

## 基于重叠网络结构的服务功能链时空优化编排策略

谷允捷\* 胡宇翔 谢记超

(国家数字交换系统工程技术研究中心 郑州 450002)

**摘要:** 网络功能虚拟化(NFV)的引入大幅降低了互联网业务的运营成本。针对现有的服务功能链(SFC)编排方法无法在优化底层资源的同时保证业务时延性能的问题, 该文提出一种基于重叠网络结构的SFC时空优化编排策略。在将计算、网络资源与细粒度时延约束纳入考虑的基础上, 该策略通过建立重叠网络模型实现了计算与网络资源的分离, 将构建SFC所需的资源开销与相关时延共同抽象化为重叠网络链路权重, 从而使SFC编排问题转化为易于求解的最短路径问题。对于需要批量处理的SFC集合设计了基于重叠网络的模拟退火迭代优化编排算法(ONSA)。通过对比实验证明了该策略下编排方案的平均端到端时延、链路资源占用率与运营开销相对其他方案分别降低29.5%, 12.4%与15.2%, 请求接受率提高22.3%, 虚拟网络功能(VNF)负载均衡性能得到显著提升。

**关键词:** 网络功能虚拟化; 时空优化; 重叠网络; 迭代优化

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2019)11-2675-09

DOI: [10.11999/JEIT190145](https://doi.org/10.11999/JEIT190145)

## A Spatial and Temporal Optimal Method of Service Function Chain Orchestration Based on Overlay Network Structure

GU Yunjie HU Yuxiang XIE Jichao

(National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China)

**Abstract:** With the introduction of Network Function Virtualization (NFV), the operating costs of operators can be greatly reduced. However, most existing Service Function Chain (SFC) orchestration researches can not optimize the resources utilization while guaranteeing the performance of service delay. A spatial and temporal optimal method of Service Function Chain (SFC) orchestration based on an overlay network structure is proposed. Based on the consideration of the restrictions such as computing resource, network resource and fine-grained end to end delay, this method separates the computing resource and network resource. The resources cost and related delay of SFC can be abstracted into the links weight of overlay network, which can help to convert the SFC orchestration problem into the shortest path problem that can be easily solved. As for the SFC requests set requiring batch processing, an Overlay Network based Simulated Annealing iterative optimal orchestration algorithm(ONSA) is designed. The simulation results demonstrate that the proposed orchestration scheme can reduce the end-to-end delay, the utilization ratio of link bandwidth resource and the operational expenditure by 29.5%, 12.4% and 15.2%, and the acceptance ratio of requests set can be improved by 22.3%. The performance of Virtual Network Function (VNF) load balancing can be significantly improved.

**Key words:** Network Function Virtualization (NFV); Spatial and temporal optimization; Overlay network; Iterative optimization

收稿日期: 2019-03-13; 改回日期: 2019-06-18; 网络出版: 2019-06-25

\*通信作者: 谷允捷 lizardwhite@163.com

基金项目: 国家重点研发计划(2017YFB0803204), 国家自然科学基金(61521003, 61872382), 广东省重点领域研发计划(2018B010113001)

Foundation Items: The National Key Research and Development Project(2017YFB0803204), The National Natural Science Foundation of China (61521003, 61872382), The Research and Development Program in Key Areas of Guangdong Province (2018B010113001)

## 1 引言

随着新型互联网业务的快速涌现,僵化的传统网络架构愈发难以满足网络服务的性能需求。扩建基础设施的对策引发网络规模急剧膨胀,运营成本提升而收益严重下滑的趋势迫使运营商不得不面对网络架构重构<sup>[1]</sup>。网络功能虚拟化(Network Function Virtualization, NFV)是一种改变网络业务运维模式的革新技术<sup>[2]</sup>,通过将网络功能(Network Function, NF)与专用硬件设备的解耦, NFV可借助通用计算平台与云资源池运行虚拟网络功能(Virtual Network Function, VNF)以提供网络服务,极大地降低了业务运营成本。但将正处于起步阶段的NFV投入运用的过程中也面临着诸多挑战,其中,服务功能链(Service Function Chain, SFC)的编排优化问题已成为当前产学研研究热点<sup>[3]</sup>。一方面, SFC的构建消耗了网络带宽与计算资源,运营商为降低运营开销需要使SFC的编排方式能够提高资源利用率;另一方面, VNF实例与链路负载的升高会增加处理时延与排队时延,而为保证业务性能,运营商必须使SFC的端到端累计时延低于其容忍时延。为适应新型网络体系中高流量与低时延的发展目标,运营商在设计SFC编排方案时需要兼顾资源节约与业务性能。

现有的SFC编排方案在降低运营开销方面取得了丰富的成果。Moens等人<sup>[4]</sup>以最小化大容量服务器(High Volume Server, HVS)数量为目标,构建整数线性规划(Integral Linear Programming, ILP)模型并在小规模网络中精确求解。Bari等人<sup>[5]</sup>综合虚拟机(Virtual Machine, VM)部署,运行和转发代价等因素,设计ILP与基于Viterbi的启发式算法,取得较优的快速编排效果。Jang等人<sup>[6]</sup>注重多条服务功能链的接受率和能源开销优化,通过对线性约束的放缩在多项式时间内求解。史久根等人<sup>[7]</sup>设计多种面向OPEX的优化算法,并通过实验充分证实了基于遗传算法(Genetic Algorithm, GA)的优势与有效性。陈卓等人<sup>[8]</sup>针对资源碎片优化问题提出一种SFC构建策略,实现更低的传输延迟。Wen等人<sup>[9]</sup>基于VM合并思想,设计适用于NFV环境的贪婪合并方案,平均降低了32%的VNF实例数目。以上文献在优化资源利用与降低运营开销方面有突出的贡献,但忽略了VNF实例与链路负载对SFC端到端时延的影响。Pei等人<sup>[10]</sup>在建立动态映射模型时考虑到了VNF、交换机以及链路负载变化对映射开销产生的影响,设计多层网络映射算法提升了节点负载均衡性能,但优化目标仍停留在降低运营开销的空间维度。陈卓等人<sup>[11]</sup>提出了一种启发式SFC

迁移重配置策略,在保证服务质量的同时明显改善了资源利用效率。但该方法主要适用于SFC迁移与重配置。Arteaga等人<sup>[12]</sup>在基于NFV架构的演进分组核心网(Virtual Evolved Packet Core, VEPC)中结合Q-learning与高斯回归,实现VNF实例数目随流量变化动态调整,完成了时空2维优化,但仅适用于单个VNF弹性伸缩的情况。可见,当前对该问题的研究文献仍非常匮乏,亟需进行深入研究。

与上述研究工作不同,本文提出一种基于重叠网络结构的服务功能链时空优化编排策略,通过构建重叠网络实现底层网络中计算资源与网络资源的分离,将构建SFC所需的资源开销与相关时延共同抽象化为重叠网络链路权重,从而使SFC编排时空优化问题转化为易于求解的最短路径问题。对需要批量处理的SFC请求集合,设计了基于重叠网络的模拟退火迭代优化编排算法(Overlay Network based Simulated Annealing iterative optimal orchestration algorithm, ONSA),提升SFC请求集合的接受率与时空优化效果。

## 2 问题描述

对运营商而言, SFC编排方案直接决定了NFV系统的资源供应与端到端时延,进而影响运营收益与业务性能。图1展示了SFC请求及两种编排方式。每条SFC请求由源、目的节点, VNF请求序列及相应资源需求量构成。编排方案1注重运营开销优化,该方案只需在底层网络中部署4个VNF实例,能够显著降低部署开销与链路传输开销,但节点4中的VNF工作负载已经达到90%,且节点2,4之间的链路极易发生拥塞。过度复用的VNF实例与拥塞的链路会明显增加VNF处理时延与链路排队时延,严重影响业务性能。编排方案2为避免上述问题,在节点2,3中分别增加了IDS, NAT实例,并改变了部分传输路径以均衡链路负载。该方案将VNF工作负载控制在60%以下,且不会引起链路拥塞,但链路传输总开销相对增加,且需要部署并维护更多的VNF,运营开销因此提高。

综上所述,仅面向“空间”维度优化的SFC编排方案极易导致底层资源负载不均,从而降低服务质量,而仅考虑“时间”维度的性能优化又会增加运营开销。为满足业务请求的资源供应与时延约束, SFC编排方案必须兼顾业务性能相关的“时间”优化与运营收益相关的资源“空间”优化。

## 3 模型建立

本节针对NFV系统中的重要元素建立数学模型,确定SFC时空编排约束及优化目标。

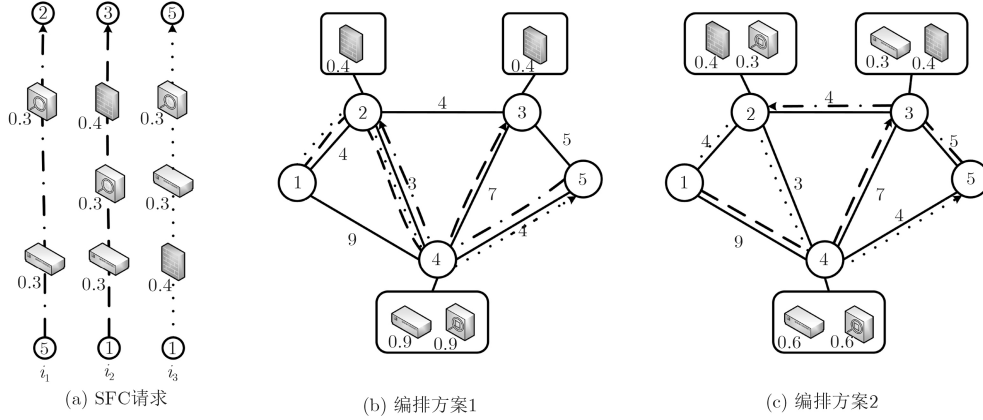


图1 SFC请求及两种编排方案

### 3.1 系统元素

#### (1) 底层网络

运营商网络的物理资源可抽象化为一张无向赋权图  $G = (V, E, A_V, A_E)$ 。其中  $V$  为节点集合,  $A_V$  为节点属性, 可表示为  $A_V = \{C_v, L_v\}$ ,  $C_v$  表示节点  $n$  中计算资源总量,  $L_v$  代表节点位置信息。  $E$  为连接这些节点的链路集合,  $A_E = \{B_e, \lambda_e, \delta_e\}$  表示链路属性, 其中  $B_e$  为链路带宽容量,  $\lambda_e$  代表链路长度,  $\delta_e$  为链路时延。

#### (2) 虚拟网络功能

VNF类型用变量  $K$  来表示。VNF实例具有属性  $A_k = \{D_k, C_k, P_k\}$ , 其中  $D_k$  为部署一个  $k$  型VNF实例所需的计算资源,  $C_k$  为  $k$  型VNF实例计算能力上限, 多条SFC可共享一个VNF实例进行业务处理, 直到SFC累积计算资源需求达到该实例计算能力上限,  $P_k$  为  $k$  型VNF实例数据处理速度。NFV系统中所有VNF实例构成的集合为  $M$ ,  $\delta_m$  为VNF实例  $m$  的处理时延。

#### (3) 服务功能链

定义  $I$  为SFC集合, 6元数组  $A_I = \{s_i, o_i, \beta^{K(i)}, c_i^k, B_i, \delta_i^{s,o}\}$  为SFC属性。  $s_i, o_i \in V$  分别为链  $i$  的源节点和目的节点,  $\beta_i^k \in \beta^{K(i)}$  为组成该SFC的NF请求,  $K(i)$  为该链的NF序列, 如  $\text{Firewall}(\beta_{i_1}^1) \rightarrow \text{IDS}(\beta_{i_1}^2) \rightarrow \text{Proxy}(\beta_{i_1}^3)$ 。  $c_i^k$  为链  $i$  中NF请求  $\beta_i^k$  对应的计算资源需求,  $B_i$  为链  $i$  的带宽需求,  $\delta_i^{s,o}$  为SFC容忍时延, 若SFC业务流的累计时延超出该值则视为SLA违例, 该SFC请求将被拒绝。

定义决策变量  $q_m^k$  表示VNF实例  $m$  的类型是否为  $k$ ,  $x_m^v$  表示VNF实例  $m$  是否部署于节点  $v$ , 变量  $y_i^{e(u,v)}$  表示SFC  $i$  是否经过链路  $e(u, v)$ ,  $y_{i,k}^m$  表示SFC  $i$  的NF请求  $\beta_i^k$  是否由VNF实例  $m$  处理。决策变量  $w_i^m = \{1, 0\}$  表示在接收链  $i$  后, VNF实例  $m$  是否部署在底层网络中,  $\hat{w}_i^m = \{1, 0\}$  代表在接收链  $i$  前, 实例  $m$  是否已经部署在底层网络中, 则  $[w_i^m - \hat{w}_i^m]^+ =$

$\{0, 1\}$  可代表VNF实例  $m$  是否在接收链  $i$  后部署, 其中  $[ \cdot ]^+$  运算符定义为  $[f(x)]^+ = \max\{f(x), 0\}$ 。

$$q_m^k = \begin{cases} 1, \text{VNF实例 } m \text{ 类型为 } k \\ 0, \text{VNF实例 } m \text{ 类型不为 } k \end{cases} \quad (1)$$

$$x_m^v = \begin{cases} 1, \text{VNF实例 } m \text{ 部署于节点 } v \\ 0, \text{VNF实例 } m \text{ 未部署于节点 } v \end{cases} \quad (2)$$

$$y_i^{e(u,v)} = \begin{cases} 1, \text{SFC } i \text{ 经过链路 } e(u, v) \\ 0, \text{SFC } i \text{ 不经过链路 } e(u, v) \end{cases} \quad (3)$$

$$y_{i,k}^m = \begin{cases} 1, \text{NF } \beta_i^k \text{ 经过VNF实例 } m \text{ 处理} \\ 0, \text{NF } \beta_i^k \text{ 不经过VNF实例 } m \text{ 处理} \end{cases} \quad (4)$$

### 3.2 约束条件

#### (1) 节点资源约束

底层网络节点为VNF部署与SFC处理提供计算资源, VNF实例占用计算资源量不仅包括自身部署所需的计算资源  $D_k$ , 还包括处理NF请求  $\beta_i^k$  所需的计算资源  $c_i^k$ 。任意节点内部署的VNF实例所需计算资源总和不得超出该节点所能提供的最大计算资源量  $C_v$ , 则有

$$\sum_{m \in M} \sum_{k \in K} x_m^v q_m^k D_k + \sum_{m \in M} \sum_{i \in I} \sum_{k \in K(i)} x_m^v y_{i,k}^m c_i^k \leq C_v, \quad \forall v \in V \quad (5)$$

对于单个VNF实例而言, 可容纳NF请求总量不得超出其计算能力上限  $C_k$ , 则有

$$\sum_{i \in I} \sum_{k \in K(i)} y_{i,k}^m c_i^k \leq \sum_{k \in K} q_m^k C_k, \quad \forall m \in M \quad (6)$$

#### (2) 业务流相关约束

底层网络中的链路具有带宽容量约束, 经过一条链路的所有SFC带宽需求累计不得超出带宽容量

$$\sum_{i \in I} y_i^{e(u,v)} B_i \leq B_e, \quad \forall e \in E \quad (7)$$

为避免业务流分离可能引起的性能退化,规定SFC只能沿一条路径传输,不采用多路径分流方式。因此对每条SFC除源、目的节点外的物理结点必须满足流量守恒约束,故有

$$\begin{aligned} & \sum_{w \in N} y_i^{e(v,w)} B_i - \sum_{w \in N} y_i^{e(w,v)} B_i \\ & = \begin{cases} 0, \forall v \in V \setminus \{s_i, o_i\}, \forall i \in I \\ B_i, v = s_i, \forall i \in I \\ -B_i, v = o_i, \forall i \in I \end{cases} \end{aligned} \quad (8)$$

### (3) 时延约束

业务流*i*的端到端时延主要由链路时延 $\delta_{e(u,v)}^i$ 与VNF处理时延 $\delta_m^i$ 构成。运营商为保证业务性能,对已接收SFC的端到端累计时延 $\delta_i$ 需满足容忍时延约束

$$\begin{aligned} \delta_i & = \sum_{e \in E} y_i^{e(u,v)} \delta_{e(u,v)}^i + \sum_{k \in K(i)} \sum_{m \in M} y_{i,k}^m \delta_m^i \leq \delta_i^{s,o}, \\ & \forall i \in I \end{aligned} \quad (9)$$

链路时延 $\delta_{e(u,v)}^i$ 由传输时延 $\delta_{e(u,v)}^{i,tx}$ 、传播时延 $\delta_{e(u,v)}^{i,prop}$ 以及排队时延 $\delta_{e(u,v)}^{i,que}$ 共同决定。由于传播时延 $\delta_{e(u,v)}^{i,prop}$ 仅与链路物理长度及传播介质相关且数值相对较小,本文对 $\delta_{e(u,v)}^i$ 只考虑传输时延以及与链路负载强度 $B_{load}^{e(u,v)}$ 相关的排队时延,可表示为

$$\begin{aligned} \delta_{e(u,v)}^i & = \delta_{e(u,v)}^{i,tx} + \delta_{e(u,v)}^{i,que} \\ & = \delta_{e(u,v)}^{i,tx} + \frac{B_{load}^{e(u,v)}}{1 - B_{load}^{e(u,v)}} \cdot \delta_{e(u,v)}^{i,tx} \\ & = \frac{1}{1 - B_{load}^{e(u,v)}} \cdot \delta_{e(u,v)}^{i,tx} \end{aligned} \quad (10)$$

依据排队模型<sup>[13]</sup>可知,链路时延 $\delta_{e(u,v)}^i$ 可简化为

$$\delta_{e(u,v)}^i = \frac{B_e \delta_{e(u,v)}^{i,tx}}{B_e + B_i - \sum_{\tilde{i} \in I} y_i^{e(u,v)} B_{\tilde{i}}} \quad (11)$$

由通用处理器共享定义的处理时延模型<sup>[13]</sup>可知,VNF处理时延 $\delta_m^i$ 取决于包处理时间 $\delta_{proc}^{m,i}$ 、负载强度 $C_{load}^m$ 与当前接收的NF请求*k*所需计算资源量 $C_i^k$ ,可表示为

$$\delta_m^i = \frac{(C_{load}^m)^2 C_k}{C_i^k} \cdot \delta_{proc}^{m,i} \quad (12)$$

当VNF实例*m*接收NF请求 $\beta_i^k$ 时,计算资源负载强度可表示为

$$C_{load}^m = \frac{\sum_{\tilde{i} \in I} \sum_{k \in K(\tilde{i})} y_{\tilde{i},k}^m C_{\tilde{i}}^k}{C_k} \quad (13)$$

则VNF处理时延 $\delta_m^i$ 可简化为

$$\delta_m^i = \frac{\left( \sum_{\tilde{i} \in I} \sum_{k \in K(\tilde{i})} y_{\tilde{i},k}^m C_{\tilde{i}}^k \right)^2 L_i}{C_i^k C_k \sum_{k \in K} q_m^k P_k} \quad (14)$$

### 3.3 优化目标

本文将业务性能与运营开销同时纳入SFC编排的优化目标。设单位计算资源开销为 $r_c$ ,单位链路传输开销为 $r_e$ 。当NFV系统接收一条SFC请求时,增加的运营开销可表示为

$$\begin{aligned} \varphi_i & = r_c \sum_{m \in M} \sum_{k \in K(i)} \{ y_{i,k}^m C_i^k + q_m^k D_k [w_i^m - \hat{w}_i^m]^+ \} \\ & + r_e \sum_{e \in E} \{ y_i^{e(u,v)} \lambda_e \} \end{aligned} \quad (15)$$

由式(9)可知,业务流*i*的端到端累计时延可表示为 $\delta_i$ ,则SFC编排时空优化目标可表示为式(16),权值 $\varepsilon \in (0, 1)$ ,用于调目标函数的影响权重。

$$\begin{aligned} \min \quad \text{Object} & = \sum_{i \in I} [\varepsilon \varphi_i + (1 - \varepsilon) \delta_i], \\ \text{s.t.} \quad & \text{式(5) - 式(9)} \end{aligned} \quad (16)$$

## 4 算法设计

VNF部署问题已被证明是NP难问题<sup>[14]</sup>,SFC编排时空优化问题可约化为VNF部署问题,因此也是NP难问题。受文献<sup>[10]</sup>启发,本文通过构建重叠网络实现计算资源与网络资源的分离,将SFC编排问题转化为易于求解的最短路径问题。对于需要批量处理的SFC集合,设计基于重叠网络的迭代优化编排算法,借助模拟退火算法提高迭代效率,从而提升SFC请求集合的接受率与时空优化效果。

### 4.1 重叠网络的构建

如图2所示,重叠网络 $G_{Overlay}$ 为3维结构,由水平层 $G_{hor}$ 和垂直层 $G_{ver}$ 共同构成。 $G_{hor}$ 与 $G_{ver}$ 的

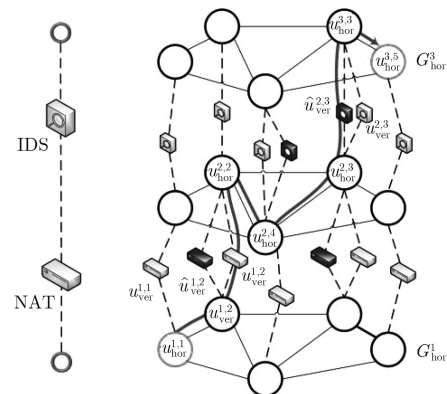


图2 SFC请求及其对应重叠网络

层数均取决于 $\beta^{K(i)}$ 的NF请求数量 $k = |K(i)|$ ，其中 $G_{hor}$ 共 $k + 1$ 层， $G_{ver}$ 共 $k$ 层。

$G_{hor}^s = (V_{hor}^s, E_{hor}^s)$ 为自下而上的第 $s$ 水平层，每层节点 $V_{hor}^s$ 、链路 $E_{hor}^s$ 以及拓扑连接方式与底层网络完全相同。每条SFC的起点位于 $G_{hor}^1$ ，终点位于 $G_{hor}^{k+1}$ ，如图2所示，以链 $i\{1 \rightarrow NAT \rightarrow IDS \rightarrow 5\}$ 为例，起点位于 $G_{hor}^1$ 的 $v_{hor}^{1,1}$ ，终点位于 $G_{hor}^3$ 的 $v_{hor}^{3,5}$ 。 $e_{hor}^s$ 为SFC可能经过的链路，其权重 $w_{hor}^e$ 由传输开销和链路时延共同构成。可表示为式(17)

$$w_{hor}^e = \varepsilon r_e \lambda_e + (1 - \varepsilon) \delta_{e(u,v)}^i \quad (17)$$

$G_{ver}^s = (V_{ver}^s, E_{ver}^s)$ 为自下而上的第 $s$ 垂直层， $V_{ver}^s$ 为位于该层的节点，代表VNF实例。 $V_{ver}^s$ 通过节点分割法区分SFC对VNF实例的部署与复用，即依据 $[w_i^m - \hat{w}_i^m]^+$ 的取值决策 $V_{ver}^s$ 是否分割为两个节点。以链 $i$ 为例，图3代表了部署链 $i$ 前，底层网络中已部署的VNF实例分布，由于底层网络节点2中存在已部署的NAT实例 $m$ ，即 $[w_i^m - \hat{w}_i^m]^+ = 0$ ，则位于垂直层 $G_{ver}^1$ 的 $V_{ver}^{1,2}$ 被分割为两个节点，分别为 $\hat{v}_{ver}^{1,2}, v_{ver}^{1,2}$ ，其中 $\hat{v}_{ver}^{1,2}$ 表示节点2中已部署且负载最低的NAT， $v_{ver}^{1,2}$ 则表示当SFC经过该节点时需要部署一个新的NAT。同理 $V_{ver}^{1,3}, V_{ver}^{2,3}, V_{ver}^{2,4}$ 均因底层节点内存在相应已部署的VNF实例而被分割为两个节点。对于其余垂直节点，由于不存在相应已部署的VNF，则仅含一个元素。垂直层的链路 $E_{ver}^s$ 用于连接相邻层的水平节点 $v_{hor}^{s,n}$ 与 $v_{hor}^{s+1,n}$ 且穿过垂直层节点 $v_{ver}^{s,n}$ ，链路权重 $w_{ver}^e$ 由穿过垂直节点的计算资源开销及其处理时延共同构成，可表示为式(18)

$$w_{ver}^e = \varepsilon r_c \{C_i^k + D_k[w_i^m - w_i^m]^+\} + (1 - \varepsilon) \delta_m^i \quad (18)$$

以链 $i$ 为例，图3中节点 $v_{hor}^{1,1}$ 与 $v_{hor}^{3,5}$ 之间的任意一条路径均可作为链 $i$ 的编排策略，设两点间最短路径为 $v_{hor}^{1,1} \rightarrow v_{hor}^{1,2} \rightarrow v_{ver}^{1,2} \rightarrow v_{hor}^{2,2} \rightarrow v_{hor}^{2,4} \rightarrow v_{hor}^{2,3} \rightarrow \hat{v}_{ver}^{2,3} \rightarrow v_{hor}^{3,3} \rightarrow v_{hor}^{3,5}$ ，则说明系统接收链 $i$ 时需要在节点2部署

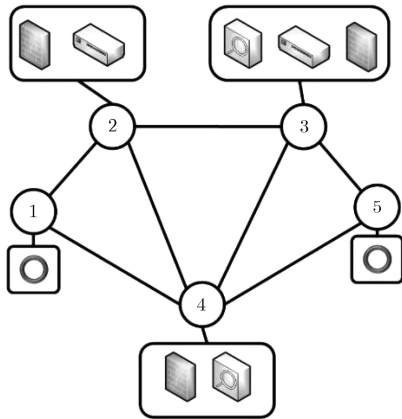


图3 已部署的VNF在底层网络分布情况

一个新的NAT实例 $v_{ver}^{1,2}$ ，同时业务流 $i$ 需要复用位于节点3中已部署的IDS实例，且在图3底层网络的传输路径为 $1 \rightarrow 2 \rightarrow NAT \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow IDS \rightarrow 3 \rightarrow 5$ 。

**定理 1** 链 $i$ 编排的时空优化问题等价于在 $G_{Overlay}$ 中求 $v_{hor}^{1,s_i}$ 与 $v_{hor}^{k+1,o_i}$ 之间的最短路径。

**证明** 在重叠网络 $G_{Overlay}$ 中，链 $i$ 源、目的节点 $(v_{hor}^{1,s_i}, v_{hor}^{k+1,o_i})$ 之间的最短路径权重累加和可以表示为式(19)，决策变量 $y_i^{e(u,v)}$ 代表 $G_{Overlay}$ 中的链路 $e(u,v)$ 是否包含于业务流 $i$ 的最短路径。

$$\min \sum_{u,v \in V_{hor}} w_{e(u,v)} y_i^{e(u,v)} \quad (19)$$

$$\begin{aligned} \text{其中, } \min \sum_{u,v \in V_{hor}} w_{e(u,v)} y_i^{e(u,v)} &= \min \sum_{e(u,v) \in e_{hor}} \{\varepsilon r_e \lambda_e + (1 - \varepsilon) \delta_{e(u,v)}^i\} \cdot y_i^{e(u,v)} + \sum_{e(u,v) \in e_{ver}} \\ &\{\varepsilon r_c \{C_i^k + D_k[w_i^m - w_i^m]^+\} + (1 - \varepsilon) \delta_m^i\} y_i^{e(u,v)} = \\ &\min \sum_{e(u,v) \in e_{hor} \cup e_{ver}} \{\varepsilon \{r_c \{C_i^k + D_k[w_i^m - w_i^m]^+\} + \\ &r_{e(u,v)} \lambda_e\} y_i^{e(u,v)} + \sum_{e(u,v) \in e_{hor} \cup e_{ver}} (1 - \varepsilon) \{\delta_{e(u,v)}^i + \delta_m^i\} \\ &y_i^{e(u,v)} = \min\{\varepsilon \varphi_i + (1 - \varepsilon) \delta_i\}. \end{aligned}$$

由式(16)可知，链 $i$ 编排的时空优化问题为 $\min\{\varepsilon \varphi_i + (1 - \varepsilon) \delta_i\}$ ，定理得证。

#### 4.2 基于重叠网络的模拟退火迭代优化编排算法 (ONSA)

为提升请求集合 $I$ 的编排时空优化效果，本文设计一种基于重叠网络的模拟退火迭代优化编排策略(ONSA)，借助模拟退火算法(SA)思想提高效率。算法流程如表1所示。除 $G = (V, E)$ 、请求集合 $I$ 及网络状态外，ONSA还需输入SA算法必需的4个参数：初始温度 $\tau_0 \in R^+$ ，终止温度 $\tau_{min} \in R^+$ ，降温因子 $\rho \in (0, 1)$ ，Markov链长度 $L$ 。

##### (1) 初始编排策略

ONSA算法首先生成初始编排策略 $OS_{init}$ 。SA算法的求解性能与初始状态没有强依赖关系<sup>[15]</sup>，但需要在 $OS_{init}$ 基础上添加非随机性扰动产生新解，为避免求解过程陷入局部最优，ONSA为其NF请求 $\beta_i^k$ 随机分配部署节点生成初始编排方案 $OS_{init}$ 。

##### (2) 新解重构

为提高算法效率，ONSA算法交替使用两种非随机扰动方式产生重映射集合进行新解重构。

(a) 负载均衡重构：计算VNF实例的平均负载强度 $\bar{C}_{load}$ ，通过各实例负载强度 $C_{load}^m$ 偏离 $\bar{C}_{load}$ 的程度计算其释放概率 $p_{rem}$ ，依据 $p_{rem}$ 选择需要释放的VNF实例 $m$ (第6行)，将所有经过 $m$ 处理的SFC请求纳入重映射集合 $I_{re}$ ，用于新解重构(第7行)。其中实例 $m$ 的释放概率 $p_{rem}^m$ 为

表1 基于重叠网络的模拟退火迭代优化编排策略

输入:	$G = (V, E)$ , $I$ , 底层网络状态 SA参数 $\{\tau_0, \tau_{\min}, \rho, L\}$
输出:	集合 $I$ 的时空优化编排方案 $OS_{\text{opt}}$
(1)	$\forall i \in I$ , 随机部署 $\beta_i^k$ 生成 $OS_{\text{mit}}$
(2)	初始化参数, $\tau \leftarrow \tau_0$ , $Object_{\text{now}} \leftarrow Object(OS_{\text{mit}})$
(3)	while $\tau > \tau_{\min}$ do
(4)	for $l = 1 : L$ do
(5)	if (rand $< 0.5$ )
(6)	计算 $\bar{C}_{\text{load}}$ 与 $p_{\text{rem}}$ 并选择释放实例 $m$
(7)	将经 $m$ 处理的请求纳入 $I_{\text{re}}$ 并释放 $m$
(8)	else
(9)	$\forall i \in I$ , 将 $\delta_i > \delta_i^{s,o}$ 的请求纳入 $I_{\text{re}}$
(10)	end for
(11)	for $i$ in $I_{\text{re}}$ do
(12)	依据 $\beta_i^k$ 与底层网络状态构造 $G_{\text{Overlay}}$
(13)	删除资源不足的节点与链路
(14)	计算链路权重 $w_{\text{ver}}^e, w_{\text{hor}}^e$
(15)	$Path_{\text{Overlay}}^i \leftarrow \text{Dijkstra}(v_{\text{hor}}^{1,s_i}, v_{\text{hor}}^{k+1,o_i})$
(16)	$Path_i \leftarrow Path_{\text{Overlay}}^i$
(17)	end for
(18)	$Object(OS_{\text{new}}) = \sum_{i \in I} [\varepsilon \varphi_i + (1 + \varepsilon) \delta_i]$
(19)	if $\Delta_{\text{Obj}} < 0$
(20)	$OS_{\text{now}} \leftarrow OS_{\text{new}}$
(21)	else
(22)	$P_{\text{acc}} = \exp(-\Delta_{\text{Obj}}/T)$
(23)	$\tau \leftarrow \tau \rho$
(24)	end while
(25)	$OS_{\text{opt}} \leftarrow OS_{\text{now}}$

$$p_{\text{rem}}^m = \frac{|C_{\text{load}}^m - \bar{C}_{\text{load}}|}{\sum_{\tilde{m} \in M} |C_{\text{load}}^{\tilde{m}} - \bar{C}_{\text{load}}|} \quad (20)$$

(b) 超时重构: 当SFC端到端累计时延  $\delta_i$  超出容忍时延  $\delta_i^{s,o}$  时, 说明这些SFC传输路径中包含拥塞链路或高负载VNF实例, 为提高NFV系统的接受率, 超时重构策略将所有超出容忍时延的SFC请求纳入重映射集合  $I_{\text{re}}$  (第9行)。

$\forall i \in I_{\text{re}}$ , ONSA依据  $\beta_i^k$  与底层网络状态构造重叠网络  $G_{\text{Overlay}}$ , 删除资源余量低于链  $i$  需求的垂直节点  $v_{\text{ver}}^s$  与链路  $e_{\text{hor}}^s, e_{\text{ver}}^s$ , 并依据负载强度与相关时延计算水平、垂直链路的权重  $w_{\text{ver}}^e, w_{\text{hor}}^e$  (第12~14行)。在  $G_{\text{Overlay}}$  中执行最短路径算法得到链  $i$  的源、目的节点  $v_{\text{hor}}^{1,s_i}$  与  $v_{\text{hor}}^{k+1,o_i}$  间的最短路径  $Path_{\text{Overlay}}^i$ ,

转化路径  $Path_i$  即可得到链  $i$  的重映射方式 (第15, 16行)。ONSA遍历  $I_{\text{re}}$  即可获得重构后的新解  $OS_{\text{new}}$ 。

### (3) 适应度与接受概率

ONSA算法依据优化目标  $Object(OS_{\text{new}}) = \sum_{i \in I} [\varepsilon \varphi_i + (1 - \varepsilon) \delta_i]$  计算新解  $OS_{\text{new}}$  的适应度, 并结合Metropolis准则决策新解的取舍 (第18~22行), 其接受概率  $P_{\text{acc}}$  可表示为式(21)。

$$P_{\text{acc}} = \left\{ \begin{array}{ll} 1, & \Delta_{\text{Obj}} < 0 \\ \exp(-\Delta_{\text{Obj}}/T), & \Delta_{\text{Obj}} > 0 \end{array} \right\} \quad (21)$$

$$\Delta_{\text{Obj}} = Object(OS_{\text{new}}) - Object(OS_{\text{now}})$$

### (4) 算法复杂度分析

ONSA算法的复杂度主要取决于重叠网络的规模与链路权重的计算。设  $\max\{K(i), \forall i \in I\} = K$ , 则  $G_{\text{Overlay}}$  的节点规模为  $(3K + 1)|V|$ , 链路规模为  $(K + 1)|E| + 2K|V|$ 。设VNF实例集合为  $M$ , 计算链路权重的复杂度为  $O(|E| + |M|)$ , 降温次数为  $R$ , 则ONSA算法复杂度为  $O(|M| + K(|E| + |V| \lg K|V|)RL)$ 。由于参数  $R$  与  $L$  均为常数, 且由重叠网络结构可知  $M$  与  $K|V|$  规模相同, 则当问题规模增大时, 影响ONSA算法的主要因素为节点规模, 即  $O(K|V| \lg K|V|)$  项, 由文献[16]可知  $K$  的取值往往不超过4, 则ONSA算法复杂度最终可化简为  $O(K|V| \lg |V|)$ 。

## 5 仿真分析

为评估算法的有效性, 本文在实验中与VNFOP算法[5]和Greedy算法[9]进行对比分析, 其中, VNFOP算法[5]复杂度为  $O(|V|^2K)$ , 与本文所提的ONSA算法相比复杂度较高。Greedy算法[9]的复杂度  $O(K^2|V|)$ , 由于  $K$  远小于  $|V|$ , 则Greedy算法的复杂度最小, 相同问题规模下所需的运行时间最短。

### 5.1 仿真参数

#### (1) 拓扑选择

仿真实验建立在真实的Geant网络拓扑中。Geant网络包含22个节点与36条链路[15]。其中每个节点计算资源量为1600百万指令每秒 (Million Instructions Per Second, MIPS), 链路带宽容量300 Mbps, 网络中所有节点兼备计算与转发能力。

#### (2) NFV系统参数

实验使用4种VNF参数如表2所示。为了确定SFC请求数量的变化范围, 本文首先根据文献[10]设定服务请求所遵循的参数: 每条SFC包含2~4个NF请求[16], 每个NF被选中的概率相同; NF所需计算资源服从2~8 MIPS的均匀分布; 请求带宽服从

0.4~0.7 Mbps 的均匀分布；容忍时延服从 900~1100 ms 的均匀分布。依据以上设定，当 SFC 请求强度变化范围为 100~1600 时，底层网络节点平均资源占用率的变化范围为 8%~74%，该范围能够较为充分地反映出 SFC 请求数量变化对 NFV 系统负载带来的影响，利于算法性能验证。同时，预先设定权重调节参数取值为  $\epsilon = 0.733$ ，该参数的合理性将通过实验进行验证。

### 5.2 仿真验证与分析

**实验 1** 为证明 ONSA 算法的时间优化性能，实验 1 对比了不同请求强度下 SFC 平均端到端时延与请求接受率。图 4 可以得出 ONSA 算法相对其他算法具有较好的端到端时延优化性能。借助的链路权重所包含的时延信息，ONSA 算法在构建 SFC 路径时可避开拥塞链路，此外，ONSA 可自适应地决策是否需要新增 VNF 实例以均衡负载，从而降低处理时延。VNFOP 算法能及时供应 SFC 请求所需 VNF 实例，且前期 VNF 实例多处于低负载状态，因此具有一定程度的时延优化效果，但由于没有考虑链路拥塞对端到端时延产生的影响，导致后期请求强度较高时，业务流的平均端到端时延明显提高。Greedy 力求通过复用或合并已部署的 VNF 实例降低计算资源开销，由于忽略链路拥塞程度以及 VNF 负载，导致请求平均端到端时延提升明显。与 VNFOP、Greedy 算法相比，ONSA 算法的端到端时延优化性能分别平均提升 29.5% 与 21.6%。

通过图 5 可知，ONSA 算法的请求接受率始终维持在较高水平，这是因为 ONSA 算法的新解重构

方式不仅能对超出容忍时延的 SFC 进行重映射，还以相对较大的概率释放了过载的 VNF 实例，降低 SFC 请求因超出容忍时延被拒绝的可能性。当请求强度达到 800, 1100 以上时，VNFOP 的请求接受率相对 ONSA 降低 6.7%, 12.4%，造成这一现象的原因是仅通过增加 VNF 实例降低处理时延的方式无法起到缓解链路拥塞的作用。Greedy 算法仅注重计算资源优化，在合并/复用 VNF 实例时易引起 VNF 过载从而增加处理时延，导致接受率相对 ONSA 算法平均降低 22.3%。

**实验 2** 通过统计 VNF 实例负载分布，计算链路资源利用率，实验 2 对比分析了 ONSA 算法空间优化的性能。图 6 统计了 3 种算法在相同请求强度下 VNF 实例的负载分布情况。相对 VNFOP 算法而言，ONSA 算法能较好地均衡 VNF 实例负载，原因是通过  $G_{Overlay}$  垂直层的节点分割方法，ONSA 算法能够权衡复用已有实例与新增 VNF 实例对构建 SFC 最优路径的利弊，此外，新解重构过程中的负载均衡重构方式也有助于 VNF 实例的负载均衡。当 SFC 请求数量较多时 (1200 以上)，ONSA 算法能将大多数 VNF 实例负载稳定在 67.2%~86.2%，负载强度不再随请求的增加而提升以避免实例过载。VNFOP 算法为保证 SFC 接受率借助增广网络及时供应 VNF 实例，但大量部署 VNF 实例极易引起负载不均。Greedy 算法通过尽可能复用已部署的 VNF 实例能

表 2 VNF 参数

VNF	实例化开销 (MIPS)	计算资源 (MIPS)	Packet 处理时间 ( $\mu s$ )
Firewall	5	60	15
Encryption	5	60	15
IDS	8	40	20
NAT	4	60	15

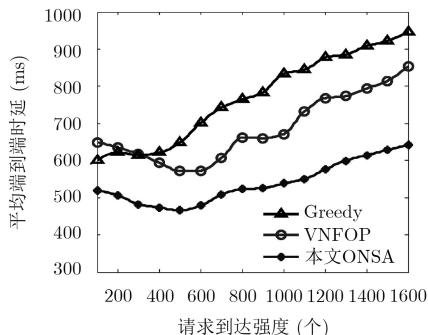


图 4 服务请求处理时间

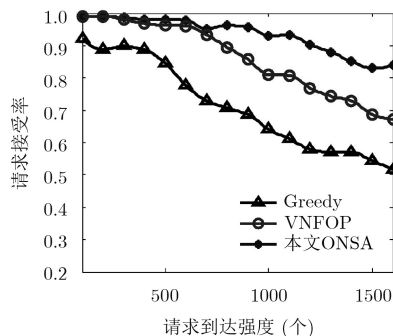


图 5 请求接受率

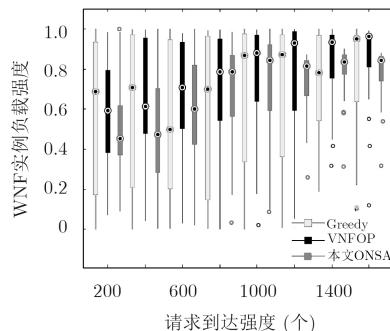


图 6 VNF 负载强度分布变化

够提高计算资源利用率,随请求强度的增加,VNF实例利用率也明显提升,但容易造成部分VNF实例过载从而增加处理时延。图7对比了链路资源利用率,可见在提升VNF计算资源利用率的同时,ONSA算法与Greedy算法均需付出相应的链路资源(带宽)代价,但ONSA算法相对Greedy算法能节省至少12.4%的链路资源。

**实验3** 该实验通过对比NFV系统运营开销来评价不同编排方案的资源优化效果,最后通过改变参数 $\varepsilon$ 的取值分析时空优化效果的变化。不同请求强度下NFV系统运营开销的变化如图8所示,可见随请求强度的增加,3种策略的运营开销都在升高。Greedy算法通过聚合分散的VNF实例有效降低了VNF数量与实例化开销,提高计算资源利用率,但SFC为到达聚合后的VNF实例难免需要延长传输路径,增加链路资源开销。VNFOP算法为保证请求接收率,前期部署VNF实例数量相对较多,实例化开销较大,但VNF利用率会随请求强度的增加逐渐提升,因此后期NFV运营开销增幅相对Greedy算法有所下降。相比之下,基于ONSA算法的运营开销相对Greedy与VNFOP算法平均降低13.7%与15.2%。

参数 $\varepsilon$ 用于调节时间与空间维度在优化目标中所占比重,进而影响NFV系统的运营开销与请求接受率。图9展示了改变 $\varepsilon$ 的取值对时空优化效果的

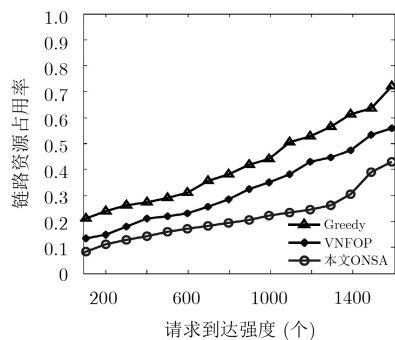


图7 链路资源利用率

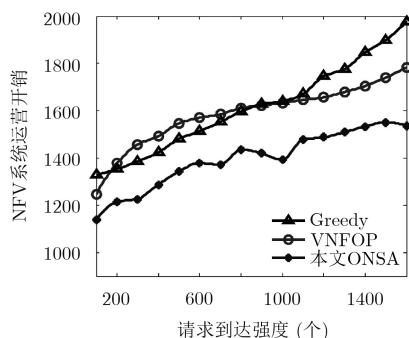


图8 NFV系统运营开销

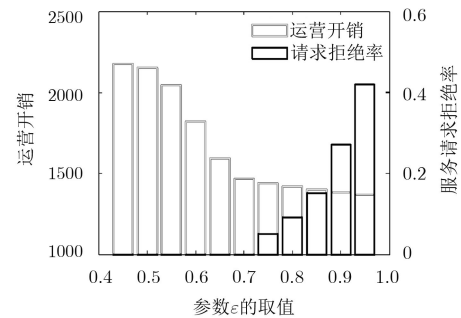


图9 请求接受率

影响,此时固定请求到达数量为800。可见当 $\varepsilon$ 取值较小时( $\varepsilon < 0.55$ ),ONSA算法所得编排方案计算资源相对充足,时间维度优化效果较好,不存在拒绝请求的情况,但由于虚拟资源的过量部署,运营开销达到2000以上;随 $\varepsilon$ 的增加( $0.55 < \varepsilon < 0.7$ ),运营开销明显降低,说明空间优化效果逐渐增强;而当 $\varepsilon > 0.75$ ,空间优化效果降低,且时间维度处于明显劣势,服务请求拒绝率增幅显著,原因是此时计算资源相对紧缺,VNF实例数量相对较少且多处于过载状态,SFC端到端累计时延易超出容忍时延。综上,参数 $\varepsilon$ 取值以接近0.7为宜。

## 6 结束语

本文提出一种基于重叠网络结构的SFC时空优化编排策略,通过建立重叠网络模型实现了计算与网络资源的分离,将SFC编排的NP难问题转化为易于求解的最短路径问题。设计了ONSA算法解决批量部署SFC集合的时空优化问题。仿真表明了ONSA算法在能够兼顾SFC的端到端时延优化与NFV系统资源优化。后续将结合深度强化学习算法,通过感知重叠网络状态自适应决策SFC编排。

## 参考文献

- [1] 郭江兴. 新型网络技术发展思考[J]. 中国科学: 信息科学, 2018, 48(8): 1102-1111. doi: 10.1360/N112018-00062. WU Jiangxing. Thoughts on the development of novel network technology[J]. *CIENTIA SINICA Informationis*, 2018, 48(8): 1102-1111. doi: 10.1360/N112018-00062.
- [2] HAN Bo, GOPALAKRISHNAN V, JI Lusheng, et al. Network function virtualization: Challenges and opportunities for innovations[J]. *IEEE Communications Magazine*, 2015, 53(2): 90-97. doi: 10.1109/mcom.2015.7045396.
- [3] QUINN E and NADEAU T. Problem statement for service function chaining[R]. IETF RFC-Informational, 2015.
- [4] MOENS H and DE TURCK F. VNF-P: A model for efficient placement of virtualized network functions[C]. Proceedings of the 10th International Conference on

- Network and Service Management and Workshop, Rio de Janeiro, Brazil, 2014: 418–423. doi: [10.1109/CNSM.2014.7014205](https://doi.org/10.1109/CNSM.2014.7014205).
- [5] BARI F, CHOWDHURY S R, AHMED R, *et al.* Orchestrating virtualized network functions[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(4): 725–739. doi: [10.1109/TNSM.2016.2569020](https://doi.org/10.1109/TNSM.2016.2569020).
- [6] JANG I, SUH D, PACK S, *et al.* Joint optimization of service function placement and flow distribution for service function chaining[J]. *IEEE Journal on Selected Areas in Communications*, 2017, 35(11): 2532–2541. doi: [10.1109/JSAC.2017.2760162](https://doi.org/10.1109/JSAC.2017.2760162).
- [7] 史久根, 张径, 徐皓, 等. 一种面向运营成本优化的虚拟网络功能部署和路由分配策略[J]. *电子与信息学报*, 2019, 41(4): 973–979. doi: [10.11999/JEIT180522](https://doi.org/10.11999/JEIT180522).
- SHI Jiugen, ZHANG Jing, XU Hao, *et al.* Joint optimization of virtualized network function placement and routing allocation for operational expenditure[J]. *Journal of Electronics & Information Technology*, 2019, 41(4): 973–979. doi: [10.11999/JEIT180522](https://doi.org/10.11999/JEIT180522).
- [8] 陈卓, 冯钢, 刘蓓, 等. 运营商网络中面向资源碎片优化的网络服务链构建策略[J]. *电子与信息学报*, 2018, 40(4): 763–769. doi: [10.11999/JEIT170641](https://doi.org/10.11999/JEIT170641).
- CHEN Zhuo, FENG Gang, LIU Bei, *et al.* Construction policy of network service chain oriented to resource fragmentation optimization in operator network[J]. *Journal of Electronics & Information Technology*, 2018, 40(4): 763–769. doi: [10.11999/JEIT170641](https://doi.org/10.11999/JEIT170641).
- [9] WEN Tao, YU Hongfang, SUN Gang, *et al.* Network function consolidation in service function chaining orchestration[C]. Proceedings of 2016 IEEE International Conference on Communications, Kuala Lumpur, Malaysia, 2016: 512–519. doi: [10.1109/ICC.2016.7510679](https://doi.org/10.1109/ICC.2016.7510679).
- [10] PEI Jianing, HONG Peilin, XUE Kaiping, *et al.* Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2018, 11(8): 367–379. doi: [10.1109/TPDS.2018.2880992](https://doi.org/10.1109/TPDS.2018.2880992).
- [11] 陈卓, 冯钢, 刘蓓, 等. 运营商网络中面向时延优化的服务功能链迁移重配置策略[J]. *电子学报*, 2018, 46(9): 2229–2237. doi: [10.3969/j.issn.0372-2112.2018.09.026](https://doi.org/10.3969/j.issn.0372-2112.2018.09.026).
- CHEN Zhuo, FENG Gang, LIU Bei, *et al.* Delay optimization oriented service function chain migration and re-deployment in operator network[J]. *Acta Electronica Sinica*, 2018, 46(9): 2229–2237. doi: [10.3969/j.issn.0372-2112.2018.09.026](https://doi.org/10.3969/j.issn.0372-2112.2018.09.026).
- [12] ARTEAGA C H T, RISSOI F, and RENDON O M C. An adaptive scaling mechanism for managing performance variations in network functions virtualization: A case study in an NFV-based EPC[C]. Proceedings of the 13th International Conference on Network and Service Management, Tokyo, Japan, 2017: 1–7. doi: [10.23919/CNSM.2017.8255982](https://doi.org/10.23919/CNSM.2017.8255982).
- [13] LI Tong, BAUMBERGER D, HAHN S, *et al.* Efficient and scalable multiprocessor fair scheduling using distributed weighted round-robin[C]. Proceedings of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Raleigh, USA, 2009: 65–74. doi: [10.1145/1504176.1504188](https://doi.org/10.1145/1504176.1504188).
- [14] AGARWAL S, MALANDRINO F, CHIASSERINI C F, *et al.* VNF placement and resource allocation for the support of vertical services in 5G networks[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(1): 433–446. doi: [10.1109/TNET.2018.2890631](https://doi.org/10.1109/TNET.2018.2890631).
- [15] LANGE S, GRIGORJEW A, ZINNER T, *et al.* A multi-objective heuristic for the optimization of virtual network function chain placement[C]. Proceedings of the 29th International Teletraffic Congress, Genoa, Italy, 2017: 152–161. doi: [10.23919/ITC.2017.8064351](https://doi.org/10.23919/ITC.2017.8064351).
- [16] WANG Xiaoke, WU Chuan, LE F, *et al.* Online VNF scaling in datacenters[C]. Proceedings of the IEEE 9th International Conference on Cloud Computing, San Francisco, USA, 2016: 140–147. doi: [10.1109/CLOUD.2016.0028](https://doi.org/10.1109/CLOUD.2016.0028).
- 谷允捷: 男, 1994年生, 博士生, 研究方向为新型网络体系结构, 网络功能虚拟化。
- 胡宇翔: 男, 1982年生, 副研究员, 研究方向为新型网络体系结构, 软件定义网络。
- 谢记超: 男, 1995年生, 硕士生, 研究方向为网络安全、网络功能虚拟化与云安全。