 较

TOO 策略低 7.93%。

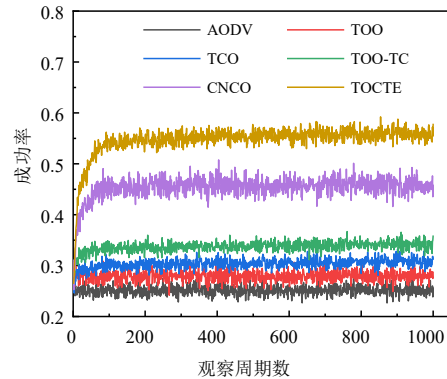


图 21 真实任务执行成功率

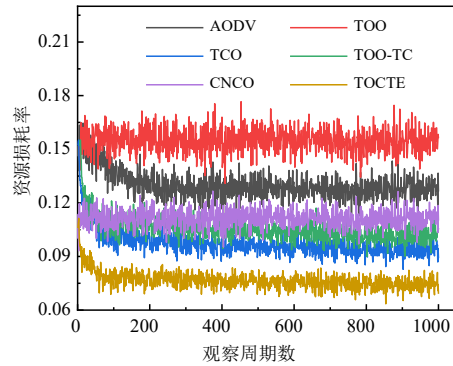


图 22 资源损耗率

表 10 不同策略卸载综合性能对比

| 性能     | 真实任务执行成功率 |        | 资源损耗率  |        |
|--------|-----------|--------|--------|--------|
|        | 最高        | 平均     | 最低     | 平均     |
| AODV   | 0.2804    | 0.2516 | 0.1103 | 0.1276 |
| TOO    | 0.3086    | 0.2797 | 0.1309 | 0.1546 |
| TOO-TC | 0.3668    | 0.3392 | 0.09   | 0.1033 |
| TCO    | 0.3323    | 0.3054 | 0.0828 | 0.095  |
| CNCO   | 0.5076    | 0.458  | 0.0953 | 0.1112 |
| TOCTE  | 0.5922    | 0.5552 | 0.0635 | 0.0753 |

实验结果表明,TOCTE 策略模型通过基于卸载请求行为评价的任务过滤和基于卸载协作综合评价的卸载决策,在任务执行和资源利用方面明显优于其他对比模型,有效提高了边缘计算簇真实任务执行成功率,

降低网络资源损耗。

#### 4 结束语

针对边缘计算任务卸载中任务请求者和资源提供者的可信问题,设计了一种基于综合信任评价的任务卸载策略,建立节点信任遗忘函数和节点行为奖惩算子,构建信任评价机制,衡量节点行为可靠性。以提高资源利用和卸载服务质量为目标,在边缘计算簇间实现基于卸载请求信任的虚假任务过滤,降低非合作节点不良行为对网络性能的影响;设计卸载协作信任模型,将边缘计算簇内的卸载协作节点择优问题转化为卸载协作信任最大化问题,以基于搜索发现概率的粒子群算法寻找协作节点,实现边缘计算的任务可靠卸载。仿真结果表明,所提策略不仅解决了任务来源可信问题,同时保证了边缘节点的卸载协同服务质量。

由于智能终端设备资源有限,任务卸载决策存在一定的延迟和能源开销,迁移、强化学习等方法可有效提高卸载决策效率,但终端训练模型将产生大量能耗,需将学习模型部署在边缘服务器,通过模型裁剪、感知压缩等方法下推模型,实现任务的高效卸载、精准预测。在下一步工作中,我们将研究边缘计算的智能卸载和模型下推等问题。

#### 参考文献

- [1] 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.  
SHI W S, ZHANG X Z, WANG Y F, 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)
- [2] 李智勇,王琦,陈一凡,等.车辆边缘计算环境下任务卸载研究综述[J].计算机学报,2021,44(5):963-982.  
LI Z Y, WANG Q, CHEN Y F, et al. A survey on task offloading research in vehicular edge computing[J]. Chinese Journal of Computers, 2021, 44(5): 963-982. (in Chinese)
- [3] 蔡绍滨,韩启龙,高振国,等.基于云模型的无线传感器网络恶意节点识别技术的研究[J].电子学报,2012,40(11):2232-2238.  
CAI S B, HAN Q L, GAO Z G, et al. Research on cloud trust model for malicious node detection in wireless sensor network[J]. Acta Electronica Sinica, 2012, 40(11): 2232-2238. (in Chinese)
- [4] LIN Y H, SHEN H Y. CloudFog: leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service[J]. IEEE Transactions on Parallel and Distributed Systems, 2017, 28(2): 431-445.
- [5] 王妍,韩笑,曾辉,等.边缘计算环境下服务质量可信的任务迁移节点选择[J].计算机科学,2020,47(10):240-246.  
WANG Y, HAN X, ZENG H, et al. Task migration node selection with reliable service quality in edge computing environment[J]. Computer Science, 2020, 47(10): 240-246. (in Chinese)
- [6] 乐光学,戴亚盛,杨晓慧,等.边缘计算多约束可信协同任务迁移策略[J].电信科学,2019,35(11):36-50.  
YUE G X, DAI Y S, YANG X H, et al. Multi-constrained trusted cooperative task migration strategy for edge computing[J]. Telecommunications Science, 2019, 35(11): 36-50. (in Chinese)
- [7] YAN J J, WU D P, SANYAL S, et al. Trust-oriented partner selection in D2D cooperative communications[J]. IEEE Access, 2017, 5: 3444-3453.
- [8] 邓晓衡,关培源,万志文,等.基于综合信任的边缘计算资源协同研究[J].计算机研究与发展,2018,55(3):449-477.  
DENG X H, GUAN P Y, WAN Z W, et al. Integrated trust based resource cooperation in edge computing[J]. Journal of Computer Research and Development, 2018, 55(3): 449-477. (in Chinese)
- [9] 赫尔曼·艾宾浩斯.记忆的奥秘[M].王迪菲,译.北京:北京理工大学出版社,2013.
- [10] 王妍,韩笑,曾辉,等.边缘计算环境下服务质量可信的任务迁移节点选择[J].计算机科学,2020,47(10):240-246.  
WANG Y, HAN X, ZENG H, et al. Task migration node selection with reliable service quality in edge computing environment[J]. Computer Science, 2020, 47(10): 240-246. (in Chinese)
- [11] 张磊,张希,安成锦,等.基于信誉的延迟容忍网络激励方案[J].电子学报,2014,42(9):1738-1743.  
ZHANG L, ZHANG X, AN C J, et al. A reputation-based incentive scheme for delay tolerant networks[J]. Acta Electronica Sinica, 2014, 42(9): 1738-1743. (in Chinese)
- [12] KULKARNI R V, VENAYAGAMOORTHY G K. Particle swarm optimization in wireless-sensor networks: A brief survey[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C(Applications and Reviews), 2011, 41(2): 262-267.
- [13] YANG X S, DEB S. Cuckoo search: Recent advances and applications[J]. Neural Computing and Applications, 2014, 24(1): 169-174.

- [14] 张以文, 汪开斌, 严远亭, 等. 基于覆盖随机游走算法的服务质量预测[J]. 计算机学报, 2018, 41(12): 2756-2768. ZHANG Y W, WANG K B, YAN Y T, et al. Service quality prediction based on covering random walk algorithm[J]. Chinese Journal of Computers, 2018, 41(12): 2756-2768. (in Chinese)
- [15] KUM D W, LE A N, CHO Y Z, et al. An efficient on-demand routing approach with directional flooding for wireless mesh networks[J]. Journal of Communications and Networks, 2010, 12(1): 67-73.
- [16] WANG B, XIE H X, XIA X D, et al. A NSGA-II algorithm hybridizing local simulated-annealing operators for a Bi-criteria robust job-shop scheduling problem under scenarios[J]. IEEE Transactions on Fuzzy Systems, 2019, 27(5): 1075-1084.



**杨晓慧** 女, 1996年5月出生于山西运城, 博士研究生, CCF 会员, 主要研究方向为边缘计算与协同服务.



**杨忠明** 男, 1998年8月出生于贵州贵阳, 硕士研究生, 主要研究方向为边缘计算与协同服务.

#### 作者简介



**熊小峰** 男, 1965年6月出生于江西赣州, 江西理工大学教授, 主要研究方向为数学建模与算法优化.  
E-mail: xxf\_gz@163.com



**黄淳岚** 女, 1997年4月出生于浙江嘉兴, 硕士研究生, 主要研究方向为边缘计算与协同服务.  
E-mail: huangcl\_lan@163.com



**乐光学(通讯作者)** 男, 1963年11月出生于贵州天柱, 博士, 嘉兴学院教授, IEEE 会员, CCF 会员, 研究领域包括边缘计算、多云融合与协同服务、无线 Mesh 网络和移动云计算.  
E-mail: gxyue@zjxu.edu.cn



**戴亚盛** 男, 1993年8月出生于浙江宁波, 博士研究生, 主要研究方向为边缘计算多云融合与协同服务.