DOI:10.20079/j.issn.1001-893x.221013005

移动边缘网络中基于用户移动的服务功能链迁移策略*

周冬杨[1],潘显兵[2]

(1.重庆移通学院 数字经济与信息管理学院,重庆 401520;2.公共大数据安全技术重庆市重点实验室,重庆 401420)

摘　要:移动边缘计算(Mobile Edge Computing,MEC)通过在网络边缘部署服务器,提供计算和存储资源,可为用户提供超低时延和高带宽业务。网络功能虚拟化(Network Function Virtualization,NFV)与 MEC 技术相结合,可在 MEC 服务器上提供服务功能链(Service Function Chain,SFC),提升用户的业务体验。为了保证移动用户的服务质量,需要在用户跨基站移动时将 SFC 迁移到合适的边缘服务器上。主要以最小化用户服务的端到端时延和运行成本为目标,提出了 MEC 网络中具有资源容量约束的 SFC 迁移策略,以实现移动用户业务的无缝迁移。仿真结果表明,与现有方案相比,该策略具有更好的有效性和高效性。

关键词:移动边缘计算(MEC);网络功能虚拟化(NFV);服务功能链(SFC);用户移动性;无缝迁移

开放科学(资源服务)标识码(OSID):

微信扫描二维码
听独家语音释文
与作者在线交流
享本刊专属服务

中图分类号:TN915　　**文献标志码**:A　　**文章编号**:1001-893X(2024)01-0008-06

Migration Strategy of Service Function Chain Based on User Mobility in Mobile Edge Networks

ZHOU Dongyang[1], PAN Xianbing[2]

(1. School of Digital Economy and Information Management, Chongqing College of Mobile Communication, Chongqing 401520, China; 2. Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing 401420, China)

Abstract: Mobile edge computing (MEC) provides ultra-low-latency and high bandwidth services to mobile users through deploying servers at the edge of the network and providing computation and storage resources. Along with the MEC technology, network function virtualization (NFV) can implement service function chain (SFC) in the MEC servers to improve the user experience. In order to ensure the quality of service (QoS) of the network, the SFC should be migrated to the applicable edge servers while mobile users move across the base stations. The authors propose a SFC migration policy considering the resources capacity constraints to achieve seamless migration of mobile user services as well as minimizing the sum of the end-to-end delay and the running costs of the user services. The simulation results show that the proposed scheme has better effectiveness and efficiency compared with the existing schemes.

Key words: mobile edge computing (MEC); network function virtualization (NFV); service function chain (SFC); user mobility; seamless migration

0　引　言

物联网传感器的爆炸式增长和智能设备的广泛普及造成移动网络的流量迅速增加。因便携式移动设备的计算和存储资源容量有限,可以将任务卸载

*　**收稿日期**:2022-10-13;**修回日期**:2022-12-03
基金项目:重庆市教育科学规划课题(K22YG224265);重庆移通学院教改项目(22JG215)
通信作者:周冬杨　　Emai:zhou88855599@126.com

到远程云进行处理,但这不可避免地会导致响应延迟,降低用户体验[1-2]。移动边缘计算(Mobile Edge Computing,MEC)[3]是一种新兴的云架构,通过在基站(Base Station,BS)侧部署 MEC 服务器,为移动用户提供计算和存储资源,显著缩短响应延迟,满足多样化的业务需求。

网络功能虚拟化(Network Function Virtualization,NFV)技术用虚拟机中的软件替代传统硬件实现的网络功能,称为虚拟化网络功能(Virtual Network Function,VNF)。通常,业务流量需要由多个 VNF 按照逻辑顺序进行处理,称为服务功能链(service Function Chain,SFC)。NFV 与 MEC 的结合可以减少云服务提供商(Cloud Service Provider,CSP)核心网的服务延迟和流量压力,提供方便灵活的 VNF 管理,大大降低 CSP 的运营费用。

由于边缘云(Edge Cloud,EC)的资源有限,合理分配 EC 资源以满足尽可能多的用户需求是至关重要的。另外,当用户跨基站移动时,需决定由以前的 EC 或迁移到新的 EC 执行流量处理。第一种情况,SFC 没有迁移,以牺牲用户通信时延为代价节约了 CSP 的网络成本。第二种情况,用户通信延迟是最优的,但有潜在的 SFC 数据迁移成本。因此,移动用户的延迟和运营成本之间存在一个权衡。同时,为了让用户不能直接感知因 SFC 迁移过程造成的服务中断,需要进行无缝迁移[4]。

当前,部分文献主要关注 MEC 网络中 VNF 的映射[5-6],然而没有考虑到用户的移动性,这是 MEC 网络的一个基本特征。随着用户的移动,SFC 固有的映射方法可能会导致用户服务延迟严重恶化。

文献[2]主要研究用户移动时 VNF 的最优迁移时间,但以整体网络迁移成本作为优化目标,不能保证单个移动用户的 QoS 总是可靠的。文献[8]使用图划分算法最小化 VNF 在集群之间迁移的数量,不能保证具有严格延迟要求的用户服务。

与上述研究不同,本文综合考虑迁移成本和服务时延两个指标,提出了基于用户移动的 SFC 无缝迁移策略。

1　系统模型

1.1　网络模型

用无向图 $G=(B\cup V,L)$ 表示移动边缘云网络,其中 B 为 BS 集合,V 为 EC 集合,L 为 BS 之间的物理链路集合。

每个基站 $b\in B$ 起到服务接入和转发的作用。对于每个 EC $v\in V$,用 C_v 表示最大的计算能力,可用于实现移动用户的 SFC。为了节约 CSP 的成本,EC 的数量小于 BS 的数量,即不是所有的 BS 都绑定 EC。此外,BS 和同位 EC 之间的传播延迟可以忽略不计。物理链路 $l\in L$ 是一条高速光缆连接各个 BS,将其表示为五元组 $\{s,d,b_l,d_l,c_l\}$,其中 s,d 表示源 BS 和目的地 BS,b_l 是最大带宽,d_l 是固有的传播延迟,c_l 表示数据在链路 l 上的传输成本。

1.2　移动用户模型

假设 SFC 中为移动用户 i 提供的所有 VNF 都集成在相应 EC 的服务器中。移动用户 $i\in U$ 的请求表示为 $r_i=\{\mathrm{SFC}_i,D_i,L_i,\hat{T}_i\}$,其中 SFC_i 是用户 i 所需的 SFC 类型,D_i 表示用户服务请求的最大容忍延迟,L_i 是移动用户 i 的移动配置文件,$\hat{T}_i$ 是用户 i 用于无缝迁移 SFC_i 的最大容忍时间。

移动用户 i 的移动轨迹表示为 $L_i=\{l_i(t)\}$。$l_i(t)\in B$ 为用户 i 在时隙 t 时的临时位置,此时,通过最近的 BS 接入网络,服务流量 $v_i(t)$ 通过相应的 EC 处理。随着用户移动,接入的 BS 会进行切换。

2　问题描述及模型定义

2.1　问题描述

假设每个移动用户只请求一种 SFC 类型,且该用户具有较高的移动性和服务延迟敏感性。当用户 i 在时隙 t 移动到一个新的临时位置时,如果业务流量 $v_i(t+1)$ 仍然由之前的 EC 处理,可能会违反用户 u 的端到端延迟容忍。因此,需要根据用户 u 的配置文件确定最优 EC 的位置。也就是说,当用户跨越 EC,即 $v_i(t)\neq v_i(t+1)$,需要一个合理的迁移策略来决定何时以及如何选择相关的路径对 SFC 进行无缝迁移。

2.2　相关参数定义

移动用户请求 r_i 的端到端延迟主要分为四部分:接入延迟、通信延迟、执行延迟和迁移延迟。用户 i 将服务请求数据 $Q_{i,t}$(以 b 为单位)传输给相应的 EC 进行计算。

接入延迟 $d_{\mathrm{acc}}(i,t)$:表示请求 r_i 在用户 i 和附近的无线信道与 BS $l_i(t)$ 之间传输流量数据所花费的总时间。无线信道的最大数据传输速率

$r_{i,t}$(单位:b/s)为

$$r_{i,t}=W\cdot \mathrm{lb}\left(1+\frac{P_i g_t}{\sigma^2}\right)\quad \forall i\in U \tag{1}$$

式中:W 为信道带宽;P_i 为用户移动设备的发送功率;g_t 为时隙 t 无线信道增益;σ^2 为噪声功率。因此,接入延迟表示为

$$d_{\mathrm{acc}}(i,t)=\frac{Q_{i,t}}{r_{i,t}} \tag{2}$$

通信延迟 $d_{\mathrm{com}}(i,t)$ 由传播延迟和传输延迟组成。BS 到相应 EC 的流量转发路径表示为 $p^c_{i,t}$,$x_{p,l,t}(t)$是一个二进制变量,表示路径 $p^c_{i,t}$ 是否在时隙 t 对移动用户 i 使用链路 l。

$$x_{p,l,i}(t)=\begin{cases}1, & p^c_{i,t}\text{ 使用链路 }l\\ 0, & \text{否则}\end{cases} \tag{3}$$

从本地 BS $l_i(t)$ 到选定 EC,流量 $v_i(t)$ 的传播延迟和传输延迟表示为

$$d_p^{\mathrm{pro}}(i,t)=\sum_{l\in L}x_{p,l,i}(t)\cdot d_l \tag{4}$$

$$d_p^{\mathrm{trans}}(u,t)=\sum_{l\in L}x_{p,l,i}(t)\cdot Q_i/b_l \tag{5}$$

则总通信时延 $d_{\mathrm{com}}(i,t)$ 为

$$d_{\mathrm{com}}(i,t)=d_p^{\mathrm{pro}}(i,t)+d_p^{\mathrm{trans}}(i,t) \tag{6}$$

执行延迟 $d_{\mathrm{pro}}(i,t)$ 是在相应的 EC 上处理用户数据包所花费的时间。设 EC 的最大处理能力为 $f_{i,t}$(单位:Hz),处理密度为 w_i(Hz),CPU 分享率是 $\eta_{i,t}$,则处理延迟 $d_{\mathrm{pro}}(i,t)$ 表示为[9]

$$d_{\mathrm{pro}}(i,t)=Q_i\cdot\frac{w_i}{\eta_{i,t}\cdot f_{i,t}} \tag{7}$$

迁移延迟 $d_{\mathrm{mig}}(i,t)$ 和迁移成本 $C(i,t)$ 当服务 EC 切换时,迁移 SFC 会增加服务延迟和成本。受预拷贝迁移机制[10]的启发,在本文中,用户 i 请求的 SFC_i 都是在新 EC 处理 $v_i(t+1)$ 时提前激活的。然后将原 EC $v_i(t)$ 中剩余的服务数据 $Q^m_{i,t}$ 转移到目的 EC $v_i(t+1)$。在数据迁移过程中,只有当业务宕机,即数据迁移延迟过大,才会影响用户感知的延迟。因此需要确保数据迁移时间低于一定的时间阈值,以实现无缝迁移。迁移完成后,未完成处理的数据在新的 EC $v_i(t+1)$ 中继续处理。因此,总的迁移延迟包括未完成的业务数据迁移延迟和未处理数据的执行延迟。

在时隙 t 未完成的业务数据大小 $Q^m_{i,t}$ 为

$$Q^m_{i,t}=\min\{Q_{i,t},\hat{Q}_{i,t}\} \tag{8}$$

$$\hat{Q}_{i,t}=Q_{i,t}\times\left(1-\frac{\Delta t-d_{\mathrm{acc}}(i,t)-d_{\mathrm{com}}(i,t)}{d_{\mathrm{pro}}(i,t)}\right) \tag{9}$$

式(9)中:Δt 表示时间间隔的大小。

如前所述,SFC 的迁移本质上是有状态数据的传输。可以选择多条迁移路径进行 SFC_i 数据的迁移,以减少同时迁移的时间。用 $P_{i,t}$ 表示在时隙 t 的迁移路径集合。对于每条路径 $p^m_{i,t}\in P_{i,t}$,$Q^m_{i,t,p}$ 是在路径 $p^m_{i,t}$ 上传输的数据的大小,$y_{p,l,t}(t)$是一个二进制变量,表示路径 $p^m_{i,t}$ 是否使用链路 l。

$$y_{p,l,i}(t)=\begin{cases}1, & \text{迁移路径 }p^m_{i,t}\text{ 使用 }l\\ 0, & \text{否则}\end{cases} \tag{10}$$

则迁移未完成数据 $Q^m_{i,t}$ 的传播时延和传输时延可以表示为

$$\hat{d}_p^{\mathrm{pro}}(i,t)=\sum_{l\in L}y_{p,l,i}(t)\cdot d_l\quad \forall p^m_{i,t}\in P_{i,t} \tag{11}$$

$$\hat{d}_p^{\mathrm{trans}}(i,t)=Q^m_{i,t,p}/b^{\min}_{p,i,t}\quad \forall p^m_{i,t}\in P_{i,t} \tag{12}$$

式中:$b^{\min}_{p,i,t}=\min(b_l)_{l\in p^m_{i,t}}$ 表示迁移路径上链路的最小带宽。路径 $p^m_{i,t}$ 上的总迁移时间为 $\hat{d}_p(i,t)=\hat{d}_p^{\mathrm{pro}}(i,t)+\hat{d}_p^{\mathrm{trans}}(i,t)$,因此,随着用户 i 的移动,SFC_i 数据迁移的总延迟表示为

$$\hat{d}_{\mathrm{mig}}(i,t)=\max\{\hat{d}_p(i,t)\}_{p^m_{i,t}\in P_{i,t}} \tag{13}$$

未完成数据的新的执行延迟为

$$\hat{d}_{\mathrm{exe}}(i,t)=Q^m_{i,t}\cdot\frac{w_i}{\eta_{i,(t+1)}\cdot f_{i,(t+1)}}-Q^m_{i,t}\cdot\frac{w_i}{\eta_{i,t}\cdot f_{i,t}} \tag{14}$$

可以看出,当目标 EC 计算资源充足,即 $\eta_{i,(t+1)}\cdot f_{i,(t+1)}>\eta_{i,t}\cdot f_{i,t}$ 时,新的数据执行延迟 $\hat{d}_{\mathrm{exe}}(i,t)$将小于 0,这意味着 SFC_i 迁移可以减少用户请求 r_i 的执行延迟;相反,迁移将增加执行时间。

用 $z_i(t)$来指示是否触发迁移:

$$z_i(t)=\begin{cases}1, & \text{if } v_i(t)\neq v_i(t+1)\\ 0, & \text{otherwise}\end{cases} \tag{15}$$

因此,最终在时隙 t 的迁移延迟 $d_{\mathrm{mig}}(i,t)$ 表示为

$$d_{\mathrm{mig}}(i,t)=\hat{d}_{\mathrm{mig}}(i,t)\cdot z_i(t)+\hat{d}_{\mathrm{exe}}(i,t) \tag{16}$$

用 $C(i,t)$ 表示移动用户 i 的 SFC_i 迁移成本。$C(i,t)$也是 SFC_i 状态数据在路径上的传输成本,用以下公式计算:

$$C(i,t)=z_i(t)\cdot\sum_{p^m_{i,t}\in P}\sum_{l\in L}c_l\cdot Q^m_{i,t,p}\cdot y_{p,l,i}(t) \tag{17}$$

2.3 问题模型设计

基于 $t-1$ 时隙迁移的 SFC 映射,为时隙 t 需迁移的用户 i 找到一个最优的迁移路径集 $P_{i,t}$,目标是使迁移成本与用户请求 r_i 的端到端延迟之和最小。

根据上一节的描述,用户请求 r_i 在时隙 t 的总延迟为

$$D(i,t)=d_{\text{acc}}(i,t)+d_{\text{com}}(i,t)+d_{\text{exe}}(i,t)+d_{\text{mig}}(i,t) \tag{18}$$

因此,MEC 网络中的 SFC 迁移问题用公式(19)表示,约束条件为公式(20)~(25)。

$$O(S)=\alpha_1 D(i,t)+\alpha_2 C(i,t) \tag{19}$$

$$\text{s.t. C1}: x_{p,l,i}(t)\in\{0,1\}, y_{p,l,i}(t)\in\{0,1\} \tag{20}$$

$$\text{C2}: D(i,t)\leqslant D_i \tag{21}$$

$$\text{C3}: \hat{d}_{\text{mig}}(i,t)\leqslant \hat{T}_i \tag{22}$$

$$\text{C4}: \sum_{p_{i,t}^m\in P_{i,t}} Q_{i,t,p}^m \geqslant Q_{i,t}^m \tag{23}$$

$$\text{C5}: \sum_{i\in U}\sum_{p_{i,t}^m\in P_{i,t}} y_{p,l,i}(t)\cdot b_{p,i,t}^{\min}\leqslant b_l \quad \forall l\in L \tag{24}$$

$$\text{C6}: \sum_d y_{p,l,i}^{s,d}(t)-\sum_d y_{p,l,i}^{d,s}(t)=\begin{cases}1, & s\text{ 是源}\\ -1, & s\text{ 是目的地}\\ 0, & \text{其他}\end{cases} \tag{25}$$

式(19)~(25)中:$\alpha_1>0$,$\alpha_2>0$,且 $\alpha_1+\alpha_2=1$ 为两个权重参数表示延迟和成本的重要性;$\hat{T}_i$ 是无缝迁移的最大时间容忍;$Q_{i,t}$ 是在 t 时隙需要迁移的数据量;约束 C2 为服务延迟约束;约束 C3 为确保 SFC_i 的无缝迁移;约束 C4 表示所有迁移路径上的数据量之和大于待迁移 SFC_i 的总数据量;约束 C5 表示带宽容量约束;约束 C6 保证迁移路径的连续性。

3　算法设计

3.1　算法概述

给定一个 MEC 网络 $G=(B\cup V,L)$ 和一个支持 NFV 的用户请求 r_i,目标是最小化用户在边缘网络上移动时的请求延迟和网络成本。要解决这个问题,有两个挑战:一是在整个时间范围 T 中触发 SFC 迁移的时机;二是如何选择 SFC 的迁移路径。

3.2　迁移触发

通过设计合理的 SFC 迁移触发机制,可以避免不必要的 SFC 迁移(为云服务提供商节省迁移成本)。当服务请求延迟大于延迟容忍阈值 D_i 时,就会触发 SFC 跨节点迁移,而不是在移动用户 i 跨 BS 移动时就考虑迁移,从而影响移动用户的 QoS。换句话说,SFC 在时隙 t 满足 $D(i,t)>D_i$ 时迁移将被触发。

3.3　迁移算法

SFC 的迁移实际上可以看作是有状态数据的迁移。给出迁移源 EC 和目标 EC 后,可以选择多个数据迁移路径并发执行,保证服务数据的无缝迁移。为此,提出了一种基于 Dijkstra 的迁移算法。迁移算法(Migration Strategy Algorithm, MSA)的伪代码如下:

Input MEC 网络 $G=(B\cup V,L)$,源 EC m,目的 EC n,用户请求 r_i,数据 $Q_{i,t}^m$

Output 数据迁移路径的集合:$P_{i,t}$;

1 G'←通过阶段 1 推导出网络加权图;

2 初始化:$Q'_{i,t}=Q_{i,t}^m$,$b_{p,i,t}^{\min}=0$,$P_{i,t}=\text{null}$,$H=0$;

3 while $b_{p,i,t}^{\min}\cdot\hat{T}_i<Q'_{i,t}$ do

4　通过 Dijkstra 算法找到 m 和 n 之间的最短路径 $p_{i,t}^m$(阶段 2);

5　if $p_{i,t}^m$ 不存在 then

6　　return null;

7　end

8　将 $p_{i,t}^m$ 添加到 $P_{i,t}$;

9　$b_{p,i,t}^{\min}=\min(b_l)_{l\in p_{i,t}^m}^m$,$H=H+1$;

10 更新网络加权图 G'(阶段 4);

11 end

12 return $P_{i,t}$;

MSA 核心思想基于以下四个阶段:

阶段 1:得到 MEC 网络的加权图(代码第 1 行)。考虑到网络链路的两个参数(数据迁移延迟和代价)的不同性质,将这两个参数集成为一个新的边缘参数。首先将这两个参数归一化到一个统一的量级,以得到相同规格的链路传播延迟和传播代价。归一化公式表示为

$$t'_l=\frac{t_l-t_l^{\min}}{t_l^{\max}-t_l^{\min}}, c'_l=\frac{c_l-c_l^{\min}}{c_l^{\max}-c_l^{\min}} \tag{26}$$

因此,构造的新加权图 G'的边权为 $w_l=w_1t'_l+w_2c'_l$,其中 w_1 和 w_2 是两个参数的权因子。

阶段 2:找到最短路径(代码第 4~8 行)。将 Dijkstra 算法应用于阶段 1 得到的加权图 G',求出 m 与 n 之间的最短路径 $p_{i,t}^m$,并加入迁移路径集 $P_{i,t}$。

阶段 3:MSA 算法最优停止(代码第 3 行)。计算通过阶段 2 得到的路径 $p_{i,t}^m$ 的有效传输带宽,$b_{p,u,t}^{\min}=\min(b_l)_{l\in p_{u,t}^m}^m$。当满足式(27)时,算法停止,$P_{i,t}$ 为时隙 t 处的迁移路径集。

$$b_{p,i,t}^{\min}\cdot\hat{T}_i<Q'_{i,t} \tag{27}$$

注意,公式(27)中的 $Q'_{i,t}$ 为第四阶段更新后需要转移的剩余 SFC_i 数据。总数据迁移延迟可由式

(13)计算。如果式(27)不满足,继续执行算法到阶段 4。

阶段 4:更新加权图(代码第 10 行)。在这个阶段,需要更新网络图 G'。提取出经过阶段 2 的最短路径 $p_{i,t}^{m}$,即可计算出 MEC 网络上剩余的链路带宽,$b_l = b_l - b_{p,u,t}^{\min}$。当网络链路 l 的剩余带宽为 0 时,将该链路对应的边从加权图 G'中删除。同时,在经过阶段 3 的判断后,算法继续寻找迁移路径。因此,更新剩余需要迁移的 SFC_i 数据量 $Q'_{u,t}$, $Q'_{i,t} = Q'_{i,t} - b_{p,i,t}^{\min} \cdot \hat{T}_i$。随后,MSA 算法返回到阶段 2 执行。

3.4 算法时间复杂度分析

用 $|N|$表示 BSs 的数量,则 MEC 网络 G 的链路数最多为$\frac{1}{2}|N|(|N|-1)$,辅助图 G'的节点数最多为 $|N|$。Dijkstra 算法的时间复杂度为 $O(|N|^2)$。算法第 10 行主要负责更新迁移路径上链路的剩余带宽,需要更新的链路数量通常比 G'中的链路总数小得多,因此,即使在最坏的情况下,也只需 $O(\frac{1}{2}|N|(|N|-1))$复杂度。假设找到的数据迁移路径数为 H,则 MSA 算法的时间复杂度为

$$O(H \cdot (|N|^2 + \frac{1}{2}|N|(|N|-1))) = O(H \cdot |N|^2)$$

4 性能评估

4.1 仿真设置

在一台 CPU 为 i7-8550U CPU,RAM 为 16 GB 的计算机上对算法进行评估。由 Python NetworkX 2.4 网络库产生有 40 个节点和 85 条链路的 MEC 网络,并将 20 个 EC 随机分配到网络中。随机选择移动用户,每个场景运行 20 次取平均值,参考文献[11-12],设置每个 EC 的计算能力为 25 GHz。每个用户请求 SFC_i 时,EC 的最大处理能力 $f_{i,t}$ 服从[0,25]GHz 的均匀分布。设每条链路的带宽容量 b_l 为[600,1 000]Mb/s,传播延迟 d_l 为[2,4]ms,传播代价 c_l 为[10,30]单位。每个用户请求的最大延迟容忍 D_i 为 50~100 ms,无缝迁移每个请求 r_i 的时延阈值 $\hat{T}_i$ 设为 15 ms,相关参数的无线信道带宽 W 设为 20 MHz,噪声功率 $\sigma^2 = 2\times10^{-13}$ W,移动设备发射功率 P_i 设为 0.5 W。无线信道增益为 $127 + 30\lg d$,d 为临时位置与基站之间的距离。设置时延和成本的权重比例为 $\alpha_1 = \alpha_2 = 0.5$。对比算法 1 为最优延迟算法(Optimal Delay Algorithm, ODA)[7]:只考虑用户请求的延迟需求而忽略迁移成本进行迁移决策;对比算法 2 为基于贪婪的算法(Greedy-Based Algorithm, GBA):根据最小的用户请求延迟和成本选择迁移目的 EC;对比算法 3 为基于随机的算法(Random-based Algorithm, RBA):RBA 随机选择目标 EC 进行 SFC 迁移。

4.2 结果分析

平均 SFC 延迟如图 1 所示,MSA 算法的结果是最好的,ODA 的性能好于 RBA 和 GBA。随着移动用户数量的增加,SFC 请求的平均时延基本保持稳定,表明该模型和算法在不同规模场景下可以有效地改善用户体验。

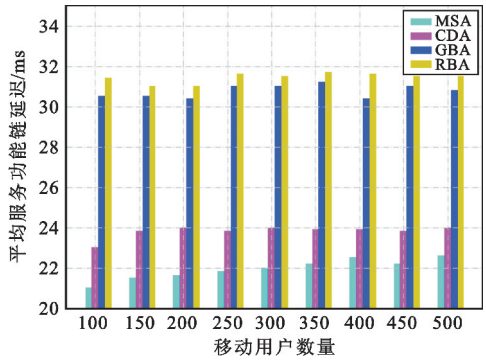

图 1 SFC 的平均延迟比较

Fig. 1 Average latency comparison of SFC

整体迁移成本如图 2 所示,与移动用户数量近似成比例。原因是用户数量越多,在相同的时间间隔内迁移的数量就越多。此外,MSA 的迁移代价最小,RBA 和 GBA 代价最大。

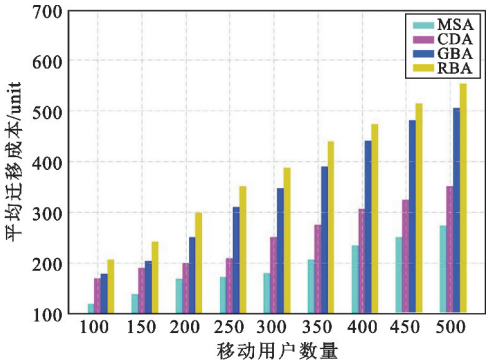

图 2 迁移代价比较

Fig. 2 Comparison of migration costs

执行时间如图 3 所示,MSA 执行时间优于 GBA 算法;与 ODA 算法相似,随着网络节点的增加,每个

算法的执行时间都会增加。

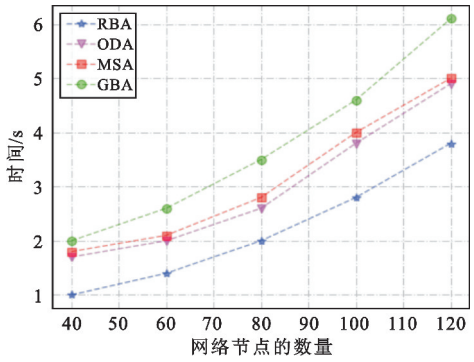

图 3　算法时间对比

Fig. 3 Algorithm time comparison

图 4 显示了权重参数 α 对 MSA 算法的平均延迟和总代价的影响。通过将 α_1 从 0 增加到 1,MSA 算法更加关注平均 SFC 延迟,从而平均 SFC 延迟减小,然而总体迁移成本会增加。因此,CSP 可以通过合理调整两个目标的权重参数来强调自己的偏好。

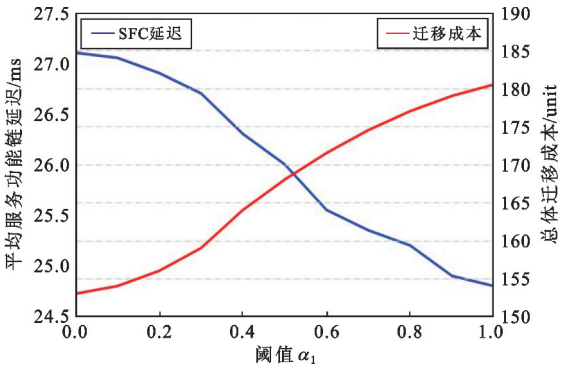

图 4　权重参数 α 对 MSA 算法的影响

Fig. 4 The impact of weight parameter α on MSA algorithm

5　结　论

本文将 MEC 网络中的 SFC 迁移问题表述为一个考虑用户移动性的数学模型。为了实现无缝迁移,提出了 MSA 算法来计算 SFC 迁移路径,目标是最小化 SFC 的平均延迟和 CSP 的迁移成本。仿真结果表明,考虑用户移动性的 MSA 算法能够在保证服务延迟的同时有效地降低网络成本。

参考文献:

[1] 韩晓非,宋青芸,韩瑞寅,等. 移动边缘计算卸载技术综述[J]. 电讯技术,2022,62(9):1368-1376.

[2] MACH P,BECVAR Z. Mobile edge computing:a survey on architecture and computation offloading[J]. IEEE Communications Surveys and Tutorials,2017,19(3):1628-1656.

[3] MAO Y,YOU C,ZHANG J,et al. A survey on mobile edge computing:the communication perspective[J]. IEEE Communications Surveys and Tutorials,2017,19(4):2322-2358.

[4] MAHADEV S,PARAMVIR B,RAMON C,et al. The case for VM-based cloudlets in mobile computing[J]. IEEE Pervasive Computing,2009,8(4):14-23.

[5] GUO Y,WANG S G,ZHOU A,et al. User allocation-aware edge cloud placement in mobile edge computing[J]. Software:Practice and Experience,2020,50(5):489-502.

[6] LI B Y,CHENG B,CHEN J L. A multi-stage approach for virtual network function migration and service function chain reconfiguration in NFV-enabled networks[C]// Proceedings of 2020 IEEE International Conference on Web Services. Beijing:IEEE,2020:207-215.

[7] CZIVA R,CHRISTOS A,DIMITRIOS P. Dynamic,latency-optimal VNF placement at the network edge[C]// Proceedings of 2018 IEEE Conference on Computer Communications. Honolulu:IEEE,2018:693-701.

[8] SONG S,LEE C G,JUN H,et al. Clustered virtualized network functions resource allocation based on context-aware grouping in 5G edge networks[J]. IEEE Transactions on Mobile Computation,2020,19(5):1072-1083.

[9] SHIN M,CHONG S,RHEE I. Dual-resource TCP/AQM for processing-constrained networks[J]. IEEE/ACM Transactions on Networking,2008,16(2):435-449.

[10] YI B,WANG X W LI K Q,et al. A comprehensive survey of network function virtualization[J]. Computer Networks,2018,133:212-262.

[11] SUN Y,ZHOU S,XU J. EMM:energy-aware mobility management for mobile edge computing in ultra dense networks[J]. IEEE Journal on Selected Areas in Communications,2017,35(11):2637-2646.

[12] PEI J,HONG P L,XUE K P,et al. Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system[J]. IEEE Transactions on Parallel Distributed System,2019,30(10):2179-2192.

作者简介:

周冬杨　男,1986 年生于重庆,2016 年获硕士学位,现为副教授,主要研究方向为普适计算机、大数据深度学习、边缘计算。

潘显兵　男,1971 年生于重庆,硕士,副教授,主要研究方向为运筹与决策分析、信息处理。