

多域 SFC 部署中基于强化学习的多目标优化方法



王珂¹ 曲桦^{1,2} 赵季红^{2,3}

1 西安交通大学软件学院 西安 710049

2 西安交通大学电子与信息工程学院 西安 710049

3 西安邮电大学通信与信息工程学院 西安 710061

(215545230@qq.com)

摘要 随着网络虚拟化技术的发展,多域网络中的服务功能链部署为服务功能链优化部署问题带来了新的挑战。传统的部署方法通常对单一目标进行优化,不适用于多目标优化问题,且无法对优化目标间权重进行衡量及平衡。因此,为了对大规模服务功能链部署请求下的时延、网络负载均衡性及接受率进行同步优化,提出了一种数据归一化处理方案,并设计了基于强化学习的两步 SFC 部署算法。该算法以传输时延与负载均衡性为反馈参数,平衡了两者的权重关系,并对其进行了同步优化,同时利用强化学习框架优化了 SFC 接受率。实验结果表明,所提算法在大规模请求数下,相比时延感知方法时延降低了 71.8%,相比多域部署方法接受率提高了 4.6%,相比贪心算法平均负载均衡性提高了 39.1%,保证了多目标优化效果。

关键词: 多域;服务功能链;数据归一化;强化学习;多目标优化

中图法分类号 TP393

Multi-objective Optimization Method Based on Reinforcement Learning in Multi-domain SFC Deployment

WANG Ke¹, QU Hua^{1,2} and ZHAO Ji-hong^{2,3}

1 School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China

2 School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

3 School of Communication and Information Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710061, China

Abstract With the development of network virtualization technology, the deployment of service function chain in multi-domain network brings new challenges to the optimization of service function chain. The traditional deployment method usually optimizes a single target, which is not suitable for multi-objective optimization, and cannot measure and balance the weight among optimization targets. Therefore, in order to optimize the delay, network load balancing and acceptance rate of large-scale service function chain deployment requests synchronously, a data normalization processing scheme is proposed, and a two-step SFC deployment algorithm based on reinforcement learning is designed. The algorithm takes transmission delay and load balancing as feedback parameters and balances the weight relationship between them, and the SFC acceptance rate is optimized by using reinforcement learning framework simultaneously. The experimental results show that, the delay of the algorithm is reduced by 71.8% compared with LASP method, the acceptance rate is increased by 4.6% compared with MDSP method, and the average load balancing is increased by 39.1% compared with GREEDY method under the large-scale requests. The multi-objective optimization effect is guaranteed.

Keywords Multi-domain, Service function chain, Data normalization, Reinforcement learning, Multi-objective optimization

1 引言

在传统网络中,大量依托于专用硬件部署的网络功能组成多种网络服务,随着网络服务请求的数量及多样性急剧增长,各运营商传统的专用硬件部署方式已无法满足当前的复杂多变的网络需求,而网络功能虚拟化(Network Function

Virtualization, NFV)技术的出现解决了这一问题^[1]。NFV 通过软硬件解耦及功能抽象,使网络设备的功能不再依赖于专用设备, NFV 技术能够充分、灵活地共享网络资源,从而降低高昂的网络设备成本,且能实现新业务的快速开发和部署,并基于实际业务需求进行自动部署、弹性伸缩、故障隔离和自愈等^[2]。

收稿日期:2020-11-23 返修日期:2021-04-30 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家重点研发计划项目(2018YFB1800305)

This work was supported by the National Key R & D Program of China(2018YFB1800305).

通信作者:曲桦(qh@mail.xjtu.edu.cn)

在 NFV 中,虚拟网络功能(Virtualized Network Function, VNF)以一定的顺序部署在服务器节点上,构成一条可以处理特定网络业务的网络功能序列,即服务功能链(Service Function Chain, SFC)^[3]。服务功能链作为网络与服务的重要承载形式,通过合理编排虚拟化的网络服务功能,从而构建完整的端到端网络服务^[4]。

随着 5G 时代的来临,网络规模逐渐扩大,网络组成方式也由单域网络扩展为多域协同网络, SFC 请求通常分布部署于多个域的网络运营商中^[5]。在 5G 网络中,低时延通信作为一种典型应用场景,对多域 SFC 部署提出了降低传输时延的服务质量(Quality of Service, QoS)需求。同时为保证网络整体的稳定性,类似于单域网络,多域网络也存在负载均衡性要求,包括各域内网络及域间网络的负载均衡性。另外随着 SFC 请求规模增大,网络用户对 SFC 的部署成功率也提出了更高的要求^[6]。因此,如何对多个相关指标进行同步优化,从而优化多域网络中的 SFC 部署过程,达到相关 QoS 需求,已成为当前多域 SFC 部署领域的重点研究问题之一。

针对 SDN 多域网络中的 SFC 部署问题,现有方法主要以时延、能耗等单一指标作为优化目标,未针对多目标进行同步优化;相关多目标优化方法也并未考虑数据间的权重关系,目标权重难以衡量;且大部分方法在不同服务功能链请求数下的接受率较为稳定,不适用于大规模的 SFC 部署问题。为解决上述问题,本文提出了多目标优化间的数据归一化处理方案,以平衡权重关系,并利用强化学习框架的优势,在多次训练后能够逐渐提高 SFC 部署成功率,进而设计了一种基于强化学习的两步多目标优化 SFC 部署算法。本文的主要工作包括:

(1)完成了多域网络中 SFC 部署问题的数学建模,给出了 SFC 传输时延、网络负载均衡性及 SFC 接受率等指标的相应优化目标。

(2)提出了 SFC 传输时延及网络负载均衡方差两方面数据归一化的处理方案,平衡两类数据的数量级与影响权重。

(3)将强化学习框架应用于多域 SFC 部署问题中,设计并实现了一种两步多目标优化 SFC 部署算法,即多域强化学习算法 MDRL(Multi-Domain Reinforcement Learning)。实验结果表明,该算法在多目标优化上有一定优势。

2 相关研究

SFC 部署问题针对网络性能进行优化,主要包括时延、负载、吞吐量、SFC 构建时间及构建成功率等,优化目标主要涉及 SFC 自身性能优化以及其面向的业务性能优化两方面。自身性能优化包括降低 SFC 部署时间、提高部署成功率等;业务性能优化包括降低时延、平衡网络负载、提高网络吞吐量等^[7]。

为降低时延,文献[8]将 VNF 调度建模为混合整数线性规划问题,其考虑了 VNF 传输及处理时延,并通过启发式算法进行解决。文献[9]将时延敏感的 SFC 部署问题划分为映射、路由及调度等子问题,并依次进行解决。文献[10]在边缘计算环境下解决了最小化响应时间的 SFC 部署问题。文献[11]同时考虑了 VNF 部署及流量路由,在边缘云中解决了

全局时延最小化的 SFC 部署问题。文献[12]通过队列模型解决了端到端的 SFC 处理及传输时延最小化问题。文献[13]通过贪心算法对 SFC 处理时延、收益及能耗进行了联合优化,但其在多个目标同时优化时并未考虑权重关系,因此目标权重难以进行衡量和平衡。文献[14]在最小化资源消耗的同时,保证了 SFC 部署时端到端时延的优化目标,但其在多个目标同时优化时并未考虑权重关系,且多目标优化效果难以适应大规模 SFC 部署请求。上述研究均在一定程度上解决了 SFC 部署的时延最小化问题,但并未考虑网络负载均衡性问题。

为平衡网络负载,文献[15]通过粒子群优化算法解决了 SFC 部署的网络负载均衡性问题。文献[16]将 SFC 部署问题转换为最优路径选择问题,并在此基础上解决了网络负载不均衡问题。文献[17]提出了基于时变图聚合的服务链负载均衡方法,通过启发式算法解决了 SFC 部署中局部资源负载不均衡的问题。上述研究均在一定程度上解决了 SFC 部署的网络负载不均衡问题,但并未考虑时延最小化问题。另外,文献[8-17]均为单域网络下的 SFC 优化部署,并未考虑多域网络中网络复杂度高、网络参数多等限制条件。

在多域网络环境下的 SFC 部署问题中,文献[18]在解决多域 SFC 部署问题时,以响应时延为主要优化目标,使用竞标机制在最小化时延与资源消耗间选择最终 SFC 部署路径,同时实现子网的负载均衡。文献[19]以最小化端到端时延为目标,选择一条具有最低时延的路径以部署服务功能,从而优化面向 5G 应用的跨域 SFC 部署问题。但文献[18-19]仅针对 SFC 传输时延进行了优化,并未考虑时延、复杂均衡性和 SFC 接受率的同步优化。文献[20]在解决多域网络下 SFC 部署问题时,以提高收益-成本比率为主要优化目标,同时有效降低了 SFC 拒绝率并在一定程度上达到了节点负载均衡要求。文献[21]在优化多运营商网络服务映射问题时,主要关注最小化业务成本与最大化收益,在两者之间寻求平衡,以提高收益-成本比率。文献[22]以最小化业务成本为主要目标,同时满足负载均衡性要求,优化多运营商 SFC 部署问题。但文献[20-22]在进行最小化业务成本和提升网络负载均衡性时,并未考虑优化目标间的权重关系,因此目标权重难以进行衡量和平衡。文献[23]以最小化网络资源消耗为主要优化目标,同时考虑最小化 SDN 域数目需求,以此解决多 SDN 域下 SFC 部署问题,但其在多个目标同时优化时并未考虑权重关系,且多目标优化效果难以适应大规模 SFC 部署请求。文献[24]在解决多域 SDN 环境下 SFC 部署问题时,以最小化全局能耗为优化目标,通过多目标优化方案选择最优路径部署 VNF。文献[25]利用深度强化学习算法解决 SFC 的跨域映射问题,对传输时延、资源开销和接受率进行了同步优化,但其并未考虑多个优化目标间的权重关系。上述研究均在某些方面解决了多域 SFC 部署优化问题,但并未考虑 SFC 传输时延、网络负载均衡性及 SFC 接受率的同步优化,且在优化过程中未考虑多目标间的数量级统一化,从而导致目标权重难以平衡。

综上所述,尽管目前 SFC 部署问题已经取得了很多研究

成果,但在多域网络环境下的多目标智能化 SFC 部署依然存在不足。因此,利用强化学习方法提升 SFC 部署效果有一定的研究价值。

3 问题建模

异构多域网络中,各域传输时延及节点能力差异可能较大,会使得虚拟网络功能部署不具有同样的可行性和必要性。因此,本文在研究多域网络中 SFC 部署问题时,假设多域网络的网络结构类似,节点计算能力及链路时延数量级相当,以简化部署问题,本文以此建立多目标优化模型。

假设底层多域网络拓扑 $G=(D,V,E)$ 为无向图,其中 D 代表 SDN 域集合, V 代表节点集合, E 代表域间及域内链路集合。网络节点上主要以 vCPU 资源对 VNF 进行承载,本文假设每一个 VNF 部署时均占用一个 vCPU。为量化网络负载均衡性,以各域及各节点的 vCPU 负载百分比的方差值作为量化指标,各域内负载方差值的计算如式(1)所示:

$$L_{\text{intra}} = \sum_{i=1}^N \left(\frac{n_i}{N_i} - \frac{\sum_{j=1}^N \frac{n_j}{N_j}}{N} \right)^2 / N \quad (1)$$

其中, n_i 代表域内各节点已占用的 vCPU 数, N_i 代表该节点的 vCPU 总数, N 代表域内总节点数。同时式(1)需满足以下约束条件:

$$\begin{cases} N_u \leq N \\ \forall V_i \in V \quad u_i \leq n_i \end{cases} \quad (2)$$

其中, N_u 代表域内的非空闲节点数。式(2)要求非空闲节点数不超过域内总节点数,且对于任一节点,已占用 vCPU 数不超过该节点 vCPU 总数。

各域间的负载为域内各节点负载之和占该域总资源的百分比,域间负载方差值的计算如式(3)所示:

$$L_{\text{inter}} = \sum_{i=1}^M \left(\sum_{j=1}^N n_{ij} / \sum_{j=1}^N N_{ij} - \sum_{m=1}^M \left(\sum_{n=1}^N n_{mn} / \sum_{n=1}^N N_{mn} \right) / M \right)^2 / M \quad (3)$$

其中, n_{ij} 代表域 i 内第 j 个节点的已占用 vCPU 数, N_{ij} 代表该节点的 vCPU 总数, M 代表 SDN 域总数。

本文为优化 SFC 传输时延,在强化学习的反馈函数中引入时延指标。由于在多域网络环境下部署 SFC 请求,因此 SFC 传输时延由域内链路传输时延和域间链路传输时延两部分组成,其公式为:

$$D = \sum_{i=1}^n \sum_{j=1}^{m-1} (d_{\text{inter}}(V_{f_{i,j}}, V_{f_{i,j+1}}) + d_{\text{intra}}(V_{f_{i,j}}, V_{f_{i,j+1}})) \quad (4)$$

其中, $f_{i,j}$ 代表第 i 条 SFC 的第 j 个 VNF, $V_{f_{i,j}}$ 代表 $f_{i,j}$ 部署的节点, $d_{\text{inter}}(V_1, V_2)$ 和 $d_{\text{intra}}(V_1, V_2)$ 分别代表节点 V_1 和 V_2 间的最短域间和域内链路传输时延。

本文方法 MDRL 需对网络负载均衡性及 SFC 传输时延指标同时进行优化,为提升网络负载均衡性,MDRL 方法倾向于将 VNF 分散部署在多个节点上,以降低负载方差,同时为降低 SFC 传输时延,MDRL 方法倾向于将 VNF 部署在时延较小的节点上。因此,MDRL 方法以最小化负载方差与传输时延之和为整体优化目标。对网络负载均衡性及 SFC 传输时延分配权重系数,将两个优化目标进行组合,分别得到域间及域内的优化目标,表示为:

$$\text{Min:} \begin{cases} a * L_{\text{inter}} + b * \left(\sum_{i=1}^n \sum_{j=1}^{m-1} d_{\text{inter}}(V_{f_{i,j}}, V_{f_{i,j+1}}) / \sum_{i=1}^n f_i \right) \\ c * L_{\text{intra}} + d * \left(\sum_{i=1}^n \sum_{j=1}^{m-1} d_{\text{intra}}(V_{f_{i,j}}, V_{f_{i,j+1}}) / \sum_{i=1}^n f_i \right) \end{cases} \quad (5)$$

其中, a, b, c, d 为权重系数,同时式(5)需满足以下约束条件:

$$\begin{cases} a + b = 1 \\ c + d = 1 \\ a > 0, b > 0, c > 0, d > 0 \end{cases} \quad (6)$$

该函数以最小化加权后的负载均衡方差和时延数据为目标,能够同时降低网络域间及域内传输时延与负载均衡方差,提高负载均衡性,因此能在一定程度上对网络负载均衡性、SFC 传输时延进行同步优化。另外,为了同时优化 SFC 接受率,MDRL 方法利用强化学习框架,借助强化学习的无监督学习特点,对实验数据进行多次训练后,能明显提升算法对 SFC 的部署成功率,因此该方法能够同时对 SFC 接受率进行优化。

4 算法设计

4.1 数据归一化

为平衡两类数据的数量级,平衡不同数据对优化结果的影响权重,本文提出了对负载数据及链路传输时延数据进行归一化的处理方案。数据归一化方法如下:

(1) 负载数据归一化

在每次计算网络负载方差时,对每个域及每个节点的负载数据进行归一化。首先计算每个域及每个节点的 vCPU 负载,并分别计算出其中的最大值 $\max$ 和最小值 $\min$,再针对负载数据中的每个元素 n ,对其进行归一化处理,其表达式为:

$$n = (n - \min) / (\max - \min) \quad (7)$$

负载方差利用归一化后的负载数据进行计算。

(2) 链路传输时延数据归一化

域间和域内的链路传输时延数据保存在邻接矩阵中,在 SFC 请求到达前,先对域间及域内的时延数据进行归一化。首先计算每个邻接矩阵的最短距离矩阵,再分别计算每个矩阵的最大值 $\max$ 和最小值 $\min$,之后针对矩阵中的每个元素 n ,根据式(7)对其进行归一化处理。

4.2 算法建模

MDRL 方法利用强化学习框架解决多域 SFC 部署问题,并选择 Q-learning 算法进行建模,其域间及域内的状态集、动作集、反馈函数及 Q 矩阵更新策略如下。

(1) 状态集

本文方法主要考虑节点 vCPU 资源是否足够部署 VNF,因此将节点及其所在域分为满载和非满载两种状态,并且由于服务功能链中 VNF 间的依赖性要求,当前需要部署 VNF 的节点位置与上一节点位置相关,所以 M 个域的状态数为 2^M ,域间总状态数为 $2^M M$;假设第 i 个域的节点数为 N_i ,则第 i 个域的状态数为 2^{N_i} ,各域内总状态数为 $2^{N_i} N_i$ 。

(2) 动作集

在部署每一个 VNF 时,均需要从 M 个域中选择一个域,

再从该域中选择一个节点,并占用该节点上的一个 vCPU 资源进行部署,所以域间动作集的总动作数为域总数 M ,各域内动作集的总动作数为各域内节点数 N_i 。

(3) 反馈函数

根据问题优化目标描述,算法的反馈函数与网络负载均衡性及 SFC 传输时延相关,因此域间反馈函数表示为:

$$R_{\text{inter}} = a * (L_{\text{inter}}(S_{t+1}) - L_{\text{inter}}(S_t)) + b * (D_{\text{inter}}(S_{t+1}) - D_{\text{inter}}(S_t)) \quad (8)$$

各域内反馈函数表示为:

$$R_{\text{intra}} = c * (L_{\text{intra}}(S_{t+1}) - L_{\text{intra}}(S_t)) + d * (D_{\text{intra}}(S_{t+1}) - D_{\text{intra}}(S_t)) \quad (9)$$

其中, S_t 代表 t 时刻的状态,因此 t 时刻域间和域内的网络负载均衡性分别为 $L_{\text{inter}}(S_t)$ 和 $L_{\text{intra}}(S_t)$, SFC 传输时延分别为 $D_{\text{inter}}(S_t)$ 和 $D_{\text{intra}}(S_t)$ 。

此外,若 MDRL 算法对某一 VNF 选择部署的域或节点为满载状态,则该条 SFC 将部署失败,此时设域间反馈值 R_{inter} 或域内反馈值 R_{intra} 为 -100,为保证针对后续 VNF,尽可能避免选择使 SFC 部署失败的域和节点,提升 SFC 部署的成功率。

(4) Q 矩阵更新策略

由于本文选择 Q-learning 算法解决强化学习问题,该算法在学习前期缺少经验反馈,需要进行多次随机探索,在学习后期则倾向于根据 Q 矩阵选择动作,因此为该算法设定一个探索概率阈值 ϵ ,其公式为:

$$\epsilon = 10 / \text{count} \quad (10)$$

其中, count 表示部署 VNF 并更新 Q 矩阵的次数。由于 MDRL 算法分为域选择及域内节点选择两步实现,因此域间及各域内的 count 值独立计数。Q-learning 算法在每次选择动作时,以 ϵ 的概率随机选择域及域内节点部署 VNF,以 $1-\epsilon$ 的概率根据 Q 矩阵选择域及域内节点部署 VNF。

在选定动作后, Q-learning 算法执行选择的动作并跳转至下一状态,并得到反馈值 R ,同时根据式(11)更新公式对 Q 矩阵进行更新:

$$Q(S_t, \text{action}) = R(S_t, \text{action}) + \gamma * \max[Q(S_{t+1}, \text{action})] \quad (11)$$

其中, action 代表选择的动作,折扣因子 γ 设为 0.8。

4.3 MDRL 算法的伪代码

本文算法的主要流程为:对每条服务功能链 f_i 部署请求,首先释放生命周期已结束的 SFC 占用的网络资源,之后选择负载最小的 SDN 域并在域中选择负载最小的节点部署头 VNF $f_{i,1}$ 。对于后续 VNF,通过两步算法进行部署。第一步,以随机或 Q 矩阵方式选择 SDN 域以预留资源,同时计算反馈值 R_{inter} 并更新域间 Q_{inter} 矩阵。第二步,判断该域与上一 VNF 部署域是否相同,若不同,则在该域中选择负载最小的节点部署 VNF;若相同,则以随机或 Q 矩阵方式在该域中选择节点部署 $f_{i,j}$,同时计算反馈值 R_{intra} 并更新域内 Q_{intra} 矩阵。本文算法的伪代码如算法 1 所示。

算法 1 基于强化学习的两步 SFC 部署算法

输入:网络状态及 SFC 请求

输出:SFC 中各 VNF 部署节点位置

1. 初始化域间及域内 Q 矩阵为零矩阵
2. for (每一条 SFC 部署请求)
3. 释放生命周期已结束的 SFC
4. 在负载百分比最小的域中选择负载百分比最小的节点部署头 VNF
5. for (后续 VNF 部署请求)
6. STEP 1:
7. 对各域负载及域间传输时延数据进行归一化
8. 选择 Q 值最大的 SDN 域并预留资源
9. 计算反馈值并更新域间 Q 矩阵
10. STEP 2:
11. if 当前域与上一部署域不同
12. then 在当前域中选择负载最小的节点部署 VNF
13. else 对当前域内各节点负载及节点间传输时延数据进行归一化
14. 在当前域中选择 Q 值最大的节点部署 VNF
15. end
16. end
17. end

5 算法仿真与结果分析

5.1 时间复杂度

假设网络域数为 M ,网络总节点数为 N ,传统多域 SFC 部署方法通过一步方法直接选择合适的节点对 VNF 进行部署,因此需要遍历所有节点,探索节点的时间复杂度为 $O(N)$ 。而本文所设计的两步部署方法首先选择合适的网络域,之后在该网络域中选择合适的节点对 VNF 进行部署,不需要一次性遍历所有节点,该算法探索网络域的时间复杂度为 $O(M)$,假设该域中节点数为 n ,则探索节点的时间复杂度为 $O(n)$,总时间复杂度为 $O(M) + O(n)$ 。由于一般多域网络中 M 和 n 远小于 N ,因此 $O(M) + O(n) < O(N)$,本文算法的探索时间开销有所保证。

5.2 仿真环境

本文通过 MATLAB 软件实现所提算法与对比算法的仿真。MDRL 算法主要分为三大模块:1)网络拓扑和节点资源的模拟生成;2)SFC 的生命周期管理;3)基于强化学习的两步 SFC 多域部署方案。

(1) 网络拓扑模拟

网络拓扑使用随机拓扑生成算法产生,SDN 域数设为 5,各域内节点数分别随机设为 5,4,5,6,4,算法根据 SDN 域数和节点数产生随机拓扑及其邻接矩阵。在 Floyd 算法中输入该邻接矩阵,计算得到任意两节点间最短距离,并设为传输时延权重。各节点的 vCPU 数从[4,7]中随机选取,并设定每个 VNF 占用一个 vCPU。本次实验随机生成的域间传输时延权重矩阵为:

$$D = \begin{bmatrix} 0 & 2 & 1 & 1 & 2 \\ 2 & 0 & 3 & 1 & 1 \\ 1 & 3 & 0 & 2 & 3 \\ 1 & 1 & 2 & 0 & 1 \\ 2 & 1 & 3 & 1 & 0 \end{bmatrix}$$

各域内节点间的传输时延权重矩阵为:

$$D1=\begin{bmatrix}0 & 2 & 4 & 3 & 1 \\ 2 & 0 & 2 & 1 & 1 \\ 4 & 2 & 0 & 1 & 3 \\ 3 & 1 & 1 & 0 & 2 \\ 1 & 1 & 3 & 2 & 0\end{bmatrix}, D2=\begin{bmatrix}0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0\end{bmatrix},$$
$$D3=\begin{bmatrix}0 & 3 & 3 & 2 & 1 \\ 3 & 0 & 2 & 1 & 2 \\ 3 & 2 & 0 & 1 & 2 \\ 2 & 1 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0\end{bmatrix}, D4=\begin{bmatrix}0 & 3 & 1 & 2 & 2 & 1 \\ 3 & 0 & 2 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 1 & 2 \\ 2 & 2 & 1 & 1 & 0 & 2 \\ 1 & 3 & 1 & 2 & 2 & 0\end{bmatrix}$$
$$D5=\begin{bmatrix}0 & 3 & 2 & 1 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0\end{bmatrix}$$

各域内节点的 vCPU 数为：

$$N1=[4 \ 6 \ 5 \ 5 \ 4], N2=[5 \ 6 \ 6 \ 7],$$
$$N3=[4 \ 4 \ 5 \ 7 \ 5], N4=[6 \ 5 \ 6 \ 4 \ 5 \ 7],$$
$$N5=[4 \ 5 \ 4 \ 6]$$

(2)SFC 生命周期管理

假设 SFC 长度从[3,6]中随机选取,设定任意一条 SFC 的生命周期为 20 条 SFC 请求到达的总时间,即当第 i 条 SFC 请求到达时,释放第 $i-20$ 条 SFC 占用的 vCPU,以此保证网络中存在适量的空闲资源以部署后续 SFC。

(3)基于强化学习的两步 SFC 多域部署算法仿真

本文利用强化学习 Q-learning 算法完成 VNF 的部署仿真,针对每条 SFC 部署请求,从头 VNF 开始依次部署后续 VNF,每完成一次 SDN 域选择,即对 Q_{inter} 矩阵进行一次更新,每在当前域中完成一次 VNF 部署,即对该域的 Q_{intra} 矩阵进行一次更新;通过对不同数量的 SFC 请求进行仿真,完成对本文算法与对比算法的评估。

(4)参数设定

本文中式(5)为优化目标,式(8)、式(9)为反馈函数,并通过数据归一化方法平衡了域间和域内的负载均衡方差及传输时延数据间的数量级,因此,根据式(6)设权重系数 a, b, c, d 均为 0.5,以此保证算法对负载均衡性及传输时延的均衡优化效果。

5.3 对比算法

本文共对比分析以下 4 种算法。

MDRL:基于强化学习的两步 SFC 多域部署算法,即本文算法。

GREEDY:贪心算法,为有效对比 MDRL 算法性能,本文将 GREEDY 算法的贪心目标调整为传输时延与网络负载的均衡性优化^[13]。

LASP:时延感知服务部署算法^[14]。

MDSP:多域 SDN 网络服务部署算法^[23]。

实验中将 3 种对比算法应用于本文设定的多域网络,并对相关逻辑进行调整以适应多域网络中的 SFC 部署问题,保证其优化目标不变。

5.4 评价指标

本文提出的算法为多目标算法,以传输时延、负载均衡性和 SFC 接受率作为多目标优化目标。

(1)SFC 平均传输时延

为方便测量,在本实验中为域间及域内链路的传输时延设置权重。由于实验中各条 SFC 长度不一,因此本文以相邻 VNF 间的平均传输时延权重进行衡量。权重越小表明 SFC 传输时延越小,权重越大表明 SFC 传输时延越大。

(2)网络负载均衡性

在所有 SFC 部署请求处理完成后,得到各域及各节点的已占用 vCPU 数,并分别计算域间及各域内的负载方差,方差越大表明网络负载均衡性越差,方差越小表明网络负载均衡性越好。

(3)SFC 接受率

在所有 SFC 部署请求处理完成后,计算部署成功的 SFC 数占总请求数的比例,从而得到算法的 SFC 接受率。接受率越高表明算法稳定性越好;接受率越低表明算法稳定性越差。

5.5 实验结果

实验中分别随机产生 500,1000,1500,2000,3000,4000,6000,8000,10000 条 SFC 请求,并得到各自请求数下的平均传输时延、网络负载均衡性及 SFC 接受率指标值。

(1)平均传输时延结果

MDRL 算法与 3 种对比算法的平均传输时延结果的对比如图 1 所示。对比结果表明,MDRL 算法在较高的 SFC 请求数下的平均传输时延最小,因为该算法根据强化学习反馈值,更倾向于选择距上一 VNF 部署节点时延最低的域及节点部署当前 VNF,所以 SFC 平均传输时延最低。LASP 算法的效果仅次于 MDRL 算法,因为 LASP 算法主要以时延作为优化目标,但其主要考虑 SFC 处理时延而非 SFC 传输时延,所以平均传输时延较高。GREEDY 算法以传输时延及负载均衡性为标准,均选择效果最好的域及节点部署下一 VNF,因此其平均传输时延结果次于 LASP 算法但优于 MDSP 算法。MDSP 算法效果最差,因为该算法并未以传输时延作为优化指标,整体平均传输时延最高。

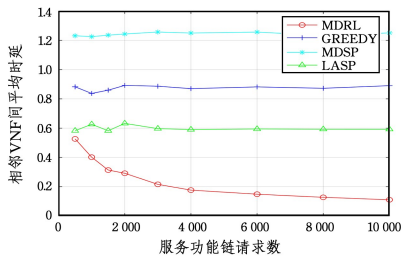


图 1 4 种算法在不同请求数下相邻 VNF 间的平均时延
Fig. 1 Average delay between adjacent VNF of four algorithms under different number of requests

MDRL 算法在不同 SFC 请求数下的平均传输时延结果如图 2 所示。结果表明,随着请求数增大,平均传输时延呈递减趋势,优化效果显著。因为 MDRL 算法是以强化学习为算法框架,训练数据越多,算法得到的效果越好,所以整体 SFC 传输时延越低,且由于强化学习方法在学习前期会以较大概

率随机选择动作部署 VNF,因此在请求数较少时,该算法的时延优化效果较差,但在进行一定时间的学习之后,该算法在高请求数下的时延优化效果得到显著提升。

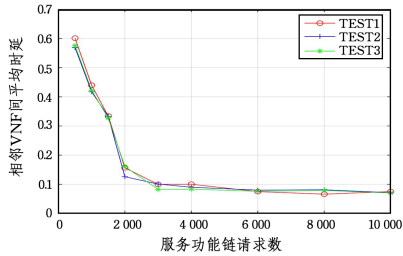


图 2 MDRL 方法中不同请求数下相邻 VNF 间的平均时延
Fig. 2 Average delay between adjacent VNF under different number of requests in MDRL method

(2)网络负载均衡性结果

MDRL 算法与 3 种对比算法的网络负载均衡性结果如图 3、图 4 所示。图 3、图 4 的结果表明,MDRL 算法的域间及各域内负载均衡方差均为最低,因为 MDRL 算法在强化学习反馈值中考虑了负载均衡因素,使得不同 SFC 部署时倾向于分别部署在不同的域及节点上,所以网络负载均衡方差最低,网络负载均衡性最好。GREEDY 算法的效果仅次于 MDRL 算法,因为 GREEDY 算法在部署时将负载均衡作为考虑因素,所以网络负载均衡性效果较好。MDSP 算法与 LASP 算法并未考虑到负载均衡性因素,因此二者的网络负载均衡性容易出现波动,且负载均衡性相对较差。

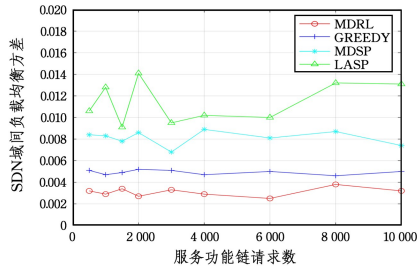


图 3 不同请求数下的域间负载均衡方差
Fig. 3 Inter-domain load balancing variance under different number of requests

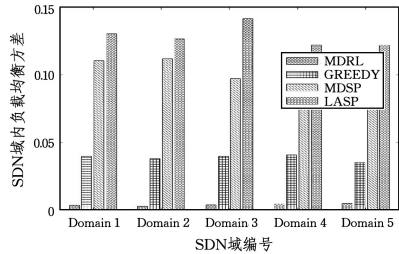


图 4 不同请求数下的域内负载均衡方差
Fig. 4 Intra-domain load balancing variance under different number of requests

(3)SFC 接受率结果

MDRL 算法与 3 种对比算法的 SFC 接受率对比结果如图 5 所示。结果表明,MDRL 算法的 SFC 接受率在不同 SFC 请求数下呈递增趋势,且在较大规模的请求数下接受率最高。

因为 MDRL 算法倾向于选择反馈值 R 最大的域及节点部署 VNF,且随着训练次数的增加,算法会尽量避免选择反馈值 R 为一 100 的域及节点,即算法会更加倾向于成功部署各个 VNF,所以 SFC 接受率呈递增趋势,且在较大规模的请求数下效果最好。MDSP, GREEDY, LASP 3 种对比算法的 SFC 接受率在不同请求数下较稳定,因为这 3 种算法并未利用机器学习框架,算法缺乏强化训练效果,所以接受率结果不会随着请求数的增大而提高。另外, GREEDY 算法及 LASP 算法以时延为主要优化目标之一, LASP 算法倾向于将 VNF 集中部署于同一域内的同一节点上,这会造成网络负载不均衡,而 GREEDY 算法同时考虑了负载均衡性,因此 LASP 算法的接受率结果最差, GREEDY 算法的接受率结果略优于 LASP 算法。

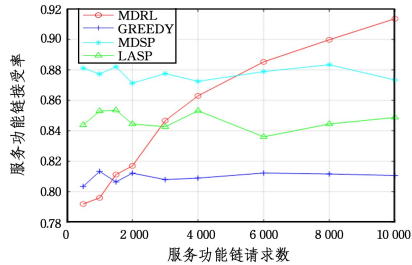


图 5 不同请求数下的 SFC 接受率
Fig. 5 SFC acceptance rate under different number of requests

结束语 本文主要研究了多 SDN 域网络环境下 SFC 部署问题,提出了数据归一化方案,并设计了一种基于强化学习的两步 SFC 部署算法,对多个评价指标进行了同步优化,解决了多域网络中 SFC 部署的多目标优化问题。仿真结果表明,本文算法的 SFC 传输时延、网络负载均衡性及 SFC 接受率等指标在较大规模的 SFC 请求数下取得了一定的优化效果。

本文在实验仿真中以随机方式生成网络拓扑,链路传输时延及各节点资源数同样随机设置。并且由于在实际情况下运营商的各种 SFC 部署请求中,各服务器部署位置随机且网络参数差异较大,因此该仿真具有一定的参考价值。

本文研究仍存在不足之处。第一,实验是在 MATLAB 平台进行模拟仿真,缺少实际网络环境下的实验结果。下一步研究将在实际网络环境中,将算法部署于物理硬件上进行实验分析,以优化得到更符合实际情况的实验结果。第二,本文使用强化学习方法解决 SFC 部署问题,在 SDN 域数及节点数较大时,会导致算法的状态集过大,算法性能极低,因此本文方法不适用于网络规模较大的情况。针对该问题,将提出简化状态集的工作思路,利用深度强化学习,将大规模的离散状态集进行简化,在大规模多域网络中解决 SFC 部署优化问题。

参 考 文 献

[1] YI B, WANG X W, LIK Q, et al. A comprehensive survey of Network Function Virtualization [J]. Computer Networks, 2018, 133: 212-262.
[2] JOSHI K, BENSON T. Network Function Virtualization [J].

IEEE Internet Computing, 2016, 20(6):7-9.

[3] HALPERN J, PIGNATARO C. Service Function Chaining Architecture, document RFC 7665 of the IETF Service Function Chaining Working Group [EB/OL]. <http://datatracker.ietf.org/doc/rfc7665/>.

[4] LI Y, CHEN M. Software-defined network function virtualization: a survey[J]. IEEE Access, 2015, 3:2542-2553.

[5] BERNINI G, GIARDINA P G, SPADARO S, et al. Multi-Domain Orchestration of 5G Vertical Services and Network Slices [C]//2020 IEEE International Conference On Communications Workshops. Dublin, Ireland, 2020:6.

[6] WIBOWO F X A, GREGORY M A, AHMED K, et al. Multi-domain Software Defined Networking: Research status and challenges[J]. Journal of Network and Computer Applications, 2017, 87:32-45.

[7] CHEN W H, YIN X, WANG Z L, et al. Placement and Routing Optimization Problem for Service Function Chain: State of Art and Future Opportunities[J]. arXiv:1910.02613.

[8] QU L, ASSI C, SHABAN K. Delay-Aware Scheduling and Resource Optimization With Network Function Virtualization[J]. IEEE Transactions on Communications, 2016, 64(9):3746-3758.

[9] ALAMEDDINE H A, QU L, ASSI C. Scheduling Service Function Chains for Ultra-Low Latency Network Services[C]//13th International Conference on Network and Service Management. Tokyo, Japan, 2017:9.

[10] SUN G, LI Y Y, LI Y, et al. Low-latency orchestration for workflow-oriented service function chain in edge computing[J]. Future Generation Computer Systems-the International Journal of Science, 2018, 85:116-128.

[11] GOUAREB R, FRIDERIKOS V, AGHVAMI A H. Virtual Network Functions Routing and Placement for Edge Cloud Latency Minimization[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(10):2346-2357.

[12] YE Q, ZHUANG W H, LI X, et al. End-to-End Delay Modeling for Embedded VNF Chains in 5G Core Networks[J]. IEEE Internet of Things Journal, 2019, 6(1):692-704.

[13] MIJUMBI R, SERRAT J, GORRICO J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions[C]//2015 1st IEEE Conference on Network Softwarization. London, UK, 2015:9.

[14] ALLEG A, AHMED T, MOSBAH M, et al. Delay-aware VNF placement and chaining based on a flexible resource allocation approach[C]//2017 13th International Conference on Network and Service Management. Tokyo, Japan, 2017:7.

[15] SHI Z, WU Z H, ZENG Y. A Method of Service Function Chain Arrangement for Load Balancing[C]//9th International Conference on Computer Engineering and Networks. Changsha, China, 2019:35-42.

[16] HAN H Y, MENG X R, YU Z H, et al. A Service Function Chain Deployment Method Based on Network Flow Theory for

Load Balance in Operator Networks[J]. IEEE Access, 2020, 8:93187-93199.

[17] XIANG Y F, WU M, WU J, et al. A Load Balancing Method of Virtualization Service Function Chain Based on Time-varying Graphs Integration[J]. Journal of Fujian Normal University (Natural Science Edition), 2018, 34(3):14-20.

[18] SUN G, LI Y, LIAO D, 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.

[19] XU Q, GAO D Y, LI T X, et al. Low Latency Security Function Chain Embedding Across Multiple Domains[J]. IEEE Access, 2018, 6:14474-14484.

[20] LI G L, ZHOU H C, FENG B H, et al. Context-Aware Service Function Chaining and Its Cost-Effective Orchestration in Multi-Domain Networks[J]. IEEE Access, 2018, 6:34976-34991.

[21] DIETRICH D, ABUJODA A, RIZK A, et al. Multi-Provider Service Chain Embedding With Nestor[J]. IEEE Transactions on Network And Service Management, 2017, 14(1):91-105.

[22] ABUJODA A, PAPADIMITRIOU P. DistNSE: Distributed Network Service Embedding Across Multiple Providers[C]//8th International Conference on Communication Systems And Networks. Bangalore, India, 2016:8.

[23] ZHANG C, WANG X W, LI F W, et al. Network Service Chains Deployment Across Multiple SDN Domains[J]. International Journal of Communication Systems, 2018, 31(18):e3826.1-e3826.25.

[24] KAUR K, GARG S, KADDOUM G, et al. An Energy-driven Network Function Virtualization for Multi-domain Software Defined Networks[C]//IEEE Conference on Computer Communications. Paris, France, 2019:121-126.

[25] ZHU G H, LI Q, LIANG S L. Cross-domain mapping algorithm of service function chain based on deep reinforcement learning [J]. Application Research of Computers, 2021, 38(6):1834-1837, 1842.



WANG Ke, born in 1992, Ph.D. His main research interests include software-defined network, network function virtualization and service function chain technology.



QU Hua, born in 1982, Ph.D, professor, is a member of China Computer Federation. His main research interests include mobile internet, network protocol design, control strategies for supporting emerging applications in ubiquitous networks,

and radio resource management in 5G radio communications systems.