

基于成本的服务链组成与部署联合优化策略

刘雅丽, 史久根

(合肥工业大学 计算机与信息学院, 合肥 230009)

摘要: 网络功能虚拟化(NFV)通过将虚拟网络功能(VNF)部署在虚拟设备中,提高了网络管理的灵活性,但随着服务需求的扩大,网络供应商消耗的运营支出(OPEX)也不断增加。由于 VNF 改变流大小的特性、VNF 间的依赖性以及组成和部署方式的复杂性,面向 OPEX 的 VNF 组成和部署问题充满挑战。提出一种面向成本的虚拟网络链组成和部署联合优化策略,将节点映射成本、链路映射成本、激活成本和能耗成本公式化为 OPEX,构建混合整数非线性规划模型。为分析影响成本的不同因素,同时提高特殊依赖情况下的处理效率,根据不同依赖关系将 VNF 请求集分为完全无序、部分有序和完全有序 VNF 集合进行分析,并设计 3 种相应优化算法。实验结果表明,在完全无序、部分有序情况下,算法性能优于首次适应算法、随机拟合算法等同类算法,对于完全有序算法,当节点资源配比在 50% 以上时,可获得小规模网络下近似线性规划模型精确解的方案。

关键词: 网络功能虚拟化;虚拟网络功能链组成;虚拟网络功能链放置;运营支出;服务功能链

开放科学(资源服务)标志码(OSID):



中文引用格式:刘雅丽,史久根.基于成本的服务链组成与部署联合优化策略[J].计算机工程,2022,48(4):106-112.

英文引用格式:LIU Y L, SHI J G. Joint optimization strategy for cost-based service chain composition and deployment[J].

Computer Engineering, 2022, 48(4): 106-112.

Joint Optimization Strategy for Cost-Based Service Chain Composition and Deployment

LIU Yali, SHI Jiugen

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

[Abstract] Network Function Virtualization (NFV) improves network management flexibility by deploying Virtual Network Functions (VNF) in virtual devices. With the continuous expansion of service demand, the Operational Expenditure (OPEX) consumed by network providers is increasing. The efficient chaining and placing of VNF is challenging based on their traffic-changing effects and dependency relationships, as well as the diversity of available server choices. This paper proposes a cost-oriented joint optimization strategy for virtual network chain composition and deployment, formulates node mapping cost, link mapping cost, activation cost, and energy consumption cost as OPEXs, and establishes a mixed-integer nonlinear programming model. To analyze the different factors that affect the cost and improve the processing efficiency of special dependence, the VNF request set is divided into non-ordered, partially ordered, and totally ordered VNF according to different dependencies and three corresponding optimization algorithms are designed. Experimental results demonstrate that the performance of the proposed algorithm is better than that of first time adaptive algorithm and random fitting algorithms in the cases of complete non-order and partial order. When the node resource ratio is greater than 50%, the totally ordered VNF algorithm can obtain a solution that approximates the exact solution of the linear programming model in a small-scale network.

[Key words] Network Function Virtualization (NFV); Virtual Network Functions (VNF) chain composition; VNF chain placement; Operational Expenditure (OPEX); Service Function Chaining (SFC)

DOI: 10.19678/j.issn.1000-3428.0061217

0 概述

网络功能虚拟化(Network Function Virtualization,

NFV)技术将网络功能的实现从昂贵的硬件设备中转换到服务器上的虚拟设备中,旨在克服网络运营商(Internet Service Provider, ISP)日益增加的运营支

基金项目:国家重大科学仪器设备开发专项(2013YQ030595)。

作者简介:刘雅丽(1996—),女,硕士研究生,主研方向为网络功能虚拟化;史久根,副教授。

收稿日期:2021-03-22 修回日期:2021-05-07 E-mail:2018170913@mail.hfut.edu.cn

出^[1](Operational Expenditure, OPEX)问题。NFV将不同的网络功能(如防火墙、深度包检测、加密、解密等)与专用网络服务器分离,通过编排不同的虚拟网络功能(Virtual Network Functions, VNF)并将其映射到物理网络设施中以实现网络服务^[2],从而有效地提高部署和管理的灵活性^[3]。但由于物理位置的选择复杂多样、VNF改变流量的特性以及VNF之间存在依赖关系^[4],因此充满了挑战性。

编排VNF并将其合理部署在物理网络中是NFV的关键技术之一^[5],现有大量研究致力于VNF网络功能的实现,其中多数仅考虑了部署问题^[6-7],例如,文献[8]制定了VNF中间件放置问题,其目的是最大程度地减少提供特定服务的实例总数,并提出具有恒定逼近率的渐近最优贪婪算法。文献[9]提出一种渐近最优的在线算法用于服务链的准入控制和中间件的增量部署。除集中讨论部署阶段的研究外,也有部分学者考虑了组成与部署联合优化问题,文献[10]探讨VNF资源优化的多阶段问题,研究服务功能链的协调分配和流量调度问题,提出一种基于CoordVNF的启发式方法来协调组成和部署两个阶段。另外,文献[11]考虑并探讨了具体流量变化的影响,文献[12]则进一步分析了VNF改变流的特性,并针对不同依赖关系状态提出了对应的放置算法。文献[13]证明了VNF改变流量的特性对成本的影响。但这些文献都未考虑VNF变化的相

对资源需求对成本的影响,没有综合分析影响运营成本的具体因素。

为联合VNF组成与VNF部署两阶段共同优化,分析影响OPEX的具体因素,本文综合考虑VNF改变流量的特性、相对资源消耗、依赖关系以及部署方式的选择,提出一种面向成本的虚拟网络链组成和部署联合优化策略。将节点映射成本、链路映射成本、激活成本和能耗成本公式化为OPEX,建立混合整数非线性规划(Mixed-Integer Nonlinear Programming, MINLP)成本模型。根据依赖关系将请求集分成完全无序、部分有序和完全有序的VNF,并设计3种对应的优化算法,分析影响成本的不同因素。

1 虚拟网络功能部署问题

虚拟网络功能(NFV)通过组成服务功能链(Service Function Chaining, SFC)并将其部署到物理网络中的方式来实现服务,完成了网络功能与物理设备的脱钩^[14]。不同SFC组成与部署方式对成本的影响如图1所示(彩图效果见《计算机工程》官网HTML版)。每个物理节点对应一个物理设备(Physical Machine, PM),如图1(a)所示,当网络功能部署到未激活的PM上时,对应PM需要分配一定的资源,如CPU、内存等,用以实现对应类型的虚拟设备(Virtual Machine, VM),以实例化VNF^[15]。

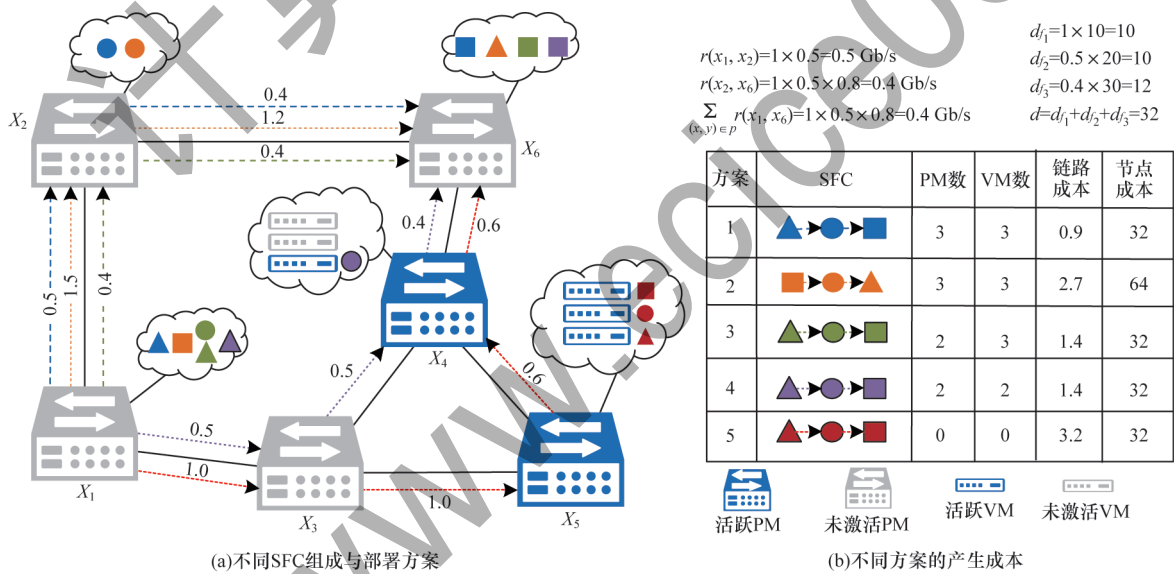


图1 不同SFC组成与部署方式对成本的影响

1.1 虚拟网络功能

网络功能是执行特定功能的网络实体,本文用 f_i 表示VNF i , $F = \{f_i | i = 1, 2, \dots, n\}$ 表示网络功能集合。VNF多数是流处理设备,常以不同方式修改网络流^[4]。例如Citrix Cloud Bridge广域网优化器,通过压缩流量将流量减少了80%^[16]。将VNF的流量缩放比定义为 $\lambda_f = r_{out}/r_{in}$, r_{in} 和 r_{out} 分别为VNF的输入、输出

流。运行VNF需要一定资源(如CPU、内存等),所需资源量 d_f 通常与该VNF的相对数据速率 d_f^{rel} 和通过的流相关^[17], $d_f = r_{in} \cdot d_f^{rel}$ 为处理一个VNF所需数据资源量。

1.2 服务功能链

服务功能链是服务流所需要遍历的一系列网络功能。假设共有 K 个服务请求, SFC请求集为 $S =$

$\{s_k | k=1, 2, \dots, K\}$, 每个 SFC 由 n 种类型的 VNF 组成, 即 $s_k = \{f_1^k, f_2^k, \dots, f_n^k | f_i^k \in F\}$ 。VNF 之间可能存在依赖关系: IPSec 解密器通常放置在 NAT 网关之前^[18]。本文用 $f_i^{\text{de}} \prec f_i$ 表示 f_i 依赖于 f_i^{de} 。为分析影响成本的不同因素, 同时提高特殊依赖情况下的处理效率, 本文将请求集分为完全无序、部分有序和完全有序的不同 VNF 集合进行讨论。完全无序的请求集为 $\{f_1, f_2\}$, 部分有序的请求集为 $\{f_3, f_4 \prec f_5\}$, 完全有序的请求集为 $\{f_6 \prec f_7 \prec f_8\}$ 。

1.3 问题描述

面向成本的 VNF 链组成与部署问题是通过编排合适的 SFC 并合理部署, 从而最小化网络产生的 OPEX。问题示例见图 1, 请求集 $\{f_1, f_2, f_3\}$ 完全无序, VNF 的缩放比分别为 0.5、0.8、1.5, 相对数据速率为 10、20、30, 请求初始流为 1 Gb/s, 源点为 x_1 , 终点为 x_6 。各方案对应成本如图 1(b) 所示, 通过比较可知: 不同 VNF 的缩放比、相对数据速率、SFC 的组成和路由选择都会对 OPEX 造成影响, 本文通过考虑各个因素, 综合建模以优化 OPEX。

2 混合整数非线性模型

将物理网络拓扑抽象表示为无向图 $G(E, V)$, E 和 V 分别表示物理链路和物理节点的集合。 G 中每个节点可以部署一定数量的虚拟机, 每个虚拟机只能实例化一种类型的 VNF。其中: $z_x \in \{0, 1\}$ 表示节点 $x \in V$ 是否被激活; $\alpha_{f_i, x}^k \in \{0, 1\}$ 表示请求 $k \in K$ 的 f_i 是否被映射到了物理节点 x 上; $\beta_{(x, y)}^{(f_i, f_j), k} \in \{0, 1\}$ 表示虚拟链路 (f_i, f_j) 是否被映射到了物理链路 (x, y) 。OPEX 中各成本描述如下:

1) 节点映射成本。部署一个 VNF 实例需要消耗一定成本, 不同类型的 VNF 有不同的资源消耗需求, 本文将这种资源需求称为节点映射成本, 将服务链 s_k 中 f_i 映射到节点所需的数据资源为 $d_{f_i}^k$ 。部署 K 个服务功能链的总节点成本如下^[19]:

$$C_{\text{node}} = \sum_k \sum_{i=1}^n d_{f_i}^k \quad (1)$$

$$d_{f_i}^k = r_{(f_{i-1}, f_i)}^k d_{f_i}^{\text{rel}} = r_0^k \prod_j \lambda_{f_j} d_{f_i}^{\text{rel}}, \forall j < i \quad (2)$$

其中: $r_{(f_{i-1}, f_i)}^k$ 为请求 $k \in K$ 的虚拟链路 (f_{i-1}, f_i) 上的流量。假设服务链的源节点为 f_0 , r_0^k 表示虚拟链路 (f_0, f_1) 上的流量, 为请求的初始流量。通过分析影响节点映射成本的具体因素, 将 $d_{f_i}^k$ 取对数 $w(d_{f_i}^k) = \ln(d_{f_i}^k)$ 可得到:

$$w(d_{f_i}^k) = \ln(r_0^k) + \ln(d_{f_i}^{\text{rel}}) + \sum_{j < i} \ln(\lambda_{f_j}) \quad (3)$$

$$\sum_i w(d_{f_i}^k) = n \ln(r_0^k) + \sum_i \ln(d_{f_i}^{\text{rel}}) + \sum_i (n-i) \ln(\lambda_{f_i}) \quad (4)$$

由式(4)可知: SFC 的节点成本变化与流和 VNF 无关, 仅与 VNF 间的相对顺序 i 有关。当 $\lambda_{f_i} > 1$ 时, VNF 增大流, 对应 $\ln(\lambda_{f_i}) > 0$; 当 $\lambda_{f_i} < 1$ 时, VNF 减小

流, 对应 $\ln(\lambda_{f_i}) < 0$; 当 $\lambda_{f_i} = 1$ 时, VNF 不改变流, 对应 $\ln(\lambda_{f_i}) = 0$ 。所以, $\lambda_{f_i} > 1$ 的 VNF 放置顺序越大, $\lambda_{f_i} < 1$ 的 VNF 对应放置顺序越小, 节点成本也越小。

2) 链路映射成本。将虚拟链路映射到物理链路中需要消耗一定带宽资源, 不同的链路组成方式、映射方式会导致不同的链路成本。部署 K 条服务功能链的总链路成本可以表示如下:

$$C_{\text{link}} = \sum_k \sum_{(x, y) \in p_k} r_{(x, y)}^k \quad (5)$$

$$r_{(x, y)}^k = r_0^k \prod_{f_i} \lambda_{f_i}, \forall x', (x, y) \in p_k, x' \prec y, \alpha_{f_i, x'}^k = 1 \quad (6)$$

其中: $r_{(x, y)}^k$ 为物理链路 (x, y) 消耗的带宽资源量, 表示请求 s_k 部署在 (x, y) 上产生的链路成本。同样地, 为分析影响链路成本的具体因素, 将 $r_{(x, y)}^k$ 取对数 $w(r_{(x, y)}^k) = \ln(r_{(x, y)}^k)$, 可得^[13]:

$$w(r_{(x, y)}^k) = \ln(r_0^k) + \sum_{\alpha_{f_i, x'}^k = 1} \ln(\lambda_{f_i}), \forall x' \prec y \quad (7)$$

$$\sum_{(x, y) \in p_k} w(r_{(x, y)}^k) = (|p_k| \ln(r_0^k) + \sum_{\alpha_{f_i, x}^k = 1} (|p_k| - h_{xk}) \ln(\lambda_{f_i})) \quad (8)$$

从式(8)可以看出: SFC 的链路成本变化仅与已部署 VNF 之间的物理路径跳数 h_{xk} 有关。综合式(4)、式(8)分析可知, SFC 顺序决定了节点映射成本, SFC 部署方式决定了链路映射成本。

3) 激活成本。在部署 VNF 之前, 每个物理节点都需要消耗一定的成本来完成预配置等前提准备工作。激活成本由所激活节点的数量决定, 用 c_x 表示各节点的激活代价。总激活成本如下:

$$C_{\text{prov}} = \sum_x z_x c_x \quad (9)$$

4) 能耗成本。PM、VM 一旦被激活, 就会消耗一定的资源用于维持自身运转^[20], 而没有被激活的休眠 PM、休眠 VM 基本不会产生能耗, 一般不考虑。活跃 PM 与活跃 VM 产生的总能耗成本如下:

$$C_{\text{energy}} = \sum_x (e^{\text{PM}} z_x + \sum_i e^{\text{VM}} \eta_{f_i}^x) \quad (10)$$

综上, 本文优化目标为最小化网络服务实现的总运营成本 OPEX, 表示为:

$$C_{\text{OPEX}} = \mu_1 C_{\text{opex}} + \mu_2 C_{\text{band}} + \mu_3 C_{\text{prov}} + \mu_4 C_{\text{energy}}$$

OPEX 受到以下约束: 式(11)保证了每个请求的 SFC 都必须被部署到网络中, 并且每个 VNF 只需要一个结点的虚拟机来实例化。

$$\sum_x \alpha_{f_i, x}^k = 1, \forall k \in K, f_i \in F \quad (11)$$

VNF 之间的依赖关系如式(12)所示:

$$i < j, \forall f_i \prec f_j \quad (12)$$

每条物理链路上的流量之和不能超过其带宽容量 $B_{(x, y)}$, 如式(13)所示:

$$\sum_k \sum_i r_{(x, y)}^k \beta_{(f_{i-1}, f_i), k} \leq B_{(x, y)} \quad (13)$$

式(14)、式(15)确保每个节点上所有 VNF 的总资源消耗不应超过其资源容量, 即要满足各类型虚拟机的容量 $C_{\text{vm}}^{f_i}$ 和节点可供承载虚拟机的总资源量 C_x 的限制:

$$\sum_k d_{f_i}^k \alpha_{f_i, x}^k / C_{vm}^{f_i} \leq \eta_x^{f_i}, \forall k \in K, f_i \in F \quad (14)$$

$$\sum_i C_{vm}^{f_i} \leq z_x C_x, \forall k \in K, f_i \in F \quad (15)$$

每个请求流不超过时延上限 D_k , 如式(16)所示:

$$|p_k| \leq D_k, \forall k \in K \quad (16)$$

VNF 的联合链接和部署如式(17)所示:

$$\alpha_{f_{i-1}, x}^k + \alpha_{f_i, y}^k \leq \beta_{(x, y)}^{(f_{i-1}, f_i), k} + 1 \quad (17)$$

3 算法设计

3.1 完全无序的 VNF 算法设计

由于3种情况均需调用完全有序的VNF放置算法(TOCP)的部署方式, 本节主要描述组成方式的影响与算法设计。由上文可知, 将缩小流的VNF尽可能靠近源点放置, 放大流的VNF尽可能靠近终点放置, 可以将链路成本最小化。考虑到相对数据速率影响, 对部署在同一节点上的VNF再次调整, 可以在不改变链路成本的同时降低节点成本。同一节点上不同VNF部署顺序影响如图2所示, 调整节点 x 上的 $\{f_1 \rightarrow f_2 \rightarrow f_3\} \Rightarrow \{f_3 \rightarrow f_2 \rightarrow f_1\}$, 输出流量相同, 节点成本从80.8减少到了76.2。

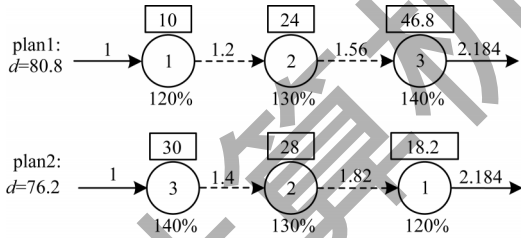


图2 同一节点上不同VNF部署顺序影响

Fig.2 Impact of different VNF deployment orders on same node

完全无序VNF的组成与放置算法(NOCP)设计如下:

1) 将 s_k 的所有 f_i 根据 λ_{f_i} 升序排序;

2) 将 $\lambda_f > 1$ 的VNF从流路径的尾部按照 λ_f 降序预部署, 将 $\lambda_f \leq 1$ 的VNF从流路径的首部按照 λ_f 升序预部署;

3) 对部署在同一节点上的VNF通过全排列算法组合, 并搜索具有最小 $\sum_x d_x$ 的SFC;

4) 通过完全有序VNF的最短路径放置算法(TOSP)确定最终方案。

算法时间复杂度与TOSP相同, 由于篇幅限制, 该部分伪代码不再赘述。

3.2 部分有序的VNF算法设计

部分有序的VNF需要考虑对应的依赖约束, 本节主要描述部分有序VNF的组成与放置算法(POCP)中独有的依赖关系处理算法——部分有序VNF的SFC组合算法(PO-C)。根据NOCP设计原则可获得无依赖约束下的最优预部署方案, 仅需对违背依赖约束的VNF进行调整。算法按照预部署的SFC顺序, 依次将违背了依赖关系的 f_i 紧随 f_i^{dc} 之后放置并组成一个新的 f_0 , f_0 继承了 f_i 和 f_i^{dc} 的依赖约

束, $\lambda_{f_0} = \lambda_{f_i^{dc}} \times \lambda_{f_i}$, $d_{f_0} = d_{f_i^{dc}} + \lambda_{f_i^{dc}} d_{f_i}$ 。将 f_0 按照 λ_{f_0} 大小插入原SFC中即可获得新的SFC队列, 调整过程如图3所示, 首先将所有VNF按照 λ 排序, 然后依次将违背了依赖关系的 f_i 放到 f_i^{dc} 后组成新的 f_0 , 再插入原SFC, 最终得到符合依赖约束的SFC组成的预部署结果。

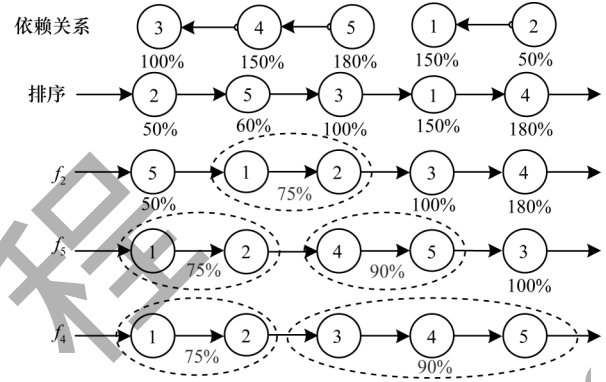


图3 依赖关系调整过程

Fig.3 Adjustment process of dependency relation

PO-C算法如算法1所示, 将违规重组的 f_0 插入有序SFC需要 $O(\log_a n)$, 重组后的 $|s_k|$ 对应减少, 因此将原SFC调整为符合依赖约束的SFC, 需要: $O(\log_a n) + O(\log_a(n-1)) + \dots + O(\log_a 1) < O(n \log_a n)$ 。因此, PO-C算法低于自相关树转换算法^[4]前瞻 k 值时的时间复杂度, 且PO-C算法不受参数 k 约束。POCP其余部分与NOCP类似, 总时间复杂度与TOCP算法的时间复杂度相同。

算法1 部分有序VNF的SFC组合算法(PO-C)

输入 请求 s_k
输出 符合约束的有序SFC队列 s_k
1. 将 s_k 的所有 f_i 根据 λ_{f_i} 升序排序, $s_k' \leftarrow s_k$
2. for each f_i do
3. if $f_i^{dc} \neq \text{null}$ then
4. $f_i \leftarrow \text{getNF}_0(s_k', f_i)$, $f_i^{dc} \leftarrow \text{getNF}_0(s_k', f_i^{dc})$
5. if $\text{Index}(s_k', f_i) > \text{Index}(s_k', f_i^{dc})$ then
6. $f_0 = \text{newNF}_0(f_i^{dc}, f_i)$, $\lambda_{f_0} = \lambda_{f_i^{dc}} \times \lambda_{f_i}$
7. $\text{listDelete}(s_k', f_i)$, $\text{listDelete}(s_k', f_i^{dc})$
8. $\text{listInsertByRatio}(s_k', f_0)$
9. $\text{Index}(s_k', f_i) = \text{Index}(s_k', f_0)$
10. $\text{Index}(s_k', f_i^{dc}) = \text{Index}(s_k', f_0)$
11. end if
12. end if
13. end for
14. return $s_k \leftarrow s_k'$

3.3 完全有序的VNF算法设计

完全有序VNF组成的SFC顺序固定, 链路成本变化取决于物理链路长度, 而能耗成本和激活成本则取决于激活的PM、VM数。为优化OPEX, 应尽可能地将VNF放置在相同或相近的物理节点上。为此, TOCP从最短路径开始, 按序依次部署VNF。优先考虑复用已激活的VM、PM数, 若其剩余资源不足, 再尝试开辟新的VM、PM数, 以免浪费。得到一组请求集的部署方案后, 再尝试在该方案中关闭活

跃度较低的节点,并迁移VM以进一步节约成本。

分析 TOCP 算法可知:寻路时使用 DFS 回溯算法,时间复杂度为 $O(|V| + |E|)$;预部署时间复杂度为 $O(n|V_{PM}|)$;迭代地寻找并关闭负载低的 PM,时间复杂度为 $O(|S_k||P_k|)$ 。因此,TOCP 总时间复杂度为 $O(n|S_k||P_k||V_{PM}|)$ 。

算法2 完全有序 VNF 的放置算法(TOCP)

输入 $G(V, E)S_k$

输出 $VM_x, finalSFC_k \langle i \in N^*, f_i \in F, x \in V \rangle, R_k$

```

1.  $VM_x \leftarrow \emptyset, finalSFC_k \leftarrow \emptyset$ 
2. while  $s_k \in S$  do
3. 获取从  $x_1^k$  到  $x_n^k$  时延  $l_k$  内的路径集合  $P_k$ , 并排序
4. while  $p_k = \{x_1^k, x_2^k, \dots, x_n^k | x_j^k \in p_k\} \in P_k$  do
5.  $i=1, j=1, preSFC_k \leftarrow \emptyset$ 
6. while  $i \leq n$  and  $x_j \in p_d$ 
7. if  $close_{x_j} = \text{true}$  then
8.  $j++$ , continue;
9. end if
10. if  $f_i \in VM_x$  and  $\sum_k d_{f_i, x}^k > C_{vm}^{f_i} \eta_x^{f_i}$ 
11.  $preSFC_k \cup \{ \langle i, f_i, x \rangle \}$ 
12. else if  $|VM_x| < C_x$ 
13.  $VM_x \cup \{ f_i \}, preSFC_k \cup \{ \langle i, f_i, x \rangle \}, \eta_x^{f_i} = 1$ 
14.  $i++$ ;
15. end if
16. end while
17. if  $|preSFC_k| = |M_k|$  the
18.  $finalSFC_k \leftarrow preSFC_k, R_k \leftarrow Path_k$ ; break
19. end if
20. end while
21. end while
22. 寻找虚拟机数最少且出入度最小的活跃节点  $x_j$ 
23. while  $x_j \neq \text{null}$  do
24.  $close_{x_j} \leftarrow \text{true}$ ; //返回 step2
25. end while
26. return  $VM_x, finalSFC_k, R_k$ ;
```

4 仿真实验结果与分析

为保证模型可行性和算法的有效性,仿真实验

建立在真实的网络拓扑中,数据来自 SNDlib^[21]。算法使用 JUN 框架搭建,运行在 2 个 Intel Core I5-8300H 处理器和 128 GB 内存的机器上。对 NOCP、POCP 算法,实验选择在较大规模和流量的城域网 newyork ($|V|=16, |E|=49, |S|=56$) 中比较;对 TOCP 算法,在小型网络 pdh ($|V|=11, |E|=34, |S|=18$) 中仿真比较。由于校验和开销,用于无线通信的 BCH(48, 36) 编码器使流量增加 1.3 倍 ($48/36 \approx 1.3$), BCH(31, 11) 编码器使流量放大 2.8 倍 ($31/11 \approx 2.8$)^[22]。综合参考文献[12-13, 22],选取并设置了 6 种不同 VNF, 对应缩放比分别为 1.2、2.8、0.2、1.0、1.3 和 0.8;对应数据比分别为 64、32、2、8、4 和 16。请求长度则按照 1~6 随机产生,请求流从 0.01 Gb/s~2.50 Gb/s 分别进行实验。同时,实验使用对应 OPEX 的 4 种成本作为性能指标进行评估。节点成本为所有请求消耗的数据资源总和;链路成本为所有请求流在物理链路中的传输流量总和;激活成本为激活节点总数;能耗成本为活跃 PM 与活跃 VM 在运行过程中所消耗的能耗总和,其中一个 PM 和 VM 在活跃状态下的能耗分别为 80.5 W、165.9 W^[20]。

4.1 完全无序的 VNF

本文引入 3 种不同的算法进行比较:

1) 首次适应 (FF)^[4] 算法。从流路径的首部开始,按照缩放比升序连续放置排序后的 VNF。

2) 随机拟合 (RF)^[11] 算法。在满足约束条件的前提下,随机组成并放置 VNF。

3) LFGL 算法^[12]。将缩减流量的 VNF 从路径首部按照缩放比升序放置;其余 VNF 从流路径尾部开始按照缩放比降序放置。

为减少部署方式的干扰,3 种比较算法与 NO-shortest 均采用最短路径算法放置,各算法性能表现如图 4 所示,其中图 4(a)是各算法比 NOCP 多的节点映射成本,可以看到 NO-shortest 各项性能均优于 3 种比较算法。并且,在相同的部署方式下,不同的组成方式造成的成本差将随着请求流的增大而不断扩大。对比 NO-shortest,采用不同部署方式的 NOCP 仅牺牲了一点链路成本,能耗成本和激活成本上性能显著提高。

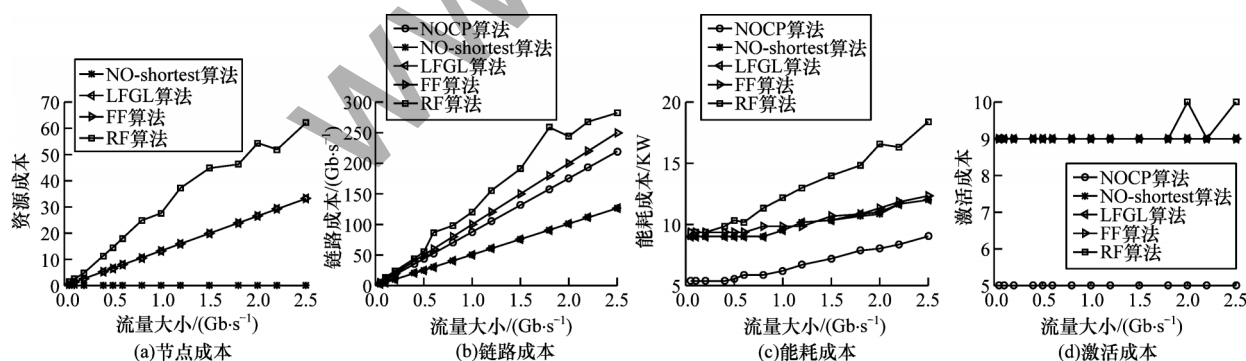


图4 完全无序 VNF 仿真结果

Fig.4 Simulation results of complete non-order VNF

4.2 部分有序的VNF

对于部分有序的VNF,实验假设VNF间存在依赖关系: $f_1 \rightarrow f_2, f_4 \rightarrow f_3$ 。LFGL、FF和RF对比方法的依赖性调整策略采用 $k=2$ 时的自相关树转换算法,各算法性能如图5所示,POCP算法可以在满足NF

之间的依赖约束的前提下有效降低总成本。图5整体趋势与图4相似,但各成本均有一定程度的增加与波动。因为相较于完全无序的VNF而言,部分有序的VNF在组成与部署的过程中约束更多,常常无法获得NOCP中的较优方案。

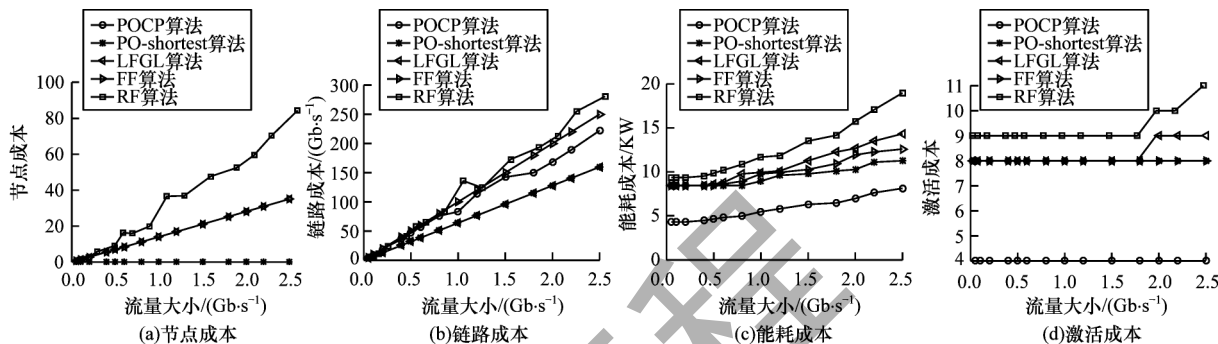


图5 部分有序VNF仿真结果

Fig.5 Simulation results of partial-order VNF

4.3 完全有序的VNF

由于TOCP在不同流量大小下的实验与NOCP、POCP结果类似,本文不做赘述。本节考虑资源受限下TOCP的性能变化。由于VNF完全有序时,式(2)、式(6)为线性关系,系统模型为混合整数线性规划(Mixed-Integer Linear Programming, MILP)模型。通过采用OPL语言建模,并使用CPLEX12.7求解得到了MILP模型结果。同时,引入采用最大限度激活VM思想的HEU^[23]算法作为对比实验进行比较。为减少不同初始SFC设

置的干扰,统一设置请求由长度为3的SFC组成: $f_1 \rightarrow f_2 \rightarrow f_3$,初始请求流率设置为1 Gb/s。节点资源配比 $p = C_x / C_{min}$ 的范围设置为(0%, 100%],其中 C_{min} 为资源不受限时请求所需的节点最小活跃VM数。各算法的性能表现如图6所示。从图6可以看出:TOCP算法的性能明显优于HEU、Shortest算法,接近MILP最优解;另外,当 $p > 50\%$ 时,TOCP算法与最优解的差距很小,但随着节点资源的逐渐不足,部署结果产生了波动,与最优解存在一定的差距。

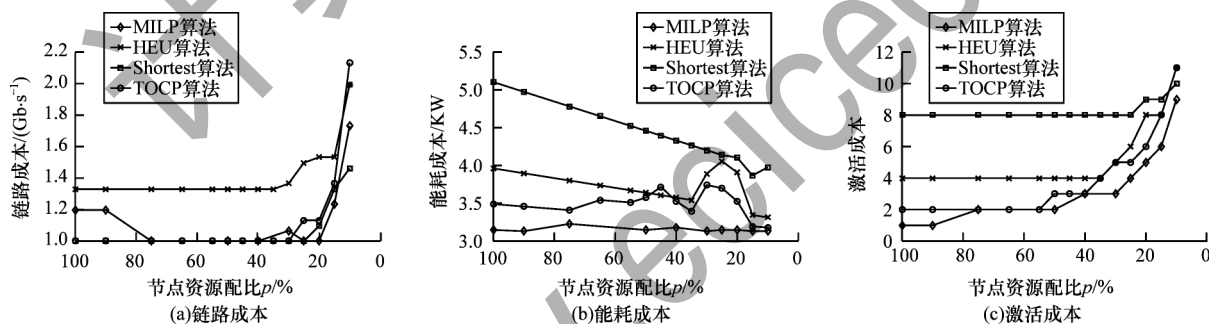


图6 完全有序VNF仿真结果

Fig.6 Simulation results of total-order VNF

5 结束语

网络功能虚拟化技术在实现网络灵活管理的同时降低了OPEX,对网络供应商有重要意义。本文考虑面向OPEX的VNF链组成与部署联合优化问题,研究不同依赖关系下的VNF组成与放置,分析流量缩放比、相对数据率、依赖关系以及部署方式对OPEX的影响。建立一种新的MINLP成本模型,并根据依赖关系的不同将请求集分成完全无序、部分有序和完全有序3种情况,设计相应的启发式算法。实验结果验证了本文算法的有效性。下一步考虑引入流量预测、网络可靠性评估、请求动态调度等算法,提高算法的可扩展性,以应对真实网络环境中的各种突发情况。

参考文献

- [1] GOLKARIFARD M, CHIASSERINI C F, MALANDRINO F, et al. Dynamic VNF placement, resource allocation and traffic routing in 5G[EB/OL]. [2021-02-20]. <https://arxiv.org/abs/2102.09426>.
- [2] BHAMARE D, SAMAKA M, ERBAD A, et al. Optimal virtual network function placement in multi-cloud service function chaining architecture[J]. Computer Communications, 2017, 102: 1-16.
- [3] SU R Y, ZHANG D Y, VENKATESAN R, et al. Resource allocation for network slicing in 5G telecommunication networks: a survey of principles and models[J]. IEEE Network, 2019, 33(6): 172-179.

- [4] MA W R, BELTRAN J, PAN Z L, et al. SDN-based traffic aware placement of NFV middleboxes [J]. IEEE Transactions on Network and Service Management, 2017, 14(3): 528-542.
- [5] YI B, WANG X, LI K, et al. A comprehensive survey of network function virtualization[J]. Computer Networks, 2018, 133: 212-262.
- [6] UMRAO B K, YADAV D K. Algorithms for functionalities of virtual network: a survey [J]. The Journal of Supercomputing, 2021, 77(7): 7368-7439.
- [7] SCHARONG F, NUNES I, SCHAEFFER-FILHO A. NFV resource allocation: a systematic review and taxonomy of VNF forwarding graph embedding [J]. Computer Networks, 2020, 185: 107-118.
- [8] SANG Y, JI B, GUPTA G R, et al. Provably efficient algorithms for joint placement and allocation of virtual network functions[C]//Proceedings of IEEE International Conference on Computer Communications. Atlanta, USA: IEEE Press, 2017: 1-9.
- [9] LUKOVSKI T, SCHMID S. Online admission control and embedding of service chains [C]//Proceedings of International Colloquium on Structural Information and Communication Complexity. Montserrat, Spain: Springer, 2015: 104-118.
- [10] BECK M T, BOTERO J F. Scalable and coordinated allocation of service function chains [J]. Computer Communications, 2017, 102: 78-88.
- [11] MA W, SANDOVAL O, BELTRAN J, et al. Traffic aware placement of interdependent NFV middleboxes [C]//Proceedings of IEEE Information Conference on Computer Communications. Atlanta, USA: IEEE Press, 2017: 1-9.
- [12] MA W R, BELTRAN J, PAN D, et al. Placing traffic-changing and partially-ordered NFV middleboxes via SDN [J]. IEEE Transactions on Network and Service Management, 2019, 16(4): 1303-1317.
- [13] CHEN Y, WU J. NFV middlebox placement with balanced set-up cost and bandwidth consumption[C]//Proceedings of the 47th International Conference on Parallel Processing. New York, USA: ACM Press, 2018: 1-10.
- [14] HALPERN J, PIGNATARO C. Service Function Chaining (SFC) architecture[EB/OL]. [2021-02-20]. <http://www.nic.funet.fi/rfc/rfc7665.txt>. •df.
- [15] MOUSTAFA N, MASHALY, ASHOUR M. Modeling and simulation of energy-efficient cloud data centers [C]//Proceedings of 2014 International Conference on Engineering and Technology. Cairo, Egypt: IEEE Press, 2014: 1-5.
- [16] CITRIX. Cloudbridge [EB/OL]. [2021-02-20]. https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/cloudbridge-technical-overview-jp.pdf.
- [17] WANG L, LU Z, WEN X, et al. Joint optimization of service function chaining and resource allocation in network function virtualization[J]. IEEE Access, 2016, 7: 8084-8094.
- [18] CISCO. NAT order of operation[EB/OL]. [2021-02-20]. <https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/6209-5.html?dtd=ossdc000283>.
- [19] BOUET M, LEGUAY J, COMBE T, et al. Cost-based placement of vDPI functions in NFV infrastructures[J]. International Journal of Network Management, 2015, 25(6): 490-506.
- [20] MARTINS J, AHMED M, RAICIU C, et al. ClickOS and the art of network function virtualization[C]//Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation. Seattle, USA: ACM Press, 2014: 459-473.
- [21] ORLOWSKI S, WESSÄLY R, PIÓRO M, et al. SNDlib 1.0-survivable network design library[J]. Networks, 2010, 55(3): 276-286.
- [22] KASHANI Z H, SHIVA M. BCH coding and multi-hop communication in wireless sensor networks [C]//Proceedings of 2006 IFIP International Conference on Wireless and Optical Communications Networks. Bangalore, India: IEEE Press, 2006: 1-5.
- [23] 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.
- SHI J G, ZHANG J, XU H, et al. Joint optimization of virtualized network function placement and routing allocation for operational expenditure [J]. Journal of Electronics and Information Technology, 2019, 41(4): 973-979. (in Chinese)

编辑 索书志