

基于性能感知的网络切片部署方法

黄开枝, 潘启润, 袁泉, 游伟, 汤红波

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

摘 要: 为了应对网络切片中共享物理资源的虚拟网络功能 (VNF) 间因资源竞争带来的性能下降问题, 提出了一种基于性能感知的网络切片部署方法。在网络切片实例部署时, 采用先虚拟节点映射再虚拟链路映射的两阶段部署方式。在虚拟节点部署阶段, 首先从资源供需的角度定义了 VNF 的性能影响因子, 以量化 VNF 的性能影响程度; 然后求出网络切片实例中所有 VNF 在可部署的物理服务器上的性能影响因子总和, 将性能影响因子总和最小的物理服务器作为映射位置, 采用模拟退火-离散粒子群算法求出节点映射结果。在虚拟链路部署阶段, 采用最短路径算法求出链路映射结果。仿真结果表明, 所提方法有效降低了其他网络切片实例对自身的性能影响。

关键词: 网络切片; 虚拟网络映射; 性能下降; 性能隔离

中图分类号: TP393.03

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019169

Method of network slicing deployment based on performance-aware

HUANG Kaizhi, PAN Qirun, YUAN Quan, YOU Wei, TANG Hongbo

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

Abstract: In order to deal with the performance degradation caused by resource contention due to the sharing of physical resources between VNF in the network slicing, a network slicing deployment method based on performance-awareness was proposed. When deploying network slice instances, first two-phase deployment that mapping virtual nodes was adopted, and then virtual links were mapped. In the virtual nodes mapping phase, the VNF performance influences factor was defined from the perspective of resource supply and demand to quantify the degree of VNF performance impact. Then the sum of the performance influence factors of all VNF on the deployable physical server was found in the network slice instance, the physical server with the smallest sum of performance influence factors was used as the mapping location, and the simulated annealing-discrete particle swarm algorithm was used to find the nodes mapping result. In the virtual links mapping phase, the shortest path algorithm was used to obtain the link mapping result. The simulation results show that the proposed method reduces the impact of other network slices on its own service performance.

Key words: network slicing, virtual network embedding, performance degradation, performance isolation

1 引言

5G 系统可以支持各种各样的垂直行业, 如自动驾驶、远程医疗、工业自动化等。传统移动网络“一刀切”的网络架构无法在垂直方向满足网络在可扩

展性、可用性和可靠性方面的性能差异要求^[1]。为适应多种类型的垂直行业的需求, 提出了网络切片 (NS, network slicing) 的概念。下一代移动网络联盟 (NGMN, the next generation mobile network alliance) 将网络切片定义为在相同的物理基础设施上运行的

收稿日期: 2018-07-05; 修回日期: 2019-06-28

通信作者: 潘启润, panqirun03@163.com

基金项目: 国家重点研发计划网络空间安全专项基金资助项目 (No.2016YFB0801605), 国家自然科学基金创新群体资助项目 (No.61521003)

Foundation Items: The National Key Research and Development Program Cyberspace Security Special (No.2016YFB0801605), The National Natural Science Foundation Innovative Groups Project of China (No.61521003)

多个虚拟网络^[2]。每个网络切片是一个独立的虚拟网络，它可以根据特定的服务需求（如超低时延、高可靠性和移动性等）选取配置相应的虚拟网络功能，创建可编程的网络切片实例（NSI, network slice instance），为不同用户提供端到端的定制化服务。

为了提高物理资源利用率，基础设施运营商将不同 NSI 的虚拟网络功能（VNF, virtual network function）部署在相同的物理服务器上，与此同时，采用资源复用技术实现资源的超额分配，这种部署方式会导致共享物理资源的 VNF 间过度竞争底层硬件资源（如 CPU、存储等），造成服务性能的严重下降，影响用户体验。但是现有的关于网络切片部署的研究主要集中在提高资源利用率或降低能源消耗^[3-5]，仅有少数文献^[6-13]考虑到上述问题。因而，在网络切片部署时如何有效降低因过度竞争资源带来的服务性能下降，已成为基础设施运营商面临的一个挑战。文献[6]研究了虚拟机（VM, virtual machine）隔离技术，通过安装多个网络适配器，降低了共存 VM 间的性能干扰。为了使底层管理者能够非侵入式地监测 VM，文献[7]设计了一种硬件计数器，该计数器可以很好地反映不同类型应用所受性能干扰的程度。文献[8]提出了一种基于层次聚类和优化搜索的功耗感知算法，从资源利用率的相关性角度量化了 2 个共存 VM 间的资源使用情况。文献[9]设计了一个通用模型来描述 VM 间的性能下降，根据不同 VM 的生命周期设计 VM 调度策略，不但能够抑制物理服务器上的性能下降，还降低了运营成本。文献[10]基于 Xen I/O 机制和虚拟 CPU（VCPU, virtual CPU）调度机制设计了一种性能估量模型，根据该模型估计出 VM 迁移前与迁移后的性能下降，并提出了一种轻量级的干扰感知的 VM 在线迁移策略，但是该策略完全基于 Xen I/O 机制和 VCPU 调度机制，并不具有通用性。文献[11]建立了 VM 的性能模型，基于该模型放置 VM 不但能够保证 VM 的性能而且能够有效降低服务器的功耗。文献[12]提出了一种 VM 的在线放置算法，综合考虑不同类型的资源使用情况，并使用基于阈值的位移策略来降低因服务器过载而导致的性能下降。为了解决虚拟化环境中因超额预订带来的性能波动，文献[13]设计了一种基于服务等级的 VM 部署策略，根据实时监测得到的网络吞吐量和服务响应时间的变化情况对不同服务等级的 VM 进行资源分配，以实现 VM 间的性能隔离，但是该方法未能有效度量虚拟节点的性能下降，导致执行时服务性能已经存在不同程度的下降，所以该方

法对用户忍耐服务性能下降的能力要求很高。

针对文献[13]中存在的问题，本文提出了一种基于性能感知的网络切片部署方法。在网络切片实例部署时采用先虚拟节点映射再虚拟链路映射的两阶段部署方式。在虚拟节点映射中，为有效度量 VNF 的性能下降，从 VNF 所需资源与物理服务器可提供资源之比的角度定义了 VNF 的性能影响因子，优先将 VNF 部署在能够使网络切片实例性能影响因子总和最小的物理服务器上，然后利用模拟退火-离散粒子群算法求解节点映射结果，从而能在多项式时间内求出最佳部署结果。在虚拟链路映射中，选取满足链路资源约束条件的最短路径作为链路映射结果。仿真结果表明，本文提出的方法在全网性能影响因子总和及性能影响程度方面均低于其他对比方法，证明了该方法可以在相同物理基础设施的多个网络切片实例间实现有效的性能隔离。

2 问题分析

2.1 网络切片实例化及部署模型建立

网络切片实例是一个专用的虚拟网络，一个网络切片实例由一条或多条服务功能链（SFC, service function chain）^[14-15]组成，不同的服务功能链由多个带顺序约束的 VNF 组成。网络切片实例化流程如图 1 所示。当用户发起服务请求时，网络管理和编排器根据用户服务请求的需求特点组合 VNF 形成 SFC；然后，结合物理网络 G_s 的剩余物理资源的状态信息，尝试为 SFC 分配物理资源，如果底层物理网络可以满足服务请求的节点和链路的虚拟资源需求，则底层资源池为其分配物理资源完成实例化，并获取可以部署的位置逻辑视图 G_v ，更新物理网络的剩余资源信息，配置 NSI 的生命周期并监控运行状态；最后，按照 $f_1 \rightarrow f_2 \rightarrow f_3$ 的顺序完成数据的传输和处理，为用户请求提供服务^[16]。

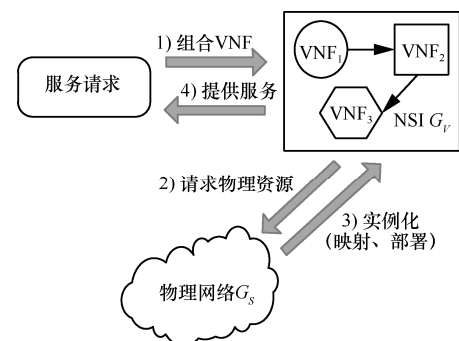


图 1 网络切片实例化流程

物理网络用一个有权无向图 $G_S = (N_S, L_S, C_N, C_L)$ 表示, 其中, N_S 和 L_S 分别为物理节点集合和物理链路集合, $n_s \in N_S$ 表示通用物理服务器, $l_s(m, n) \in L_S$ 是连接服务器 m 和服务器 n 间的物理链路; C_N 和 C_L 分别表示物理服务器和物理链路的资源属性, 物理服务器可以提供多种物理资源, $C_N = (c_1, c_2, \dots, c_m)$ 表示物理服务器可以提供的物理资源集合, $c_m \in C_N$ 表示物理资源类型, 在本文中物理服务器的资源包括计算资源、存储资源和网络资源, C_L 表示物理链路可以提供的带宽资源。

假设在物理网络上部署了 K 个网络切片实例, 其中, 第 k 个网络切片实例可以表示为 $G_V^k = (N_V^k, L_V^k, R_N^k, R_L^k)$, $k \in [0, K]$ 。同样地, N_V^k 和 L_V^k 表示第 k 个网络切片实例的 VNF 集合和虚拟链路集合。 $R_N^k = (r_1^k, r_2^k, \dots, r_m^k)$ 指第 k 个网络切片实例中 VNF 所需的虚拟资源集合; $r_m^k \in R_N^k$ 表示虚拟资源类型, 包括 VNF 所需的计算资源、存储资源和网络资源, R_L^k 表示虚拟链路所需的带宽资源。

网络切片实例 G_V^k 到物理网络 G_S 的映射称为网络切片实例部署, 在满足资源约束条件下, 将 VNF 映射到物理服务器上, 将虚拟链路映射到一组物理链路上。变量 $x_{n_s^k, k, n_s}$ 和 $y_{l_s^k, k, P}$ 表示网络切片实例的 VNF 和虚拟链路是否映射成功, 值为 1 表示映射成功, 值为 0 表示映射失败, 如式(1)和式(2)所示。

$$x_{n_s^k, k, n_s} = \begin{cases} 1, & n_s^k \text{ 成功映射到 } n_s \text{ 上} \\ 0, & n_s^k \text{ 未成功映射到 } n_s \text{ 上} \end{cases} \quad (1)$$

$$y_{l_s^k, k, P} = \begin{cases} 1, & l_s^k \text{ 成功映射到一组物理链路 } P \text{ 上} \\ 0, & l_s^k \text{ 未成功映射一组物理链路 } P \text{ 上} \end{cases} \quad (2)$$

2.2 性能下降分析

当多个网络切片实例部署到同一物理网络上时, 必然会存在多个 VNF 部署在同一物理服务器上的情况, 如图 2 中服务器 C 上的 VNF_b 和 VNF_e, 在开放共享物理资源的环境下, VNF_b 和 VNF_e 可能由于误操作或过载而占用大量物理资源, 影响其他 NSI 的服务性能。当 NSI 的服务性能下降时, 会给网络带来多方面的影响, 如表 1 所示。为应对该问题, 文献[13]根据实时监测网络吞吐量 λ 和服务响应时间

t , 在资源分配时通过最小化 $\frac{\alpha_1(\bar{\lambda} - \lambda)}{\lambda} + \frac{\alpha_2(t - \bar{t})}{\bar{t}}$ 来避免性能下降, $\bar{\lambda}$ 和 \bar{t} 分别代表用户期望的网络

吞吐量和 服务响应时间; 但是该方法在检测到 λ 降低和 t 增加时才采取行动, 这时虚拟节点的性能已经存在下降, 如何有效度量虚拟节点的性能下降并在网络切片部署时采取有效方法控制虚拟节点的性能下降已成为关键问题。虚拟节点的性能下降主要是由多虚拟节点间物理资源的过度竞争引起的^[9], 如图 2 所示, 在服务器 B 上只有 VNF_a, 没有其他的 VNF 占用物理资源, 故 VNF_a 基本可以达到期望的服务性能;但是在服务器 C 上, VNF_b 和 VNF_e 间存在物理资源的竞争, 故可能会导致虚拟节点的性能下降, 因此可以根据服务器的资源使用情况来对虚拟节点的性能影响大小进行表征。

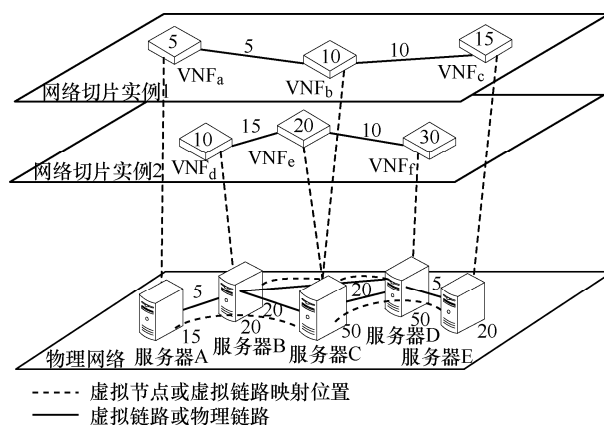


图 2 NSI 请求部署模型

表 1 性能下降所带来的影响

网络参数	影响
网络吞吐量 λ	降低
服务响应时间 t	增加
缓存失效率 μ	增加

本文提出了一种基于性能感知的网络切片部署方法, 一方面从多维度的资源供需角度定义了性能影响因子, 以刻画当前物理服务器中其他应用给自身应用所带来的性能影响;另一方面根据该性能影响因子为虚拟节点选择可部署的服务器, 采用改进型离散粒子群算法求出部署结果, 并且采用最短路径算法求出链路部署结果。

3 基于性能感知的网络切片部署方法

本文提出的基于性能感知的网络切片部署方法采用先虚拟节点映射再虚拟链路映射的两阶段部署方式^[17]。首先进行虚拟节点的部署, 对满足节点资源约束及唯一性约束的物理服务器, 根据性能

影响因子的定义求出部署在这些物理服务器上时 VNF 的性能影响因子,选择使 NSI 请求中所有 VNF 性能影响因子总和最小的物理服务器作为最终的部署位置,并利用改进型离散粒子群算法求解节点映射结果;然后进行虚拟链路部署,在满足链路资源约束的条件下,为相邻 VNF 间的虚拟链路选择最短的物理链路进行映射。

3.1 虚拟节点映射

在进行虚拟节点映射时,首先从 VNF 所需资源与服务器可提供资源之比的角度定义 VNF 的性能影响因子,并结合 VNF 性能影响因子的定义,建立了虚拟节点的映射模型,然后利用模拟退火-离散粒子群算法求出节点映射结果。

3.1.1 性能影响因子的定义

在物理资源超额分配的情况下,多个 VNF 间过度竞争一种或多种物理资源,使 VNF 的资源供需不匹配带来服务性能的下降。为了估计 VNF 的性能影响程度,本文将 VNF 的虚拟资源的需求与服务器的剩余物理资源的比值定义为性能影响因子。假设 VNF 运行在相同类型的硬件上,在物理服务器 $x \in N_s$ 上存在 n 个 VNF,第 i 个 VNF 的性能影响因子 η_x^i 如式(3)所示。

$$\eta_x^i = \theta_1 \frac{V_{\text{CPU}}^i}{P_{\text{CPU}}^{i,x}} + \theta_2 \frac{V_{\text{mey}}^i}{P_{\text{mey}}^{i,x}} + \theta_3 \frac{V_{\text{net}}^i}{P_{\text{net}}^{i,x}}, i \in n \quad (3)$$

其中, V_{CPU}^i 、 V_{mey}^i 和 V_{net}^i 分别表示 VNF _{i} 所需的虚拟 CPU、存储和网络的资源; $P_{\text{CPU}}^{i,x}$ 、 $P_{\text{mey}}^{i,x}$ 和 $P_{\text{net}}^{i,x}$ 分别表示物理服务器 x 上在未部署 VNF _{i} 时,剩余的物理 CPU 资源、存储资源和网络资源, $P_{\text{CPU}}^{i,x} = \Phi_{\text{CPU}}(x) - \sum_{j=i} r_{\text{CPU}}(j)$, $j \in n$, $\Phi_{\text{CPU}}(x)$ 表示物理服务器 x 可提供的 CPU 资源总量, $\sum_{j=i} r_{\text{CPU}}(j)$ 表示未在物理服务器 x 上部署 VNF _{i} 时,已部署的 VNF 的 CPU 资源使用总量; $P_{\text{mey}}^{i,x}$ 和 $P_{\text{net}}^{i,x}$ 定义同上。因不同类型的 VNF 所需的不同类型的资源大小存在差异,采用权重系数 θ 进行调整,且 $\theta_1 + \theta_2 + \theta_3 = 1$ 。

3.1.2 虚拟节点的映射模型建立

基于性能影响因子的定义,当一个 NSI 请求到达时,为了保证服务质量、避免 VNF 下降明显,在选择物理服务器时,需要保证 VNF 的性能影响因子较小。故将优化目标定为最小化全网所有 VNF 的性能影响因子总和,如式(4)所示。此外,还需满

足如式(5)~式(9)所示的约束条件。其中,当第 k 个 SFC 的虚拟网络功能 n_v^k 映射到服务器 n_s 上时,要保证物理服务器的剩余 CPU 资源、剩余存储资源及剩余网络资源不小于 VNF 所需的相应的虚拟资源,如式(5)~式(7)所示;式(8)表示一个 VNF 只能部署在一个物理服务器上,不能对其进行分割;式(9)保证了服务请求的所有 VNF 全部被部署到物理网络上。

$$\min \sum_{n_s \in N_S} \sum_{n_v^k \in N_V^k} (x_{n_v^k, k, n_s} \eta(n_v^k)) = \min \sum_{n_s \in N_S} \sum_{n_v^k \in N_V^k} \left(x_{n_v^k, k, n_s} \left(\theta_1 \frac{r_{\text{CPU}}(n_v^k)}{c_{\text{CPU}}(n_s)} + \theta_2 \frac{r_{\text{mey}}(n_v^k)}{c_{\text{mey}}(n_s)} + \theta_3 \frac{r_{\text{net}}(n_v^k)}{c_{\text{net}}(n_s)} \right) \right) \quad (4)$$

$$\sum_{n_v^k \in N_V^k} (x_{n_v^k, k, n_s} r_{\text{CPU}}(n_v^k)) \leq c_{\text{CPU}}(n_s), n_s \in N_S \quad (5)$$

$$\sum_{n_v^k \in N_V^k} (x_{n_v^k, k, n_s} r_{\text{mey}}(n_v^k)) \leq c_{\text{mey}}(n_s), n_s \in N_S \quad (6)$$

$$\sum_{n_v^k \in N_V^k} (x_{n_v^k, k, n_s} r_{\text{net}}(n_v^k)) \leq c_{\text{net}}(n_s), n_s \in N_S \quad (7)$$

$$\sum_{n_s \in N_S} x_{n_v^k, k, n_s} = 1, \forall n_v^k \in N_V^k \quad (8)$$

$$\sum_{n_v^k \in N_V^k} x_{n_v^k, k, n_s} \leq 1, \forall n_s \in N_S \quad (9)$$

3.1.3 虚拟节点映射结果求解

网络切片的虚拟节点映射是一个 NP-hard 问题,在多项式的时间内无法利用传统方法进行求解,因此采用智能算法进行求解。在实际网络中,网络管理和编排器作为网络切片中统筹全局的管理者,负责网络切片的映射问题,智能算法被写入编排器。因网络切片在实现时融入了软件定义网络(SDN, software defined network)的技术,基于 SDN 架构并借助 OpenFlow 协议可以在高度分布化和动态化的网络中实现快速的信息交互。如图 3 所示,网络管理和编排器接收网络切片实例和基础设施层发来的数据,将网络切片实例的服务描述、网络拓扑等信息载入网络服务目录中,将 VNF 所需的虚拟资源等描述类的信息载入 VNF 目录中,更新 NFV 实例库的信息,将物理网络拓扑和剩余资源等信息载入基础设施资源库中,待所有信息完备后,启动智能算法,开始求解映射结果。

本文使用基于模拟退火(SA, simulated annealing)的离散粒子群(DPSO, discrete particle

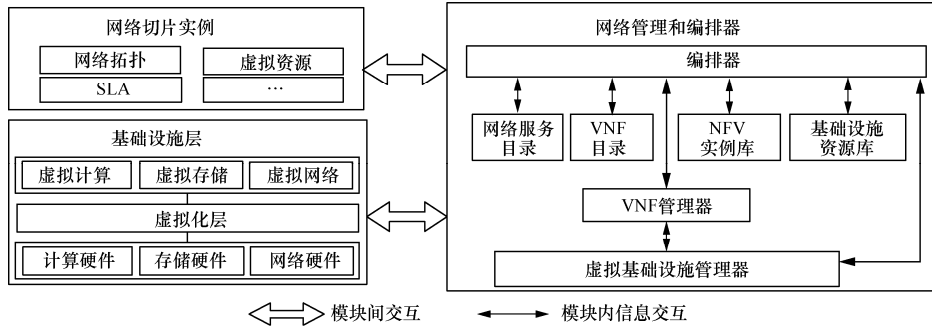


图 3 网络管理和编排器

swarm optimization) 算法求解节点映射结果。DPSO 优化算法是一种群智能优化算法, 通过研究鸟类在觅食过程中的行为特点, 把所研究问题的搜索空间视为作鸟类飞行空间, 每一个粒子都是所求问题的一个候选解, 最优解即为鸟类最终找到的食物, 在寻找最优解时利用不同个体之间的协作和信息的共享^[18]。

当多个网络切片实例映射到相同的物理网络上时, 为了保证不同网络切片实例间的 VNF 不会因为资源过度竞争而产生性能下降, 本文将当前 NSI 请求的所有 VNF 性能下降因子总和作为算法的适应度函数, 如式(10)所示。为了求出映射结果, 首先, 初始化由 n 个粒子组成一个种群, 这里的每一个粒子都是一个可能解, 第 i 个粒子可以用向量 $\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{ia})$ 来表示, 其中, i 是 $[1, n]$ 的整数, n 是种群规模, a 是当前 NSI 请求中包含的 VNF 总个数, x_{ia} 是当前 NSI 请求中第 a 个 VNF 映射到的物理服务器的序号。然后, 将 \vec{x}_i 代入式(10)求出当前映射方案的全网 VNF 性能下降影响总和, 因本文求的是最小值, 适应度越低, 代表当前映射方案越优。与此同时, 每当计算出一个粒子的适应度, 算法都会更新当前的个体最优解 \vec{P}_{ia} 和全局最优解 \vec{P}_{ga} 。为了使粒子快速而准确地飞向最优解, 每个粒子对应一个飞行速度, 第 i 个粒子的飞行速度可以用向量 $\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{ia}), i \in [1, n]$ 表示。本文采用带压缩因子的 DPSO 算法, 利用式(11)进行速度的更新; 压缩因子 χ 由式(12)求出; 学习因子 c_1 和 c_2 分别代表粒子本身和其他粒子所获经验, 调整学习因子能够控制粒子在全局探测和局部开采的平衡; 利用式(13)进行位置更新。

$$Fitness(G_V^k) = \sum_{n_s \in N_s} \sum_{n_v^k \in N_v^k} (x_{n_s^k, k, n_s} \eta(n_v^k)) =$$

$$\sum_{n_s \in N_s} \sum_{n_v^k \in N_v^k} \left(x_{n_s^k, k, n_s} \left(\theta_1 \frac{r_{CPU}(n_v^k)}{c_{CPU}(n_s)} + \theta_2 \frac{r_{mey}(n_v^k)}{c_{mey}(n_s)} + \theta_3 \frac{r_{net}(n_v^k)}{c_{net}(n_s)} \right) \right) \quad (10)$$

$$v_{ij}(k+1) = \chi \left[v_{ij}(k) + c_1 r_1 (p_{ij}(k) - x_{ij}(k)) + c_2 r_2 (p_{gj}(k) - x_{ij}(k)) \right] \quad (11)$$

$$\chi = \frac{2}{\left| 2 - C - \sqrt{C^2 - 4C} \right|}, C = c_1 + c_2, C > 4 \quad (12)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), j = 1, \dots, n \quad (13)$$

DPSO 算法是一种全局搜索算法, 易跳入局部极值点, 而 SA 算法可以有效避免该问题。SA 算法把系统降温时的能量看成是目标函数, 将优化过程模拟成系统降到最低温度的过程。利用 SA 算法对个体最优解进行领域搜索, 采用轮盘赌输策略来选择是否利用个体最优解替换全局最优解。突跳概率可利用式(14)进行计算, N 是种群大小, f 是适应度函数。速度更新式如式(15)所示。

$$\frac{e^{-\frac{Fitness(p_j) - Fitness(p_g)}{t}}}{\sum_{j=1}^N e^{-\frac{Fitness(p_j) - Fitness(p_g)}{t}}} \quad (14)$$

$$v_{ij}(k+1) = \chi \left[v_{ij}(k) + c_1 r_1 (p_{ij}(k) - x_{ij}(k)) + c_2 r_2 (p'_{ij}(k) - x_{ij}(k)) \right] \quad (15)$$

3.2 虚拟链路映射

完成网络切片实例的所有 VNF 的映射后, 基于最短路径^[19]的思想完成不同 VNF 间的链路映射。根据 SFC 的各个 VNF 间的顺序约束得到 VNF 的部署顺序 $\varphi_k = \{f_1, f_2, \dots, f_i, \dots, f_n\}$, 假设与虚拟节点 f_i 相连的上一级节点是 $f_{i_1}, f_{i_2}, \dots, f_{i_q}$, 而且经过同一 VNF 的 f_i 的所有的虚拟链路 $l_v^k(f_{i_p}, f_i), \forall p \in [1, q]$ 的映射优先级相同。第 k 个 SFC 的虚拟链路 l_v^k 映射

到物理链路 p 上，主要分为以下几步。

1) 设 f_{i_p} 和 f_i 分别映射到物理服务器 a 和 b 上 ($f_{i_p} \mapsto a, f_i \mapsto b$)，若由 a 到 b 需要经过 hop 条物理链路，利用最短路径算法，将虚拟链路 $l_v^k(f_{i_p}, f_i)$ 映射到经过最小跳的物理链路 $p_{a \rightarrow n \rightarrow b}$ 上， n 为所经过的中间物理服务器。

2) 检查是否满足链路资源约束，保证每条物理链路的剩余带宽资源不小于虚拟链路所需的带宽资源，如式(16)所示。换句话说，链路的资源约束由子物理链路的最小剩余带宽决定，如式(17)所示。当满足链路资源约束时， $l_v^k(f_{i_p}, f_i)$ 成功映射到 $p_{a \rightarrow n \rightarrow b}$ 上，保存链路映射结果；否则转至 3)。

3) 若该链路与其他虚拟链路存在复用物理链路，查看是否可以释放该链路的物理资源以满足链路资源约束，如果可以满足， $l_v^k(f_{i_p}, f_i)$ 成功映射到 $p_{a \rightarrow n \rightarrow b}$ 上，保存链路映射结果；否则转至 4)。

4) 如果不存在链路复用或是释放复用链路资源仍旧无法满足约束，则利用算法求出次短路径，重复上述步骤进行判断。如果全部路径均无法满足链路约束条件，则当前虚拟链路映射失败。

$$\sum_{l_v^k \in L_v^k} (y_{l_v^k, k, p} r_{\text{bw}}(l_v^k)) \leq c_{\text{bw}}(p), p \in P \quad (16)$$

$$r_{\text{bw}}(f_{i_p}, f_i) \leq \min\{c_{\text{bw}}(a, n_1), \dots, c_{\text{bw}}(n_{\text{hop}-1}, b)\} \quad (17)$$

4 部署方法流程

基于性能感知的网络切片部署方法流程如算法 1 所示。

算法 1 基于性能感知的网络切片部署

输入 物理网络 G_s 与 K 个网络切片实例请求 G_v^k

输出 最佳映射方案

1) 设置种群规模 Scope 、最大迭代次数 maxg 、 c_1 和 c_2 ；

2) 为 NIND 个粒子初始化位置向量 \vec{x} 、速度向量 \vec{v} ；

3) 根据式(10)，计算初始种群中所有粒子的性能下降因子总和作为适应度值，并求出个体最优值 pb 、全局最优值 gb ；

4) 设置初始退火温度 $T = -\frac{\text{Fitness}(pg)}{\log(0.2)}$ ；

5) 当迭代次数在区间 $[1, \text{maxg}]$ 时，重复步骤 6) 和步骤 7)；

6) 当种群规模在区间 $[1, \text{Scope}]$ 时，利用 SA 算法对粒子的 pb 进行领域搜索，使用轮盘赌策略，计算 $\text{Fitness}(pj) - \text{Fitness}(pg)$ ，当

$$\min\left\{1, \exp\left(-\frac{(\text{Fitness}(pj) - \text{Fitness}(pg))}{t_j}\right)\right\} >$$

$\text{random}[0, 1]$ 时，用 pb 替换 gb ；

7) 当种群规模在区间 $[1, \text{Scope}]$ 时，利用式(15)及式(13)更新粒子的速度向量 \vec{v}_d 、位置向量 \vec{x}_d ，并更新个体最优值 pb 、全局最优值 gb ；

8) 存储 NSI 请求 G_v^k 最佳的节点映射结果；

9) 由 VNF 的顺序约束得到 NSI 的虚拟节点部署顺序 $\varphi_k = \{f_1, f_2, \dots, f_i, \dots, f_n\}$ ；

10) 对 $\forall i \in n$ 获取与 VNF 相连的上一级 VNF： $f_{i_1}, f_{i_2}, \dots, f_{i_q}$ ；

11) 对于经过 VNF 的 f_i 的所有的链路 $l_v^k(f_{i_p}, f_i), \forall p \in [1, q]$ ，利用最短路径算法进行相关链路映射；

12) 当满足 $r_{\text{bw}}(f_{i_p}, f_i) \leq \min\{c_{\text{bw}}(a, n_1), \dots, c_{\text{bw}}(n_{\text{hop}-1}, b)\}$ 时， $l_v^k(f_{i_p}, f_i)$ 成功映射到 $p_{a \rightarrow n \rightarrow b}$ 上，转入步骤 15)；否则，转入步骤 13)；

13) 若该链路与其他虚拟链路存在复用，且可以释放该链路的物理资源以满足链路约束，转入步骤 15)；

14) 若释放资源仍无法满足约束或是不存在链路复用，检查剩余的次短路径，重复步骤 12)~步骤 14)；

15) 存储相关虚拟链路的映射结果；

16) 输出最佳映射结果。

5 仿真分析

5.1 实验环境

本文使用 GT-ITM 工具生成物理网络及网络切片实例请求，运行在配置为 Intel(R) Pentium(R) 3.4 GHz CPU、4 GB 内存的 PC 机上，并使用 Matlab 对结果进行分析。将本文提出的方法 (Min-Dep) 与最小化部署成本 (Min-Cost)^[20] 及最小化虚拟网间的共存度 (Min-CoNum) 这 2 种方法从请求接受率、收益开销比、性能影响因子总和及性能影响程度四方面进行对比。网络参数设置如表 2 所示，仿

真参数设置如表 3 所示。

网络参数	取值
物理节点数量	50
物理节点资源	[50,100]
物理链路资源	[50,100]
每个 NSI 请求 VNF 数量	[2,5]
VNF 节点资源	[5,20]
虚拟链路资源	[5,20]

网络参数	取值
最大迭代次数	60
种群规模	50
c_1	2
c_2	2
λ	0.6

5.2 仿真结果及性能分析

图 4 是不同部署方法的收益开销比的曲线。从图 4 中可以看出，3 种方法随着 NSI 请求数量增多，收益开销比均逐渐趋于稳定，最终稳定结果显示 Min-Cost 方法的收益开销比最大，而且本文方法 (Min-Dep) 的收益开销比稍高于 Min-CoNum 方法。这主要是因为 Min-Cost 方法统筹考虑了节点和链路的成本消耗，使底层物理网络的碎片资源数量相对较少，提高了收益开销比。而其他 2 种方法部署的目标函数都是仅从虚拟节点考虑，在虚拟链路的映射上在满足资源约束的条件下，寻找最小跳的路径，这样增多了节点上的未被使用的碎片资源，资源利用率降低，导致收益开销比较低。

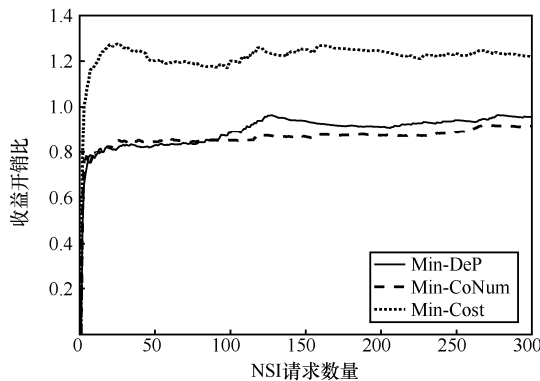


图 4 收益开销比

图 5 反映了 3 种方法随着 NSI 请求数量增多请

求接受率的变化情况。从图 5 中可以看出，3 种方法的请求接受率均出现了不同程度的下降，这是因为部署到物理网络上的 NSI 请求逐渐增多，剩余物理资源逐渐减少，导致部分新到达的 NSI 请求无法成功映射到物理网络上。相较于 Min-CoNum 方法，Min-Dep 方法和 Min-Cost 方法的请求接受率下降更为明显，且在部署了大概 250 个 NSI 请求后，Min-Dep 方法的请求接受率稍高于 Min-Cost 方法。原因是 Min-CoNum 方法在映射时因受节点共存数量的限制，那些部署较少 VNF 的服务器会作为优先选择对象，在一定程度上避免了因节点资源不足造成的映射失败的情况，提高了请求接受率。而本文方法统筹 3 种资源的供需比，Min-CoNum 方法期望部署成本最小化，均未考虑底层网络资源的均衡性，故使请求接受率下降较多。

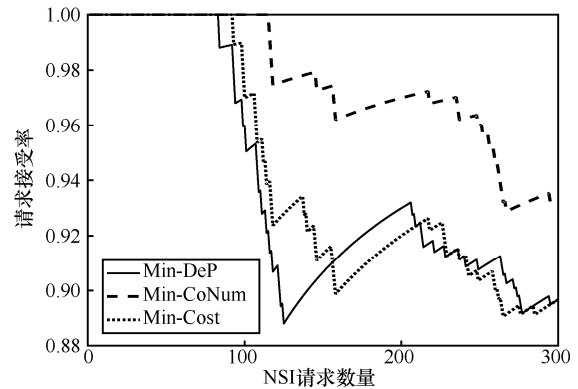


图 5 网络请求接受率

图 6 反映了 3 种方法的性能影响因子总和的对比情况。从图 6 中可以看出，Min-Dep 方法相较于其他 2 种方法，性能影响因子总和维持在一个相对较低的水平。主要是由于 Min-Cost 方法在部署时将最小化部署成本作为目标函数，并未考虑共享物理资源的 VNF 间的性能影响，导致性能影响因子总和处于一个相对较高的水平。而 Min-CoNum 方法在部署时仅是从物理服务器上共存 VNF 数量考虑，并未考虑它们之间的资源供需情况，因此性能影响因子总和水平较高。

为了进一步证明 Min-Dep 方法的有效性，除了考虑性能影响因子总和的对比情况，本文还对性能影响程度进行了对比。运行在 VM 中的 VNF 的标准性能^[14,20]定义为在理想情况下运行时的性能，即物理服务器的其他 VNF 全部处于闲置状态。在实际运行时，VNF 受到的性能影响程度可以表示为它在实际情况下的性能与它在理想环境下的性能之

比。在物理服务器 $x \in N_s$ 上存在 n 个 VNF，第 i 个 VNF 在理想情况下运行时的性能设为 $\text{ideal_}\eta_x^i$ ，在实际情况下运行时的性能设为 η_x^i ，则第 i 个 VNF 性能影响程度 ω_x^i 可以表示为

$$\omega_x^i = \frac{\eta_x^i}{\text{ideal_}\eta_x^i} \quad (18)$$

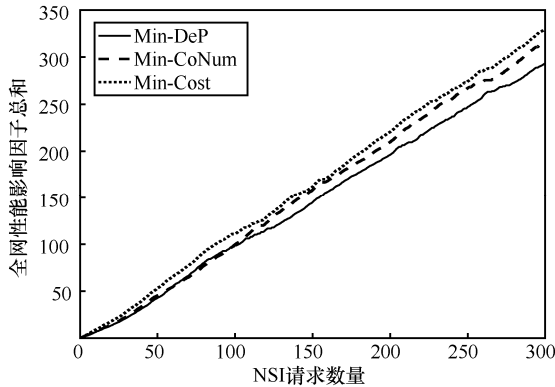


图 6 性能影响因子总和

图 7 对比了这 3 种方法的性能影响程度的变化情况。在请求初始到达阶段，因物理网络剩余资源丰富，3 种方法的性能影响程度相差不大；随着请求数量的增多，底层物理资源剩余量的减少，3 种方法在性能影响程度上均呈现增长趋势，但是 Min-DeP 方法性能影响程度的增长速率最慢，而且始终低于其他 2 种方法，这主要是由于 Min-DeP 方法在部署时优先将 VNF 部署在性能影响因子较小的物理服务器上，相较于其他方法，Min-DeP 方法可以在部署阶段有效降低其他 VNF 对自身性能的影响。

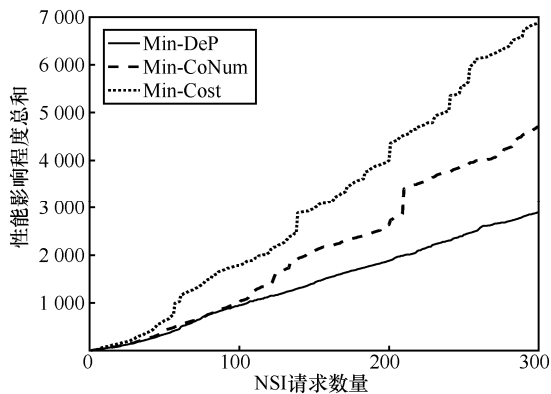


图 7 性能影响程度总和比较

6 结束语

网络切片是 5G 十分有发展潜力的关键技术，

旨在通过建立端到端的解决方案定制化满足不同用户的需求。由于网络切片是基于虚拟化技术实现的，在同一物理基础设施上部署的网络切片实例间仍存在因资源过度竞争带来的性能下降问题，故本文提出了一种基于性能感知的网络切片部署方法，并验证了该方法的有效性。在虚拟节点部署阶段，首先从多维度的资源供需的角度量化了 VNF 的性能影响大小，然后将全网性能影响程度总和最小作为目标函数，并采用群智能优化算法进行迭代寻优，得到最终的节点部署结果；在虚拟链路部署阶段，采用了最短路径方法进行虚拟链路映射。最终证明了该方法可以在一定程度上降低其他网络的虚拟网络功能对自身网络的性能影响，但是在网络请求接受率和收益开销比方面优势不太明显，在今后的研究中将从这 2 个方面进行改进。

参考文献：

- [1] ORDONEZ-LUCENA J, AMEIGEIRAS P, LOPEZ D, et al. Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges[J]. IEEE Communications Magazine, 2017, 55(5):80-87.
- [2] NGMN Alliance. Description of network slicing concept[S]. 2017.
- [3] 刘彩霞, 卢干强, 汤红波, 等. 一种基于 Viterbi 算法的虚拟网络功能自适应部署方法[J]. 电子与信息学报, 2016, 38(11):2922-2930.
- [4] LIU C X, LU G Q, TANG H B, et al. Adaptive deployment method for virtualized network function based on viterbi algorithm[J]. Journal of Electronics & Information Technology, 2016, 38(11):2922-2930.
- [5] 汤红波, 袁泉, 卢干强, 等. 一种支持节点分割的 vEPC 虚拟网络功能部署模型[J]. 电子与信息学报, 2017, 39(3):546-553.
- [6] TANG H B, YU Q, LU G Q, et al. A model for virtualized network function deployment based on node-splitting in vEPC[J]. Journal of Electronics & Information Technology, 2017, 39(3):546-553.
- [7] 刘彩霞, 李凌书, 汤红波, 等. 基于子图同构的 vEPC 虚拟网络分层协同映射算法[J]. 电子与信息学报, 2017, 39(5):1170-1177.
- [8] LIU C X, LI L S, TANG H B, et al. Hierarchical coordination strategy for vEPC virtual network embedding based on subgraph isomorphism[J]. Journal of Electronics & Information Technology, 2017, 39(5):1170-1177.
- [9] SHIEH A, KANDULA S, GREENBERG A, et al. Seawall: performance isolation for cloud datacenter networks[C]//USENIX Conference on Hot Topics in Cloud Computing. USENIX Association, 2010:1.
- [10] 王卅, 张文博, 吴恒, 等. 一种基于硬件计数器的虚拟机性能干扰估算方法[J]. 软件学报, 2015, 26(8):2074-2090.
- [11] WANG S, ZHANG W B, WU H, et al. Approach of quantifying virtual machine performance interference based on hardware performance counter[J]. Journal of Software, 2015, 26(8): 2074-2090.
- [12] ZHU Q, ZHU J, AGRAWAL G. Power-aware consolidation of scientific workflows in virtualized environments[C]//High PERFORMANCE Computing, Networking, Storage and Analysis. 2010:1-12.
- [13] JIN X, ZHANG F, WANG L, et al. Joint optimization of operational

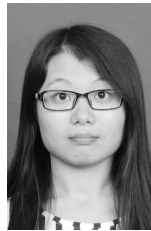
cost and performance interference in cloud data centers[J]. IEEE Transactions on Cloud Computing, 2017, 5(4):697-711.

- [10] XU F, LIU F, LIU L, et al. iAware: making live migration of virtual machines interference-aware in the cloud[J]. IEEE Transactions on Computers, 2014, 63(12):3012-3025.
- [11] HUI Z, JING W, FENG L, et al. Power-aware and performance-guaranteed virtual machine placement in the cloud[J]. IEEE Transactions on Parallel & Distributed Systems, 2018, 29(6): 1385-1400.
- [12] JI S, MING D L, JI N, et al. An online virtual machine placement algorithm in an over-committed cloud[C]// IEEE International Conference on Cloud Engineering (IC2E). 2018:106-112.
- [13] TOMÁS L, LAKEW E B, ELMROTH E. Service level and performance aware dynamic resource allocation in overbooked data centers[C]// International Symposium on Cluster, Cloud and Grid Computing, 2016:42-51.
- [14] 李丹, 兰巨龙, 王鹏, 等. 基于最优加权图匹配的服务功能链部署方法[J]. 通信学报, 2019,40(3):10-18.
LI D, LAN J L, WANG P, et al. Service function chain deployment algorithm based on optimal weighted graph matching[J]. Journal on Communications, 2019,40(3):10-18.
- [15] 兰巨龙, 金子晋, 孙鹏浩, 等. 基于可靠性的服务功能链构建算法[J]. 通信学报, 2019, 40(1):68-74.
LAN J L, JIN Z J, SUN P H, et al. Service function chain construct algorithm based on reliability[J]. Journal on Communications, 2019, 40(1): 68-74.
- [16] 袁泉, 汤红波, 黄开枝, 等. 基于 Q-learning 算法的 vEPC 虚拟网络功能部署方法[J]. 通信学报, 2017,38(8):172-182.
YUAN Q, TANG H B, HUANG K Z, et al. Deployment method for vEPC virtualized network function via q-learning[J]. Journal on Communications, 2017,38(8):172-182.
- [17] 王健, 赵国生, 李志新. 面向 SDN 的虚拟网络映射算法研究[J]. 通信学报, 2017, 38(10):26-35.
WANG J, ZHANG G S, LI Z X. Research on mapping algorithm of virtual network oriented to SDN[J]. Journal on Communications, 2017, 38(10):26-35.
- [18] PAN Q K, WANG L, TASGETIREN M F, et al. A hybrid discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem with makespan criterion[J]. International Journal of Advanced Manufacturing Technology, 2008, 38(34):337-347.
- [19] JOHNSON D B. A note on dijkstra's shortest path algorithm[J]. Journal of the ACM, 1973, 20(3):385-388.
- [20] GONG L, WEN Y, ZHU Z, et al. Towards profit-seeking virtual network embedding algorithm via global resource capacity[C]// IEEE Conference on Computer Communications. 2014:1-9.

[作者简介]



黄开枝 (1973-), 女, 安徽滁州人, 国家数字交换系统工程技术研究中心博士生导师, 主要研究方向为无线移动通信和无线物理层安全。



潘启润 (1993-), 女, 河北石家庄人, 国家数字交换系统工程技术研究中心硕士生, 主要研究方向为新一代移动通信技术、网络切片安全。



袁泉 (1991-), 男, 山东青岛人, 国家数字交换系统工程技术研究中心博士生, 主要研究方向为移动通信网络和网络功能虚拟化。



游伟 (1984-), 男, 江西丰城人, 国家数字交换系统工程技术研究中心讲师, 主要研究方向为密码学和 5G 网络安全。



汤红波 (1968-), 男, 湖北孝感人, 国家数字交换系统工程技术研究中心博士生导师, 主要研究方向为移动通信网络和新型网络体系结构。