移动边缘计算中隐私感知的在线任务卸载机制

邓慧娜[1,2], 叶阿勇[1,2], 刘燕妮[1,2], 孙明辉[1,2]

[1]福建师范大学计算机与网络空间安全学院 福州 中国 350117
[2]福建省网络安全与密码技术重点实验室 福州 中国 350117

摘要 移动边缘计算是一种新型的计算范式, 它将云计算能力从集中式云分布到网络边缘, 可有效解决云计算实时性低及移动终端计算能力不足等问题。但由于用户移动的不确定性以及边缘服务器的覆盖范围的有限性, 使得实现高效率的任务卸载面临挑战, 并且现有可用性优先的任务卸载算法容易造成用户轨迹隐私泄露。针对上述问题, 本文考虑了迁移成本、轨迹隐私与可用性三者之间的矛盾关系, 基于信息论提出一种高可用性的在线隐私感知任务卸载机制。首先, 基于真实轨迹与发布轨迹之间的互信息量化轨迹隐私泄露程度, 并将该任务卸载问题转换为多目标优化问题; 然后, 进一步提出一种基于马尔可夫链的任务卸载方案来求解该优化问题; 最后, 在多约束场景下设计了面向设备端的轻量级在线任务卸载算法, 解决了在迁移成本约束下轨迹隐私与感知时延的加权平衡问题, 以及迁移成本与感知时延双重约束下的轨迹隐私泄露最小化问题。实验结果表明, 本文提出的隐私感知任务卸载方案在不同约束场景下的安全性均优于其他方案, 能以较低的感知时延实现轨迹隐私保护, 适用于资源受限的移动设备进行快速决策与卸载。

关键词 移动边缘计算; 轨迹隐私; 迁移成本; 可用性; 任务卸载
中图法分类号 TP391 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2023.07.09

Privacy-aware online task offloading mechanism in mobile edge computing

DENG Huina[1,2], YE Ayong[1,2], LIU Yanni[1,2],SUN Minghui[1,2]

[1] College of computer and Cyber Security, Fujian Normal University,Fuzhou, China, 350117
[2] Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou, China, 350117

Abstract Mobile edge computing is a new computing paradigm, it pushes the computing power of cloud servers from the upper-layer centralized cloud to the lower-layer network edge, which can effectively solve the problems of low real-time performance of cloud computing and insufficient computing power of mobile devices. But due to the uncertainty of the user's movement and the limited signal coverage of the edge server, it has become a very challenging problem to get the task offload to be completed efficiently. At the same time, there are currently some usability-first task offloading algorithms, which ignore user trajectory privacy and security in order to obtain the lowest delay. In view of these challenges, we considered the contradictory relationship among the three aspects of migration cost, trajectory privacy, and usability. And we propose a privacy-aware online task offloading mechanism with high availability on the basis of information theory. First of all, we quantify the degree of leakage of trajectory privacy by calculating the mutual information between the real trajectory and the released trajectory, and formulate this task offloading problem as a multi-objective optimization problem. Next, we further propose a task offloading scheme based on Markov chain to solve this optimization problem. At last, we designed a lightweight online task offloading algorithm for the device side in a multi-constraint scenario. We solve the weighted balance problem between trajectory privacy and perceived delay under the constraints of migration cost, and the problem of minimizing trajectory privacy leakage under the dual constraints of migration cost and perceived delay. Our experimental results show that, the privacy-aware task offloading scheme proposed in this paper is more secure than other schemes in different constrained scenarios. It can realize trajectory privacy protection while ensuring low perception delay, and it is suitable for fast decision-making and offloading on those mobile devices with limited resources.

Key words mobile edge computing; track privacy; migration cost; usability; task offloading

通讯作者: 叶阿勇, 博士, 教授, 博士生导师 Email:yay@fjnu.edu.cn。
本课题得到资助国家自然科学基金(No. 61972096, No. 61771140, No. 61872088, No. 61872090), 福建省科技厅高校产学合作计划项目(No. 2022H6025)。
收稿日期: 2022-01-07; 修改日期: 2022-02-21; 定稿日期: 2023-04-18

1 引言

移动边缘计算[1-2](Mobile Edge Computing, MEC)是一种随着 5G 技术发展衍生出的新技术, 它通过将云计算能力和 IT 服务环境下沉到移动通信网络边缘, 就近向用户提供服务, 用户通过无线通信信道直接将计算任务卸载到移动设备周围的边缘服务器上, 从而构建出一个具备高性能、低时延与高带宽的电信级服务环境[3]。目前已经在许多现实场景中得到应用, 例如: 智慧工厂、智能电网、智能驾驶、健康医疗、娱乐和数字媒体[4-6], MEC 技术能在满足计算密集型应用的计算需求的同时保证极小的处理时延[7]。

MEC 技术提高了用户的体验质量, 但 MEC 系统中不合理的任务卸载策略会导致服务器空闲或者过载。现有的任务卸载策略虽然考虑了计算资源、隐私、时延、能耗、成本等限制因素, 但仍以下几个问题没有得到解决: (1)用户的移动轨迹易暴露。可用性优先的调度机制往往将任务卸载到最近的边缘服务器, 以获得最小通信时延, 但也容易暴露用户移动轨迹。如图 1 所示, 攻击者根据路网信息以及用户连续接入的服务器位置, 就可以知道用户的家庭住址以及去哪个医院的隐私信息。(2)轨迹隐私、迁移成本[8]与可用性三者之间相互制约。为了获得更低的时延, 用户设备在快速移动过程中会通过频繁的迁移将任务卸载到最近的服务器, 而计算任务在多个服务器之间迁移时会消耗大量网络资源与能源, 造成服务提供商较大的迁移成本压力。同时攻击者根据现有卸载机制易获得用户的敏感隐私信息, 而进行隐私保护则要选择距离更远的服务器, 此时会牺牲一定的可用性以及负担更高的迁移成本。

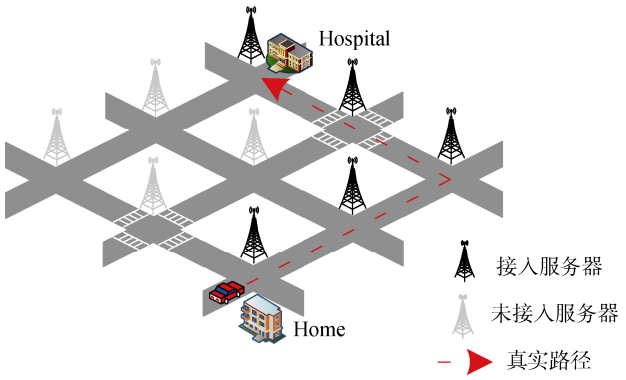

图 1 用户移动轨迹泄露图

Figure 1 Diagram of user trajectory privacy leak

因此, 本文研究在多场景下用户移动卸载过程中受迁移成本约束的任务卸载问题, 主要贡献为以下 3 个方面:

1) 考虑了迁移成本、轨迹隐私与可用性三者之间的矛盾关系, 引入互信息量化隐私泄露度, 将任务卸载问题转换为多目标优化问题, 然后基于马尔可夫链提出一种高可用性的隐私感知在线任务卸载机制。

2) 在两种约束场景下设计了面向设备端的在线任务卸载算法: 在迁移成本约束下提出一种多目标优化的在线任务卸载算法, 实现轨迹隐私与感知时延之间的平衡; 在迁移成本与时延的双重约束下, 提出一种在线最优调度算法, 实现轨迹隐私泄露的最小化。这两种算法的计算复杂度呈线性, 在计算受限的移动设备上具有更高的可用性。

3) 搭建仿真环境, 考虑任务数量、迁移成本约束、时延约束对算法性能的影响, 从 $G(t)$值、隐私泄露度、响应时间等方面分析本文的任务卸载算法。

2 相关工作

目前许多学者在计算卸载策略上展开了深入的研究, 他们的工作可以分为时延优先卸载策略和隐私感知的卸载策略。

2.1 时延优先的计算卸载

时延是服务器性能的体现, 是评估计算卸载策略的一个关键要素, 文献[9-12]针对 MEC 系统中计算卸载的时延问题展开研究。

文献[9]研究基于设备到设备(Device to Device, D2D)的分布式 MEC 系统, 提出一种联合任务分配与功率分配的优化方案以最小化任务处理时延。设计出一个基于遗传算法(GA)的进化方案来解决该混合整数非线性规划(MINLP)问题, 进一步提出一种启发式的移动任务感知调度算法来获得低复杂度的有效任务分配。文献[10]研究在用户移动过程中长期迁移成本约束下的服务器性能优化问题, 以实现用户感知时延最小化。基于 Lyapunov 优化将该长期优化问题分解为不需要先验知识的实时优化问题, 然后基于 Markov 近似算法来寻求这个 NP-hard 问题的近似最优解, 最后提出了一种低时间复杂度的分布式近似方案来解决大规模场景下的计算卸载问题。文献[11]研究边缘计算架构下的用户分配问题, 通过考虑无线信号传输的复杂性与距离对于用户体验质量的影响, 来实现最低延迟和最大化用户 QoE(Quality of Experience, QoE)的目标, 在小规模场景下提出了最优解算法(DEUA-O)和在大规模场景下提出了次优解算法(DEUA-H)。文献[12]从运营商的角度来解

决用户分配问题, 认为感知时延是体现用户体验质量的决定性因素, 通过为每个用户分配不同的QoS(Quality of Service, QoS)级别来实现用户体验质量的最大化, 用整数线性规划技术来求解这个NP-hard 问题, 在大规模场景中提出一种启发式算法来求解它的次优解。

2.2 隐私保护的计算卸载

现有的大部分策略都是时延优先的计算卸载策略, 但是由于分布式的 MEC 设备容易受到攻击, 攻击者通过服务器窃取、推测、利用用户的私人信息来造成损失, 因此一些工作加入隐私保护来制定卸载策略, 文献[13-16]针对计算卸载过程中用户隐私保护问题展开研究。

文献[13]研究 MEC 系统的用户隐私安全问题, 考虑在MEC系统中现有的应用程序无法抵御攻击者对用户卸载进行的推断攻击, 基于 Lyapunov 优化框架提出一种成本效益-隐私权衡的用户任务卸载方案, 来实现在保证最佳用户体验的同时保护用户隐私。文献[14]研究移动边缘计算架构中隐私保护与任务卸载的联合优化问题, 在保护用户隐私的同时获得最低的时延与能耗。将该优化问题表述为一个半参数 MBA(Multi-armed Bandit)问题, 基于转换汤普森采样(TS)结构提出一种面向用户侧的隐私感知在线任务卸载(PAOTO)算法, 来得到最佳的延迟和能耗性能以及保护用户隐私。文献[15]考虑在物联网中的隐私安全问题, 在系统层面上提出一种基于强化学习(Reinforcement Learning)的具有隐私感知的任务卸载方案, 在物联网设备上做出任务卸载决策的同时保护 MEC 系统的用户位置隐私和使用模式隐私。文献[16]针对用户服务信任的问题, 提出一个具有信任感知的任务卸载机制。通过过滤掉用户不信任的服务器, 然后根据卸载策略让用户在可信 MEC 服务器中选择服务器来提供服务, 以减少延迟与能耗。

上述文献从隐私保护和时延优先两个方面提出了相应的计算卸载策略, 但也有一些局限的地方, 他们没有考虑到: 1)现实场景中用户通常是动态移动的; 2)现有的卸载机制易造成轨迹隐私的泄露。因此我们提出了一种针对用户移动过程中对轨迹隐私的保护方案, 主要解决用户在移动过程因为选择服务器位置泄露而造成用户轨迹隐私泄露的问题, 同时提出一种具有隐私感知的实时在线决策算法来解决这个问题。

3 系统模型与问题定义

考虑在单用户多 MEC 网络场景中, 用户 u 在路径 L 上前进时, 周围有 $S=\{s_1, s_2, \cdots, s_\alpha\}$边缘服务器为用户提供计算服务, 每个 MEC 边缘服务器具有一定的计算资源, 且用户可以通过无线接入点或者基站(5G基站)来获服务。我们定义时间$t\in\mathbf{T}=\{1, 2, \cdots, T\}$, 其中 T 为轨迹长度。同时定义 $n\in\mathbf{N}=\{1, 2, \cdots, N\}$ 为用户在一次服务申请时需要向MEC服务器的卸载的任务数, 具体符号含义如表 1 所示。为简化问题, 本系统有以下几个假设:

1) 每个用户每次移动只能请求一次计算任务, 且需大量计算资源以及可接受一定延迟。

2) 每个 MEC 服务器的计算资源与整个任务的迁移成本受限。

3) 接入服务器必须完成用户的每次服务请求。

表 1 主要符号与含义

Table 1 Main symbols and meanings

符号	含义
u	用户
s_i	边缘服务器 s_i
S	服务器集合
T	轨迹长度
n	时刻 t 中, 用户卸载给服务器 s_i 的任务数量
L	长度为 T 的真实轨迹
Γ	长度为 T 的发布轨迹
l_t	时刻 t 的用户真实位置
τ_t	在时刻 t 的接入服务器位置
$x_i^u(t)$	在时刻 t 选择 s_i 服务器的决策
$G(t)$	隐私泄露度与时延的加权函数
$m_{j,i}^u$	用户 u 的任务在 t 时刻从服务器 s_j 迁移到服务器 s_i 所需要的成本
$M^{provide}$	运营商提供的最大迁移成本

3.1 威胁模型

攻击者可分为内部攻击者和外部攻击者, 假定内部攻击者为边缘服务器, 它是诚实但好奇的, 它会遵循整个任务卸载的过程, 但是它可能利用用户上传的位置数据跟踪用户的移动轨迹, 推断用户的隐私信息。外部攻击者为第三方全局攻击者, 它能够通过侧信道攻击监听、推测边缘服务器上的用户信息。

本文考虑存在一个攻击者, 它具有以下能力: 1)获得所有边缘服务器的位置; 2)获得用户所有接入边缘服务器的信息, 包括服务器 ID、接入时间以及服务时长; 3)掌握所有路网信息; 4)推测用户隐私信息。在一定的时间段 1, 2, …, T 内, 用户在移动的过程中向周围的 MEC 服务器进行任务请求, 定义任务卸载过程中的接入MEC服务器位置集合$\Gamma=\{\tau_1, \tau_2, \cdots,$

$\tau_T\}$为发布轨迹(其中 τ_T 为 T 时刻的接入服务器位置), 攻击者通过跟踪用户行进过程中的发布轨迹, 来推测用户的真实轨迹。同时, 为了减少时延与能耗, 用户会选择最近的服务器进行请求和卸载, 攻击者依据这个卸载机制能更精确获得用户在某一时刻的位置, 继而综合路网信息、停留时间等推测出用户的私人信息(如个人职业, 家庭地址, 工作单位等)。因此, 我们的目标是在知道这类情况下, 阻止因用户任务卸载导致的个人隐私信息泄漏。

3.2 轨迹隐私泄露度模型

给定一个时间段 1, 2, $\cdots$, T, 用 $L=\{l_1, l_2, \cdots, l_T\}$ 来表示用户在时间段内长度为 T 的真实轨迹, l_T 表示用在 T 时刻的真实位置。由于服务器位置是公开的, 攻击者根据发布轨迹获得用户的真实轨迹, 因此定义真实轨迹与发布轨迹的相似度为轨迹隐私泄露度, 通过计算发布轨迹与真实轨迹之间的互信息来度量, 公式如下:

$$\begin{aligned}\Phi_{privacy}(t) &= I(L;\Gamma) = I(l_1,l_2,\cdots,l_T;\tau_1,\tau_2,\cdots,\tau_T)\\ &= H(L) - H(L\mid\Gamma)\end{aligned} \tag{1}$$

3.3 迁移成本模型

移动用户在行进过程中会不停对MEC服务器的服务请求, 任务在不同的MEC服务器之间进行迁移, 由此产生额外的操作成本, 因此对迁移成本进行定义。首先, 定义 $m_{j,i}^u$ 为用户 u 的一个单位任务在 t 时刻从服务器 j 迁移到服务器 i 所需要的成本, 因此当 u 经过一条长度为 T(即在 T 个服务器之间迁移)的轨迹时, 需要将 n 个单位任务在不同的服务器之间迁移, 其迁移成本可以表示为:

$$M^u(t) = \sum_{i=1}^{\alpha}\sum_{j=1}^{\alpha} x_j^u(t-1)x_i^u(t)m_{j,i}^u \times n \tag{2}$$

其中, $x_i^u(t)$ 为在时刻 t 用户做的决策, $x_i^u(t)=\begin{cases}1, \text{选择服务器} s_i\\ 0, \text{未选择服务器} s_i\end{cases}$, 即选择服务器 s_i 为任务卸载服务器。服务提供商希望尽可能减用户尽可能减少迁移来降低运营成本, 因此给出一个迁移成本限制 $\sum_{t=1}^{T} M^u(t) \leqslant M^{provider}$, $M^{provide}$ 为最大迁移成本, 如果用户总的迁移成本超过这个值, 运营商将停止本次服务。

3.4 QoS 模型

移动边缘服务器上的计算资源是受限且有差异的, 因此不同的MEC服务器提供给用户的QoS有区别, 在本文的MEC系统结构中, 服务器的QoS由用户感知延迟来体现[5], 用户感知延迟由计算延迟和通信延迟组成。

(1) 计算时延模型

在每个MEC服务器上, 可能会有多个移动用户共享服务器的计算资源, 但服务器的计算资源有限, 当它收到多个任务请求时, 有些用户的请求可能会进入服务器的排队队列中或者被提供低质量的服务。因此, 为了更好的体现出服务器的QoS, 我们建立一个简化的计算时延模型来度量它。在时刻 t 用户 u 将 n 个任务卸载到服务器 s_i 上计算所需时间为: $o(t)=\frac{\delta_n^t}{f_i}$, 在时刻 t 的计算时延为:

$$O(t) = \sum_{i=1}^{\alpha} x_i^u(t)o(t) = \sum_{i=1}^{\alpha} x_i^u(t) \times \frac{\delta_n^t}{f_i} \tag{3}$$

其中 f_i 为服务器的 CPU 计算能力, δ_n^t 是在 t 时刻计算 n 个任务所需要的计算资源。

(2) 通信时延模型

在 MEC 系统中, 在数据传输的过程中移动设备与服务器之间因距离而产生通信时延, 为了便于说明问题, 本文系统模型中不考虑由于信号衰减、网络质量等因素出现丢包等情况。使用一个参数模型来描述任务 $N_t^u=\{\alpha_u,\beta_u,\delta_u,\gamma\}$, 其中 α_u 表示任务上传的数据量, β_u 表示任务下载的数据量, δ_u 表示处理该任务所需要的 CPU 资源, γ 是一个标识数, 表示该任务是否处理成功, 如果成功则为 1, 否则为 0。参考文献[17-18], 用户与 MEC 服务器之间的上传数据速率可以表示为:

$$y_{up}(m,u) = Z_m \log_2\left(1+\frac{p_u \mathrm{B}_{m,u}}{\xi^m}\right) \tag{4}$$

设置输出数据下行信道的网络带宽与输入数据的上传信道的网络带宽表示为相等, 在忽略设备损耗, 环境等其他因素的影响下, 假设同一信道的信道增益和噪声功率是不变的, 因此输出数据的下载速率可以表示为:

$$y_{down}(m,u) = Z_m \log_2\left(1+\frac{p_u \mathrm{B}_{m,u}}{\xi^m}\right) \tag{5}$$

其中, Z_m 是上行信道的网络带宽, p_u 为信道传输功率, $\mathrm{B}_{m,u}$ 为用户 u 和 MEC 节点 m 之间的信道增益, ξ^m 为白噪声功率。计算移动用户 u 输入数据到 MEC 节点 m 之间的传输时间以及当 MEC 服务器完成任务计算以后, 从服务器 m 将输出数据传输给用户 u 的下载时间为:

$$h_{up}(t)=\frac{\alpha_u}{y_{up}(m,u)} \tag{6}$$

$$h_{down}(t)=\frac{\beta_u}{y_{down}(m,u)} \tag{7}$$

因此, 结合公式(6)和(7)总的通信时延为:

$$H(t)=h_{up}(t)+h_{down}(t) \tag{8}$$

综合任务的计算时延(3)和通信时延(8), 在 t 时刻的用户感知时延可表示为:

$$Z(t)=O(t)+H(t) \tag{9}$$

3.5 问题表述

本文主要关注的是在多 MEC 服务器的环境中, 用户在移动过程中进行任务卸载时的服务器选择问题, 涉及用户敏感信息的保护。本文设计了一个迁移成本约束下的具有隐私感知的任务卸载方案, 将迁移成本约束下的任务卸载问题转化为一个联合优化问题, 设置权重w_1, w_2来表示用户对于隐私保护和时延两者不同的偏好, 将轨迹隐私与感知时延(即 QoS)转化成同一维度, 实现轨迹隐私与感知时延的加权最小化, 该问题可定义为:

$$Object: \min_{\sum_{t=1}^{T}M^u(t)\leqslant M^{provide}} \sum_{i=1}^{\alpha} G(t)x_i^u(t) \tag{10}$$

$$G(t)=w_1\Phi_{privacy}(t)+w_2Z(t) \tag{11}$$

$$st: \quad w_1+w_2=1 \tag{12}$$

$$\sum_{i=1}^{\alpha}\sum_{n=1}^{N}x_i^u(t)*\delta_n^t(t)\leqslant C_i \tag{13}$$

$$\sum_{i=1}^{\alpha}x_i^u(t)=1 \tag{14}$$

$$x_i^u(t)\in\{0,1\} \tag{15}$$

其中, 目标(10)是在服务提供商给出的迁移成本约束下, 实现轨迹隐私-感知时延加权函数的最小化, $G(t)$代表加权函数, $\sum_{t=1}^{T}M^u(t)\leqslant M^{provider}$ 表示迁移成本限制, $x_i^u(t)$ 表示在 t 时刻选择服务器 s_i 的决策, $x_i^u(t)*$表示 $G(t)$值最小的最佳决策。公式(11)是 $G(t)$的组成部分, 由轨迹隐私泄露度与时延加权构成。限制(12)表示总权重为 1, 权重数 w_1, w_2 在任务卸载前由用户决定。限制(13)表示用户卸载任务所需的计算资源总和不能超过该服务器的最大容量, 限制(14)和(15)确保每个用户最多被分配给一个具有足够计算资源的边缘服务器。

3.6 NP-hard 问题证明

在 3.5 小节已提出本文的优化问题, 现证明迁移成本约束下的轨迹隐私与时延的联合优化问题是一个 NP-hard 问题, 以下给出证明。

定理 1.迁移成本约束下的轨迹隐私-时延函数最小化问题是 NP-hard 问题。

证明.首先, 介绍一个经典的NP-hard问题: 容量受限的设施选址问题(CFLP)。定义设施为 F, 设施的容量为 C, 任务需求为 R, 花费成本为 $Cost$。此 CFLP 问题可以定义为以下:

$$\min: \sum_{i\in F}\sum_{j\in R}\cos t_{i,j}y_{i,j}+\sum_{i\in F}f_ix_i \tag{16}$$

$$st: \quad \sum_{i\in F}y_{i,j}=1 \tag{17}$$

$$\sum_{j\in R}r_jy_{i,j}\leqslant c_ix_i \tag{18}$$

$$x_i, y_{i,j}\in\{0,1\} \tag{19}$$

其中 $y_{i,j}$ 表示设施 i 是否满足需求 r_j, x_i 表示设施 i 是否打开, $cost_{i,j}$ 是将需求 r_j 分配给设施 i 的花费, c_i 表示设施 i 的容量, f_i 代表开放成本。

我们将 CFLP 问题简化为一个具有隐私感知的任务调度问题, 由于本系统中没有服务器的启动成本问题, 因此 $\sum_{i\in F}f_ix_i=0$。我们给定一个实例 CFLP($Cost$, R, C, F)和本文实例 POTO($G(t)$, N, $M^{provide}$, S), 其中$|R|=|N|$, $|F|=|S|$, $M^{provider}$代表成本限制, $G(t)$是加权最小值。可以得到, 满足目标函数(16)和约束(19)的解决方案也能满足目标函数(10)和约束(15)。每个用户对于服务器的需求不一样, 所以对于成本和时延的权重不一样, 所以满足约束(17)的解决方案也满足约束(12)。当服务器不满足用户的任务需求时, 即不会将任务卸载给服务器, 所以可以将约束(13)投影到约束(18)。

如果一个解决方案能解决 CFLP 问题, 那该方案也能解决本文的优化问题, 因此证明迁移成本约束下的轨迹隐私-时延联合问题也是一个 NP-hard 问题。

4 隐私感知的任务卸载机制设计

一般来说, 用户的轨迹隐私泄露的越少、感知时延越大, 则接入服务器的可用性就越小。因此, 在迁移成本的约束下实现具有隐私感知的任务卸载机制时, 如何达到轨迹隐私与可用性之间的动态平衡仍存在挑战, 目前可采用全局最优的离线方法与局部最优的在线方法来实现两者间的动态平衡。本文的任务卸载机制设计思路如图 2 所示。

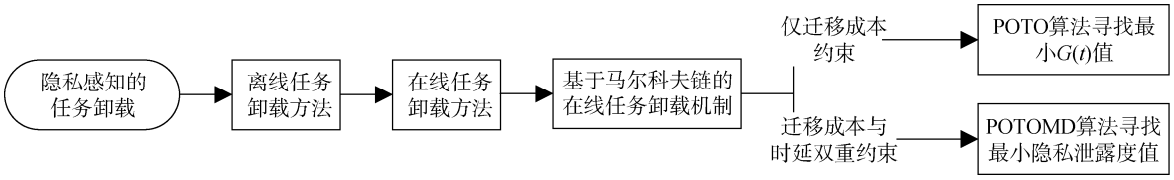

图 2　隐私感知的任务卸载机制设计思路

Figure 2　Design of privacy aware task unloading mechanism

4.1　离线/在线轨迹隐私-可用性权衡命题

命题 1.(离线轨迹隐私-可用性权衡)在一定时间段 1, 2, …, T 内, 给定长度为 T 的真实轨迹 $L=(l_1, l_2, \cdots, l_T)$、长度为 T 的发布轨迹 $\Gamma=\{\tau_1, \tau_2, \cdots, \tau_T\}$, 在迁移成本约束下离线轨迹隐私与感知时延加权最小化函数为:

$$G(t)_{offline}=\min_{\sum_{t=1}^{T}M^u(t)\leqslant M^{provider}} w_1\Phi_{privacy}^{offline}(t)+w_2Z(t) \quad (20)$$

$$\Phi_{privacy}^{offline}(t)=I(L;\Gamma)=I(l_1,l_2,\cdots,l_T;\tau_1,\tau_2,\cdots,\tau_T) \quad (21)$$

其中 $\Phi_{privacy}^{offline}(t)$ 为计算整条发布轨迹与真实轨迹的互信息。

假设一个简化的离线场景, 用户 u 在一条长度为 T 的轨迹上移动, 在每个 t 时刻用户周围有 N 个服务器可选择。用户设备在移动开始前通过全局规划计算出所有发布轨迹的 $G(t)$值, 然后找出最小的 $G^*(t)$ 值对应的发布轨迹, 将该发布轨迹中对应的服务器作为卸载服务器, 从而得到全局最优的任务调度决策。但为了得到最优解, 该方式必须计算出所有发布轨迹与真实轨迹可能组合 $(l,\tau)\in L\times\Gamma$ 的互信息, 计算量为 N^T, 则计算复杂度将随着轨迹长度、服务器密度的增加而呈指数增加, 从而使得计算受限的移动设备无法做出实时决策。因此, 我们提出另一个命题, 即在线轨迹隐私-可用性权衡, 用于分析在线轨迹隐私泄露度-可用性的加权最小化问题。

命题 2.(在线轨迹隐私-可用性权衡)在一定的时间段 1, 2, …, T 内, 迁移成本约束下在线轨迹隐私-感知时延的加权最小化函数为

$$G(t)_{online}=\min_{\sum_{t=1}^{T}M^u(t)\leqslant M^{provider}} w_1\Phi_{privacy}^{online}(t)+w_2Z(t) \quad (22)$$

$$\Phi_{privacy}^{online}(t)=\sum_{t=1}^{T}I(L^t;\tau_t\mid\Gamma^{t-1}) \quad (23)$$

其中 $\Phi_{privacy}^{online}(t)$ 为计算在 t 时刻发布轨迹与真实轨迹的互信息。

定理 2: $G(t)_{offline}\leqslant G(t)_{online}$

证明: 离线任务卸载方法可得到全局最优的任务卸载决策, 在线任务卸载方法得到是局部次优任务卸载决策, 线卸载决策集中包括离线卸载决策。

因为 $Z(t)$值为计算移动设备与边缘服务器之间的感知时延, 离线与在线卸载方法中的 $Z(t)$相等, 因此主要证明 $\Phi_{privacy}^{offline}\leqslant\Phi_{privacy}^{online}$。

$$\begin{aligned}\Phi_{privacy}^{offline}&=I(L;\Gamma)\\&=\sum_{t=1}^{T}I(L;\tau_t\mid\Gamma^{t-1})\\&=\sum_{t=1}^{T}\{I(L^t;\tau_t\mid\Gamma^{t-1})+I(L_{t+1},\cdots,L_T;\tau_t\mid\Gamma^{t-1},L^t)\}\\&\overset{(a)}{=}\sum_{t=1}^{T}I(L^t;\tau_t\mid\Gamma^{t-1})\end{aligned}$$

步骤(a)是因为在在线任务卸载的服务器位置选择设置中, 当前的接入服务器位置 τ_t 独立于由未来的真实位置 $L_{t+1},\cdots,L_T$ 给定的 Γ^{t-1},L^t。因此可以得到, 最小化的和小于或等于单个最小化的和, 因此有 $\Phi_{privacy}^{offline}\leqslant\Phi_{privacy}^{online}$, 由此证明 $G(t)_{offline}\leqslant G(t)_{online}$。

在在线方法中, 感知时延很容易由 t 时刻用户发出请求与收到回复之间的时间差计算得到, 因此目标函数中 $G(t)$的值取决于对隐私泄露度的求解。对于 $\Phi_{privacy}^{online}(t)$, 只需计算时刻 t 内的发布轨迹与每个时刻 t 真实轨迹之间的互信息 $I(L^t;\tau_t\mid\Gamma^{t-1})$, 同时在时刻 t 计算得到的数据会作为时刻 t+1 互信息计算的输入数据, 而不是类似于离线方式需计算整条发布轨迹与真实轨迹间的互信息。

但是, 在线卸载决策仍需要计算时刻 t 内接入服务器位置与所有时刻 t 真实位置之间所有可能组合 $(\tau^t,l^t)\in(\Gamma,L)$ 的互信息值, 即函数(23)的计算仍与轨迹长度 T 有关, 与离线方式一样, 它的计算复杂度仍为 N^T, 呈指数增长。

为了解决上述问题, 我们提出一种基于马尔可夫链的在线任务卸载机制。通过引入马尔可夫链来得到隐私泄露度的上界与下界, 消除轨迹长度对于隐私泄露度计算的影响, 使函数(23)的计算与轨迹长度无关, 从而降低计算复杂度。

4.2 迁移成本约束下基于马尔可夫链的在线任务卸载机制

4.2.1 基于马尔可夫链的隐私泄露度算法

在马尔可夫链中, 当前系统的状态仅仅与之前几个时刻的状态相关, 而与其他时刻状态无关, 极大的简化了一些复杂问题[19]。因此在本文的轨迹隐私模型中, 我们认为方案中的服务器发布位置只与上一时刻的位置状态有关, 而与之前的位置状态无关。假设用户的移动轨迹可以由一种概率分布来生成, 这种概率分布由用户的初始位置概率分布与移动模型计算得, 用长度为 n 的向量 p_1 作为用户的初始位置概率分布, 用户的移动模型可以看作是由马尔可夫转移矩阵 M 表示的一阶马尔可夫链[20-21]。基于这种认知, 本方案在时刻 t 选择的服务器 τ_t 仅取决于当前用户真实位置 l_t, t–1 时刻的服务器位置 τ_{t-1} 和用户真实位置 l_{t-1}, 如图 3 所示。

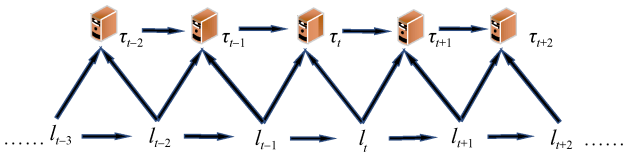

图 3 基于马尔可夫链的发布轨迹

Figure 3 Location release mechanism with Markov chain

根据文献[20]可知, 当前接入服务器位置的选择受制于上一时刻接入服务器与用户的位置, 因此基于马尔可夫链对本方案的接入服务器选择范围进行限制。针对命题 2, 我们通过放宽隐私泄露度函数得到在线真实隐私泄露度 $\Phi_{privacy}^{online}$ 在迁移成本约束下的上界与下界, 使隐私泄露度函数的上下界位置数与轨迹长度无关。因此提出以下定义:

定理 3: (基于马尔可夫限制的轨迹隐私泄露度上下界)通过对接入服务器位置选择机制的马尔可夫限制, 在线隐私泄露度的上下界定义如下:

$$\Phi_{lower}^{Markov} \leqslant \Phi_{privacy}^{online} \leqslant \Phi_{upper}^{Markov} \tag{24}$$

$$\Phi_{upper}^{Markov}(t) = \sum_{t=1}^{T} I(l_t, l_{t-1}; \tau_t \mid \tau_{t-1}) \tag{25}$$

$$\Phi_{lower}^{Markov}(t) = \sum_{t=1}^{T} I(l_t; \tau_t \mid \tau_{t-1}, l_{t-1}) \tag{26}$$

其中, $\Phi_{upper}^{Markov}(t)$ 为在线隐私泄露度 $\Phi_{privacy}^{online}(t)$ 的上限, $\Phi_{lower}^{Markov}(t)$ 为在线隐私泄露度 $\Phi_{privacy}^{online}(t)$ 的下限。

证明: 1) $\Phi_{privacy}^{online} \leqslant \Phi_{upper}^{Markov}$

$$\begin{aligned} I(L^t; \tau_t \mid \Gamma^{t-1}) &\overset{(b)}{=} I(l_{t-1}, l_t; \tau_t \mid \tau_1, \tau_2, \cdots, \tau_{t-1}) \\ &+ I(l_1, l_2, \cdots, l_{t-2}; \tau_t \mid \tau_1, \tau_2, \cdots, \tau_{t-1}, l_{t-1}, l_t) \\ &= I(l_{t-1}, l_t; \tau_t \mid \tau_1, \tau_2, \cdots, \tau_{t-1}) \\ &= H(\tau_t \mid \tau_1, \tau_2, \cdots, \tau_{t-1}) - H(\tau_t \mid l_t, l_{t-1}, \tau_{t-1}) \\ &\leqslant H(\tau_t \mid \tau_{t-1}) - H(\tau_t \mid l_t, l_{t-1}, \tau_{t-1}) \\ &= I(l_t, l_{t-1}; \tau_t \mid \tau_{t-1}) = \Phi_{upper}^{Markov}(t) \end{aligned}$$

步骤(b)的第二项等于零, 是因为在马尔可夫限制中当前的接入服务器位置 τ_t 只取决于 l_t, l_{t-1}, τ_{t-1}, 因此 $\Phi_{privacy}^{online}$ 总是小于等于 Φ_{upper}^{Markov}。

2) $\Phi_{privacy}^{online} \geqslant \Phi_{lower}^{Markov}$

$$\begin{aligned} I(L^t; \tau_t \mid \Gamma^{t-1}) &= H(\tau_t \mid \tau_1, \tau_2, \cdots, \tau_{t-1}) - H(\tau_t \mid \tau_{t-1}, l_t, l_{t-1}) \\ &\geqslant H(\tau_t \mid \tau_1, \tau_2, \cdots, \tau_{t-1}, l_{t-1}) - H(\tau_t \mid \tau_{t-1}, l_t, l_{t-1}) \\ &= H(\tau_t \mid \tau_{t-1}, l_{t-1}) - H(\tau_t \mid \tau_{t-1}, l_t, l_{t-1}) \\ &\overset{(c)}{=} I(l_t; \tau_t \mid \tau_{t-1}, l_{t-1}) \end{aligned}$$

因为在马尔可夫限制中当前接入服务器位置 τ_t 只取决于 l_t, l_{t-1}, τ_t, 因此对步骤(c)的第一项进行了相应的转换, 因此 $\Phi_{privacy}^{online}$ 总是大于等于 Φ_{lower}^{Markov}。

在基于马尔可夫限制计算 $\Phi_{privacy}^{online}(t)$ 时, 只需计算每个时刻 t 内 N 个发布位置与真实位置之间的互信息, 其计算复杂度为 N 呈线性增长。相较于离线卸载方法与在线卸载方法, 基于马尔可夫链的在线任务卸载方案具有更低的计算复杂度、更高的可用性, 适用于资源受限的移动设备做出快速决策。

下面给基于马尔可夫链的在线隐私泄露度算法, 虽然公式(24)给出了在线隐私泄露度的上下限, 但我们只考虑最大的隐私泄露度, 因此只计算马尔可夫上限而不再计算下限。根据文献[22]可知

$$I(l_t, l_{t-1}; \tau_t \mid \tau_{t-1}) = \sum_{\tau_t, l_t, \tau_{t-1}, l_{t-1}} p(\tau_t, l_t, \tau_{t-1}, l_{t-1}) \log \frac{q(\tau_t \mid l_t, \tau_{t-1}, l_{t-1})}{r(\tau_t \mid \tau_{t-1})} \tag{27}$$

$$q(\tau_t \mid l_t, \tau_{t-1}, l_{t-1}) = \frac{r(\tau_t \mid l_{t-1}) e^{-1-\lambda d(l_t, \tau_t)}}{\sum_{\tau_t} r(\tau_t \mid l_{t-1}) e^{-1-\lambda d(l_t, \tau_t)}} \tag{28}$$

$$p(l_t, \tau_{t-1}, l_{t-1}) = \sum_{\tau_{t-2}, v_{t-1}} p(\tau_{t-1}, l_{t-1}, \tau_{t-2}, l_{t-2}) p(l_t \mid l_{t-1}) \tag{29}$$

$$r(\tau_t \mid \tau_{t-1}) = \sum_{v_t, v_{t-1}} p(l_t, l_{t-1} \mid \tau_{t-1}) q(\tau_t \mid l_t, \tau_{t-1}, l_{t-1}) \tag{30}$$

因此, 根据 Φ_{upper}^{Markov} 计算出 $\Phi_{privacy}^{online}$ 的最大值, 以下给出其隐私泄露度算法, 如算法 1 所示:

算法 1. 隐私泄露度算法

输入: 拉格朗日乘子 λ, t–1 时刻内发布服务器位

置与真实位置的联合分布 $p(\tau_{t-1},l_{t-1},\tau_{t-2},l_{t-2})$, 发布服务器位置的边界分布 $p(\tau_{t-1})$, 失真矩阵 $d(\tau_t,l_t)$

输出: 时刻 t 的发布服务器位置的条件分布 $q(\tau_t\,|\,l_t,\tau_{t-1},l_{t-1})$, 发布服务器位置的边界分布 $p(\tau_i)$, 发布服务器位置与真实位置的联合分布 $p(\tau_t,l_t,\tau_{t-1},l_{t-1})$, 最大隐私度量 $\Phi_{upper}^{Markov}(t)$

1: 初始化 $r_0(\tau_t\,|\,\tau_{t-1})$ 作为均匀分布

2: FOR τ_t,l_t in Γ,L do

3: 通过公式(28)计算 $q_0(\tau_t\,|\,l_t,\tau_{t-1},l_{t-1})$

4: 通过公式(29)计算出 $p(l_t,\tau_{t-1},l_{t-1})$

5: 通过公式(30)计算出 $r(\tau_t\,|\,\tau_{t-1})$

6: 通过公式(27)计算出 $I(l_t,l_{t-1};\tau_t\,|\,\tau_{t-1})$

7: 通过公式(21)计算出 $\Phi_{upper}^{Markov}(t)$

8: END FOR

4.2.2 隐私感知的在线任务卸载算法

在计算出 $\Phi_{privacy}^{online}(t)$ 后, 我们提出一种具有隐私感知的在线任务卸载算法(Privacy-aware of Online Task Offloading, POTO), 通过计算每个决策的 $G(t)$ 值来快速做出最佳服务器选择决策, 如算法 2 所示:

算法 2. **隐私感知的在线任务卸载算法 POTO**

输入: 用户 u, 边缘服务器集合 S, 任务量 N, 迁移成本限制 $M^{provide}$

输出: 用户的服务器的最佳分配策略: $u\rightarrow S$

1: 初始化用户的内置服务器列表 *user_X*

2: FOR 在时刻 t

3: 搜寻用户 u 周围的可用服务器

4: IF $S_{cap}^{i}\geqslant N_{cap}^{j}$

5: 将 s_i 服务器加入列表 *user_X*

6: END IF

7: FOR s_i in *user_X* do

8: 根据公式(2), 计算出 $\sum_{t=1}^{T}M^{u}(t)$

9: IF $\sum_{t=1}^{T}M^{u}(t)\leqslant M^{provider}$

10: 将 s_i 服务器加入列表 *user_Y*

11: END IF

12: FOR s_i in *user_Y* do

13: 根据公式(11), 计算出选择每个当前策略 $x_i^u(t)$ 的 $G(t)$ 值

14: END FOR

15: 在策略集 $x^u(t)$ 中选择出 $G(t)$ 值最小的作为最佳分配策略 $x_i^u(t)*$ 执行

16: END FOR

输入一组服务器 S、用户 u 和需卸载的任务量 N, 用户 u 在设备内部建立一个内置的服务器列表 *user_X*, 用来记录周围拥有完成其任务所需的计算资源的服务器(第 1 行), 搜寻周围有哪些可用的服务器(第 2~3 行), 然后挑选出符合要求的服务器(第 4 行, 其中 S_{cap}^{i} 代表服务器的计算资源, N_{cap}^{j} 代表完成任务 N 所需的计算资源)加入移动设备的内置列表中(第 5 行)。前面这部分主要是先选择出符合条件的服务器。

第二部分是对内置列表中的服务器进行 $G(t)$ 值的计算, 每一个选择策略 $x_i^u(t)$ 计算出一个迁移成本值, 判断它是否小于总的成本限制(第 7~11 行), 如果小于再计算符合条件的服务器的 $G(t)$ 值(第 12~14 行), 最后比较选出 $G(t)$ 值最小的服务器作为最佳决策 $x_i^u(t)*$(第 15 行)。

本算法的一个示例如图 4 所示, 用户沿着一条轨迹前进, 在 t 时刻经过探索周围有四个服务器可供卸载。首先构造一个能够满足任务需求的服务器列表, 再根据用户需求从列表中选择出最合适的服务器。表 2 列出用户任务卸载到边缘服务器 s_1, s_2, s_3, s_4 所需的传输时延、计算时延、隐私泄露度与 $G(t)$ 值(设置 w_1=0.9,w_2=0.1)。

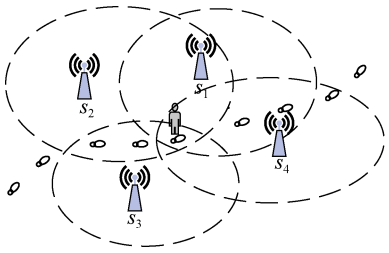

图 4 示例图

Figure 4 Diagram of example

表 2 示例表

Table 2 Table of example

服务器(s_i)	传输时间	计算时延	隐私泄露度	$G(t)$
s_1	1.3	4.3	5.9	7.67
s_2	2.4	∞	7.5	∞
s_3	1.7	3.2	6.8	6.61
s_4	2.6	3.5	7.8	5.83

由表可见, 用户在 s_1, s_2, s_3, s_4 服务器的可用范围内, 服务器与用户的距离差异导致其传输时间不一样, 其中 s_1 与用户之间的距离最近, s_4 最远。同时

不同计算资源的服务器处理任务的时间也不相同, 从计算时延来看, s_3 的 CPU 处理最强大, s_2 的计算能力无法满足用户的任务需求, 因此计算时延无穷。在本次服务申请中, 权重值设置为 w_1=0.9, w_2=0.1, 可见用户更注重隐私的保护, 通过计算得出 s_4 的 $G(t)$ 值最小, 虽然它的感知时延不是最小的, 但其隐私保护程度是最强的, 所以 s_4 是最符合用户需求的服务器。

假设此处有 n 个边缘服务器, 其中可用边缘服务器个数为 m 个, 其中 $m \leqslant n$, 那么 POTO 算法的计算复杂度为 $O(nm)$, 与用户周围的服务器数量乘以可用服务器的数量成正比, 是一种线性关系, 因此 POTO 算法效率较高。

4.3 迁移成本与时延约束下的在线任务卸载算法

在用户任务卸载过程中, 除了存在服务提供商的迁移成本约束, 同时包含用户对服务质量的要求, 即希望在保证基本服务性能的同时尽可能获得更低的感知时延。因此, 本文提出一个最大感知时延 T_{accept}, 即 $Z(t) \leqslant T_{accept}$, 目标是在迁移成本与感知时延双重约束下实现轨迹隐私泄露最小化, 目标函数可表示为:

$$Object: \min_{\substack{Z(t)\leqslant T_{accept};\\ \sum_{t=1}^{T} M^u(t)\leqslant M^{provider}}} : \Phi_{privacy}^{online}(t)x_i^u(t) \tag{31}$$

该问题只需解决隐私泄露度的最小化, 不再是 NP-hard 问题, 因此我们提出一种迁移成本与时延约束的隐私感知在线任务卸载算法(Privacy-aware Online Task Offloading with Migration cost and Delay constraints, POTOMD), 实现过程如算法 3 所示。

算法 3. **迁移成本与时延约束的在线卸载算法 POTOMD**

输入: 用户 u, 边缘服务器集合 S, 任务量 N, 迁移成本限制 $M^{provide}$, 时延限制 T_{accept}

输出: 用户的服务器的最佳分配策略 $x_i^u(t)*$: $u \rightarrow S$

1: 初始化用户的内置服务器列表 *user_X*, *user_Y*

2: FOR 在时刻 t

3: 搜寻用户 u 周围的可用服务器

4: IF $S_{cap}^i \geqslant N_{cap}^j$

5: 将 s_i 服务器加入列表 *user_X*

6: END IF

7:FOR s_i in *user_X* do

8: 根据公式(2), 计算出 $\sum_{t=1}^{T} M^u(t)$

9: 根据公式(9), 计算出 $Z(t)$

10: IF $\sum_{t=1}^{T} M^u(t) \leqslant M^{provider}$ 且 $Z(t) \leqslant T_{accept}$

10: 将 s_i 服务器加入列表 *user_Y*

11: END IF

12: END FOR

13: FOR s_i in *user_Y* do

14: 根据隐私泄露度算法, 计算出选择每个当前每个策略 $x_i^u(t)$ 的 $\Phi_{privacy}^{online}$ 值

15: END FOR

16: 在策略集 $x^u(t)$ 中选择出值 $\Phi_{privacy}^{online}$ 值最小的作为最佳 $x_i^u(t)*$ 执行策略

17: END FOR

与 POTO 算法相似, 假设此处有 n 个边缘服务器, 其中可用边缘服务器个数为 m 个, 其中 $m \leqslant n$, POTOMD 的计算复杂度为 $O(nm)$, 呈线性增长, 其计算效率较高。

5 实验与分析

我们基于 ONE 模拟器来进行现实世界的实验模拟, 它的主要道路数据是芬兰赫尔基市中心的街道和道路, 移动用户基于设置好的行径轨迹移动。将市中心分为 56 块平均区域, 在每个区域的中心配备一个 MEC 服务器来提供卸载服务, 主要参数设置如表 3 所示。设置 10000s 的模拟时间, 移动用户与 MEC 服务器之间使用 WiFi(802.11 标准)来进行通信, 每次实验进行了 50 次的重复实验, 实验结果取平均值。在实验中设置所有方案用户的移动轨迹相同, 但是发布轨迹根据方案的不同而有差异。最后与以下三个算法进行性能比较:

表 3 实验设置

Table 3 Configuration of experiment

参数设定	参数值
WiFi 服务半径/m	200
MEC 服务器计算能力/GHz	20
MEC 服务器数/个	56
移动用户速度/(m/s)	[0.5,1]
移动设备计算能力/GHz	0.5
信道增益/dBm	[–30, –10]
单位任务大小/Mbps	[0.5~1]
噪声功率/dBm	[1,2]
网络带宽/MHz	10

1. 贪心算法(Greedy, GD): 基准方案, 将任务卸载到信号范围内最近未满载的服务器上。

2. LB 算法[22]: 先评估周围服务器的负载情况, 将任务卸载到负载最小的服务器上。

3. DCCRA 算法[23]: 考虑任务迁移的影响动态选择时延最低的服务器。

使用以下五个实验来评估算法的效果:

1) 在相同的实验参数条件下, 比较本文算法与其他三种算法在不同迁移成本约束下 $G(t)$值的对比。

2) 在相同的实验参数条件下, 比较本文算法与其他三种算法在不同任务数量下的响应时间对比。

3) 在相同的实验参数条件下, 不同 w_1, w_2 取值方案在不同任务数量下的平均响应时间。

4) 在相同的实验参数条件下, 比较不同的 w_1, w_2 取值方案以与其他三种方案的安全性。

5) 在相同的实验参数条件下, 比较本文算法与其他三种算法在不同时延约束下的安全性。

5.1 发布轨迹图

图 5 展示了在不同 w_1, w_2 取值方案中的发布轨迹图, 其中 w_1, w_2 代表不同的权重(w_1 越大, 即用户对隐私保护的需求更大; w_2 越大, 即用户对响应速度要求更高)椭圆中的数值代表 MEC 服务器的标号, 箭头表示两节点之间的信息交互, 每个节点位置连在一起形成了发布轨迹。由此可见, 不同程度的隐私保护方案生成的轨迹图是不相同的。

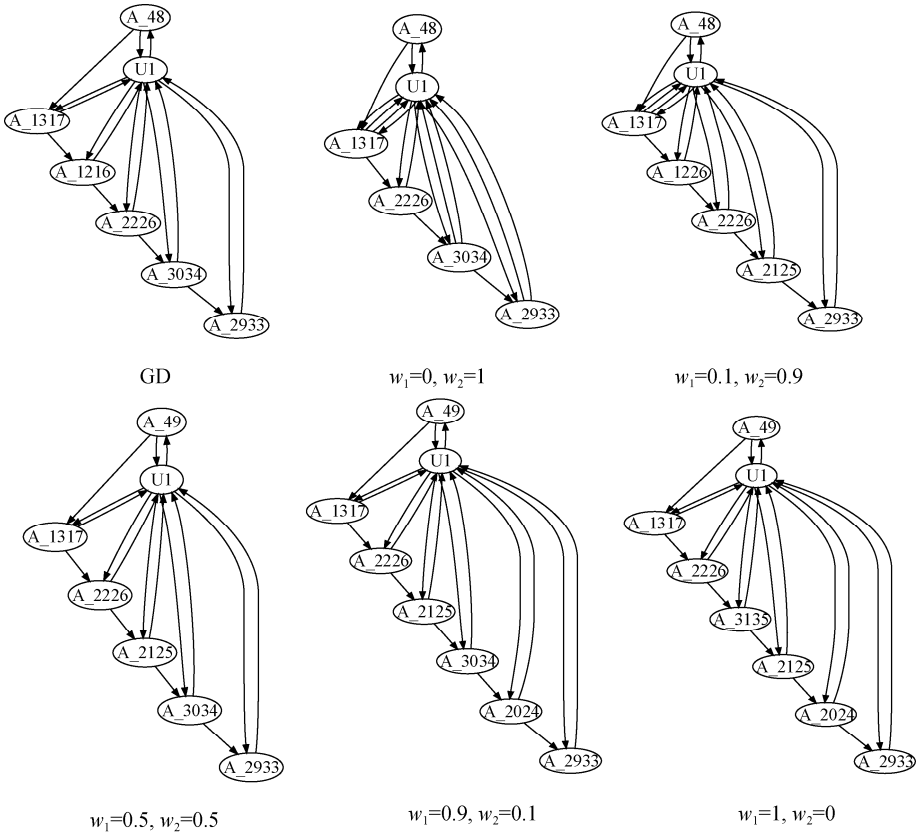

图 5 不同方案信息交互图

Figure 5 Information interaction diagram of different schemes

5.2 不同迁移成本约束对 $G(t)$的影响

$G(t)$值是轨迹隐私泄露度与时延的加权函数值, 它主要考虑用户在隐私与时延之间的权衡, 其值越小说明该调度策略越合理且更符合用户要求。图 6 表示的是不同的调度方案对 $G(t)$值的影响, 我们考虑了不同的成本约束 70、140、210、280, 同时设置权重值为 w_1=0.5, w_2=0.5, 用户任务卸载数量为 15。可以看到随着迁移成本约束值的增加, $G(t)$值也随之降低, 在迁移成本约束为 70、140 时, DCCRA、LB、POTO 方案的 $G(t)$值相近, 这是因为成本的约束, 使用户的可选服务器被限制; 在迁移成本约束为 210、280 时, 它们的 $G(t)$值在逐渐降低, 这是因为随着成本的增加, 用户的可选择服务器范围更广, POTO 可选择隐私泄露度更低的服务器, 其他方案可以选择时延更低的服务器, 同时相较于其他三个方案, POTO 方案的 $G(t)$值更小, 因为虽然 POTO 算法的时

延较高, 但是其隐私泄露度是最小的。

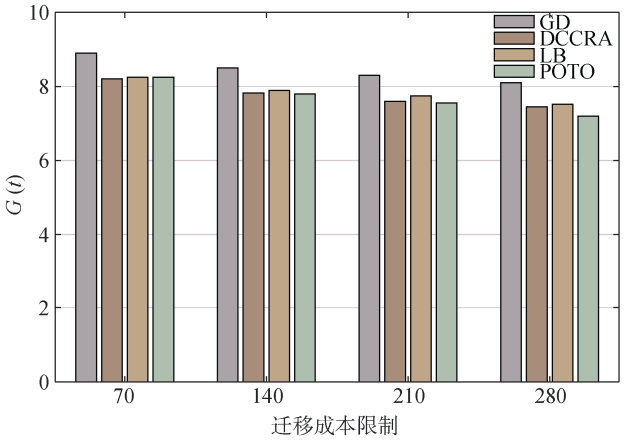

图 6 迁移成本对 $G(t)$值的影响

Figure 6 Influence of migration cost on $G(t)$

5.3 不同算法的响应时间

在迁移成本限制值固定为 280 的情况下, 考虑任务数量的影响, 进一步评估四种调度算法的可用性。它们的平均响应时间如图 7 所示, 随着任务量从 1 增加到 15 的过程中, 这四种算法的响应时间呈近乎线性的增长趋势。总体来看, 相较于其他三种算法, POTO 算法的平均响应时间要长一点(其中相对于 GD 算法平均增加 0.42s, 相对于 LB 算法平均增加 0.69s, 相较于 DCCRA 算法平均增加 0.95s), 这是因为 POTO 算法将隐私安全考虑进去了, 它通过选择相对距离更远的服务器来进行隐私保护, 因此导致更大的通信开销。同时可以看到虽然我们的方案响应时间更长, 但时间差几乎都是在 1s 内, 用户的感知延迟并不明显, 因此这个时差是可以接受的。

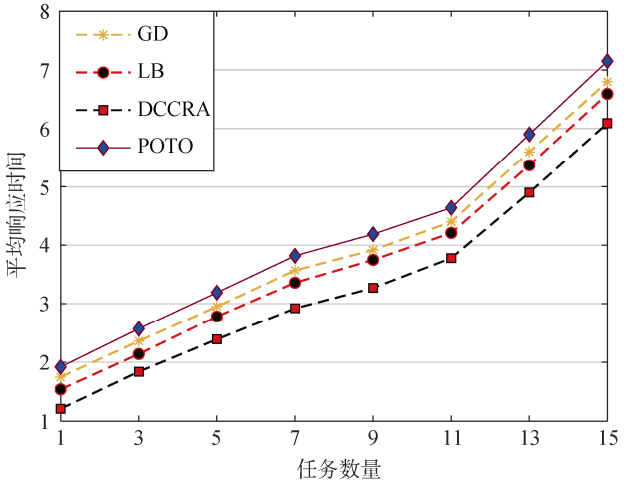

图 7 四种算法平均响应时间

Figure 7 Average response time of four algorithms

5.4 不同 w_1,w_2 取值方案的平均响应时间

在迁移成本为 280 的约束条件下, 本实验选取五个具有代表性的w_1, w_2取值, 通过获得的响应时间数据来分析不同取值之间性能差异的原因。图 8 是不同取值方案完成不同任务数量所需的的平均响应时间, 由图可见, w_1=0, w_2=1 的时候完成服务请求所需的时间是最少的, 且随着 w_1 值的增加, w_2 值的减少, 完成计算任务所需的时间在慢慢增加, 在 w_1=1, w_2=0 的时候的完成任务所需时间最长。这是因为随着 w_1 值的增加, 表示用户对于隐私安全的更加重视, 在实现隐私保护时, 其算法会在可用服务器中选择距离最远的服务器, 因此带来更长的通信时延。在本文的调度策略中, 任务卸载所需的时间会随着隐私保护强度增大而增加, 实现隐私保护的代价是需要更长的时延, 所以我们的方案是由用户侧来决定安全的强度, 拥有更高的灵活度和自由度。

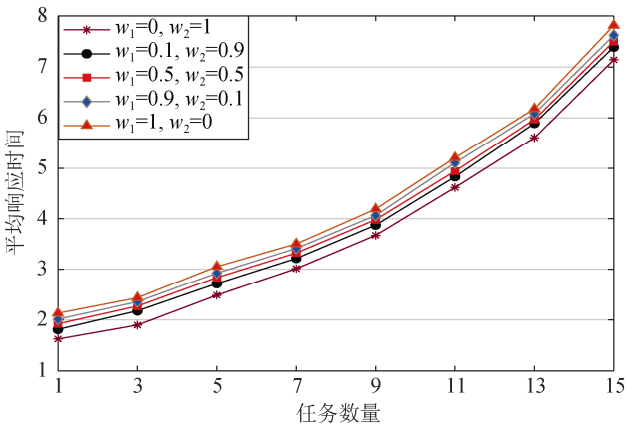

图 8 不同 w_1,w_2 取值的响应时间

Figure 8 Response time of different w_1 and w_2

5.5 发布轨迹与真实轨迹之间的隐私泄露度

隐私泄露度是指用户轨迹隐私的泄露程度, 它主要评估调度方案的安全性, 其值越小, 该方案的安全性越高。如图 9 所示, 本实验考虑在迁移成本约束为 280, 任务数量固定为 15 的情况下, 以真实轨迹为基准, 计算 DCCRA 方案、LB 方案和不同 w_1, w_2 取值方案的隐私泄露度。从图中可以看出, 随着 w_1 值的增加, 隐私泄露度值随之降低, 在 w_1=1, w_2=0 时的泄露度最小, 说明此方案的安全性最高, 其中 GD 方案的隐私泄露度最大, 安全性最低。同时, 也可看出相对于有隐私保护考量的方案来说(w_1=0.1, w_2=0.9; w_1=0.5, w_2=0.5; w_1=0.9, w_2=0.1), 没有加隐私保护的方案泄露度值要更高(GD;DCCRA; LB; w_1=0, w_2=1)。其中, LB 方案与 w_1=0, w_2=1 方案的泄露度值较为接近, 说明在不加入隐私保护时, 用户一般会选择距

离最近的服务器来服务请求。因此我们可以得到本文方案相对于其他方案来说安全性更高，更能保护用户的隐私安全。

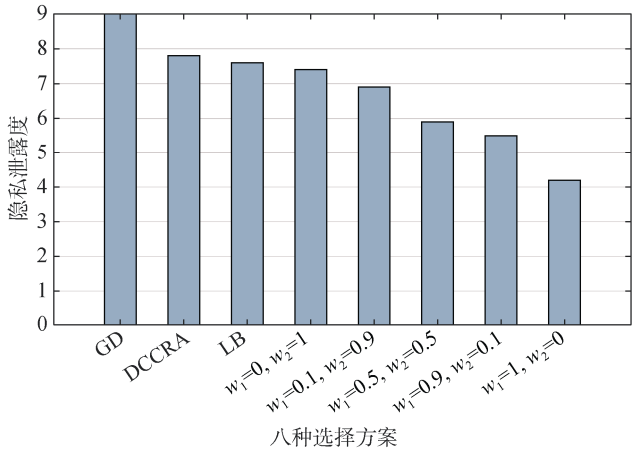

图 9　不同方案的隐私泄露度

Figure 9　Privacy disclosure of different schemes

5.6　不同的时延限制下的隐私泄露度

本实验主要是评估在迁移成本与感知时延同时约束下的调度算法安全性能，因此在迁移成本限制为 280，任务数量为 15 的情况下，计算了四种方案在不同时延约束下的隐私泄露度。如图 10 所示，随着时延限制从 6 增加到 8 的过程中，它们的泄露度值在逐步减少。同时在时延限制为 6、6.5、7 时它们的泄露度值较为接近，其中算法 2 与 GD 算法的值相同都为 10，这是因为时延的限制过小时，服务器选择过少，为了满足用户的时延限制，四种调度方案只能将任务卸载到最近的服务器。在时延限制为 7.5、8 的时候，算法 2 的隐私泄露度值呈断崖式下跌，因为此时的时延限制值能够让 POTOMD 算法在满足用户

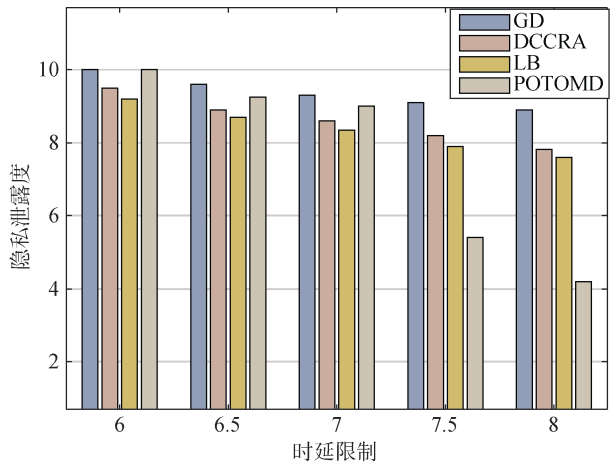

图 10　不同时延限制下的隐私泄露度

Figure 10　Privacy leakage of different delay constraints

时延需求的同时，可以选择距离更远的服务器实现轨迹隐私保护，让隐私泄露度值更低。

6　总结

本文利用互信息来度量轨迹隐私泄露度，并结合迁移成本和时延来综合考虑服务器的选择，提出了一种任务调度的动态调整策略，并设计一种轻量级的实时在线用户侧决策算法，使用户可以在移动过程中，拥有既能隐私保护又能拥有较好 QoE 体验的任务卸载方案。本文提出的具有隐私感知的任务调度方案与其他方案不同，我们在考虑用户移动性的同时，还考虑了用户的轨迹隐私。在下一步研究中我们计划在任务调度中考虑更多具体现实的场景，在轨迹数据的安全保护上做进一步的优化，同时在综合量化计算中引入更多的因素，从多个用户角度来做服务器选择。

参考文献

[1] Ahmed A, Ahmed E. A Survey on Mobile Edge Computing[C]. *2016 10th International Conference on Intelligent Systems and Control*, 2016: 1-8.

[2] Yu Y F. Mobile Edge Computing towards 5G: Vision, Recent Progress, and Open Challenges[J]. *China Communications*, 2016, 13(Supplement2): 89-99.

[3] Liu J H, Zhang Q. Code-Partitioning Offloading Schemes in Mobile Edge Computing for Augmented Reality[J]. *IEEE Access*, 2019, 7: 11222-11236.

[4] 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.
(施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. *计算机研究与发展*, 2019, 56(1): 69-89.)

[5] Xu X L, Liu X H, Xu Z Y, et al. Trust-Oriented IoT Service Placement for Smart Cities in Edge Computing[J]. *IEEE Internet of Things Journal*, 2020, 7(5): 4084-4091.

[6] Liang G J, Wang Q, Xin J F, et al. Survey of Mobile Edge Computing Resource Allocation[J]. *Journal of Cyber Security*, 2021, 6(3)227-256.
(梁广俊, 王群, 辛建芳, 等. 移动边缘计算资源分配综述[J]. *信息安全学报*, 2021, 6(3)227-256.)

[7] Chen X, Pu L J, Gao L, et al. Exploiting Massive D2D Collaboration for Energy-Efficient Mobile Edge Computing[J]. *IEEE Wireless Communications*, 2017, 24(4): 64-71.

[8] Shen C, Tekin C, van der Schaar M. A Non-Stochastic Learning Approach to Energy Efficient Mobility Management[J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(12): 3854-3868.

[9] Saleem U, Liu Y, Jangsher S, et al. Mobility-Aware Joint Task Scheduling and Resource Allocation for Cooperative Mobile Edge Computing[J]. *IEEE Transactions on Wireless Communications*,

2021, 20(1): 360-374.

[10] Tao O Y, Zhi Z, Xu C. Follow me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing[C]. *IEEE Journal on Selected Areas in Communications*, 2018: 2333-2345.

[11] Xu Z, Zou G, Xia X, et al. Distance-aware Edge User Allocation with QoE Optimization[C]. *2020 IEEE International Conference on Web Services*, 2020:66-74.

[12] Lai P, He Q, Cui G M, et al. QoE-Aware User Allocation in Edge Computing Systems with Dynamic QoS[J]. *Future Generation Computer Systems*, 2020, 112: 684-694.

[13] He X F, Jin R C, Dai H Y. Peace: Privacy-Preserving and Cost-Efficient Task Offloading for Mobile-Edge Computing[J]. *IEEE Transactions on Wireless Communications*, 2020, 19(3): 1814-1824.

[14] Zhu D , Li T , Liu H , et al. Privacy-Aware Online Task Offloading for Mobile-Edge Computing[J]. *Wireless Communications and Mobile Computing*, 2021, 2021(3):1-16.

[15] Min M H, Wan X Y, Xiao L, et al. Learning-Based Privacy-Aware Offloading for Healthcare IoT with Energy Harvesting[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 4307-4316.

[16] Wu D X, Shen G H, Huang Z Q, et al. A Trust-Aware Task Offloading Framework in Mobile Edge Computing[J]. *IEEE Access*, 2019, 7: 150105-150119.

[17] Dinh T Q, Tang J H, La Q D, et al. Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling[J]. *IEEE Transactions on Communications*, 2017, 65(8): 3571-3584.

[18] Liu J, Mao Y Y, Zhang J, et al. Delay-Optimal Computation Task Scheduling for Mobile-Edge Computing Systems[C]. *2016 IEEE International Symposium on Information Theory*, 2016: 1451-1455.

[19] Götz M, Nath S, Gehrke J. MaskIt: Privately Releasing User Context Streams for Personalized Mobile Applications[C]. *The 2012 ACM SIGMOD International Conference on Management of Data*, 2012: 289-300.

[20] Zhang W J, Li M, Tandon R, et al. Online Location Trace Privacy: An Information Theoretic Approach[J]. *IEEE Transactions on Information Forensics and Security*, 2019, 14(1): 235-250.

[21] Shokri R, Theodorakopoulos G, Le Boudec J Y, et al. Quantifying Location Privacy[C]. *2011 IEEE Symposium on Security and Privacy*, 2011: 247-262.

[22] Jia M K, Liang W F, Xu Z C, et al. Cloudlet Load Balancing in Wireless Metropolitan Area Networks[C]. *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016: 1-9.

[23] Plachy J, Becvar Z, Strinati E C, et al. Dynamic Allocation of Computing and Communication Resources in Multi-Access Edge Computing for Mobile Users[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(2): 2089-2106.

邓慧娜 于 2019 年在湖南师范大学应用化学专业获得工学学位, 现在福建师范大学计算机与网络空间安全学院网络空间安全专业攻读硕士学位。研究兴趣为移动边缘计算中的位置隐私保护, 任务卸载机制。Email:denghn220@163.com

叶阿勇 于 2009 年于西安电子科技大学计算机系统结构专业获得博士学位, 现任福建师范大学计算机与网络空间安全学院教授, 博士生导师, CCF 高级会员。主要研究方向为区块链、网络 安全和位置隐私。Email:yay@fjnu.edu.cn

刘燕妮 于 2021 年在韩山师范学院计算机科学与技术专业获得理学学士学位, 现在福建师范大学网络空间安全专业攻读硕士学位。研究领域为数据治理与区块链, 研究兴趣包括: 数据共享与数据可携性, 隐私计算。Email:2426759858@qq.com

孙明辉 于 2020 年在华北科技学院计算机科学与技术专业获得工学学士学位。现在福建师范大学网络空间安全专业攻读硕士学位。研究领域为对抗学习与隐私保护。研究兴趣包括生成对抗网络、对抗样本等。Email:ming_hui_sun@163.com