

基于 VNF 实例共享的服务功能链部署算法*

王媛滔, 舒兆港[†], 钟一文, 邱彩钰, 田佳霖

(福建农林大学 计算机与信息学院, 福州 350002)

摘要: 针对服务功能链(SFC)部署过程中存在虚拟网络功能(VNF)实例部署成本和转发路径成本难以权衡的问题,提出了基于 VNF 实例共享的 SFC 部署算法。首先针对多链 SFC 建立 VNF 和虚拟链路映射模型,并预估路径部署长度上限,保证 SFC 时延需求;其次,在路径部署长度限制范围内,尽可能使 VNF 实例共享最大化,以平衡链路转发成本和 VNF 部署成本,最终得到 SFC 部署策略。与已有的 SPH(shortest path heuristic)和 GUS(greedy on used server)部署算法相比,所提算法所得的总运营成本分别降低 6.6% 和 12.15%,且当 SFC 数量增多时,该算法的服务接受率可达 89.33%。仿真实验结果表明,提出算法可以在保证用户服务质量的同时有效降低 SFC 部署成本。

关键词: 服务功能链; 虚拟网络功能; VNF 实例共享; 成本优化

中图分类号: TP393.01

文献标志码: A

文章编号: 1001-3695(2023)06-033-1806-06

doi:10.19734/j.issn.1001-3695.2022.11.0564

Service function chain placement algorithm based on VNF instance sharing

Wang Yuantao, Shu Zhaogang[†], Zhong Yiwen, Qiu Caiyu, Tian Jialin

(College of Computer & Information Sciences, Fujian Agriculture & Forestry University, Fuzhou 350002, China)

Abstract: This paper proposed an SFC(service function chaining) placement algorithm based on VNF(virtual network function) instance sharing to address the problem of difficult tradeoff between the deployment cost of VNF instances and forwarding cost in the process of SFC placement. Firstly, for multi-link SFC, it constructed the VNF and virtual link mapping model, and estimated the upper bound of the path deployment length to ensure the SFC delay. Secondly, within the limit of the path length, it maximized the number of the reused VNF instance as much as possible to balance the forwarding cost and the deployment cost of VNF. Finally, it obtained the SFC placement strategy. Compared with the existing SPH(shortest path heuristic) and GUS(greedy on used server) placement algorithms, the total operating expense of the proposed algorithm is 6.6% and 12.15% lower, and when the number of SFC increases, the service acceptance rate of this algorithm can reach 89.33%. The simulation experimental results show that this algorithm can effectively reduce the SFC placement cost while ensuring the quality of service.

Key words: service function chain; virtual network function; VNF instance sharing; cost optimization

0 引言

在传统网络中,物理设备是一个承担着不同网元功能的黑盒子,其配置的灵活性较低。当前 5G/6G 网络需要为垂直行业灵活地提供各种类型的网络服务,而传统的网络管理架构显得力不从心。随着软件定义网络(software defined networking, SDN)^[1]和网络功能虚拟化(network function virtualization, NFV)技术的出现,大大缓解了这一问题。SDN 和 NFV 可以将网络功能灵活地部署在云服务的虚拟机上^[2],以实现低成本和灵活的网络管理^[3]。以上管理模式中,虚拟机中的网络功能称为虚拟网络功能(virtual network function, VNF),而服务功能链(service function chain, SFC)是由多个 VNF 按一定顺序连接的 VNF 实例集合(如防火墙、WAN 优化器、负载均衡器和网络地址转换器),从而形成一个 VNF 转发图(virtual network function-forwarding graph, VNF-FG)^[4]。因此,当用户请求网络服务时,网络服务提供商可以为该用户在虚拟机上部署对应的 SFC;当服务结束后,则可以将虚拟机资源进行释放,既增加了管理的灵活性,又降低了运营成本。

虽然 SFC 方案有效降低了网络服务提供商的运营成本,但 VNF 的部署和编排^[5]仍然有很多问题需要解决。主要体现在:在保证用户网络的服务质量(quality of service, QoS)的前提下(如满足时延、带宽需求),如何选择 VNF 的部署位置和共享方案以进一步优化运营成本。运营成本包括部署 VNF 实例的成本和链路转发成本两部分。目前大部分研究只考虑 VNF 实例成本,或只考虑链路转发成本,且在研究 VNF 实例共享方案时只考虑了 SFC 为单链的场景。因此,如何综合考虑以上约束条件和多链场景,并兼顾 QoS 需求,进一步优化运营成本,是当前研究的一大挑战,也是本文的主要研究目标。

对于用户的时延需求,文献[6]将两个 VNF 之间的跳数作为传输延迟;文献[7]将物理链路端到端传输延迟的总和视为请求的延迟;文献[8]通过将 VNF 实例集中在更少的物理节点上,从而使请求流通过更少的物理链路以降低传输时延;文献[9]提出一种基于深度强化学习的 SFC 映射算法,降低网络平均时延和部署失败率;文献[10]通过避免物理链路上的流量过载情况来降低请求延迟。然而,以上研究只考虑了 SFC 为单链的情况,很少有研究考虑其他类型的 VNF-FG。文献[11]

收稿日期: 2022-11-11; **修回日期:** 2023-01-09 **基金项目:** 福建省自然科学基金资助项目(2020J01574);中国高校产学研创新基金资助项目(2021FNA05003)

作者简介: 王媛滔(1999-),女,安徽合肥人,硕士研究生,主要研究方向为网络功能虚拟化;舒兆港(1981-),男(通信作者),江西鄱阳人,副教授,工学博士,主要研究方向为软件定义网络与网络功能虚拟化、机器学习与智能化网络体系结构、网络切片、云计算与边缘计算等(zgshu@fafu.edu.cn);钟一文(1968-),男,福建上杭人,教授,硕士,工学博士,主要研究方向为计算智能和计算机应用;邱彩钰(2000-),女,福建莆田人,硕士研究生,主要研究方向为软件定义网络;田佳霖(1997-),男,山东桓台人,硕士研究生,主要研究方向为软件定义网络。

考虑了多播网络服务的 VNF 放置,但是其中 VNF-FG 是仅具有一个源端和几个目的端的特例,且没有考虑延迟的需求。本文将为每个请求考虑基于 VNF-FG 部署的通用模型,并在 VNF-FG 映射到实际物理网络时考虑用户的延迟需求。

在有关增大经济效益或降低成本作为优化目标的研究中,文献[12]建立基于马尔可夫决策的随机优化模型,优化 SFC 部署成本和端到端时延;文献[13]通过最小化活动虚拟机的数量,降低了部署成本;文献[14]提出了一种基于加权图匹配的匹配算法以提高资源利用率,并通过 VNF 迁移的方法关闭空闲服务器,以降低服务器成本;文献[15]针对实时 VNF 部署和迁移,使用供需模型量化 VNF 干扰,提出一种实时自适应感知算法以最大化总收益;文献[16]提出一种在线扩展算法,调整 VNF 实例部署以平衡 VNF 运营成本、VNF 实例维护成本和 VNF 部署成本,但没有考虑请求时延,忽略了用户的 QoS;文献[17]提出基于时间窗的虚拟网络批处理映射策略,以减少虚拟网络映射成本和功耗;文献[18]通过减少部署网络服务的计算和带宽资源消耗来降低运营成本;文献[19]提出了一种新的 VNF 部署策略,解决了 VNF 部署位置不确定性导致的带宽资源浪费。

以上研究均为单个目标优化。在最小化 SFC 端到端时延没有考虑 VNF 实例的部署成本,这增大了运营成本,因为为了满足时延需求,VNF 实例尽可能地部署在源端和目的端的最短路径上,部分 VNF 实例所占据的服务器资源并没有得到合理的利用。如果服务器的计算能力足以支持 VNF 需求,则不同服务链上相同类型的 VNF 可以共享该 VNF 实例^[20],提高资源利用率,减少 VNF 实例的部署成本,但这也带来转发路径的增长,从而导致流量开销变大的问题,同时可能无法满足 SFC 的时延要求。由此可见,链路分配和 VNF 实例部署之间的相互作用在 VNF 编排中起着重要作用。随着网络状态的变化,在满足资源需求和服务质量(如时延)的前提下,如何平衡 VNF 实例部署成本和链路转发成本是目前亟待解决的问题。

目前 SFC 部署问题在现有工作中可以定义为具有不同目标的优化问题,为此提出了不同的解决方案。现有工作中,精确求解方案可以提供最优解,但其很大程度上受网络规模的影响,如文献[21]利用 CPLEX 等计算工具得到小规模网络下的精确求解,但在大规模网络下运算会导致执行时间过长。而启发式算法可以在短时间内提供近似最优解,且通常不受网络规模的影响。现有工作为了实现优化目标,通常会将线性规划,如整数线性规划(integer linear programming, ILP)或者混合整数线性规划(mixed integer linear programming, MILP),结合启发式算法进行求解。例如,文献[22]建立了整数线性规划模型,在有线和无线混合网场景下,基于资源感知的节点选取算法和带宽最小的虚拟链路映射方案,提出了一种面向移动边缘计算网络的 SFC 高效可靠的部署方法。文献[23]为了实现服务功能链的低成本高效率部署,提出了一种基于改进麻雀搜索算法的服务功能链优化映射算法,但其容易陷入局部最优。文献[24]提出了一个联合 SFC 部署和路由的优化问题。鉴于该问题是 NP-hard,设计了一种候选路径选择的近似算法,提升了求解效率。文献[9,25]提出一种基于深度强化学习的 SFC 映射算法,通过适当的奖励设计、状态和动作空间公式来决定服务链中各 VNF 的部署位置。但是在采用智能学习算法解决 SFC 部署问题时,现有工作的状态空间定义较为简单,但在实际情况下需要考虑的特征变量很多,这也会导致状态空间求解复杂度提升。

不同于已有的研究工作,本文基于 VNF 实例共享,综合考虑链路分配和 VNF 部署的问题,提出了一种 SFC 部署和路径选择算法以最小化运营成本。此外,链路分配(就路径长度而

言)和 VNF 实例部署(就共享 VNF 实例的数量而言)之间的关系在部署中是不稳定的,且在多目标多约束下求解部署策略已被证明为 NP-hard 问题^[26]。本文的主要贡献有:a)针对 VNF-FG 构建了一个基于 VNF 实例共享的 SFC 部署模型,并利用 ILP 模型对问题进行建模,在满足用户 QoS 要求的前提下最小化运营成本;b)考虑到实例部署成本和链路转发成本之间的矛盾性,提出了基于 VNF 实例共享的服务功能链部署算法,通过预估路径长度上限并在路径长度约束下最大化可共享的 VNF 实例数量,以平衡链路和 VNF 实例的部署,从而最小化运营成本。

1 问题描述与系统模型

1.1 问题描述

如图 1(a)所示,有两条 SFC(SFC_1 和 SFC_2),在不考虑 VNF 实例共享下, SFC_1 有两条转发路径,分别为 $A \rightarrow B \rightarrow E \rightarrow F \rightarrow C \rightarrow D$ 和 $A \rightarrow B \rightarrow C \rightarrow D$, SFC_2 经过节点 $I \rightarrow J \rightarrow G \rightarrow H$ 。可以发现两条 SFC 中都存在 VNF_2 ,所以可以复用 VNF_2 实例,将节点 J 上的 VNF_2 部署到节点 F 上,如图 1(b)所示,则 SFC_2 经过节点 $I \rightarrow F \rightarrow G \rightarrow H$,此时部署成本降低。但这样会存在某些 SFC 在部署 VNF 实例时,为了到达共享的 VNF 实例会映射更长的物理路径,增加了带宽资源的使用。如图 2(a)所示,在不考虑 VNF 实例共享下, SFC_1 经过节点 $A \rightarrow B \rightarrow E \rightarrow F \rightarrow C \rightarrow D$ 和 $A \rightarrow B \rightarrow C \rightarrow D$, SFC_2 经过节点 $I \rightarrow J \rightarrow K \rightarrow L$;但在图 2(b)中,若将 VNF_2 共享后, SFC_2 经过节点 $I \rightarrow F \rightarrow G \rightarrow H \rightarrow K \rightarrow L$,增加了链路带宽资源的消耗,同时还违反了 SFC 的时延要求。如何平衡链路和 VNF 实例之间的关系,是目前需要解决的问题。

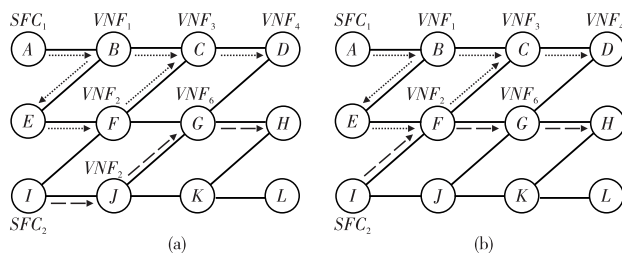


图1 无 VNF 实例共享和有 VNF 实例共享
Fig. 1 No sharing VNF instances and sharing VNF instances

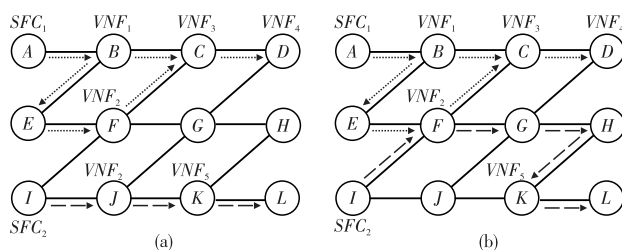


图2 VNF 实例共享引发链路增长
Fig. 2 VNF instance sharing causes link growth

1.2 系统模型

1.2.1 底层物理网络模型

底层物理网络为无向连通图,用 $G = (N, L)$ 表示, N 表示物理节点的集合,可以进行 VNF 实例化, L 表示物理链路集合,节点 $u \in N$ 的 CPU 计算资源为 C_u ,限制了共享 VNF 实例的 SFC 数量。 $B_{u,v}$ 表示连接节点 u, v 的链路 $l_{uv} \in L$ 的可用带宽资源; τ_{uv} 表示链路 $l_{uv} \in L$ 的传输延迟。

1.2.2 服务功能链

在模型中,所有种类的 VNF 集合由 F 表示,SFC 集合由 I 表示。其中 SFC 请求 $i(i \in I)$ 由一个五元组 $(S_i, T_i, G_i, D_i, F_i)$ 表示, S_i 表示源端, T_i 表示目的端, G_i 表示 SFC 的 VNF-FG, D_i

表示 SFC 允许的最大延迟, F_i 表示构成 SFC 的 VNF 集合。相同类型的 VNF 在不同 SFC 中所需资源 $r_j^i (j \in F)$ 不同。

对于 VNF-FG, 与文献[27]相同, 本文考虑三种基本拓扑, 如图 3 所示, 即单链、具有不同目的端的拆分和具有单个目的端的拆并。利用经典的图搜索算法, 如深度优先搜索可以很轻松地构建 VNF-FG 的转发路径(virtual network function-forwarding path, VNF-FP)集合。SFC 请求 i 的 VNF-FP 的集合用 $\mathcal{F}_i (i \in I)$ 表示, 而 i 的第 j 条转发路径为 $f_{i,j} \in \mathcal{F}_i$, 如图 1(a) 中 SFC_1 所需的转发路径为 $\mathcal{F}_1 = (f_{1,1}, f_{1,2})$, 其中 $f_{1,1} = (l_{AB}, l_{BC}, l_{CD})$, $f_{1,2} = (l_{AB}, l_{BE}, l_{EF}, l_{FC}, l_{CD})$ 。其中第 z 条虚拟链路为 $f_{i,j,z} \in f_{i,j}$, 它所对应的真实物理转发路径存储在 $P_{f_{i,j,z}}$ 中, $b_{f_{i,j,z}}$ 表示虚拟链路 $f_{i,j,z}$ 所需的带宽资源。第 j 条转发路径对应的真实物理转发路径集合为 $P_{f_{i,j}} \in P_{f_{i,j}}$, $F_{f_{i,j}}$ 为第 j 条转发路径上所需的 VNF 集合。

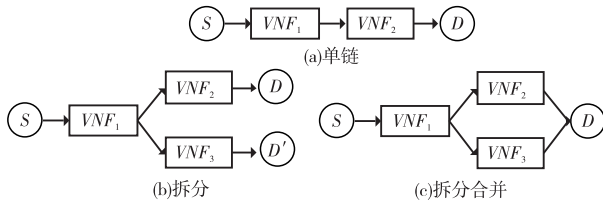


图 3 基本拓扑
Fig. 3 Basic topology

本文涉及的主要参数及意义如表 1 所示。

表 1 参数及意义

Tab. 1 Parameters and description

符号	说明
$G = (N, L)$	底层物理网络
N	可用的物理节点的集合
L	物理链路集合
C_u	节点 $u \in N$ 的计算资源
B_{uv}	链路 $l_{uv} = (u, v) \in L$ 的可用带宽资源
τ_{uv}	链路 $l_{uv} = (u, v) \in L$ 的传输延迟
F	所有种类的 VNF 集合
I	SFC 请求集合, $ I $ 为请求数量
F_i	i 的 VNF 集合
G_i	i 的 VNF-FG
S_i	i 的源端
T_i	i 的目的端
D_i	i 允许的最大延迟
$\mathcal{F}_i$	i 中转发路径的集合, $ \mathcal{F}_i $ 为转发路径数量
$F_{f_{i,j}}$	i 的第 j 条转发路径上 VNF 的集合, $ F_{f_{i,j}} $ 为转发路径上的 VNF 数量
r_j^i	i 中 VNF_j 所需的计算资源
$f_{i,j}$	i 的第 j 条转发路径
$f_{i,j,z}$	i 的第 j 条转发路径的第 z 条虚拟链路
$P_{f_{i,j,z}}$	$f_{i,j,z}$ 所对应的真实物理转发路径
$P_{f_{i,j}}$	第 j 条转发路径对应的真实物理转发路径集合, $ P_{f_{i,j}} $ 为集合中转发路径数量
$b_{f_{i,j,z}}$	虚拟链路 $f_{i,j,z}$ 所需的带宽资源
V_l	物理链路上使用单位带宽的成本
V_{cost}	部署一个 VNF 所用的成本
z_{iju}	二元变量, i 中 VNF_j 映射在节点 u 上, 则为 1
x_u	二元变量, 物理节点 u 已被实例化 VNF, 则为 1

1.2.3 运营成本

当 i 中第 j 个 VNF 实例化在物理节点 u 时, 记 $z_{iju} = 1$; 当物理节点 u 已有 VNF 实例, 则为 $x_u = 1$; V_{cost} 表示启用一个物理节点所用的成本, V_l 表示在物理链路上使用单位带宽的成本。则在物理节点 u 实例化 VNF 的成本为 V_1 , 其中若该节点已经实例化 VNF 则无须再花费部署成本:

$$V_1 = V_{cost} \times \sum_{i \in I} \sum_{j \in F_i} \sum_{u \in N} z_{iju} (1 - x_u) \quad (1)$$

则 SFC 的链路转发成本 V_2 为

$$V_2 = V_l \times \sum_{i \in I} \sum_{f_{i,j} \in \mathcal{F}_i} \sum_{f_{i,j,z} \in f_{i,j}} |P_{f_{i,j,z}}| \times b_{f_{i,j,z}} \quad (2)$$

因此, 总运营成本为

$$V_{total} = \alpha_1 V_1 + \alpha_2 V_2 \quad (3)$$

其中: α_1, α_2 是对成本组成进行优先排序的权重因子。

1.2.4 时延

对于每个 SFC 来说, 每条 VNF-FP 映射到物理链路上, 在物理链路上的传输时延为

$$d_{f_{i,j}} = \sum_{f_{i,j,z} \in f_{i,j}} \sum_{(u,v) \in P_{f_{i,j,z}}} \tau_{uv} \quad (4)$$

则每个 SFC 所需的传输时延取转发路径中所需最长的时延作为该条 SFC 的传输时延:

$$d_i = \max_{f_{i,j} \in \mathcal{F}_i} \{d_{f_{i,j}}\} \quad (5)$$

1.2.5 优化目标

本文是对服务功能链进行成本优化部署, 保证每个 SFC 的服务质量, 因此本文同时对 VNF 实例化成本和链路转发成本联合优化, 优化目标整理为

$$\begin{aligned} & \min V_{total} \quad (6) \\ \text{s. t. } & \begin{cases} C1: \sum_{i \in I} \sum_{j \in F_i} z_{iju} \times r_j^i \leq C_u \quad \forall u \in N \\ C2: \sum_{i \in I} \sum_{f_{i,j} \in \mathcal{F}_i} \sum_{f_{i,j,z} \in f_{i,j}} b_{f_{i,j,z}} \leq B_{u,v} \quad \forall (u,v) \in P_{f_{i,j,z}} \\ C3: \sum_{u \in N} z_{iju} \leq 1 \quad \forall i \in I, \forall j \in F_i \\ C4: \sum_{j \in F_i} \sum_{u \in N} z_{iju} = |F_i| \quad \forall i \in I \\ C5: d_i \leq D_i, \forall i \in I \\ C6: z_{iju} \in \{0, 1\} \quad \forall i \in I, \forall j \in F_i, \forall u \in N \\ C7: x_u \in \{0, 1\} \quad \forall u \in N \end{cases} \end{aligned}$$

本文的优化目标受到 C1 ~ C7 约束, 其中, C1 表示由于 CPU 资源, 每个服务器只能支持有限个 VNF 实例; C2 表示分配给 SFC 相应的转发路径带宽不能超过物理链路的带宽; C3 表示 SFC 中的每个 VNF 只能部署在单个物理节点上, 不能由多个物理节点服务; C4 表示对于每条 SFC, 一个物理节点上只能部署一种类型的 VNF 实例; C5 表示 SFC 的实际传输时延不可超过最大允许时延; C6 表示二进制变量约束, 当 i 中的第 j 个 VNF 实例化在物理节点 u 上时, z_{iju} 为 1, 否则为 0; C7 表示二进制变量约束, 当物理节点 u 已被实例化, 则 x_u 为 1, 否则为 0。

2 基于 VNF 共享的服务功能链部署算法

2.1 算法描述

假设时间被平均划分为多个时隙, 这些时隙编号为 1, 2, ..., T , 其中 T 为监测周期的长度。当 SFC 到达时, 它将在其到达的下一个时隙开始时被接纳或拒绝, 一旦 SFC 被接受, 将在一定时间内占用或共享部分资源, 当服务完成后会将资源释放供其他 SFC 使用。因此在时隙 $t-1$ 结束时网络中是存在部分 VNF 实例的, t 时刻到达的 SFC, 在部署时可以重用已部署的相同类型的 VNF 实例, 也可以在新的服务器上部署。

沿着更短的路径部署 VNF 意味着更少的链路转发成本, 但也代表着减少了共享 VNF 实例, 因此需要在更多的物理节点部署 VNF 实例; 相反, 共享 VNF 实例意味着产生更长的转发路径, 并且可能违反 SFC 时延约束, 这两者间是存在矛盾的。本文将这个问题分为两步解决: a) 为每个服务 SFC 转发路径设置路径长度上限, 用 $Len_{i,j}$ 表示; b) 在路径长度约束下最大化重用 VNF 实例数量。

对于 SFC 中每条转发路径, 为了确保可以连接, $Len_{i,j}$ 至少要大于两台服务器之间的最短路径。因为 SFC 中无法共享的 VNF 实例必须部署在新的节点上, 所以沿最短路径可能没有足够的空服务器来放置, 需要扩展最短路径到达更多空服务器, 并考虑所需的额外链路资源, 与此同时, 尽可能地满足 SFC

的时延需求。所以 $Len_{i,j}$ 可用下式表示:

$$Len_{i,j} = |SP_{f_{i,j}}| + \max(w_{p_{f_{i,j}}} \times (|F_{f_{i,j}}| - n_{p_{f_{i,j}}}), 0) \quad (7)$$

其中: $|SP_{f_{i,j}}|$ 表示 i 中第 j 条转发路径从源端到目的端上的最短路径长度, $w_{p_{f_{i,j}}}$ 是当前最短路径上服务器到达相邻空服务器和可共享 VNF 所在服务器的平均距离, $|F_{f_{i,j}}|$ 代表 i 中第 j 条转发路径上所需的 VNF 数量, $n_{p_{f_{i,j}}}$ 表示当前最短路径上的空服务器数量。

对于长度上限,可以根据式(7)得出,而如何最大化重用因子是本文主要解决的问题。对于请求 i ,其第 j 条转发路径为 $f_{i,j}$,需要找到一条连接源端 S_i 和目的端 T_i 的路径 $P_{f_{i,j}}$,为 $f_{i,j}$ 上的每个 $VNF_k (VNF_k \in F_{f_{i,j}})$ 进行部署, $F_{f_{i,j}}$ 为 $f_{i,j}$ 上 VNF 集合,即对 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 找到一个部署最优解 $f(P_{f_{i,j}}, F_{f_{i,j}})$,最大化重用因子(即在该条物理路径上可共享的 VNF 数量)。对于 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 可以分为两个子问题,即 $Q(P_{f_{i,j,z}}, F'_{f_{i,j}})$ 和 $Q(P_{f_{i,j,z}}, F''_{f_{i,j}})$,其中 $P_{f_{i,j,z}}$ 为未加入最后一条实际链路的集合, $P_{f_{i,j,z}}$ 为最后一条实际链路,而 $F'_{f_{i,j}} = (VNF_1, \dots, VNF_h)$, $F''_{f_{i,j}} = (VNF_{h+1}, \dots, VNF_{|F_{f_{i,j}}|})$ 。因此本文优化方法使用动态规划思想,对于 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 问题,拆分成 $Q(P_{f_{i,j,z}}, F'_{f_{i,j}})$ 和 $Q(P_{f_{i,j,z}}, F''_{f_{i,j}})$ 两个子问题进行递归求解,其问题的解组合后即为 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 的解,可以表达为 $f(P_{f_{i,j,z}}, F_{f_{i,j}}) = f(P_{f_{i,j,z}}, F'_{f_{i,j}}) + f(P_{f_{i,j,z}}, F''_{f_{i,j}})$;若子问题没有解,则 $f(P_{f_{i,j,z}}, F_{f_{i,j}}) = \emptyset$ 。由此可以得到动态方程:

$$f(P_{f_{i,j,z}}, F_{f_{i,j}}) = \arg \max_{f_h, 0 \leq h \leq |F_{f_{i,j}}|} x(f_h) \quad (8)$$

$$其中: f_h = \begin{cases} f(P_{f_{i,j,z}}, F_{f_{i,j}}) & h = |F_{f_{i,j}}| \\ f(P_{f_{i,j,z}}, F_{f_{i,j}}) & h = 0 \\ f(P_{f_{i,j,z}}, F'_{f_{i,j}}) + f(P_{f_{i,j,z}}, F''_{f_{i,j}}) & 0 < h < |F_{f_{i,j}}| \end{cases}$$

$x(f_h)$ 为 $Q(P_{f_{i,j}}, F_{f_{i,j}})$ 求解后得到的重用因子。而对于单个转发路径上重用因子的求解,利用动态规划递归求解,很容易找到子问题的解,从而可以在最大化重用因子下得到部署策略。

如算法1所示,使用动态规划(第3~17行)将当前虚拟转发路径与所找到的一条物理路径进行比较,得到当前物理路径可支持顺序部署的 VNF 数量和重用因子。如当前虚拟转发路径上 VNF 顺序集合 $F_{f_{i,j}} = \{VNF_1, VNF_2, VNF_3, VNF_4\}$,物理路径上服务器部署情况 $PF = \{VNF_1, VNF_2, \text{empty}, \text{empty}, VNF_4\}$,使用算法1可得到当前物理路径上可支持部署的 VNF 数量 Len 为4,重用因子 x 为3。

算法1 计算物理路径的可支持长度和重用因子算法

输入:虚拟转发路径上 VNF 顺序集合 $F_{f_{i,j}}$,物理路径上服务器部署情况集合 PF 。

输出:物理路径可支持部署的 VNF 数量 Len ,重用因子 x 。

```
1  初始化二维数组  $t, w$  动态记录可支持部署的 VNF 数量和重用因子,初始化一维数组  $dep$  记录部署方案;
2  //将虚拟转发路径上 VNF 与物理路径上部署情况进行比较
3  for  $k = 1$  to  $|F_{f_{i,j}}|$  do
4    for  $z = 1$  to  $|PF|$  do
5      if 虚拟转发路径上 VNF 与当前服务器上 VNF 相同 then
6         $t[k][z] = t[k-1][z-1] + 1, w[k][z] = w[k-1][z-1] + 1$ ;
7      elif 当前服务器未实例化 VNF then
8         $t[k][z] = t[k-1][z-1] + 1, w[k][z] = \max(w[k-1][z], w[k][z-1])$ ;
9      else
10        $t[k][z] = \max(t[k-1][z], t[k][z-1]), w[k][z] = \max(w[k-1][z], w[k][z-1])$ ;
11     end if
12   end for
```

```
13  if  $t[k][|PF|] < k$  then //无法按照顺序放置 VNF
14     $k = k - 1, \text{break}$ ; //停止放置
15  end if
16 end for
17  $Len = t[k][|PF|], x = w[k][|PF|]$ ; /* 物理路径可支持部署 VNF 数量和重用因子 */
18 return  $Len, x$ 
```

算法2根据算法1中记录可支持顺序部署的 VNF 数量和重用因子的二维数组,回溯得到该条虚拟路径在最优物理路径上的部署方案。如对于 $F_{f_{i,j}} = \{VNF_1, VNF_2, VNF_3, VNF_4\}$, $PF = \{VNF_1, VNF_2, \text{empty}, \text{empty}, VNF_4\}$ 使用算法2可以得到部署方案 $dep = \{1, 2, 4, 5\}$,即相对应的物理路径的位置下标。

算法2 单个转发路径的 VNF 部署算法

输入:虚拟转发路径上 VNF 顺序集合 $F_{f_{i,j}}$,物理路径上服务器部署情况集合 PF ,算法1中动态记录可支持部署的 VNF 数量和重用因子的二维数组 t, w 。

输出:部署方案 dep 。

```
1  初始化一维数组  $dep$  记录部署方案;
2  //回溯得到在当前物理路径上的部署方案
3   $index = |PF|$ ; //index 记录  $PF$  的下标
4  for  $m = |F_{f_{i,j}}|$  to 1 do //从后往前遍历虚拟转发路径上 VNF
5     $flag = 0$ ;
6    for  $n = index$  to 1 do
7      if 虚拟转发路径上 VNF 与当前服务器上 VNF 相同时且  $F_{f_{i,j}}$  中第  $m$  个 VNF 放置在物理路径第  $n$  个位置时可支持的 VNF 数量  $Len$  为  $m$ , then
8         $index = n, \text{break}$ ;
9      if 当前服务器未实例化且  $F_{f_{i,j}}$  中第  $m$  个 VNF 放置在物理路径第  $n$  个位置时可支持的 VNF 数量  $Len$  为  $m$  then
10        $index = n, flag = 1, \text{break}$ ;
11    end if
12  end for
13  if  $flag == 1$  then
14    for  $n = index$  to 0 do
15      if 虚拟转发路径上 VNF 与服务器上 VNF 相同时且  $F_{f_{i,j}}$  中第  $m$  个 VNF 放置在物理路径第  $n$  个位置时  $Len$  为  $m$  then
16        $index = n, \text{break}$ ;
17    end if
18  end for
19   $dep[m] = index, index = index - 1$ ; //得到部署方案
20 end if
21 end for
22 return  $dep$ 
```

综上所述,本文提出基于 VNF 共享的服务功能链部署算法,如算法3所示。依次遍历每条 SFC 的转发路径,计算路径长度上限(第5行),根据 SFC 的网络资源需求,在物理网络中查找源端到目的端的路径,筛选出符合路径长度上限和时延需求的可行路径(第6行),遍历可行路径,调用算法1计算可支持部署的 VNF 数量和重用因子,比较各路径的重用因子与物理路径长度的比值,取比值最高的物理路径,调用算法2进行部署(第10~18行)。若无法给 SFC 提供服务,则 SFC 被拒绝(第20~22行)。更新网络资源后(第23行),继续下一条 SFC 的部署。

算法3 基于 VNF 共享的服务功能链部署算法

输入:在 t 时刻到达的 SFC。

输出:部署策略。

```
1  初始化一维数组  $bestpath$  记录部署方案,初始化变量  $reject$  为0,记录拒绝的服务数量;
2  for  $i = 1$  to  $|I|$  do
3    for  $j = 1$  to  $|F_i|$  do
4       $g = 0$ ; //记录重用因子与物理路径长度的比值
5      根据网络中资源,使用式(7)计算路径长度上限  $Len_{i,j}$ ;
6      根据资源需求和时延需要,找出从源端到目的端符合
```

```

         $Len_{i,j}$ 和时延需求的可选路径存放在  $P_{f_{i,j}}$  中;
7    if  $P_{f_{i,j}}$  为空 then
8        break;
9    end if
10   for each  $P_{f_{i,j}}$  do
11       调用算法 1, 得到可支持部署的 VNF 数量  $Len$ , 重用因子  $x$ ;
12       if  $Len$  等于虚拟路径上所需的 VNF 数量 then
13           if  $x/|P_{f_{i,j}}| > g$  then
14                $g = x/|P_{f_{i,j}}|$ ;
15           end if
16       end if
17   end for
18   取比值最高的物理路径, 调用算法 2 返回放置方案存入  $bestpath$ ;
19 end for
20 if 无部署方案, then
21     拒绝请求
22 end if
23 更新网络资源;
24 end for
25 return 部署策略
    
```

2.2 算法复杂度分析

在 SFC 部署阶段, 共有 $|I|$ 条 SFC 到达, 其中每条 SFC 又有 $|F_i|$ 条转发路径。对于每条转发路径, 本文与 $|P_{f_{i,j}}|$ 条可选路径进行比较, 选择最佳路径, 即算法 1 所示, 而算法 1 需要进行 $|F_{f_{i,j}}| \times |PF|$ 循环, 所以算法 1 的时间复杂度为 $O(|F_{f_{i,j}}| \times |PF|)$, 其中 $|F_{f_{i,j}}|$ 为转发路径上 VNF 数量, $|PF|$ 为物理节点数量。当选出最佳部署路径时, 进行算法 2 部署, 它的时间复杂度也为 $O(|F_{f_{i,j}}| \times |PF|)$ 。因此本文算法时间复杂度为 $O(|I| \times |F| \times |P_{f_{i,j}}| \times |F_{f_{i,j}}| \times |PF|)$ 。

3 仿真与性能分析

本章对提出的基于 VNF 共享的服务功能链部署算法进行仿真和性能分析, 并与两种启发式算法进行比较, 从运营成本、VNF 利用率和服务接受率进行分析。

3.1 实验方法及仿真参数设置

本文采用含有 58 个节点、87 条链路的网络拓扑, 这个网络拓扑来源于一项致力于收集各国不同网络拓扑的研究^[28]。每个的 CPU 资源为 100%, 而物理链路的带宽为 10 Gbps^[29], 物理链路的延迟在 2~4 ms^[30]。

在单个时隙的情况下, 本文假设部分 VNF 实例已经随机放置在网络中。每条 SFC 是通过选择 N 中的节点作为源 S_i 和目标 T_i 随机生成的, 并根据文献[30]分配其所需带宽资源和延迟要求, 其中根据文献[29]设置每个网络功能所需 CPU 计算资源, 如表 2 所示。每个 SFC 的服务链的生成与文献[29]中的一致。

表 2 关键参数和值
Tab. 2 Key parameters & values

参数	值	参数	值
VNF 数量	1~5	$b_{f_{i,j},z}/\text{Mbps}$	10~120
$ F_i $	1~2	r_j^i	10%~25%
D_i/ms	40~100		

3.2 实验结果与分析

由于缺乏相关标准数据, 策略目标不同, 现有理论文献也存在方法难以横向比较等问题^[31]。为了评估本文算法的性能, 本文与文献[15]中两种启发式方法进行比较: a) shortest path heuristic (SPH), 沿着源端和目的端之间的最短路径部署 SFC 的每个 VNF-FP; b) greedy on used server (GUS), 优先在可

共享的 VNF 中放置, 以避免占用额外的未使用服务器。

a) 运营成本。作为本文的优化目标, 运营成本即为 VNF 部署成本和链路转发成本之和。在仿真中, 本文将 α_1 、 α_2 均设为 1, 这意味着链路和 VNF 实例部署一样重要。为了清晰地观察各个算法的结果, 本文对数据进行了归一化处理, 结果在 $[0, 1]$ 内。如图 4 所示, 相对于另外两种算法, 本文算法的总运营成本最低, 其中 SPH 和 GUS 所需要的平均总成本分别高于本文算法 6.6% 和 12.15%。其中链路转发成本如图 5 所示, GUS 的平均链路转发成本高于本文算法 50.38%, 该算法为了尽可能地共享 VNF 实例, 不可避免地增加了路径长度。对于 VNF 实例部署成本, 如图 6 所示, SPH 的部署成本最高, 高于本文算法 33.94%, 因为每个 SFC 都贪婪地部署在最短路径上, 而忽略了共享 VNF 实例。通过比较, 本文基于 VNF 共享的服务功能链部署算法很好地平衡了链路转发成本和部署成本, 能够有效降低运营成本。

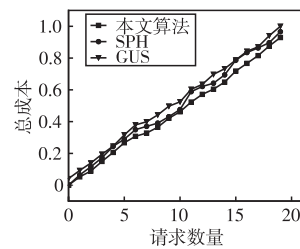


图 4 运营成本
Fig. 4 Operating expense

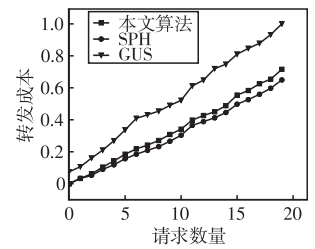


图 5 链路转发成本
Fig. 5 Forwarding cost

b) 利用率。如图 7 所示, 由于 GUS 优先考虑 VNF 实例共享, 其 VNF 实例利用率最高, 高于本文算法 42.4%, 而 SPH 优先考虑最短路径部署, 考虑较少的 VNF 实例共享, 所以它的利用率最低, 低于本文算法 23.82%。虽然本文的利用率低于 GUS, 但是牺牲了部分 VNF 实例部署成本, 从而避免了链路转发成本的过度增长。

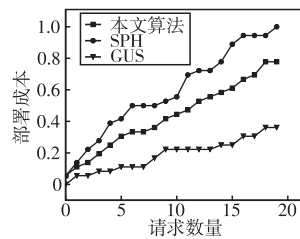


图 6 VNF 部署成本
Fig. 6 Deployment cost of VNF

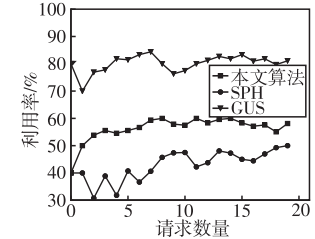


图 7 VNF 利用率
Fig. 7 Utilization rate of VNF

c) 接受率。如图 8 所示, 当 SFC 数量增大时, 本文基于 VNF 共享的服务功能链部署算法服务接受率均高于 SPH 和 GUS, 如当 SFC 数量达到 75 时, SPH 和 GUS 的服务接受率分别为 84% 和 86.67%, 而本文算法的服务接受率为 89.33%。因为 SPH 优先考虑最短路径部署, 导致网络中存在 VNF 实例独占服务器资源, 而 GUS 虽然共享了 VNF 实例, 但是因为路径不可避免地增长, 使链路带宽资源无法满足后续 SFC, 所以本文算法在网络中可以支持更多的 SFC。

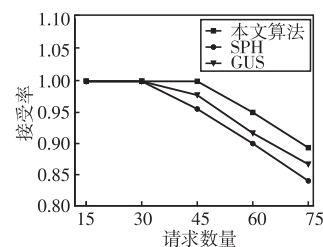


图 8 服务接受率
Fig. 8 Service acceptance rate

4 结束语

针对 SDN/NFV 架构下 SFC 部署成本优化问题,本文首先将问题建模为多目标和多约束问题模型,该模型以最小化 VNF 实例部署成本和链路转发成本为目标,同时受限于每条 SFC 时延、物理链路带宽资源和服务器 CPU 资源约束;其次为每条 VNF-FP 设置路径长度上限,以平衡链路和 VNF 实例的使用;最后提出基于 VNF 共享的服务功能链部署算法,联合 SFC 放置和路径选择问题,以节省链路转发成本并有效地重用 VNF 实例。仿真结果表明,与 SPH 和 GUS 算法相比,本文的算法可以更好地管理链路和 VNF 实例之间的关系,显著地降低了运营成本,并且可以使用有限的资源来服务更多的服务 SFC。在未来,将考虑结合 VNF 实例的水平扩展和垂直扩展,开展基于深度强化学习的服务功能链部署方法的研究。

参考文献:

- [1] Hu Fei, Qi Hao, Bao Ke. A survey on software-defined network and OpenFlow: from concept to implementation [J]. *IEEE Communications Surveys & Tutorials*, 2014, 16(4): 2181-2206.
- [2] 王进文, 张晓丽, 李琦, 等. 网络功能虚拟化技术研究进展 [J]. *计算机学报*, 2019, 42(2): 415-436. (Wang Jinwen, Zhang Xiaoli, Li Qi, et al. Network function virtualization technology: a survey [J]. *Chinese Journal of Computers*, 2019, 42(2): 145-436.)
- [3] Mijumbi R, Serrat J, Gorricho J L, et al. Network function virtualization: state-of-the-art and research challenges [J]. *IEEE Communications Surveys & Tutorials*, 2015, 18(1): 236-262.
- [4] Cao Jiuyue, Zhang Yan, An Wei, et al. VNF-FG design and VNF placement for 5G mobile networks [J]. *Science China Information Sciences*, 2017, 60(3): article No. 040302.
- [5] Bari F, Chowdhury S R, Ahmed R, et al. Orchestrating virtualized network functions [J]. *IEEE Trans on Network and Service Management*, 2016, 13(4): 725-739.
- [6] Yang Binxu, Chai W K, Xu Zichuan, et al. Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications [J]. *IEEE Trans on Network and Service Management*, 2018, 15(1): 475-488.
- [7] Huang Meitian, Liang Weifa, Ma Yu, et al. Maximizing throughput of delay-sensitive NFV-enabled request admissions via virtualized network function placement [J]. *IEEE Trans on Cloud Computing*, 2019, 9(4): 1535-1548.
- [8] Xia Ming, Shirazipour M, Zhang Ying, et al. Network function placement for NFV chaining in packet/optical datacenters [J]. *Journal of Lightwave Technology*, 2015, 33(8): 1565-1570.
- [9] 金明, 李琳琳, 张文瑾, 等. 基于深度强化学习的服务功能链映射算法 [J]. *计算机应用研究*, 2020, 37(11): 3456-3460, 3466. (Jin Ming, Li Linlin, Zhang Wenjin, et al. SFC mapping algorithm based on deep reinforcement learning [J]. *Application Research of Computers*, 2020, 37(11): 3456-3460, 3466.)
- [10] Mebarkia K, Zsóka Z. Service traffic engineering: avoiding link overloads in service chains [J]. *Journal of Communications and Networks*, 2019, 21(1): 69-80.
- [11] Xu Zichuan, Liang Weifa, Huang Meitian, et al. Efficient NFV-enabled multicasting in SDNs [J]. *IEEE Trans on Communications*, 2018, 67(3): 2052-2070.
- [12] 唐伦, 贺兰钦, 连沁怡, 等. 基于改进深度强化学习的虚拟网络功能部署优化算法 [J]. *电子与信息学报*, 2021, 43(6): 1724-1732. (Tang Lun, He Lanqin, Lian Qinyi, et al. Virtual network function placement optimization algorithm based on improve deep reinforcement learning [J]. *Journal of Electronics & Information Technology*, 2021, 43(6): 1724-1732.)
- [13] Aklamanu F, Randriamasy S, Renault E. Utility and A*-based algorithm for network slice placement and chaining [C]//Proc of IEEE Global Communications Conference. Piscataway, NJ: IEEE Press, 2019: 1-6.
- [14] Chen Xiaoxiao, Shi Jiugen, He Yong, et al. Deployment strategy of resource optimized service function chain in NFV environment [C]//Proc of International Conference on Cryptography, Network Security, and Communication Technology. Bellingham: SPIE Press, 2022: 130-137.
- [15] Zhang Qixia, Liu Fangming, Zeng Chaobing. Online adaptive interference-aware VNF deployment and migration for 5G network slice [J]. *IEEE/ACM Trans on Networking*, 2021, 29(5): 2115-2128.
- [16] Luo Ziyue, Wu Chuan. An online algorithm for VNF service chain scaling in datacenters [J]. *IEEE/ACM Trans on Networking*, 2020, 28(3): 1061-1073.
- [17] 柴蓉, 谢德胜, 陈前斌. 基于成本及功耗联合优化的 SDN 虚拟网络映射算法 [J]. *电子学报*, 2021, 49(8): 1615-1624. (Chai Rong, Xie Desheng, Chen Qianbin. Cost and power consumption joint optimization based virtual network embedding algorithm for software-defined networking [J]. *Acta Electronica Sinica*, 2021, 49(8): 1615-1624.)
- [18] Liu Junjie, Lu Wei, Zhou Fen, et al. On dynamic service function chain deployment and readjustment [J]. *IEEE Trans on Network and Service Management*, 2017, 14(3): 543-553.
- [19] Wang Fangxin, Ling Ruilin, Zhu Jing, et al. Bandwidth guaranteed virtual network function placement and scaling in datacenter networks [C]//Proc of the 34th International Performance Computing and Communications Conference. Piscataway, NJ: IEEE Press, 2015: 1-8.
- [20] Beck M T, Botero J F. Coordinated allocation of service function chains [C]//Proc of IEEE Global Communications Conference. Piscataway, NJ: IEEE Press, 2015: 1-6.
- [21] Bari F, Chowdhury S R, Ahmed R, et al. Orchestrating virtualized network functions [J]. *IEEE Trans on Network and Service Management*, 2016, 13(4): 725-739.
- [22] 陈嘉亮, 王丰, 张潇. 移动边缘计算网络下的服务功能链部署优化设计 [J]. *计算机应用研究*, 2022, 39(10): 3108-3113. (Chen Jialiang, Wang Feng, Zhang Xiao. Optimization design of service function chain deployment under mobile edge computing network [J]. *Application Research of Computers*, 2022, 39(10): 3108-3113.)
- [23] 朱国晖, 景文煊, 李世昌. 基于改进麻雀搜索算法的服务功能链优化映射算法 [J]. *计算机应用研究*, 2022, 39(7): 2120-2123, 2131. (Zhu Guohui, Jing Wenhuan, Li Shichang. Optimized mapping algorithm of service function chain based on improved sparrow search algorithm [J]. *Application Research of Computers*, 2022, 39(7): 2120-2123, 2131.)
- [24] Shang Xiaojun, Liu Zhenhua, Yang Yuanyuan. Network congestion-aware online service function chain placement and load balancing [C]//Proc of the 48th International Conference on Parallel Processing. New York: ACM Press, 2019: 1-10.
- [25] Shah H A, Zhao Lian. Multiagent deep-reinforcement-learning-based virtual resource allocation through network function virtualization in Internet of Things [J]. *IEEE Internet of Things Journal*, 2020, 8(5): 3410-3421.
- [26] Cong Peijin, Xu Guo, Wei Tongquan, et al. A survey of profit optimization techniques for cloud providers [J]. *ACM Computing Surveys*, 2020, 53(2): article No. 26.
- [27] Zhang Qixia, Liu Fangming, Zeng Chaobing. Adaptive interference-aware VNF placement for service-customized 5G network slices [C]//Proc of IEEE Conference on Computer Communications. Piscataway, NJ: IEEE Press, 2019: 2449-2457.
- [28] Knight S, Nguyen H X, Falkner N, et al. The Internet topology Zoo [J]. *IEEE Journal on Selected Areas in Communications*, 2011, 29(9): 1765-1775.
- [29] Luizelli M C, Bays L R, Burriel L S, et al. Piecing together the NFV provisioning puzzle: efficient placement and chaining of virtual network functions [C]//Proc of IFIP/IEEE International Symposium on Integrated Network Management. Piscataway, NJ: IEEE Press, 2015: 98-106.
- [30] Huang Meitian, Liang Weifa, Xu Zichuan, et al. Throughput maximization in software-defined networks with consolidated middleboxes [C]//Proc of the 41st Conference on Local Computer Networks. Piscataway, NJ: IEEE Press, 2016: 298-306.
- [31] 祖家琛, 胡谷雨, 严佳洁, 等. 网络功能虚拟化下服务功能链的资源管理研究综述 [J]. *计算机研究与发展*, 2021, 58(1): 137-152. (Zu Jiachen, Hu Guyu, Yan Jiajie, et al. Resource management of service function chain in NFV enabled network: a survey [J]. *Journal of Computer Research and Development*, 2021, 58(1): 137-152.)