

联合联邦学习与深度强化学习的 并行服务功能链部署算法

刘光远^{1,2,3}, 杜婕^{1,2,3}, 庞紫园^{1,2,3}

(1. 石家庄铁道大学信息科学与技术学院, 050043, 石家庄; 2. 河北省电磁环境效应与信息处理重点实验室, 050043, 石家庄; 3. 石家庄市人工智能重点实验室, 050043, 石家庄)

摘要: 针对多域边缘云网络中并行服务功能链(SFC)的动态部署问题,构建了一种优化的SFC并行结构,提出一种联合联邦学习(FedAvg)与深度强化学习(DRL)的新算法——FA-D3QN-PER。该方法解决了现有的单一DRL算法和集中式决策框架在解决SFC切分和部署时存在的资源分配不均和隐私泄露问题,通过允许各域内的智能体独立训练,并利用FedAvg共享模型参数,在保护数据隐私的同时优化全局策略。在部署阶段,对混合SFC的并行结构进行分析和优化;根据优化结果将优化后的混合SFC合理切分成若干个子链,并将其分配给合适的边缘域;将各子链中的虚拟网络功能(VNF)映射至目标域内的物理节点上。仿真结果表明,FA-D3QN-PER方法具有稳定性强、收敛速度快等特点,能够显著提高SFC部署的接受率,同时有效减少平均延迟和总成本,相较于FA-DQN、DFSC和MuL算法,FA-D3QN-PER算法将接受率提高了11.6%,平均延迟和总成本分别减少了17%和18.56%。

关键词: 多域边缘云网络;并行服务功能链;动态部署;联邦学习;深度强化学习

中图分类号: TP393.0 **文献标志码:** A

DOI: 10.7652/xjtub202509011 **文章编号:** 0253-987X(2025)09-0110-12

A Parallel Service Function Chain Deployment Method Combining Federated Learning and Deep Reinforcement Learning

LIU Guangyuan^{1,2,3}, DU Jie^{1,2,3}, PANG Ziyuan^{1,2,3}

(1. School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China;

2. Hebei Provincial Key Laboratory of Electromagnetic Environmental Effects and Information Processing, Shijiazhuang 050043, China; 3. Shijiazhuang Key Laboratory of Artificial Intelligence, Shijiazhuang 050043, China)

Abstract: To address the dynamic deployment of parallel service function chains (SFCs) in multi-domain edge cloud networks, this paper constructs an optimized parallel SFC architecture and proposes a novel approach combining federated learning (FedAvg) and deep reinforcement learning (DRL), termed FA-D3QN-PER. This method resolves the issues of imbalanced resource allocation and privacy leakage inherent in existing single-agent DRL and centralized decision-making frameworks when handling SFC partitioning and deployment. By enabling independent training of agents within each domain and leveraging FedAvg for model parameter sharing, it optimizes global strategies while preserving data privacy. The deployment phase

involves the analysis and optimization of the hybrid SFC parallel structure; intelligent partitioning of the optimized hybrid SFC into sub-chains and their allocation to suitable edge domains; mapping of virtual network functions (VNFs) within each sub-chain to physical nodes in target domains. Simulation results demonstrate that the FA-D3QN-PER method exhibits strong stability and fast convergence, significantly improving SFC deployment acceptance rates while effectively reducing average latency and total costs. Compared to FA-DQN, DFSC, and MuL, the FA-D3QN-PER method increases acceptance rates by 11.6%, while reducing average latency and total costs by 17% and 18.56%, respectively.

Keywords: multi-domain edge cloud networks; parallel service function chain; dynamic deployment; federated averaging; deep reinforcement learning

随着第六代移动通信系统(6G)的引入和物联网(IoT)业务多样性的增长,电信服务提供商(TSP)面临着满足用户多样化需求的挑战。传统网络架构依赖大量部署于不同地理位置数据中心的专用设备,如防火墙、路由器、网络代理等^[1],然而培训专业技术人员的资金投入与维护这些设备的复杂性,给 TSP 带来了显著的财务负担^[2]。网络功能虚拟化(NFV)技术通过将网络功能抽象为可以在标准化服务器上以软件的形式运行的虚拟网络功能(VNF),提高了硬件利用率,降低了成本,并简化了这些 VNF 的部署、管理与扩展^[3]。NFV 使得网络服务能够根据实际需求动态调整计算、存储和网络资源^[4],避免因过度配置造成的浪费,并支持集中管理和编排,简化操作流程,降低维护成本^[5]。随着网络规模不断扩大,网络结构也从单一域扩展为由地理位置分散的多个边缘云构成的多域协同网络。通过将网络划分为多个域,可以在每个域内实现更灵活和高效的资源管理与服务部署。此外,这种划分方式还允许不同地理位置的用户通过最近的边缘域获得低延迟的服务响应。NFV 结合多域边缘云环境的优势,在每个域内实施独立管理策略,既确保了安全隔离又促进了资源共享,同时通过本地化数据和服务处理提升了服务质量和用户体验。

提供网络服务的服务功能链(SFC)是由一组按特定顺序连接的 VNF 构成的。首先,SFC 通常是按照串行顺序处理的,但这种串行处理会导致 SFC 的延迟随链长的增加而线性增加,从而造成不可接受的延迟^[6]。为了解决这个问题,先前的研究工作采用并行处理的方式来提高网络部署效率^[7]。尽管多分支并行处理缓解了串行处理方法面临的瓶颈,但现有的并行化方法的并行处理服务图可能不够合理,导致并行 SFC 的结构不够优化,进而增加延迟或降低资源利用率。其次,因为边缘网络环境的高度动态

性、用户的移动性和边缘节点中资源的有限性,与中心云相比,SFC 在边缘云上的部署需要更加高效地管理有限资源,并且快速响应用户位置变化和服务请求,所以需要实现更加灵活的流量管理和业务映射。

为了解决这些挑战,研究人员探索了在多域环境中动态部署 SFC 的各种方法,包括集中式和分布式算法,但由于多域环境下各域的技术标准、安全要求和管理政策存在显著差异,因此大大增加了实现高效、安全的 SFC 部署的难度。随着 IoT 设备数量的不断增长和服务需求的多样化,对于资源的有效利用、数据安全传输及服务质量的要求也在不断提升^[8],这进一步揭示了现有方法的局限性。因此,如何在保障性能、安全与效率的同时有效应对跨域协作、资源共享等挑战,特别是在确保跨域隐私保护的同时优化并行 SFC 部署、最小化成本和延迟以及最大化请求接受率的问题尚未得到充分解决。

针对以上挑战,本文重点研究多域边缘云环境下并行 SFC 动态部署问题,提出一种利用联邦学习(FedAvg)和深度强化学习(DRL)的方法加以解决。本文的研究目标是优化 SFC 的并行化结构,平衡成本、延迟和服务接受率三者之间的关系,同时遵守严格的隐私限制,本文的主要贡献如下。

(1)构建混合 SFC 的并行结构优化模型,从而实现降低延迟,减少数据包堆积和提高资源利用率。

(2)设计 FedAvg 和 DRL 算法的协同框架,在保障跨域数据隐私的前提下,实现全局参数更新,并引入优先经验回放(PER)机制,提升算法的收敛效率;同时,建立延迟、成本和请求接受率的多目标优化函数,通过权衡三者之间的关系,提高系统的运行效率和服务质量。

(3)将本文方法与因子分析深度 Q 网络(FA-DQN)、面向供应链的设计(DFSC)和多查找表(MuL)算法进行比较,验证了本文算法的有效性。

1 混合与动态 SFC 部署研究

由于服务请求的动态性和高效资源利用的需要,在多域边缘云网络中部署 SFC 面临重大挑战。研究人员已经开展了一些研究工作,包括串行、并行和混合 SFC 的部署,然而很少有专门针对多域边缘云环境中混合 SFC 的动态部署的研究。因此,本文研究了解决混合 SFC 部署的相关文献,详细说明了为什么动态 SFC 部署是必要的,以及为什么 FedAvg 联合 DRL 算法的解决方案适用于这一研究课题。

1.1 混合 SFC 部署

对于所有的 SFC,根据其中 VNF 之间的依赖关系,通过并行部署多个相同或不同类型的 VNF,并在必要时结合串行部署方式,得到的 SFC 被称为混合 SFC。在单域环境中,Wang 等研究了 SFC 构建和备份问题,设计了 VNF 共享和可靠构建算法,以及基于中心性和可靠性的节点排名算法^[9]。Wu 等研究了在满足延迟要求的同时最大化网络服务利润的 VNF 放置问题,虽然考虑了 VNF 并行化,但未充分探讨 VNF 间的依赖关系^[10]。Rui 等提出了基于 Petri 网的 SFC 可靠性评估方法,考虑了 VNF 的并行情况,并设计了可靠性优化算法^[11]。在多域环境中,由于网络域的持续增加以及各域特有的隐私政策,研究人员难以获取详细的网络信息或实现全域控制,导致管理复杂性增大。现有方法部分解决了域自治、隐私或可扩展性问题,并提出了几种解决方案以优化延迟、吞吐量和成本等指标。一些研究人员通过考虑不同的目标,如根据端到端延迟、资源分配效率和部署成本进行最佳放置,并利用集中式算法解决这一问题。Toumi 等通过线性规划设计了成本和延迟的奖励函数,利用基于 DRL 的方法解决多域网络中的 SFC 部署问题^[12]。Joshi 等提出了一种在多域环境中保护隐私的 SFC 编排方法 pSMART,采用二进制查询响应机制来协调多个网络域,虽然解决了隐私问题,但仍需完全控制所有域^[13]。Xiang 等提出了一种高效的跨域查询算法,通过将资源可用性抽象为资源状态,仅暴露最少的网络信息^[14]。这些集中式方法依赖于全局信息的可见性和对多个域的完全控制,因此并不直接适用于多域 SFC 部署问题。此外,也有一些研究人员采用分布式方法来解决这一问题。Lin 等提出了一种基于列生成的优化算法来解决多域网络中的 SFC 部署问题,该算法不需要域内拓扑和资源容量等信息^[15]。Liu 等提出了一种分布式方法来确保多域间的公平竞争,

并通过迁移已部署的 VNF 来平衡负载^[16]。Huang 等利用联邦强化学习提出一种可扩展的方法,通过联邦学习训练全局模型,充分利用可用资源^[17]。

1.2 动态 SFC 部署

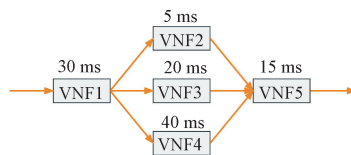
由于静态 SFC 部署机制不适用于高度动态的边缘云环境,因此研究人员对动态 SFC 部署工作进行了广泛的研究和探索。Chen 等提出了一种分布式 SFC 部署算法,使分散的边缘节点能够协同工作^[18]。Wang 等采用 DRL 技术实现自适应资源分配^[19]。Xia 等提出了一个服务提供商视角下的约束边缘数据缓存方法,能够有效地为大规模问题找到近似最优解^[20]。Shang 等研究了 SFC 部署与流量路由的联合优化问题,旨在实现成本效益最大化并有效控制网络拥塞^[21]。Mao 等开发了一个考虑流量变化的联合 SFC 部署和流量路由模型,并提出了两阶段交通敏感方案^[22]。Pham 等结合马尔科夫逼近和匹配理论,通过迭代方法解决 VNF 放置问题,以减少能源消耗和流量成本^[23]。因此,动态 SFC 部署更能适应网络资源的动态变化,提高资源利用率,满足不断变化的服务需求。

尽管现有研究已取得了一些成果,但目前的 SFC 部署工作仍不能充分解决多域边缘云环境带来的挑战。例如,现有的研究方法无法有效应对多域边缘云环境的复杂性,并且对用户移动性和网络动态变化适应能力有限,面对多样化的业务需求时灵活性不足。这导致在 SFC 部署过程中资源利用效率较低,同时服务质量也难以得到有效保障。因此,本文设计了一种新的基于并行优化的 SFC 部署算法,以最小化使用服务器数量。通过利用 FedAvg 和对决双深度网络(D3QN)的算法,在充分保障隐私的情况下,实现了并行 SFC 的高效经济部署。

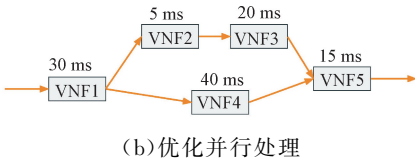
2 系统模型与问题描述

2.1 系统框架

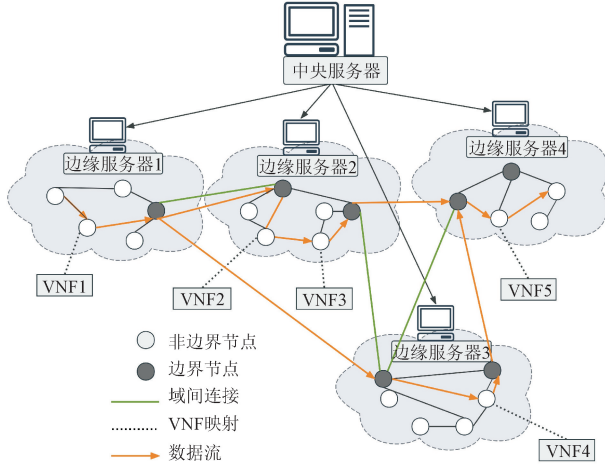
多域边缘云环境中混合 SFC 优化与部署示意图如图 1 所示,当一个新的 SFC 请求到达时,首先对其进行并行结构优化,再根据依赖关系将其切分成若干个可以独立部署的子链,然后将这些子链分配给各边缘域完成部署。



(a) 部分并行处理



(b) 优化并行处理



(c) 将并行优化后的 SFC 映射到多域边缘云环境中

图1 多域边缘云环境中混合 SFC 优化与部署示意图

Fig.1 Schematic diagram of hybrid SFC optimization and deployment in a multi-domain edge cloud environment

每个边缘域都有一个边缘服务器,它知道所有的域内信息,但只知道有关其他域的有限信息。由于隐私保护一直是多域网络中最关心的问题之一,因此,研究假设只有边界结点对其他边缘域的服务可见。

2.2 网络与服务模型

2.2.1 多域边缘云网络模型

将多域边缘云网络建模为无向图 $G = (N, L)$, 其中 N 表示物理节点集,包括边界节点和非边界节点, L 表示物理链路集。每个物理节点 $n \in N$ 都具有一定的资源容量 R_n^r , 这里 r 表示资源类型(如 CPU 或内存)。每个物理链路 $l \in L$ 有可用带宽资源容量 B_l 。用二进制变量 ϕ_n^u 表示虚拟节点 $u \in U$ 是否部署在物理节点 $n \in N$ 上,若部署则该变量取值为 1。同样,用二进制变量 ϕ_l^e 表示虚拟链路 $e \in E$ 是否被分配给物理链路 $l \in L$ 。第 m 个域的物理网络表示为 $G_m = (N_m, L_{h,m})$, N_m 为第 m 个域内物理节点的集合, $L_{h,m}$ 为第 m 个域中域内链路的集合。整个多域边缘云的物理网络可以表示为 $G = G_1 \cup G_2 \cup \dots \cup G_M \cup L_g$, 其中 M 为域的总数, L_g 为所有域间链路的集合。

2.2.2 SFC 请求模型

随机到达的服务请求集合表示为 S , 它同时也表示混合 SFC 的集合。第 i 个 SFC 表示为 S_i , 其大小

为 $|S_i|$, 可以表示为一个 5 元组 $S_i = (V_i, D_i, t_{a,i}, t_{b,i}, P_i)$, 其中, VNF 的集合由 V_i 表示, VNF 之间的预定义的相互依赖的集合由 D_i 表示, 服务请求的到达时间和离开时间分别由 $t_{a,i}$ 和 $t_{b,i}$ 表示, P_i 表示可以并行执行的 VNF 的集合。第 i 个 SFC 中 VNF 的集合表示为 $V_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,j}\}$, 其中, 第 j 个 VNF 用 $v_{i,j}$ 表示。每个 VNF 实例都有对 CPU 或内存的需求量, 表示为 R_v^r 。

2.3 问题描述

2.3.1 研究动机

SFC 部署问题通常涉及 VNF 放置和路径选择两个主要方面。VNF 放置是指将 VNF 放置到合适的物理节点上, 路径选择则是为连接 VNF 的虚拟链路确定最优传输路径。大量混合 SFC 的部署可能会导致物理节点资源的过度冗余, 进而增加部署成本并引发资源碎片化的问题。另外, 在 SFC 并行处理中, 不合理的服务图会造成资源的浪费。一方面, 过多的并行分支可能因为需要较高的带宽进行报文转发而增加成本; 另一方面, 并行分支间处理延迟的显著差异会导致数据包堆积, 即较快分支的数据包缓存在合并模块中, 等待较慢分支的数据包到达。例如, 一个部分并行处理的 SFC 如图 1(a) 所示, 其中 VNF2、VNF3 和 VNF4 可以并行执行。因此, 所有离开 VNF1 的数据包都需要被复制并发送到各并行分支, 处理后的数据包在 VNF5 之前的合并模块中进行合并。由于 VNF2 和 VNF3 的处理延迟远小于 VNF4, 因此这两个分支的数据包会更早到达合并模块, 导致合并模块需要分配额外的内存资源来缓存这些数据包。通过调整 SFC 的并行结构, 可以解决由于 VNF 的延迟差异引起的上述问题。如图 1(b) 所示, 如果将 VNF2 和 VNF3 合并为一个分支, 可以显著减小并行分支间的延迟差异, 从而减少合并模块的缓存需求。因此, 在利用 SFC 并行处理机制时, 应考虑各并行分支的处理延迟, 达到降低成本、节约资源和降低延迟的目的。

在多域边缘云网络中, 用户高移动性使得服务请求频繁切换至不同的边缘域, 因此需要定期更新服务部署策略来适应这种变化。从服务提供商的角度来看, TSP 需具备即时响应新服务请求的能力。随着越来越多的 IoT 设备需要随时接入定制化服务, TSP 希望 SFC 具备动态调整与适应变化的能力。

2.3.2 问题定义

混合 SFC 的动态部署问题被定义为一个两阶段的过程。第一阶段是对混合 SFC 进行并行结构

优化,根据服务功能的完整性要求将其切分为多个可并行处理的子链,并将这些子链合理分配到各边缘域中。第二阶段则是将这些分配到各域的子链进行域内映射,在映射时考虑域内节点的资源状况和优化目标,确保这些子链能够高效地映射到物理节点上。

2.4 约束条件和优化目标

2.4.1 约束条件

(1)节点资源容量约束。每个物理网络节点 $n \in N$ 上部署的所有 VNF 的资源需求总和不能超过该节点的资源容量 R_n^r ,表达式如下

$$\sum_{n \in N} \sum_{i \in [1, S]} \sum_{j \in [1, |S_i|]} \phi_n^{i,j} R_{i,j}^r \leq R_n^r \quad (1)$$

(2)VNF 唯一部署约束。SFC 中的每个 VNF 只能被部署在一个物理节点上,表达式如下

$$\sum_{n \in N} \sum_{i \in [1, S]} \sum_{j \in [1, |S_i|]} \phi_n^{i,j} = 1 \quad (2)$$

(3)带宽资源约束。所有映射到物理链路路上的虚拟链路所需带宽 B_e 的总和不超过该物理链路的可用带宽,表达式如下

$$\sum_{l \in L} \sum_{e \in E} \phi_l^e B_e \leq B_l \quad (3)$$

(4)虚拟链路方向约束。虚拟链路在同一时间只能映射到物理链路的一个方向,表达式如下

$$\sum_{n \in N} \sum_{l \in L} \sum_{e \in E} (\phi_{l,x \rightarrow y}^e + \phi_{l,y \rightarrow x}^e) < 2 \quad (4)$$

(5)最后期限约束。当服务请求不在其有效时间范围内时,不会被分配任何物理节点或带宽资源,表达式如下

$$\sum_{n \in N} \sum_{t \in [t_a, t_b]} \left(\sum_{i \in [1, S]} \sum_{j \in [1, |S_i|]} \phi_n^{i,j} + \sum_{l \in L} \sum_{e \in E} \phi_l^e \right) = 0 \quad (5)$$

(6)域间连接约束。当两个 VNF 分别位于不同的域时,至少存在一条跨域链路来连接这两个域,表达式如下

$$\sum_{l \in L_g} \sum_{v_1, v_2 \in V} \sum_{m_1, m_2 \in M} \phi_{v_1}^{m_1} \phi_{v_2}^{m_2} l_{m_1, m_2} \geq 1 \quad (6)$$

式中: $\phi_{v_1}^{m_1}$ 、 $\phi_{v_2}^{m_2}$ 是二进制变量,分别表示 VNF 集合中的 v_1 、 v_2 是否位于域 m_1 和域 m_2 中。

(7)隐私保护约束。为确保各边缘域的隐私安全,引入变量隐私指数 φ 来衡量多域 SFC 部署过程中的隐私保护程度,为了提高隐私保护水平,本文引入一个最小隐私保护阈值 $\pi_{\min}$,要求所有域的隐私指数必须高于这个阈值,表达式如下

$$\varphi = \sum_{m=1}^M \left(1 - \left(\frac{z_m Q_m}{m} \right) \right) / m \geq \pi_{\min} \quad (7)$$

式中: z_m 是第 m 个域的度数,表示该域在多域网络拓扑中邻接域的数量; Q_m 是第 m 个域的网络参数信息可见性。

2.4.2 优化目标

(1)最小化总延迟。总响应延迟 T 由传输延迟 T_w 、处理延迟 T_p 和排队延迟 T_q 组成,最小化总响应延迟可以表达为

$$\min \sum_{i \in [1, S]} T_{w,i} + T_{p,i} + T_{q,i} \quad (8)$$

传输延迟 T_w 指的是数据包在链路路上的传输时间,包括域内传输延迟 T_w^h 和域间传输延迟 T_w^g

$$T_w = \sum_{l \in L_g} \sum_{e \in E} \phi_l^e T_w^g + \sum_{l \in L_h} \sum_{e \in E} \phi_l^e T_w^h \quad (9)$$

处理延迟 T_p 指的是数据包在 VNF 实例上的处理时间

$$T_p = \sum_{n \in N} \sum_{i \in [1, S]} \sum_{j \in [1, |S_i|]} \phi_n^{i,j} \frac{1}{v_n} \quad (10)$$

式中: v_n 表示物理节点上处理 VNF 的速率。

排队延迟 T_q 指的是请求在队列中等待的时间。定义高优先级请求的排队延迟为 T_{q1} ,到达率为 λ_1 ,低优先级请求的排队延迟为 T_{q2} ,到达率为 λ_2 ,则有

$$(\lambda_1 + \lambda_2) T_q = \lambda_1 T_{q1} + \lambda_2 T_{q2} \quad (11)$$

(2)最小化总成本。部署一个服务请求的总成本用 C 表示,它由混合 SFC 的部署成本 C_d 和混合 SFC 的流量传输成本 C_f 组成。可以表达为

$$C = \min(\lambda C_d + \mu C_f); \lambda + \mu = 1 \quad (12)$$

式中: λ 和 μ 分别表示混合 SFC 部署成本、流量传输成本的权重。

假设 SFC 的部署成本与 SFC 中包含的 VNF 的数量成比例,则部署一个 SFC 请求的成本为完成该链中所有 VNF 部署的成本总和

$$C_d = \sum_{n \in N} \sum_{i \in [1, S]} \sum_{j \in [1, |S_i|]} \phi_n^{i,j} R_{i,j}^r c_r \quad (13)$$

式中: c_r 为每个资源单位的成本。

混合 SFC 的流量传输成本是将物理链路路上的带宽分配给虚拟链路以进行流量传输的成本总和,包括域间流量传输成本 C_f^g 和域内流量传输成本 C_f^h

$$C_f = \sum_{l \in L} \sum_{e \in E} \phi_l^e B_e c_B \quad (14)$$

式中: c_B 为每个链路带宽单位的成本。

(3)最大化请求的接受率。为了最大化接受率 α ,需要定义一个二元变量 φ_{S_i} 来表示请求 S_i 是否被接受。如果请求 S_i 被接受,则取值为 1。表达式如下

$$\varphi_{S_i} = \begin{cases} 1, & T \leq T_{\max} \\ 0, & \text{其他} \end{cases} \quad (15)$$

定义目标函数为最大化所有被接受请求的总吞吐量

$$\text{maximize } \sum_{i \in [1, S]} \varphi_{S_i} B_i \tau \tag{16}$$

式中: B_i 为请求 S_i 的带宽需求; τ 是时间间隔。

综上所述,本研究的优化目标是最小化总延迟和总成本,同时最大化请求接受率 α 。因此,为总延迟、总成本和请求接受率分配权重系数,并将这 3 个优化目标进行组合^[24],得到全局优化目标如下式

$$Z = \varepsilon \alpha - \delta C - \eta T \tag{17}$$

式中: ε 、 δ 和 η 是每个目标函数的加权因子。

通过根据实际需求设置不同优化目标的权重,实现多目标优化或平衡,从而确保在多域环境中实现高效的 SFC 部署。

3 基于联邦学习和 DRL 算法的并行 SFC 动态部署机制

在多域边缘云环境下部署混合 SFC 时,首先需要对混合 SFC 进行分析,优化并行结构,然后再对优化后的混合 SFC 进行切分,并将切分后的子链分配到合适的边缘域。由于物理节点的资源容量会随 SFC 子链的映射而变化,若当某个域的剩余资源量不足时仍在该域部署,可能会导致 SFC 的接受率降低。因此,本文设计了 FedAvg 联合 DRL 的 SFC 部署方法来应对这些挑战。

在多域边缘云环境中,网络状态、用户需求和服 务类型的动态变化都要求部署策略能够快速适应这些变化,同时在保证性能的同时平衡资源消耗和延迟需求。DRL 算法将深度学习和强化学习的优势相结合,通过利用深度神经网络(DNN)感知环境,将智能体与物理网络的高效交互和奖励机制的正负反馈相结合,使模型同时具有动态感知和决策能力。因此与数学规划算法和启发式算法相比,DRL 算法具备更好的动态适应性和鲁棒性^[19,20],能够迅速适应环境变化,在相关研究领域有较好的应用。

D3QN 算法作为 DRL 领域中新一代的算法,通过结合价值分解和双网络机制的思想,将 D3QN 的网络输出分为两部分,一部分用来评估状态的价值,另一部分用来评估每个动作的优势。通过运用双 Q 网络结构,提高了价值估计的精度和算法的整体稳定性和收敛性,然而智能体需要通过多次尝试无效或次优的部署策略来逐渐收敛到有效的策略。这导致 D3QN 算法在初期学习阶段和动态变化的环境中可能面临样本效率低和收敛过早的问题。因此,在 D3QN 中引入优先经验回放机制,通过赋予经验回放缓冲区中的样本不同的优先级,并在训练过程中选择

关键样本,协助智能体在探索未知与利用已有知识之间找到平衡,进而提高学习速率和对动态环境的适应性。此外,在跨域环境中,为了实现高效且隐私保护的模型优化,利用 FedAvg 算法在不交换原始数据的情况下协同优化全局模型。具体来说, t 时刻各域接收全局模型参数 W_t ,并利用本地数据独立训练模型, $t+1$ 时刻各域将更新后的参数 W_{t+1}^m 发送至中央服务器。通过 FedAvg 算法,中央服务器对来自各域的参数进行加权平均,更新全局模型后分发回各域。这种方法不仅充分利用了各域的数据优势,提升了模型的泛化能力和鲁棒性,还确保了敏感数据的安全与隐私,实现了跨域合作中的模型协同优化。

综上所述,本文所提 FA-D3QN-PER 算法通过 FedAvg 算法在不暴露敏感数据的情况下协作优化资源分配策略,同时利用 D3QN-PER 的高效学习和动态适应能力,解决了跨域 SFC 部署中的资源分配和路径选择问题,提高了系统的请求接受率,并且降低了成本和延迟。FA-D3QN-PER 算法的框架图如图 2 所示。

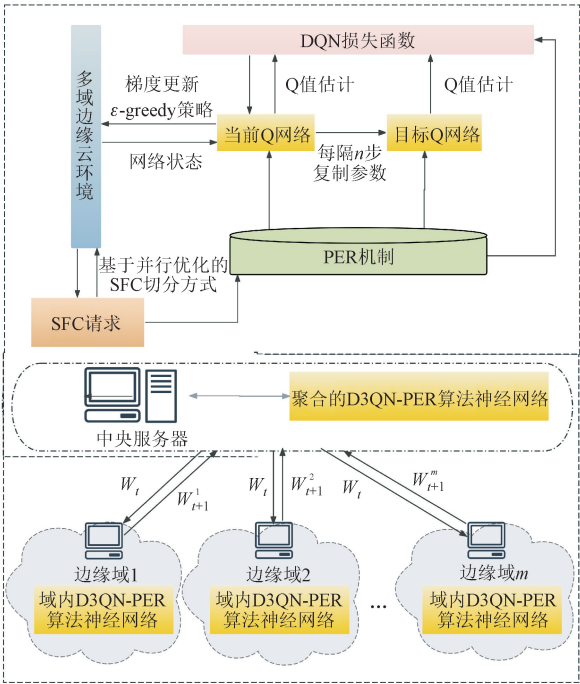


图 2 FA-D3QN-PER 算法框架图
Fig. 2 FA-D3QN-PER algorithm framework diagram

3.1 SFC 切分的马尔科夫决策过程建模

3.1.1 状态集

在任意时刻 t , 状态 S_t 被定义为各边缘域中的剩余资源和域间链路剩余传输资源 R_g^B 的集合, $S_t = \{S_{t,1}, S_{t,2}, \dots, S_{t,m}, R_g^B\}$ 。域 m 内剩余资源容量和域

内链路剩余传输资源容量分别用 R_m^r 和 R_m^B 表示, $S_{i,m} = \{R_m^r, R_m^B\}$ 。

3.1.2 动作集

由于整个网络域是由 M 个边缘域组成的, SFC 切分后要将各并行分支分配到各边缘域中, 因此, 动作 A_s 可以定义为 $A_s = \{P_{i,k}, m_{i,k}\}$, 其中, $P_{i,k}$ 表示要分配的第 i 个 SFC 的第 k 个并行分支, $m_{i,k}$ 表示选定的用于部署这一并行分支的边缘域 m 。SFC 切分的可选择动作数等于当前 SFC 的并行分支数 k 与边缘域的数量 M 的乘积, 即 $|A_s| = kM$ 。

3.1.3 奖励函数

由于多域环境下混合 SFC 切分的优化目标是降低域间流量传输成本和域间传输延迟的同时提高接受率。因此, 奖励函数如下式

$$R(S_i, A_s) = \epsilon\alpha - \delta C_f^r - \eta T_w^g \quad (18)$$

该函数用于评估在当前状态下采取某一行动后获得的奖励。

总奖励为

$$R_a = \sum \gamma R(S_{t+1}, A_{s,t+1}) \quad (19)$$

其中, $\gamma \in [0, 1]$ 为未来奖励的折现因子。

3.2 基于 D3QN-PER 算法的 SFC 并行分支切分算法

3.2.1 并行优化算法

并行优化算法通过将并行分支合理分配至资源容器, 在满足最大处理延迟约束的同时优化并行分支列表。首先, 算法计算并行分支列表中所有分支的最大处理延迟; 若列表为空, 则延迟设为 0; 否则, 遍历列表确定最大长度。接着, 基于最大处理延迟与预设参数 ρ , 初始化容器集合及其容量。然后, 尝试将每个分支放入现有容器中, 如果当前容器无法容纳该分支, 则创建新容器。最后, 将所有容器中的分支重新组织为优化后的并行分支列表 P' 。

关于该算法的时间复杂度, 计算并行分支列表中最大处理延迟需要遍历整个列表, 时间复杂度为 $O(n)$ 。将各分支放入容器时, 最坏情况需要检查所有现有容器, 时间复杂度为 $O(mw)$, 其中 w 为容器数且 $w \leq n$ 。最后, 重新组织所有容器中的分支并形成优化后的并行分支列表的时间复杂度为 $O(n)$ 。综上所述, 该算法的整体时间复杂度为 $O(n^2)$ 。

3.2.2 基于 D3QN-PER 算法的 SFC 并行分支切分算法

基于 D3QN-PER 的 SFC 并行分支切分算法用于在多域边缘云环境中将各并行分支高效分配到合

适边缘域。首先根据各边缘域的资源占用情况初始化神经网络权重和系统状态。接着检查服务请求是否在其有效时间范围内, 若请求无效则跳过并记录原因。对于有效请求, 遍历各并行分支, 若当前分支为首个并行分支, 则将其划分到资源占用率最低且满足需求的边缘域。后续分支则采用 Noisy DQN 算法加入高斯噪声, 通过随机选择或最大化价值策略评估候选域是否满足约束条件。若选定动作满足所有约束, 则执行该动作、更新状态并计算奖励函数, 然后将样本存储至经验回放缓冲区中。此外, 算法每隔固定学习时间间隔 τ 计算损失并执行梯度下降, 同时定期更新目标网络参数。

关于该算法的时间复杂度, 初始化及迭代确定初始状态的时间复杂度为 $O(N)$; 将各并行分支依次划分到各边缘域的时间复杂度为 $O(ikM)$ 。因此, 该算法的时间复杂度为 $O(NikM)$ 。

3.3 子链域内映射的马尔科夫决策过程建模

3.3.1 状态集

在任意时刻 t , 状态 M_t 被定义为各个边缘域内部物理节点中剩余资源的集合, $M_t = \{R_{n,1}^r, R_{n,2}^r, \dots, R_{n,n}^r\}$ 。

3.3.2 动作集

由于每个边缘域平均由 n 个节点组成, 子链的域内映射是将子链中的每一个 VNF 映射到域内节点上, 因此, 动作 A_m 可以定义为 $A_m = \{v_{k,j}, n_{k,j}\}$, 其中, $v_{k,j}$ 表示要映射的 VNF, $n_{k,j}$ 表示选定的用于映射该 VNF 的节点。子链的域内映射可选择的动作数等于子链中的 VNF 的平均数量 J 与域内平均节点数量 n 的乘积, 即 $|A_m| = Jn$ 。

3.3.3 奖励函数

由于子链的域内部署的优化目标为降低域内流量传输成本、VNF 部署成本、域内传输延迟、VNF 处理延迟和排队延迟的同时提高接受率。因此, 奖励函数如下式

$$R(M_t, A_m) = \epsilon\alpha - \delta(C_f^h + C_d) - \eta(T_w^h + T_p + T_q) \quad (20)$$

总奖励的计算式如式(19)所示。该函数用于评估在当前状态下采取某一行动后所获得的奖励。

3.4 基于 FA-D3QN-PER 的子链域内动态部署算法

基于 FA-D3QN-PER 的 SFC 子链域内动态部署算法用于在各域内为每个 VNF 选择合适的物理节点。迭代开始时, 各边缘域从中央服务器获取最新的全局模型参数。随后依次处理待部署的 VNF,

使用 Noisy DQN 算出生成带噪声的动作,并评估其是否满足约束条件。若满足,则执行部署操作、更新状态并计算奖励函数,并将此次交互样本存入本地经验回放缓冲区。各边缘域基于本地环境独立训练模型,计算损失并执行梯度下降更新,每隔若干步更新目标网络参数。完成一轮本地训练后,各域将更新后的模型上传至中央服务器。中央服务器对来自各域的参数进行聚合,生成新的全局模型参数,并同步回各边缘域,继续下一轮迭代。

关于该算法的时间复杂度,初始化以及迭代确定初始状态并将中央服务器的参数传输到各域的时间复杂度为 $O(N)$;遍历 VNF 为其寻找合适的部署节点的时间复杂度为 $O(kJn)$ 。最后,全局更新的时间复杂度为 $O(N)$ 。因此,该算法的整体时间复杂度为 $O(NkJn)$ 。

4 模拟实验与评估

在本节中,针对多域边缘云环境下的并行 SFC 动态部署问题,采用 FA-D3QN-PER 算法对接受率、平均延迟及总成本 3 项指标进行了评估。通过与现有文献中较优方案对比,验证了本文模型与算法的优越性。

4.1 仿真设置

模拟仿真实验在 Windows10 操作系统、i7-9750H CPU 和 8 GB 内存的环境下,使用 Python 3.8.18 平台、PyTorch 2.1.0 库进行。为了验证 FA-D3QN-PER 算法的性能,将其与 FA-DQN、MuL^[25]以及 DFSC 算法^[26]进行对比。

在模拟实验中,随机生成包含 5 个域的网络拓扑,每个域有 10 个除边界节点以外的物理节点^[27]。每个 SFC 请求可选择的虚拟节点数随机设定为 2~4,用来模拟不同的资源分配场景。VNF 的部署时间设定为 10~20 ms^[28],域内传输延迟服从 0~1 ms 的均匀分布,域间传输延迟服从 1~2 ms 的均匀分布。此外,每个 VNF 的 CPU 需求和内存需求从预定义的列表中随机选取,确保实验设置的多样性和真实性。尽管 SFC 请求以随机的方式到达系统,但是算法仍会按到达顺序依次处理这些请求。

在算法参数的选择上,首先,根据实验反馈确定最终参数,将奖励函数的参数分别设置为 $\epsilon = 0.4$ 、 $\delta = 0.3$ 和 $\eta = 0.3$ 。此外,将神经网络的隐藏层设置为包含 128 个神经元。最后,考察了 4 种不同学习率(0.01、0.005、0.001 和 0.000 5)对 DRL 模型损失值随迭代次数变化的情况^[29],采用均方误差计算损失值,用于衡量模型预测值与真实值之间的差

距,如图 3 所示。实验结果表明,学习率为 0.01 时的模型在早期迭代中迅速下降,但随后出现过冲现象,这表明当学习率为 0.01 时训练过程不稳定且有发散风险。当学习率为 0.005 时,损失在最初的几轮迭代中迅速下降,但此后出现一定程度的波动甚至回升,这表明当学习率为 0.005 时,训练过程可能会因为学习率过高出现振荡现象。相比之下,学习率为 0.001 时模型的损失函数不仅快速下降至较低水平,而且避免了高学习率带来的不稳定性,减少了训练过程中的振荡,表现出良好的收敛性和稳定性。学习率为 0.000 5 的模型虽然损失能够持续下降至接近 0,但其收敛速度过慢,在实际应用中可能会导致延迟和成本增加。因此,综合考虑性能、效率和成本等因素,将学习率设置为 0.001 来优化模型训练。

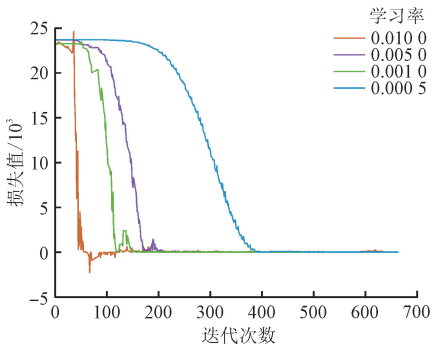


图 3 不同学习率对收敛效果的影响

Fig. 3 Effect of different learning rates on convergence

4.2 实验结果及分析

4.2.1 SFC 请求数量对算法性能的影响

图 4~图 6 分别展示了当 SFC 请求数量由 100 条增长至 600 条^[25-26],且每条 SFC 上的 VNF 的数量随机设定为 2~6 时,FA-D3QN-PER、FA-DQN、DFSC 和 MuL 算法在接受率、平均延迟和总成本方面的性能表现。这样设计的目的是评估不同算法在面对不同负载情况下的性能表现。

(1)接受率。图 4 展示了不同算法处理 SFC 请求时请求接受率随请求数量的变化。实验结果表明,FA-D3QN-PER 算法在接受率上显著优于其他算法,分别比 FA-DQN、DFSC 和 MuL 算法高出约 11.6%、12.8%和 40%。当 SFC 请求较少时,所有算法都表现良好,但随着请求数量的增加,FA-D3QN-PER 算法展现出更强的适应性和稳定性,而其他算法的接受率出现不同程度的下降。这是因为 FA-D3QN-PER 算法引入的 PER 机制,提高了学习效率 and 决策质量,并结合 FedAvg 和 D3QN 算法实现了隐私保护下的多域联合训练,适应复杂网络环

境。相比之下,因为缺乏 PER 机制,所以 FA-DQN 算法性能较差。DFSC 算法适用于小型至中型网络,但在大规模变化的网络状况下自适应能力不足。MuL 算法侧重于优化部署成本,难以应对网络故障或突然增加的请求。

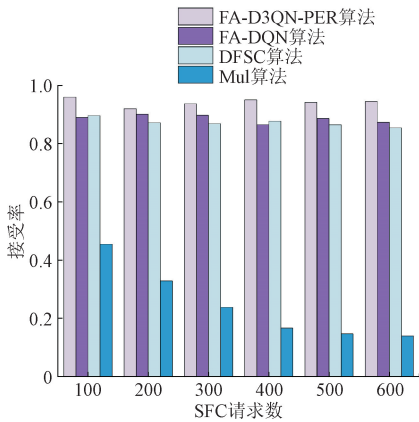


图 4 不同 SFC 请求数下 4 种算法的请求接受率对比
Fig. 4 Comparison of request acceptance rates for four algorithms under different SFC request counts

(2)平均延迟。图 5 展示了不同算法处理 SFC 请求时平均延迟随请求数量的变化。FA-D3QN-PER 算法在所有请求数下均表现出最小的平均延迟,与 FA-DQN、DFSC 和 MuL 算法相比其性能优势明显。这是因为 FA-D3QN-PER 算法通过智能选择低时延的节点进行映射,并且能够快速适应复杂网络环境,即使在高请求数的情况下也能保持较低的延迟。FA-DQN 的延迟在请求数增大后呈现线性增长的趋势。DFSC 的延迟随着请求数增加迅速增长,在超过 300 个请求后增长加快,表明其自适应能力不足。MuL 算法侧重于优化成本,在延迟方面表现较差。

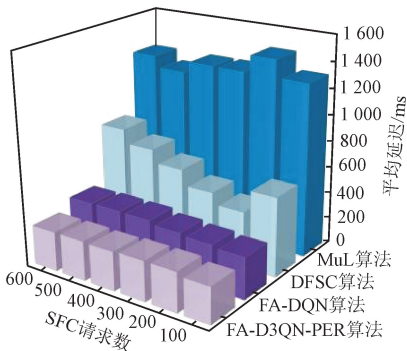


图 5 不同 SFC 请求数下 4 种算法的平均延迟对比
Fig. 5 Comparison of average latency for four algorithms under different SFC request counts

(3)总成本。图 6 展示了不同算法在处理 SFC 请求时总成本随请求数的变化情况。FA-D3QN-PER 算法在所有请求数量下都保持最低的总成本,这是因为 FA-D3QN-PER 算法通过智能资源分配策略,提高了学习效率并优化了资源配置,使总成本的变化较为平缓,即使请求数量大幅增加也能维持较低水平。FA-DQN 算法的总成本虽然也较低,但随着请求数增长,其成本上升趋势明显。DFSC 算法的总成本总体上高于 FA-D3QN-PER 和 FA-DQN 算法,因为它未充分考虑动态资源分配。MuL 算法的总成本在低请求数时较低,但在超过 300 个请求后成本随着请求数增加迅速升高,这主要是由于它难以应对网络故障或请求突然增加的情况,导致额外成本增加。

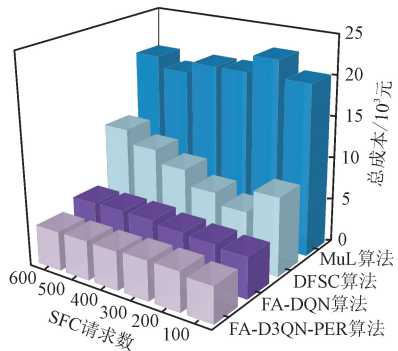


图 6 不同 SFC 请求数下 4 种算法的总成本对比
Fig. 6 Comparison of total costs for four algorithms under different SFC request numbers

4.2.2 不同链长对算法性能的影响

图 7~图 9 分别展示了当 SFC 请求数固定为 500 条^[25]且每条 SFC 上的 VNF 的数量从 4 个增长为 12 个时^[26],FA-D3QN-PER、FA-DQN、DFSC 和 MuL 算法在接受率、平均延迟和总成本方面的性能表现。这样设计的目的是评估不同算法在处理复杂度和资源需求不同的请求时的表现。

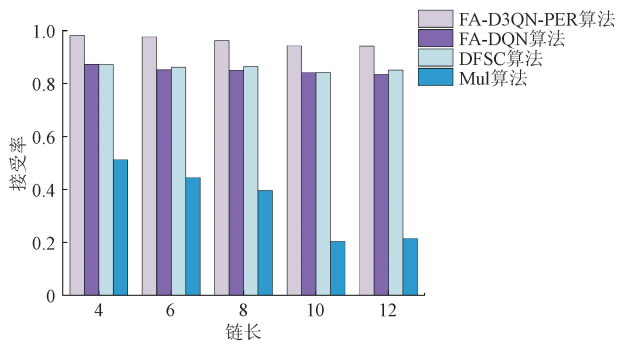


图 7 不同链长下 4 种算法的请求接受率对比
Fig. 7 Comparison of request acceptance rates for four algorithms under different chain lengths

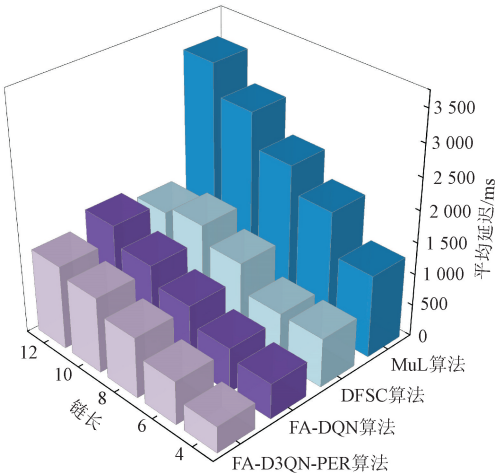


图 8 不同链长下 4 种算法的平均延迟对比

Fig. 8 Comparison of average latency for four algorithms under different chain lengths

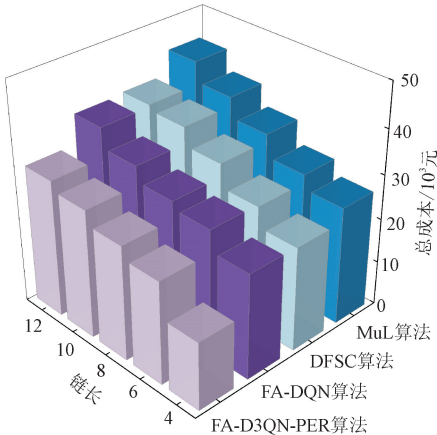


图 9 不同链长下 4 种算法的总成本对比

Fig. 9 Comparison of total costs for four algorithms under different chain lengths

(1)接受率。如图 7 所示,FA-D3QN-PER 算法在所有链长下均展现出最高的请求接受率,即使链长从 4 逐渐增加到 12,其接受率也仅有轻微下降。这得益于 PER 机制提高了学习效率和决策质量,通过结合 FedAvg 和 D3QN 算法能够更有效地适应复杂网络环境。由于缺乏 PER 机制,FA-DQN 算法在处理复杂任务时路径选择优化不足,导致其接受率低于 FA-D3QN-PER 算法。DFSC 算法适用于中小型网络,在较长链任务中自适应能力不足,难以有效管理资源分配,接受率波动较大,表现出不稳定性。MuL 算法的接受率显著低于其他 3 种算法,尤其在链长增加时迅速下降,表明其对链长变化更为敏感。

(2)平均延迟。如图 8 所示,随着链长从 4 增加

到 12,所有算法的平均延迟均呈现上升趋势,但 FA-D3QN-PER 算法的表现最为稳定,其平均延迟始终低于其他 3 种算法。FA-D3QN-PER 算法通过智能选择低时延节点进行映射,确保在处理较长 SFC 请求时仍保持较低的延迟。FA-DQN 和 DFSC 算法在链长较短时表现良好,但在链长增加后,平均延迟显著升高,显示出对较长 SFC 请求处理能力的不足。MuL 算法更侧重于成本优化,因此在链长增加时,其平均延迟迅速升高,难以应对复杂的链长变化需求。

(3)总成本。如图 9 所示,随着链长增加,所有算法的总成本都呈上升趋势,但 FA-D3QN-PER 算法的表现最优。在链长较短时,4 种算法的成本差异不大,但当链长增加至 8 时,FA-D3QN-PER 算法的成本显著低于其他算法。优化的部署和资源分配策略使其在复杂多变的网络环境中保持高效和低成本。FA-DQN 算法初期表现良好,但在处理复杂任务时未能持续优化资源配置,导致成本逐渐升高。DFSC 算法适用于小型至中型网络,在面对较长链任务时自适应能力不足,难以应对动态变化的需求,从而增加了成本。MuL 算法只在链长小于 6 时表现较好,在链长进一步增加时,MuL 算法无法快速调整资源分配,导致成本增加。

5 结 论

针对多域边缘云环境中并行 SFC 的动态部署问题,主要挑战在于如何在保证跨域隐私保护的前提下,实现低成本、低延迟和高接受率的并行 SFC 部署。为解决这一问题,构建了优化的 SFC 并行模型,在提高资源利用效率的同时降低部署延迟。基于这一模型,进一步提出了多域边缘云环境下的并行 SFC 动态部署算法 FA-D3QN-PER。该算法通过结合 D3QN 算法与 PER 机制,提高了决策过程的精确性和效率;在跨域部署时,利用 FedAvg 算法避免了因直接共享本地数据而产生的数据泄露风险,确保了跨域隐私保护。仿真结果表明,FA-D3QN-PER 算法在接受率、平均延迟及总成本方面均优于现有方法,展现出在实际应用中的潜力。此外,该方法具有良好的可扩展性,支持未来网络中不断增长的多域和多边缘场景。

参考文献:

[1] HE Fujun, SATO T, OKI E. Optimization model for backup resource allocation in middleboxes with impor-

- tance [J]. IEEE/ACM Transactions on Networking, 2019, 27(4): 1742-1755.
- [2] CAO Chen, WANG Jianhua, KWOK D, et al. webT-WAS: a resource for disease candidate susceptibility genes identified by transcriptome-wide association study [J]. Nucleic Acids Research, 2022, 50(D1): D1123-D1130.
 - [3] 刘光远, 曹晶仪, 庞紫园, 等. 一种低时延虚拟网络功能映射及调度优化算法 [J]. 西安交通大学学报, 2023, 57(2): 121-130.
LIU Guangyuan, CAO Jingyi, PANG Ziyuan, et al. An optimized algorithm with low latency for virtual network function mapping and scheduling [J]. Journal of Xi'an Jiaotong University, 2023, 57(2): 121-130.
 - [4] 孙士清, 彭建华, 游伟, 等. 5G 网络下资源感知的服务功能链协同构建和映射算法 [J]. 西安交通大学学报, 2020, 54(8): 140-148.
SUN Shiqing, PENG Jianhua, YOU Wei, et al. A co-ordinating composition and mapping algorithm for a service function chain with resource-aware [J]. Journal of Xi'an Jiaotong University, 2020, 54(8): 140-148.
 - [5] KAUR K, MANGAT V, KUMAR K. A review on virtualized infrastructure managers with management and orchestration features in NFV architecture [J]. Computer Networks, 2022, 217: 109281.
 - [6] XIE Sihao, MA Junte, ZHAO Jin. FlexChain: bridging parallelism and placement for service function chains [J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 195-208.
 - [7] AGARWAL S, CHINTAPALLI V R, TAMMA B R. FlexSFC: flexible resource allocation and VNF parallelism for improved SFC placement [C]//2022 IEEE 8th International Conference on Network Softwarization (NetSoft). Piscataway, NJ, USA: IEEE, 2022: 302-306.
 - [8] 徐冉, 王文东, 龚向阳, 等. 网络功能虚拟化中延时感知的资源调度优化方法 [J]. 计算机研究与发展, 2018, 55(4): 738-747.
XU Ran, WANG Wendong, GONG Xiangyang, et al. Delay-aware resource scheduling optimization in network function virtualization [J]. Journal of Computer Research and Development, 2018, 55(4): 738-747.
 - [9] WANG Ying, ZHANG Leyi, YU Peng, et al. Reliability-oriented and resource-efficient service function chain construction and backup [J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 240-257.
 - [10] WU Yunyi, ZHENG Weichang, ZHANG Yongbing, et al. Reliability-aware VNF placement using a probability-based approach [J]. IEEE Transactions on Network and Service Management, 2021, 18(3): 2478-2491.
 - [11] RUI Lanlan, CHEN Xushan, GAO Zhipeng, et al. Petri net-based reliability assessment and migration optimization strategy of SFC [J]. IEEE Transactions on Network and Service Management, 2021, 18(1): 167-181.
 - [12] TOUMI N, BAGAA M, KSENTINI A. On using deep reinforcement learning for multi-domain SFC placement [C]//2021 IEEE Global Communications Conference (GLOBECOM). Piscataway, NJ, USA: IEEE, 2021: 1-6.
 - [13] JOSHI K D, KATAOKA K. pSMART: a lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN [J]. Computer Networks, 2020, 178: 107295.
 - [14] XIANG Qiao, WANG Xingchen, ZHANG Jianshun, et al. Unicorn: unified resource orchestration for multi-domain, geo-distributed data analytics [J]. Future Generation Computer Systems, 2019, 93: 188-197.
 - [15] LIN Rongping, YU Song, LUO Shan, et al. Column generation based service function chaining embedding in multi-domain networks [J]. IEEE Transactions on Cloud Computing, 2023, 11(1): 185-199.
 - [16] LIU Yi, ZHANG Hongqi, CHANG Dexian, et al. GDM: A general distributed method for cross-domain service function chain embedding [J]. IEEE Transactions on Network and Service Management, 2020, 17(3): 1446-1459.
 - [17] HUANG Haojun, ZENG Cheng, ZHAO Yangmin, et al. Scalable orchestration of service function chains in NFV-enabled networks: a federated reinforcement learning approach [J]. IEEE Journal on Selected Areas in Communications, 2021, 39(8): 2558-2571.
 - [18] CHEN Lixing, SHEN Cong, ZHOU Pan, et al. Collaborative service placement for edge computing in dense small cell networks [J]. IEEE Transactions on Mobile Computing, 2021, 20(2): 377-390.
 - [19] WANG Jiadai, ZHAO Lei, LIU Jijia, et al. Smart resource allocation for mobile edge computing: a deep reinforcement learning approach [J]. IEEE Transactions on Emerging Topics in Computing, 2021, 9(3): 1529-1541.
 - [20] XIA Xiaoyu, CHEN Feifei, GRUNDY J, et al. Con-

strained app data caching over edge server graphs in edge computing environment [J]. IEEE Transactions on Services Computing, 2022, 15(5): 2635-2647.

[21] SHANG Xiaojun, LIU Zhenhua, YANG Yuanyuan. Online service function chain placement for cost-effectiveness and network congestion control [J]. IEEE Transactions on Computers, 2022, 71(1): 27-39.

[22] MAO Yingling, SHANG Xiaojun, YANG Yuanyuan. Provably efficient algorithms for traffic-sensitive SFC placement and flow routing [C]//IEEE INFOCOM 2022-IEEE Conference on Computer Communications. Piscataway, NJ, USA: IEEE, 2022: 950-959.

[23] PHAM C, TRAN N H, REN Shaolei, et al. Traffic-aware and energy-efficient vNF placement for service chaining: joint sampling and matching approach [J]. IEEE Transactions on Services Computing, 2020, 13(1): 172-185.

[24] 王珂, 曲桦, 赵季红. 多域 SFC 部署中基于强化学习的多目标优化方法 [J]. 计算机科学, 2021, 48(12): 324-330.

WANG Ke, QU Hua, ZHAO Jihong. Multi-objective optimization method based on reinforcement learning in multi-domain SFC deployment [J]. Computer Science, 2021, 48(12): 324-330.

[25] KIBALYA G, SERRAT-FERNANDEZ J, GORRICHIO J L, et al. A multi-stage graph aided algorithm for distributed service function chain provisioning across multiple domains [J]. IEEE Access, 2021, 9: 114884-114904.

[26] CHEN Chen, NAGEL L, CUI Lin, et al. Distributed federated service chaining: a scalable and cost-aware approach for multi-domain networks [J]. Computer Networks, 2022, 212: 109044.

[27] 王琛, 汤红波, 游伟, 等. 一种 5G 网络低时延资源调度算法 [J]. 西安交通大学学报, 2018, 52(4): 117-124.

WANG Chen, TANG Hongbo, YOU Wei, et al. A resource scheduling algorithm with low latency for 5G networks based on effective hybrid genetic algorithm and tabu search [J]. Journal of Xi'an Jiaotong University, 2018, 52(4): 117-124.

[28] SUN Gang, LI Yayu, LIAO Dan, et al. Service function chain orchestration across multiple domains: a full mesh aggregation approach [J]. IEEE Transactions on Network and Service Management, 2018, 15(3): 1175-1191.

[29] ZHU Yuchao, YAO Haipeng, MAI Tianle, et al. Multiagent reinforcement-learning-aided service function chain deployment for internet of things [J]. IEEE Internet of Things Journal, 2022, 9(17): 15674-15684.

(编辑 武红江)