

## 一种基于联邦学习资源需求预测的虚拟网络功能迁移算法

唐 伦 吴 婷\* 周鑫隆 陈前斌

(重庆邮电大学通信与信息工程学院 重庆 400065)

(重庆邮电大学移动通信重点实验室 重庆 400065)

**摘 要:** 针对网络切片场景下时变网络流量引起的虚拟网络功能(VNF)迁移问题, 该文提出一种基于联邦学习的双向门控循环单元(FedBi-GRU)资源需求预测的VNF迁移算法。该算法首先建立系统能耗和负载均衡的VNF迁移模型, 然后提出一种基于分布式联邦学习框架协作训练预测模型, 并在此框架的基础上设计基于在线训练的双向门控循环单元(Bi-GRU)算法预测VNF的资源需求。基于资源预测结果, 联合系统能耗优化和负载均衡, 提出一种分布式近端策略优化(DPPO)的迁移算法提前制定VNF迁移策略。仿真结果表明, 两种算法的结合有效地降低了网络系统能耗并保证负载均衡。

**关键词:** 虚拟网络功能; 预测; 迁移; 深度强化学习

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2022)10-3532-09

DOI: [10.11999/JEIT210743](https://doi.org/10.11999/JEIT210743)

## A Virtual Network Function Migration Algorithm Based on Federated Learning Prediction of Resource Requirements

TANG Lun WU Ting ZHOU Xinlong CHEN Qianbin

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

(Key Laboratory of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** In order to solve the problem of virtual network function migration caused by time-varying network traffic in network slicing, a Virtual Network Function (VNF) migration algorithm based on Federated learning with Bidirectional Gate Recurrent Units (FedBi-GRU) prediction of resource requirements is proposed. Firstly, a VNF migration model of system energy consumption and load balancing is established, and then a framework based on distributed federated learning is introduced to cooperatively train the predictive model. Secondly, considering predicting the resource requirements of VNF, an online training Bidirectional Gate Recurrent Unit (Bi-GRU) algorithm on the basis of the framework is designed. Finally, on the grounds of the resource prediction results, system energy consumption optimization and load balancing are combined, and a Distributed Proximal Policy Optimization (DPPO) migration algorithm is proposed to formulate a VNF migration strategy in advance. The simulation results show that the combination of the two algorithms reduces effectively the energy consumption of the network system and ensures the load balance.

**Key words:** Virtual Network Function(VNF); Prediction; Migration; Deep reinforcement learning

收稿日期: 2021-07-27; 改回日期: 2022-03-23; 网络出版: 2022-03-30

\*通信作者: 吴婷 2721283189@qq.com

基金项目: 国家自然科学基金(62071078), 重庆市教委科学技术研究项目(KJZD-M201800601)

Foundation Items: The National Natural Science Foundation of China (62071078), The Science and Technology Research Program of Chongqing Municipal Education Commission (KJZD-M201800601)

## 1 引言

网络功能虚拟化技术(Network Function Virtualization, NFV)可将传统网络硬件资源转化为虚拟网络资源,并在通用服务器上实例化为虚拟网络功能(Virtual Network Function, VNF)处理网络业务,这实现了网络的灵活与高效运用<sup>[1]</sup>。在NFV中,网络服务请求由一系列按既定顺序排列的VNF处理,这样有序排列的VNF集合称为服务功能链(Service Function Chain, SFC)<sup>[2]</sup>。由于网络流量的动态变化,VNF的资源需求与底层物理资源的可用性不匹配导致网络性能下降,或使某些物理节点的资源利用率较低增加多余能耗开销,利用VNF迁移技术可以保证网络性能和降低网络系统能耗。

现有许多文献已经研究了VNF的迁移问题,文献[3]研究如何优化SFC重配置以响应不断变化的网络流量需求和资源限制,并基于实时的网络信息提出一种基于禁忌搜索的VNF迁移算法,然而文献[3]缺少对VNF资源需求预测使VNF迁移存在“滞后”的时延问题。文献[4]提出一种基于在线深度信念网络的VNF资源需求预测方法,其根据预测结果提前迁移资源使用过载物理节点上的VNF保证网络性能,文献[5]提出一种基于图神经网络的VNF资源需求预测算法,并基于预测结果优先制定一种基于深度Dyna-Q学习的迁移策略,但是文献[4,5]所提多任务集中预测方法不仅会增加数据集中节点的数据训练负担从而降低数据训练效率,也会由于上传数据至集中节点训练造成数据泄露,甚至可能由于集中节点的内存不足以满足大规模数据训练要求导致数据无法训练。

综上所述,本文针对VNF迁移“滞后”带来的时延问题,引入一种双向门控循环单元(Bidirectional Gated Recurrent Unit, Bi-GRU)进行VNF资源需求预测,为缓解现有集中式机器学习存在的数据训练负担问题,本文引入联邦学习(Federated Learning, FL)框架辅助Bi-GRU模型训练,并基于此提出一种联邦双向门控循环单元(FedBi-GRU)的VNF资源需求预测算法,使各个预测模型在各自物理节点上训练从而缓解集中式节点训练负担和避免数据泄露风险。基于预测结果,以最小化VNF迁移后系统能耗为目标制定VNF迁移策略,又考虑到最小化能耗极有可能使网络资源使用不均衡,使系统难以容忍未来动态流量变化造成频繁迁移的问题,本文又引入负载均衡使系统更加稳定。由于网络流量的动态变化导致VNF的迁移决策问题多维且复杂,考虑到深度强化学习在高维复杂的空间中获得较好的优化效果,本文引入一种分布式近端策略优化(Distributed Proximal Policy Optimization, DPPO)的深度强化学习算法为VNF迁移做决策。

## 2 系统模型

### 2.1 网络场景

本文考虑5G云无线接入网的上行链路切片网络场景,该场景被分为物理层、虚拟化层及网络切片层,网络切片层由一系列VNF按既定顺序排列而成的SFC集合组成,用来处理网络业务。如图1所示,通过收集网络资源使用数据建立VNF的资源需求预测模型,预知底层资源 $\eta_R$ 使用超过阈值上限 $\eta_R^u$ 或低于阈值下限 $\eta_R^d$ 的物理节点提前制定VNF

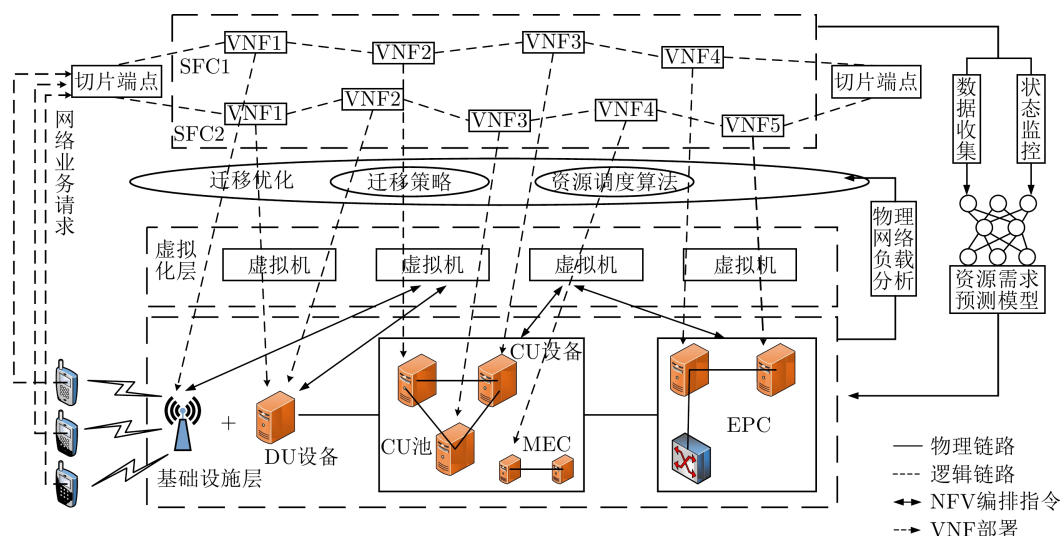


图1 网络场景图

迁移计划, 实现VNF迁移后系统能耗优化和负载均衡, 其中 $R = 1, 2, 3$ , 表示网络资源类型。

## 2.2 网络模型

### 2.2.1 物理网络

本文将物理层定义为全连接无向图 $G^P = (N^P, L^P)$ ,  $N^P$ 为底层物理节点集合,  $L^P$ 为底层物理链路集合, 令 $C_n$ 和 $M_n$ 分别表示节点 $n$ 拥有的CPU和存储资源,  $l_{nm}$ 表示物理节点 $n$ 和 $m$ 之间的物理链路, 物理链路 $l_{nm}$ 带宽容量为 $B_{n,m}$ 。

### 2.2.2 SFC

设网络切片层中有一组SFC集合, 表示为 $F = \{f_i | i = 1, 2, \dots, I\}$ 。第 $i$ 条SFC链抽象为有向图 $G_i^V = (N_i^V, L_i^V)$ ,  $N_i^V$ 为第 $i$ 条SFC上所有VNF集合,  $L_i^V$ 为第 $i$ 条SFC上所有虚拟链路集合。第 $i$ 条SFC上第 $j$ 个VNF的CPU资源需求为 $C_{i,j}^V$ , 存储资源需求为 $M_{i,j}^V$ , SFC  $i$ 上VNF $j$ 和VNF $k$ 之间的虚拟链路 $l_{jk}^V$ 带宽需求为 $B_{jk}^V$ 。设 $x_n^{i,j}$ 为VNF $j$ 是否映射到物理节点 $n$ 的二进制变量,  $x_n^{i,j} = 1$ 表示VNF $j$ 映射到物理节点 $n$ 上,  $y_{n,m}^{j,k}$ 表示虚拟链路 $l_{jk}^V$ 是否映射到物理链路 $l_{nm}$ 上,  $y_{n,m}^{j,k} = 1$ 表示 $l_{jk}^V$ 映射到 $l_{nm}$ 上。

### 2.2.3 网络能耗模型

本节研究如何优化VNF迁移后的系统能耗, 系统能耗主要来自底层物理节点能耗, 物理节点能耗包括其运行状态能耗和状态切换能耗。运行状态能耗主要为物理节点能够运行的基础能耗和负载能耗, 物理节点的负载能耗与其上的CPU资源利用率成正比<sup>[6]</sup>, 状态切换能耗主要由物理节点开断电产生<sup>[7]</sup>。因此物理节点 $n$ 的CPU资源利用率表示为

$$u_n^{\text{cpu}} = \sum_{i \in F} \sum_{j \in N_i^V} \frac{x_n^{i,j}(t) \cdot C_{i,j}^V(t)}{C_n} \quad (1)$$

设 $P_n^b$ 为物理节点 $n$ 的运行基础能耗,  $P_n^{\text{cpu}}$ 为物理节点 $n$ 的CPU资源被占满能耗, 则物理节点 $n$ 的运行能耗<sup>[8]</sup>表示为

$$P_n^{\text{ac}}(t) = Z_n(t) \cdot P_n^b + u_n^{\text{cpu}} \cdot p_n^{\text{cpu}} \quad (2)$$

$Z_n(t)$ 表示物理节点 $n$ 是否开启的二进制变量, 只要有VNF映射到物理节点 $n$ 上, 物理节点 $n$ 保持开启状态, 此时 $Z_n(t) = 1$ , 否则该物理节点进入休眠状态,  $Z_n(t) = 0$ 。由于物理节点的休眠能耗远远小于其基础运行能耗, 所以物理节点的休眠能耗可以忽略不计<sup>[9]</sup>。VNF迁移会使物理节点工作状态发生变化产生状态切换能耗<sup>[10]</sup>, 设 $S_n(t)$ 表示物理节点 $n$ 工作状态发生变化的二进制变量,  $S_n(t) = 1$ 表示物理节点 $n$ 工作状态发生变化,  $S_n(t)$ 可以表示为

$$S_n(t) = \begin{cases} 0, & Z_n(t) = Z_n(t-1) \\ 1, & Z_n(t) \neq Z_n(t-1) \end{cases} \quad (3)$$

因此, 当VNF从物理节点 $n$ 迁移到物理节点 $m$ 后, 网络系统能耗可以表示为

$$P_{\text{tot}}(t) = \sum_{m \in N^P} P_m^{\text{ac}}(t) + \sum_{m \in N^P} S_m(t) \cdot P_m^s(t) \quad (4)$$

其中,  $P_m^s$ 为节点 $m$ 的切换能耗。

### 2.2.4 负载均衡模型

均衡的网络负载让网络系统更能容忍网络流量的动态变化<sup>[11]</sup>, 本文引入负载均衡来避免网络流量动态变化造成物理节点上VNF频繁迁移, 物理节点的负载均衡能力由物理节点的网络资源方差衡量, 网络资源方差越小表明网络系统的负载均衡能力越强<sup>[12]</sup>。VNF $j$ 由物理节点 $n$ 迁移到物理节点 $m$ , 会引起物理节点 $m$ 的负载变化, 这种变化是由节点 $m$ 的CPU资源和存储资源消耗引起的, 此时物理节点 $m$ 的负载表示为

$$L_m^{\text{cpu}}(t) = \sum_{j \in N_i^V} x_m^{i,j} \cdot C_{i,j}^V(t) \quad (5)$$

$$L_m^{\text{mem}}(t) = \sum_{j \in N_i^V} x_m^{i,j} \cdot M_{i,j}^V(t) \quad (6)$$

其中,  $L_m^{\text{cpu}}$ 和 $L_m^{\text{mem}}$ 分别表示物理节点 $m$ 的CPU和存储资源负载, 则物理节点 $m$ 的CPU、存储资源负载的均值分别表示为

$$L_{\text{ave}}^{\text{cpu}}(t) = \frac{\sum_{m \in N^P} L_m^{\text{cpu}}(t) / C_m}{|N^P|} \quad (7)$$

$$L_{\text{ave}}^{\text{mem}}(t) = \frac{\sum_{m \in N^P} L_m^{\text{mem}}(t) / M_m}{|N^P|} \quad (8)$$

本文用物理节点的网络资源方差来衡量网络系统的负载均衡能力, 则物理节点 $m$ 的CPU和存储资源方差可分别表示为

$$L_{\text{var}}^{\text{cpu}}(t) = \frac{\sum_{m \in N^P} [L_m^{\text{cpu}}(t) - L_{\text{ave}}^{\text{cpu}}(t)]^2}{|N^P|} \quad (9)$$

$$L_{\text{var}}^{\text{mem}}(t) = \frac{\sum_{m \in N^P} [L_m^{\text{mem}}(t) - L_{\text{ave}}^{\text{mem}}(t)]^2}{|N^P|} \quad (10)$$

为确保CPU和存储资源在衡量网络系统负载均衡时单位统一, 本文使用负载消耗的百分比作为计量单位。因此, 网络系统的负载均衡能力 $L_{\text{total}}(t)$ 可由CPU和存储资源方差的加权和得到, 其表示为

$$L_{\text{tot}}(t) = \omega_1 L_{\text{var}}^{\text{cpu}}(t) + \omega_2 L_{\text{var}}^{\text{mem}}(t) \quad (11)$$

其中,  $\omega_1$ 和 $\omega_2$ 分别表示CPU和存储资源对网络系统负载均衡影响的权重因子, 本文假设两种资源对网络系统的影响相同, 设 $\omega_1 = \omega_2 = 0.5$ 。

### 2.3 优化目标

综上所述, 本文的优化目标为最小化VNF迁移过后的网络系统能耗和资源方差, 由于系统能耗数量级远远大于资源方差, 故对联合优化目标进行归一化, 具体表示为

$$\begin{aligned}
 & \min_{x,y,Z} \alpha \frac{P_{\text{tot}}}{P_{\text{max}}} + \beta \frac{L_{\text{tot}}}{L_{\text{max}}} \\
 & \text{s.t.} \\
 & \text{C1: } \sum_{n \in N^P} x_n^{i,j}(t) = 1, \forall i \in F, \forall j \in N_i^V \\
 & \text{C2: } \sum_{l_{nm} \in L^P} y_{n,m}^{j,k}(t) = 1, \forall i \in F, \forall l_{jk}^V \in L_i^V \\
 & \text{C3: } y_{n,m}^{j,k}(t) \leq x_n^{i,j}(t), \forall i \in F, \forall j \in N_i^V, \forall n \in N^P \\
 & \text{C4: } y_{n,m}^{j,k}(t) \leq x_m^{i,k}(t), \forall i \in F, \forall k \in N_i^V, \forall m \in N^P \\
 & \text{C5: } \sum_{i \in F} \sum_{j \in N_i^V} x_m^{i,j}(t) \cdot C_{i,j}^V(t) \leq \eta_1^{\text{up}} \cdot C_m, \forall m \in N^P \\
 & \text{C6: } \sum_{i \in F} \sum_{j \in N_i^V} x_m^{i,j}(t) \cdot M_{i,j}^V(t) \leq \eta_2^{\text{up}} \cdot M_m, \forall m \in N^P \\
 & \text{C7: } \sum_{i \in F} \sum_{l_{jk}^V \in L_i^V} y_{n,m}^{j,k}(t) \cdot B_{j,k}^V(t) \leq \eta_3^{\text{up}} \cdot B_{n,m}, \forall j, k \in N_i^V, \forall l_{nm} \in L^P \\
 & \text{C8: } \sum_{i \in F} \sum_{j \in N_i^V} x_m^{i,j}(t) \geq 1, \text{ 则 } z_m(t) = 1, \text{ 否则 } z_m(t) = 0, \forall m \in N^P \\
 & \text{C9: } \sum_{i \in F} \sum_{j \in N_i^V} \sum_{n,m \in N^P} y_{n,m}^{j,k}(t) \cdot T(n,m) + \sum_{i \in F} \sum_{j \in N_i^V} \sum_{n \in N^P} x_n^{i,j}(t) \cdot T_n^{i,j} \leq T_{\text{tot}}
 \end{aligned} \tag{12}$$

其中,  $\alpha$ 和 $\beta$ 为调节优化目标的权重因子, 本文取 $\alpha = \beta = 0.5$ ,  $P_{\text{max}}$ 为所有物理节点开启能耗之和,  $L_{\text{max}}$ 为网络系统的最大资源方差,  $\eta$ 为网络资源使用阈值, C1确保每个VNF只能映射到1个物理节点上, C2确保相邻VNF之间的虚拟链路只能映射到1条物理链路上, C3和C4确保虚拟链路重新映射时, 其上的VNF必须映射到虚拟链路所在物理链路对应的物理节点上, C5~C7分别表示物理节点的CPU、存储和物理链路的带宽资源限制, C8表示物理节点有VNF映射时, 该物理节点处于开启状态, C9表示SFC的端到端时延限制,  $T(n,m)$ 为物理链路 $l_{nm}$ 的传输时延,  $T_n^{i,j}$ 为节点 $n$ 的资源处理时延,  $T_{\text{tot}}$ 为SFC最大端到端时延。

## 3 基于FedBi-GRU的VNF资源需求预测算法

### 3.1 基于在线训练的Bi-GRU预测算法

本文首先利用Bi-GRU算法对VNF的资源需求进行预测以便制定后续迁移计划, Bi-GRU是一种双向的门控循环单元(Gated Recurrent Unit, GRU), 其计算原理与GRU相同<sup>[13]</sup>。

基于Bi-GRU的资源预测算法过程为: 首先,

物理节点采集其上VNF的CPU、存储及带宽资源历史数据。对于SFC $i$ 上的VNF $j$ , 其资源需求特征可以表示为 $r_j = \{r_j^C, r_j^M, r_j^B\}$ , 其中 $r_j^C, r_j^M, r_j^B$ 分别为VNF $j$ 的CPU、存储、带宽资源特征。其次, 将VNF $j$ 资源需求特征 $r_j$ 输入Bi-GRU中训练和进行资源需求预测。

然而由于网络流量实时变化, 基于VNF历史资源使用数据训练得到的Bi-GRU模型对VNF未来网络资源需求的预测可能存在较大误差, 因此本文引入一种在线训练的方法来微调Bi-GRU模型, 该预测方法结合离线训练和在线训练联合优化Bi-GRU模型。

离线训练阶段利用大量VNF网络资源使用历史数据进行Bi-GRU模型的预训练, 通过采集VNF $j$ 的资源历史使用数据可得一系列数据特征样本集合 $o = \{o_1, o_2, \dots, o_s, \dots\}$ , 其中 $o_s$ 为VNF $j$ 的第 $s$ 个特征样本集,  $o_s = \{r_j(t), r_j(t-1), \dots, r_j(t-\varepsilon+1)\}$ ,  $\varepsilon$ 表示特征样本集 $o_s$ 的长度。

在线训练阶段将离线训练所得模型参数作为训练的初始参数, 并采集实时的VNF资源使用数据对Bi-GRU预测模型进行反向微调优化。在线训练样本更新机制采用滑动窗口的方式更新, 即每时刻



增添新的资源样本特征  $r_j = \{r_j^C, r_j^M, r_j^B\}$ , 就删除最旧时刻的资源样本特征, 保持滑动窗口长度  $\varepsilon$  不变。

### 3.2 联邦多任务训练过程

网络流量流经一条SFC, 其上的VNF会存在丰富关联信息, 而上述预测方法只针对单个VNF的资源需求预测, 这忽略了SFC上VNF之间信息的关联性, 致使资源预测模型泛化性不足。由此文献[4,5]针对此问题提出一种集中式多任务模型训练方法, 然而这种集中式模型训练方式存在数据泄露风险、数据集中节点数据存在训练负担, 甚至会由于集中节点的内存不足以支持大规模数据训练导致数据无法训练。

因此, 本文针对上述问题, 提出一种基于联邦学习的分布式VNF网络资源需求预测框架——FedBi-GRU, 该框架使VNF在各自部署的物理节点上根据本地数据进行Bi-GRU预测模型训练, 省去数据上传过程, 只需上传模型训练参数, 并在SDN控制器处对各个VNF模型参数进行聚合以保证SFC上各个VNF的丰富关联信息, 训练过程如下所示:

首先SDN控制器下发初始参数  $w_0$  给SFC $i$ 上的各个VNF, 各个VNF结合其本地收集的样本集合  $O_j$ 、真实输出值  $Y_j$ 、隐藏层状态  $H$  及初始参数  $w_0$  进行在线Bi-GRU模型训练, 随后各个VNF将训练所得参数  $w_1, w_2, \dots, w_j$  上传至SDN控制器进行联邦平均[14]得到全局模型参数  $\bar{w}(t)$ ,  $\bar{w}(t)$  可表示为

$$\bar{w}(t) = \frac{\sum_{j=0}^{K_i} D_j w_j(t)}{D} \quad (13)$$

其中,  $K_i$  为SFC $i$ 上VNF的个数,  $D_j$  为第 $j$ 个VNF训练样本数量,  $D$  为SFC $i$ 上所有VNF的训练样本总数。随后SDN控制器将全局模型参数  $\bar{w}(t)$  下发给SFC $i$ 上的所有VNF进行本地Bi-GRU模型更新。以VNF $j$ 为例, 其本地梯度  $g_{j,w}$  和参数  $w_j(t+1)$  更新方式为

$$g_{j,w} = \nabla \frac{1}{D_j} \sum_{j=1}^{D_j} (f_j(r_j) - y_j)^2 \quad (14)$$

$$w_j(t+1) = \bar{w}(t) - \frac{\lambda}{D_j} \sum_{j=1}^{D_j} g_{j,w}(t) \quad (15)$$

其中,  $\lambda$  为Bi-GRU模型的学习率。

## 4 基于DPPO的VNF迁移算法

根据上述FedBi-GRU算法得到VNF资源需求预测结果, 计算底层物理节点资源利用率, 根据资源使用阈值判定过载或轻载的物理节点并根据优化目

标制定迁移计划。因此, 本节提出一种基于DPPO的深度强化学习算法求解上述迁移优化目标。该算法首先将上述优化目标转化成马尔可夫决策过程(Markov Decision Process, MDP)模型, 其次通过DPPO算法求解该MDP模型得到优化目标函数的近似最优解。

### 4.1 MDP模型

本文将优化目标转化为离散时间马尔可夫决策过程(Discrete Time Markov Decision Process, DTMDP), DTMDP被定义为一个4元组  $(S, A, P, R)$ , 分别表示状态空间、动作空间、状态转移概率和奖励函数。定义  $s(t) = \{\psi(t), \xi(t), C(t), M(t), B(t)\} \in S$  表示时刻 $t$ 的网络系统状态空间, 其中 $\psi(t)$ 表示物理节点的状态空间,  $\xi(t)$ 表示物理链路状态空间,  $C(t), M(t), B(t)$ 分别表示VNF的CPU、存储和虚拟链路资源需求状态空间。动作空间 $A$ 表示VNF可采取的映射动作集合, 在时刻 $t$ VNF的映射动作  $a(t) \in A$ 。在当前状态  $s(t) \in S$  采取动作  $a(t) \in A$  后, 转移到下一时刻状态  $s(t+1)$  的概率, 表示为  $p(s(t+1)|s(t), a(t))$ 。

系统在状态  $s(t)$  采取动作  $a(t)$  会生成动作映射策略  $\pi$ , 如果动作映射策略  $\pi$  满足式(12)的约束, 则根据优化目标可得瞬时奖励  $r(t)$ , 表示为

$$r(t) = -\alpha \frac{P_{\text{tot}}}{P_{\text{max}}} - \beta \frac{L_{\text{tot}}}{L_{\text{max}}} \quad (16)$$

若不满足式(12)的约束, 即时奖励  $r(t) = -1/\varepsilon$ , 其中 $\varepsilon$ 为惩罚因子, 其为无穷小的正实数。系统通过值函数  $V^\pi$  来评估当前时刻VNF映射策略  $\pi$  的好坏,  $V^\pi$  表示为

$$V^\pi(s(t), a(t)) = E[r(t) + \gamma_v V^\pi(s(t+1), a(t+1)) \cdot |s(t), a(t)] \quad (17)$$

其中,  $\gamma_v \in [0, 1]$  为折扣因子, 表示未来的奖励对当前奖励的重要程度。则VNF迁移的最优策略可表示为

$$\pi^* = \arg \max_a V^\pi(s, a), \forall s, a \quad (18)$$

### 4.2 DPPO训练过程

本文结合了A3C的异步优势和近端策略优化(Proximal Policy Optimization, PPO)的收敛性能优势, 提出一种基于DPPO的VNF迁移算法。DPPO算法利用A3C多智能体训练的框架, 拥有1个全局网络和多个智能体网络, 全局网络和多智能体网络拥有相同的PPO结构。各个智能体被放置在不同的线程中独立进行随机模型训练, 并定期向全局PPO网络推送损失函数的梯度, 全局PPO网络将所收集梯度用来更新全局参数, 智能体可从全局网

络下载参数。DPPO采用了PPO新旧动作策略限制更新幅度的方式，因此DPPO需要考虑新旧动作策略的放置，本文将新生成的动作策略放入全局PPO网络中，而智能体的本地PPO网络则采用旧的动作策略与环境进行交互更新网络参数，详细过程如下所述：

本文以每条SFC作为一个智能体。各个智能体与网络系统交互并从环境中获得物理节点和链路状态信息、VNF的CPU、存储和带宽需求信息。基于环境反馈信息，智能体采取相应的VNF映射动作，并计算瞬时奖励。PPO网络结构中的Critic部分是基于状态值函数和优势函数评估Actor网络输出VNF映射策略的好坏。对于智能体 $n$ 来说，它的状态值函数 $V(s_n(t))$ 表示为

$$V(s_n(t)) = E_{\pi(s(t)|a(t))}(r_n(t) + \gamma_v V(s_n(t+1))) \quad (19)$$

DPPO作为一种策略梯度强化学习方法，其引入泛化优势函数估计评估当前策略的合理性<sup>[15]</sup>，优势函数表示为

$$A(s_n(t), a_n(t)) = \delta_n(t) + (\gamma_v \lambda) \delta_n(t+1) + \dots + (\gamma_v \lambda)^{(T-t+1)} \delta_n(T-1) \quad (20)$$

其中， $T$ 为一轮训练中的迭代步长， $\lambda \in (0, 1)$ 为衰减因子， $\delta_n(t) = r_n(t) + \gamma_v V(s_n(t+1)) - V(s_n(t))$ 为时序差分误差， $r_n(t)$ 为智能体 $n$ 在时刻 $t$ 的瞬时奖励。智能体 $n$ 的Critic网络评估过程为最小化时序差分误差的损失函数，其损失函数表示为

$$L(\theta_c^n) = r_n(t) + \gamma_v r_n(t+1) + \dots + \gamma_v^{T-t+1} r_n(T-1) + \gamma_v^{T-1} V(s_n(T+1)) - V(s_n(T))^2 \quad (21)$$

DPPO网络智能体的本地网络利用PPO新旧策略的比率限制动作策略更新幅度，使得Actor网络动作更新策略不会太大，也使网络能在较大的学习率下稳健训练。PPO网络的Actor部分在当前状态会以一定概率产生一个策略 $\pi(s_n(t)|a_n(t), \theta_a^n)$ ，设 $\eta(\theta_a^n)$ 为智能体 $n$ Actor网络中新旧策略的比率，可以表示为

$$\eta(\theta_a^n) = \frac{\pi(s_n(t)|a_n(t), \theta_a^n)}{\pi_{old}(s_n(t)|a_n(t), \theta_a^n)} \quad (22)$$

基于裁剪优化目标的PPO算法性能效果优于自适应散度惩罚系数的PPO算法<sup>[16]</sup>，因此本文采用裁剪优化目标的PPO算法以避免新的VNF映射策略严重偏离旧的VNF映射策略。基于裁剪优化目标的PPO算法的损失目标函数为

$$L_{cut}(\theta_a^n) = E_t[\min(\eta(\theta_a^n) A(s_n(t), a_n(t)), \text{cut}(\eta(\theta_a^n), 1 - \sigma, 1 + \sigma) A(s_n(t), a_n(t)))] \quad (23)$$

参照文献<sup>[15]</sup>，超参数 $\sigma$ 设为0.2，使新旧策略

的变化率在0.8~1.2。如果新旧策略的比率 $\eta(\theta_a^n)$ 落在区间 $(1 - \sigma, 1 + \sigma)$ 外面，则损失函数需要被裁剪。裁剪方式为：若 $\eta(\theta_a^n) > 1 + \sigma$ ， $\eta(\theta_a^n)$ 取值为 $1 + \sigma$ ；若 $\eta(\theta_a^n) < 1 - \sigma$ ， $\eta(\theta_a^n)$ 取值为 $1 - \sigma$ ；若 $\eta(\theta_a^n)$ 在 $(1 - \sigma, 1 + \sigma)$ 内，则取本来的 $\eta(\theta_a^n)$ 值。智能体 $n$ 本地Critic和Actor网络的损失函数用于更新全局网络的相应参数，全局Critic网络梯度 $\Delta\theta_c$ 和参数 $\theta_c$ 更新方式为

$$\Delta\theta_c = \Delta\theta_c + \frac{\partial L(\theta_c^n)}{\partial \theta_c^n} \quad (24)$$

$$\theta_c = \theta_c + \varepsilon_c \Delta\theta_c \quad (25)$$

其中， $\theta_c^n$ 为智能体 $n$ 的Critic网络参数， $\varepsilon_c$ 为全局Critic网络的学习率。全局Actor网络的梯度 $\Delta\theta_a$ 和参数 $\theta_a$ 的更新方式为

$$\Delta\theta_a = \Delta\theta_a + \frac{\partial L_{cut}(\theta_a^n)}{\partial \theta_a^n} \quad (26)$$

$$\theta_a = \theta_a + \varepsilon_a \Delta\theta_a \quad (27)$$

其中， $\theta_a^n$ 为智能体 $n$ 的Actor网络参数， $\varepsilon_a$ 为全局Actor网络的学习率。待一轮训练完毕，将全局网络参数 $\theta_c^n$ 和 $\theta_a^n$ 代替各个智能体旧的网络参数进行下一轮的训练，始终保证全局网络使用新的策略，智能体网络使用旧的策略。

基于DPPO的VNF迁移算法如表1所示。

## 5 仿真与性能分析

### 5.1 仿真参数设置

本文针对VNF资源需求预测和迁移方案进行了实验仿真，有关网络的仿真参数如表2所示。

### 5.2 VNF资源需求预测

本文将所提FedBi-GRU算法与多任务Bi-GRU、单任务Bi-GRU预测算法进行了性能比较，这3种算法都在真实数据集archive<sup>[17-19]</sup>上进行评估。本文以预测一条SFC上4个VNF的CPU资源需求为例，4个VNF分别使用10000条CPU数据，训练集与测试集按7:3进行。Bi-GRU网络选用隐藏层数为2，隐藏单元为20，时间步长为5，学习率为0.005的结构。多任务Bi-GRU在两层Bi-GRU的基础上添加回归层sigmoid<sup>[4,5]</sup>进行集中式训练，3种算法最大迭代次数为300。本文使用均方根误差(Root Mean Square Error, RMSE)衡量3种算法的预测性能，RMSE公式为

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (28)$$

其中， $y_t$ 和 $\hat{y}_t$ 分别代表真实值和预测值，如果RMSE的值越小则表明算法的预测精度越高。

表 1 基于DDPO的VNF迁移算法

|   |
|---|
| 输入: VNF的资源需求预测结果 $r_{t+1} = \{r_{t+1}^C, r_{t+1}^M, r_{t+1}^B\}$ , 物理网络图 $G^P = (N^P, L^P)$ , SFC网络图 $G_i^V = (N_i^V, L_i^V)$                                       |
| 输出: VNF映射策略 $\pi$   |
| (1) 根据VNF的资源需求预测结果, 计算各个物理节点的资源利用率 $\eta_R$   |
| (2) <b>if</b> $\eta_R \leq \eta_R^d \& \eta_R \geq \eta_R^{up}$ <b>then</b>   |
| (3) 初始化全局参数 $(\theta_c, \theta_a)$ , 局部参数 $(\theta_c^n, \theta_a^n)$ , 全局PPO网络最大迭代次数 $K_{max}$ , 局部PPO网络最大迭代次数 $M$ , 线程数 $N$ , 学习率 $(\varepsilon_c, \varepsilon_a)$ |
| (4) <b>for</b> thread = 1, 2, ..., $N$ <b>do</b>  |
| (5) <b>for</b> episode = 1, 2, ..., $M$ <b>do</b>   |
| (6)         从本地Actor网络的策略 $\pi(s_n(t)   a_n(t), \theta_a^n)$ 中选取映射动作 $a(t)$   |
| (7) <b>if</b> $\eta_1 \in (\eta_1^d, \eta_1^{up}) \& \eta_2 \in (\eta_2^d, \eta_2^{up}) \& \eta_3 \in (\eta_3^d, \eta_3^{up}) \& T \leq T_{tot}$ <b>then</b>        |
| (8)             执行动作 $a(t)$ , 根据式(16)得到瞬时奖励 $r(t)$ , 并转移到状态 $s(t+1)$  |
| (9)             从本地Actor网络获得优势函数 $A(s_n(t), a_n(t))$  |
| (10) <b>else</b>  |
| (11)             式(16)瞬时奖励 $r(t) = -1/\varepsilon$ , 从本地Actor网络重新选取动作 $a(t)$  |
| (12) <b>end if</b>  |
| (13) <b>end for</b>   |
| (14)     根据式(24)更新全局PPO的Critic网络累计梯度 $\Delta\theta_c$   |
| (15)     根据式(26)更新全局PPO的Actor网络累计梯度 $\Delta\theta_a$  |
| (16)     将 $\Delta\theta_c$ 和 $\Delta\theta_a$ 推送至全局PPO网络进行异步更新   |
| (17) $\theta_c \leftarrow \theta_c + \varepsilon_c \Delta\theta_c, \theta_a \leftarrow \theta_a + \varepsilon_a \Delta\theta_a$                                     |
| (18) <b>end for</b>   |
| (19)     同步全局PPO网络参数至本地PPO网络参数: $\theta_c^n = \theta_c, \theta_a^n = \theta_a$  |
| (20)     继续执行步骤4—步骤17   |
| (21) <b>until</b> $K \geq K_{max}$  |
| (22) <b>end if</b>  |

表 2 仿真参数

| 仿真参数        | 描述                 | 取值                      |
|-------------|--------------------|-------------------------|
| $N^P$       | 物理节点数量             | 22                      |
| $P_n^b$     | 物理节点待机能耗           | Uniform[100,150](W)     |
| $P_n^{cpu}$ | 物理节点CPU满载能耗        | Uniform[250,300](W)     |
| $P_m^s$     | 物理节点状态切换能耗         | Uniform[15,25](W)       |
| $C_n$       | 物理节点CPU资源容量        | Uniform[200,300](units) |
| $M_n$       | 物理节点存储资源容量         | Uniform[300,400](Mbps)  |
| $B_{nm}$    | 物理链路 $l_{nm}$ 带宽容量 | Uniform[80,100](Mbps)   |
| $F$         | SFC集合数量            | 30个                     |
| $N_i^V$     | VNF集合长度            | Uniform[3,5](个)         |
| $T_{tot}$   | SFC端到端时延限制         | 30 ms                   |

本文截取SFC上的一个VNF的30个CPU资源预测数据进行3种算法的对比, 如图2和图3所示, 不难看出FedBi-GRU与多任务Bi-GRU对VNF的CPU资源需求预测效果相差不大, 而单任务Bi-GRU的预测效果最差。这是因为多任务Bi-GRU和FedBi-GRU使多个相关训练任务共享VNF之间的

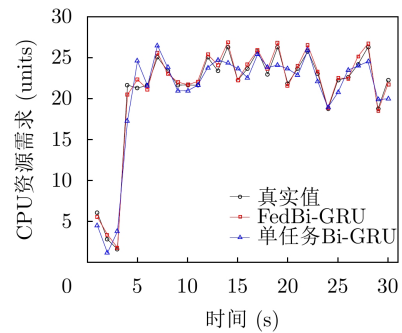


图 2 FedBi-GRU与单任务Bi-GRU预测对比

丰富关联信息, 在提高预测精度的同时也提升了Bi-GRU模型的泛化性。但是FedBi-GRU作为一种分布式机器学习方法, 不仅减轻了数据集中节点的训练负担也提高了数据训练效率, 同时避免了内存短缺问题, 因此FedBi-GRU算法在VNF的资源需求预测上具有可行性。

### 5.3 VNF迁移算法

基于DPPO的VNF迁移算法, 线程设置为4,  $K_{max} = M = 500$ , 本地PPO网络学习步长为50,

折扣因子 $\gamma_v = 0.99$ , Actor和Critic学习率都取0.001。

从图4、图5可以看出当CPU资源使用阈值上限为0.5, 下限为0.1时, 网络系统的能耗最低和网络资源方差最小, 这是因为随着CPU资源使用阈值的下降, 物理节点迸发的VNF迁移数量也越多, DPPO迁移算法能有效利用网络系统中物理节点的CPU资源, 使整个网络系统的能耗和网络资源方差越小, 网络系统的负载均衡能力越好。

本文将所提DPPO算法与文献[4]所提基于禁忌搜索的VNF迁移算法(TDMA)和A3C算法做了对比, 图6和图7展示了不同算法在CPU资源使用阈值上下限为0.5, 0.1时的对比, 由此可知本文所提DPPO迁移算法在网络系统能耗和负载均衡方面取得了更好的效果。TDMA迁移算法在能耗和负载均衡方面效果较差, 这是因为TDMA算法为一种

启发式算法, 随着SFC数量的增加, VNF映射的解搜索空间变得大而复杂, 导致TDMA算法极易陷入局部最优解, 使得更多迁移中的VNF映射到其邻域物理节点, 使整个网络系统的资源利用不均衡, 所以其能耗和网络资源方差相对其他算法来说较大。A3C与DPPO算法利用多个智能体全面探索网络环境, 能寻找更丰富多样的VNF映射解空间, 因此随着SFC数量增多, VNF数量映射动作增多时, 能够更加有效利用网络资源, 使得物理节点资源使用更加均衡, 更加节省网络系统的能耗, 所以两种算法的性能优于单智能体的PPO算法。DPPO算法的基础框架采用了PPO架构, A3C算法采用的是演员-评论家(Actor-Critic, AC)学习架构, PPO作为改进的AC算法, 其利用新旧动作策略限制了VNF映射策略的幅度, 使得VNF重映射更加稳定, 所以提高了DPPO算法在VNF重映射的性能。

## 6 结束语

本文研究了网络切片环境下的VNF迁移问题, 首先针对VNF迁移“滞后”性问题, 本文提出一种基于分布式联邦学习的双向门控循环单元预测VNF的资源需求问题, 并基于预测结果提出了一种分布式近端策略优化算法提前制定VNF迁移计划以保证网络服务的连续性。仿真结果表明, 本文所提分布式资源预测机制有较高的预测精度。另外, 本文所提分布式近端策略优化算法在降低网络系统能耗和保证负载均衡方面也起到了较好的作用。

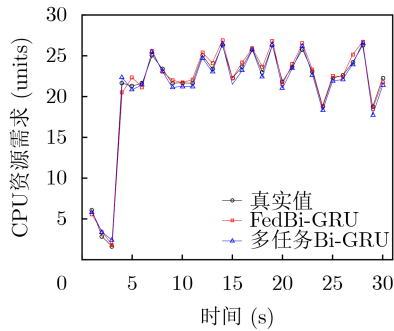


图3 FedBi-GRU与多任务Bi-GRU预测对比

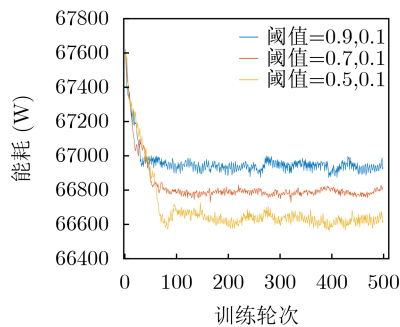


图4 不同CPU阈值的网络系统能耗

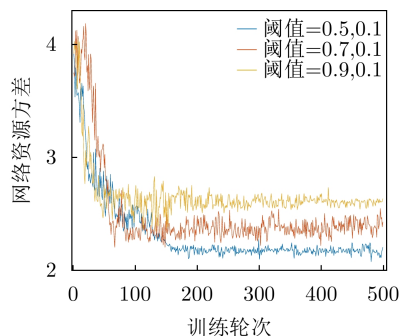


图5 不同CPU阈值的网络资源方差

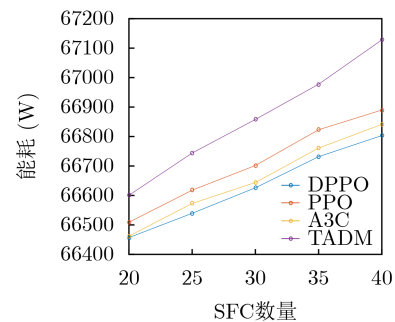


图6 网络系统能耗对比

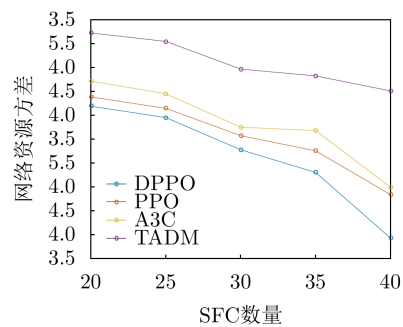


图7 网络资源方差对比



## 参 考 文 献

- [1] LI Defang, HONG Peilin, XUE Kaiping, *et al.* Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy[J]. *IEEE Transactions on Network and Service Management*, 2019, 16(4): 1651–1664. doi: [10.1109/TNSM.2019.2936505](https://doi.org/10.1109/TNSM.2019.2936505).
  - [2] QU Kaige, ZHUANG Weihua, YE Qiang, *et al.* Dynamic flow migration for embedded services in SDN/NFV-enabled 5G core networks[J]. *IEEE Transactions on Communications*, 2020, 68(4): 2394–2408. doi: [10.1109/TCOMM.2020.2968907](https://doi.org/10.1109/TCOMM.2020.2968907).
  - [3] LIU Yicen, LU Hao, LI Xi, *et al.* An approach for service function chain reconfiguration in network function virtualization architectures[J]. *IEEE Access*, 2019, 7: 147224–147237. doi: [10.1109/ACCESS.2019.2946648](https://doi.org/10.1109/ACCESS.2019.2946648).
  - [4] TANG Lun, HE Xiaoyu, ZHAO Peipei, *et al.* Virtual network function migration based on dynamic resource requirements prediction[J]. *IEEE Access*, 2019, 7: 112348–112362. doi: [10.1109/ACCESS.2019.2935014](https://doi.org/10.1109/ACCESS.2019.2935014).
  - [5] LIU Yicen, LU Yu, LI Xi, *et al.* On dynamic service function chain reconfiguration in IoT networks[J]. *IEEE Internet of Things Journal*, 2020, 7(11): 10969–10984. doi: [10.1109/JIOT.2020.2991753](https://doi.org/10.1109/JIOT.2020.2991753).
  - [6] HUANG Yuzhe, XU Huahu, GAO Honghao, *et al.* SSUR: An approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center[J]. *IEEE Transactions on Green Communications and Networking*, 2021, 5(2): 670–681. doi: [10.1109/TGCN.2021.3067374](https://doi.org/10.1109/TGCN.2021.3067374).
  - [7] DAYARATHNA M, WEN Yonggang, and FAN Rui. Data center energy consumption modeling: A survey[J]. *IEEE Communications Surveys & Tutorials*, 2015, 18(1): 732–794. doi: [10.1109/COMST.2015.2481183](https://doi.org/10.1109/COMST.2015.2481183).
  - [8] ERAMO V, AMMAR M, and LAVACCA F G. Migration energy aware reconfigurations of virtual network function instances in NFV architectures[J]. *IEEE Access*, 2017, 5: 4927–4938. doi: [10.1109/ACCESS.2017.2685437](https://doi.org/10.1109/ACCESS.2017.2685437).
  - [9] HAN Zhenhua, TAN Haisheng, WANG Rui, *et al.* Energy-efficient dynamic virtual machine management in data centers[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(1): 344–360. doi: [10.1109/TNET.2019.2891787](https://doi.org/10.1109/TNET.2019.2891787).
  - [10] ZHANG Zhongbao, CAO Huafeng, SU Sen, *et al.* Energy aware virtual network migration[J]. *IEEE Transactions on Cloud Computing*, 2022, 10(2): 1173–1189. doi: [10.1109/TCC.2020.2976966](https://doi.org/10.1109/TCC.2020.2976966).
  - [11] GUO Zehua, XU Yang, LIU Yafeng, *et al.* AggreFlow: Achieving power efficiency, load balancing, and quality of service in data center networks[J]. *IEEE/ACM Transactions on Networking*, 2020, 29(1): 17–33. doi: [10.1109/TNET.2020.3026015](https://doi.org/10.1109/TNET.2020.3026015).
  - [12] LI Biyi, CHENG Bo, LIU Xuan, *et al.* Joint resource optimization and delay-aware virtual network function migration in data center networks[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(3): 2960–2974. doi: [10.1109/TNSM.2021.3067883](https://doi.org/10.1109/TNSM.2021.3067883).
  - [13] ZHANG Kunpeng, WU Lan, ZHU Zhaoju, *et al.* A multitask learning model for traffic flow and speed forecasting[J]. *IEEE Access*, 2020, 8: 80707–80715. doi: [10.1109/ACCESS.2020.2990958](https://doi.org/10.1109/ACCESS.2020.2990958).
  - [14] LIU Yi, JAMES J J Q, KANG Jiawen, *et al.* Privacy-preserving traffic flow prediction: A federated learning approach[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 7751–7763. doi: [10.1109/JIOT.2020.2991401](https://doi.org/10.1109/JIOT.2020.2991401).
  - [15] ZHANG Zhenyu, LUO Xiangfeng, LIU Tong, *et al.* Proximal policy optimization with mixed distributed training[C]. The 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, USA, 2019: 1452–1456. doi: [10.1109/ICTAI.2019.00206](https://doi.org/10.1109/ICTAI.2019.00206).
  - [16] SCHULMAN J, WOLSKI F, DHARIWAL P, *et al.* Proximal policy optimization algorithms[J]. arXiv preprint arXiv: 1707.06347, 2017.
  - [17] BEN YAHIA I G, BENDRISS J, SAMBA A, *et al.* CogNitive 5G networks: Comprehensive operator use cases with machine learning for management operations[C]. 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 2017: 252–259. doi: [10.1109/ICIN.2017.7899421](https://doi.org/10.1109/ICIN.2017.7899421).
  - [18] BENDRISS J, BEN YAHIA I G, and ZEGHLACHE D. Forecasting and anticipating SLO breaches in programmable networks[C]. 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 2017: 127–134. doi: [10.1109/ICIN.2017.7899402](https://doi.org/10.1109/ICIN.2017.7899402).
  - [19] BENDRISS J. Cognitive management of SLA in software-based networks[D]. [Ph. D. dissertation], Institut National des Télécommunications, 2018.
- 唐 伦: 男, 教授, 博士, 研究方向为下一代无线通信网络、异构蜂窝网络、软件定义无线网络等。
- 吴 婷: 女, 硕士生, 研究方向为5G网络切片、服务功能链部署和重配置、机器学习算法。
- 周鑫隆: 男, 硕士生, 研究方向为网络切片、资源分配、深度学习算法。
- 陈前斌: 男, 教授, 博士生导师, 研究方向为个人通信、多媒体信息处理与传输、异构蜂窝网络等。

责任编辑: 余 蓉