计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191000116

基于软件定义网络资源优化的虚拟网络功能部署策略

黄梅根[1] 汪 涛[1] 刘 亮[2] 庞瑞琴[1] 杜 欢[1]
1 重庆邮电大学计算机科学与技术学院 重庆 400065
2 重庆邮电大学通信与信息工程学院 重庆 400065
(huangmg@cqupt.edu.cn)

摘 要 随着软件定义网络(Software Define Network,SDN)和网络功能虚拟化(Network Function Virtual,NFV)技术的不断发展,防火墙、入侵检测等硬件中间件被动态部署在特定服务器上的虚拟网络功能(Virtual Network Function,VNF)所替代。为了满足流量安全和性能策略,网络流请求通常需要经过特定的 VNF 序列,称为服务功能链(Service Function Chain,SFC),这使得 VNF 的动态部署问题成为目前软件定义网络中的一个研究热点。学术界提出了多种部署策略,但由于大部分是在单一资源约束条件下进行的部署研究,无法实现全局网络资源的负载均衡。文中提出了充分考虑全局网络资源的虚拟网络功能部署策略。首先,给出了网络模型的整体结构,并用整数线性规划模型对该问题进行数学建模,由于该问题是一个 NP 难问题,因此,设计了一个高效的启发式搜索算法(Heuristic Search Algorithm,HSA)来对原问题进行求解,该算法能够在满足全局网络资源的约束下高效地利用网络资源实现 VNF 的动态部署并实现负载均衡。实验仿真结果表明,该部署算法能够很好地降低负载均衡度,并提高流请求接收率。

关键词:软件定义网络;网络功能虚拟化;部署;负载均衡;请求接收率
中图法分类号 TP393

Virtual Network Function Deployment Strategy Based on Software Defined Network Resource Optimization

HUANG Mei-gen[1],WANG Tao[1],LIU Liang[2],PANG Rui-qin[1] and DU Huan[1]
1 School of Computer Science and Technology,Chongqing University of Posts and Telecommunications,Chongqing 400065,China
2 School of Communication and Information Engineering,Chongqing University of Posts and Telecommunications,Chongqing 400065,China

Abstract With the continuous development of software-defined network and network function virtualization technology,hardware middleware such as firewall and intrusion detection is replaced by virtual network functions dynamically deployed on specific servers. In addition,in order to meet the traffic security and performance policy,network traffic requests usually need to go through a specific VNF sequence,known as service function chain,which makes the dynamic deployment of VNF become a hot topic in software defined network. Many deployment strategies have been proposed in the academic circle. However,most of the deployment studies are conducted under the constraint of a single resource,and the load balancing of global network resources cannot be achieved. Therefore,this paper proposes a virtual network function deployment strategy that fully considers the global network resources. Firstly,the whole structure of the network model is given,and an integer linear programming model is introduced for mathematical modeling. Then,an improved model solving algorithm is proposed,which can effectively utilize network resources and achieve load balancing under the constraint of global network resources. Finally,the simulation results show that the proposed deployment algorithm can reduce the load balance and improve the request reception rate.

Keywords Software define network,Network function virtual,Deploy,Load balancing,Request reception rate

1 引言

网络功能虚拟化是近年来兴起的一种将软件与硬件设备分离的技术。它能显著降低运营资本支出,并通过降低实现时间来促进新服务部署的灵活性[1-2]。软件定义网络是一种将控制平面与数据平面解耦的新型网络范式。SDN 控制器通过中央控制和灵活的管理,根据获取的全局网络信息有效地控制网络功能间的数据转发[3]。这两种技术的结合可以增强网络的灵活性和可扩展性,达到基础设施资源共享的效果。另外,为了满足流量安全和性能策略,在启用 NFV 的 SDN 网

本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:重庆市科委基础研究与前沿探索项目(cstc2018jcyjA0743,cstc2018jcyjA0644);重庆市教委科学技术研究项目(KJQN201800640,KJ1502003)
This work was supported by the Natural Science Foundation Project of CQ CSTC (cstc2018jcyjA0743,cstc2018jcyjA0644) and Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN201800640,KJ1502003).

通信作者:刘亮(liuliang@cqupt.edu.cn)

络中，网络流请求通常需要经过特定的虚拟网络功能序列，NFV 和 SDN 的发展为动态服务功能链技术的发展提供了有利条件。图 1 为启用 NFV 的网络中服务功能链配置的简单示例。图 1 中共有 7 个网络节点，分别为节点 1－节点 7，其中节点 3、节点 5、节点 6 连接了物理服务器，可以部署 VNF，称之为功能节点。节点 1、节点 2、节点 4、节点 7 没有连接物理服务器，作为交换机节点。其中功能节点 6 部署了 VNF1 和 VNF2，功能节点 5 部署了 VNF4，功能节点 3 部署了 VNF3。从 A 到 C 的数据流（A－＞VNF1－＞VNF4－＞C）通过 VNF 实例 VNF1 和 VNF4，而从 B 到 C 之间的数据流通过实例 VNF2 和 VNF3。其中 VNF1 和 VNF2 在同一服务器上共存。

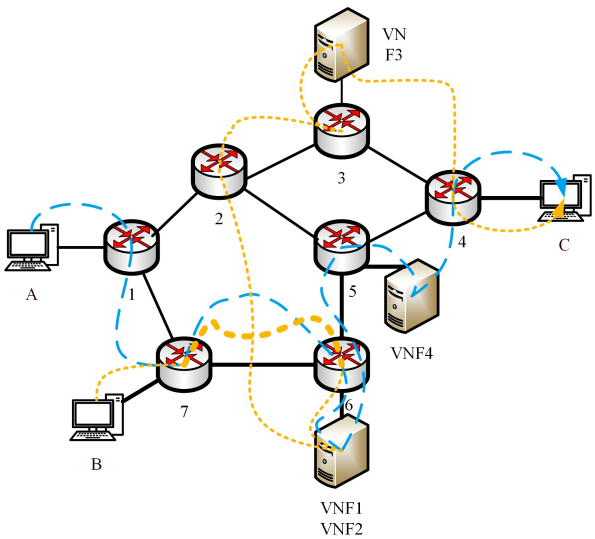

图 1　SFC 配置的简单示例

Fig. 1　Simple example of SFC configuration

目前，基于软件定义网络的动态服务功能链技术还处于研究阶段，其中考虑网络资源约束，最小化网络资源消耗的 VNF 动态部署是当前研究的热点之一。在现有的 VNF 部署策略研究中，大多数方法仅考虑了全局网络资源中的一种单一网络资源约束来实现 VNF 的部署，没有综合考虑全局网络资源如交换机流表条目、服务器计算资源、链路带宽等对 VNF 动态部署的影响。由于每个 SDN 交换机节点的流表通常由昂贵且耗电的三元内容寻址内存实现（Ternary Content Addressable Memory，TCAM），TCAM 一般只能存储几千个流表条目，因此在进行 VNF 动态部署时，交换机节点的流表条目资源是不可忽视的一种网络资源。本文将网络节点区分为功能节点和交换机节点。其中功能节点是指交换机连接了物理服务器的网络节点，如图 1 中的节点 3，VNF 只允许部署在功能节点上并需要消耗功能节点上的计算资源，而图 1 中的交换机节点 1 只负责交换路由信息。由于微数据中心和云数据中心可以作为功能节点[4-5]，与交换机节点相比，其可以不考虑功能节点上的流表消耗，仅关注其计算资源的消耗。而交换机节点可以忽略它的计算资源消耗，仅关注其流表条目资源的消耗。本文从全局网络角度出发，不仅考虑了功能节点计算资源和链路带宽资源的消耗，还考虑了交换机节点的流表条目资源的消耗，充分利用虚拟网络资源灵活分配这一特性，为具有 SFC 请求的流选择合适的 VNF 部署位置和服务路径，从而在满足资源和服务功能链顺序的约束条件下有效地部署 VNF 以达到全局网络资源的负载均衡。本文首先给出了网络模型的整体结构，然后引入整数线性规划模型对该问题进行数学建模。由于已知此类问题是 NP 难的[6]，因此对该问题设计了一个高效的启发式搜索算法（Heuristic Search Algorithm，HSA）进行求解，并得到了较好的实验效果。

2　相关工作

近年来，虚拟网络功能 VNF 的部署问题被学术界广泛研究。文献[7]提出了一种基于禁忌搜索算法的 VNF 部署方法，该方法通过设置禁忌表的方式在全局网络中搜索服务功能最优的部署位置，但该方法只考虑了节点资源利用率。文献[8]提出了一种面向 NFV 环境下的服务功能管理机制，该机制主要通过维特比算法在候选节点中选择部署开销最小的节点作为功能节点，有效地降低了服务功能的部署成本，但其缺乏对链路资源开销的考虑。文献[9]提出了一种服务功能部署模型，该模型采用（Auxiliary Frequency Slot Matrix，AFM）启发式算法进行求解，但 AFM 仅以物理节点的计算资源容量作为选择策略，未从全局服务路径资源的利用方面考虑 VNF 的部署问题。文献[10]通过 Greedy 算法穷举全网满足资源约束及连通性的所有服务路径，并选用链路资源占用量最小的服务路径，但这种方法只侧重于链路的优化选择问题。文献[11]提出 SDN/NFV 架构下的服务功能部署机制，该机制构建了分层图模型，将时延最小化作为服务路径选择依据，虽然降低了服务功能链的处理时间，但缺少对节点资源的考虑。

3　网络模型与问题描述

3.1　具有网络功能虚拟化的 SDN 网络模型

本文将物理网络定义为无向图 $G=(V,L)$，其中 V 和 L 分别表示节点集和链路集，$u,v\in V$ 是物理节点，$uv\in L$ 代表由节点 u 和节点 v 连接形成的一条物理链路。如果物理节点承载了 VNF 功能，则将 $V_m\subset V$ 定义为功能节点集，否则将 $V_s\subset V$ 定义为普通节点集。定义集合 M 表示 VNF 集，其中 $m\in M$ 表示在 VNF 集中的一个虚拟网络功能。G 中有一个 SDN 控制器，它监控 G 网络中功能节点的计算资源、链路带宽资源以及交换机流表条目资源，并执行资源分配，以满足每条请求流的资源需求。假设一条具有 SFC 请求的流到达，如果可以满足该条流请求的所需资源，则该请求将被接受；否则，该请求被拒绝。

本文以 F 表示具有 SFC 请求的流的集合，其中请求流 $f_i\in F$。C_u^{ft} 表示交换机节点的流表容量，当 SFC 请求流到达时，交换机节点的剩余流表条目数的比率用 $r_{i,u}^{ft}$ 表示。功能节点上的计算资源总量表示为 C_u^{cpu}，链路 $uv\in L$ 的带宽资源容量用 C_{uv}^{bw} 表示。当 SFC 请求流到达时，功能节点上剩余计算资源的比率以及链路 uv 上剩余带宽资源的比率分别用 $r_{i,u}^{cpu}$ 和 $r_{i,uv}^{bw}$ 表示。

3.2　具有服务功能链的流请求

在具有虚拟网络功能的 SDN 网络中，由用户发起的流应该始终遍历一组按预定义顺序连接的 VNF 实例，以满足用

户的需求。本文假设所有的流都带有 SFC 请求，每个 SFC 请求由一个入口节点、一个出口节点和一系列 VNF 请求组成。本文用一个五元组来表示 SFC 请求流，SFC 请求流以及服务功能链的公式如下：

$$p_{r_i}=(s_{r_i},t_{r_i};SC_{r_i},bw_{r_i},CP_{r_i}) \tag{1}$$

$$SC_{r_i}=\langle SC_{r_i,1},SC_{r_i,2},\cdots,SC_{r_i,l}\rangle,l_{SC_{r_i}}=|SC_{r_i}| \tag{2}$$

式(1)的 p_{r_i}，s_{r_i} 表示请求流的入口节点，t_{r_i} 表示请求流的出口节点，每个数据包从源 s_{r_i} 到目标 t_{r_i} 必须依次通过相应的节点。请求流的 VNF 服务链序列被定义为 SC_{r_i}，其中，$SC_{r_i}=(SC_{r_i,1},SC_{r_i,2},\cdots,SC_{r_i,l})$ 表示流必须通过的有序 VNF 序列(如：Firewall→IDS→Proxy)；$l_{SC_{r_i}}=|SC_{r_i,l}|$ 表示请求流的服务功能链长度，即一条请求的 VNF 请求总数；bw_{r_i} 为请求流的链路带宽资源需求量；CP_{ri} 为 SFC 请求流的计算资源需求量。

3.3 逻辑功能链路图

根据系统模型，由 SDN 服务链部署模型编排平面中的 SFC 策略[12]将 SFC 流请求相应的虚拟网络功能组合起来构建一个逻辑功能链图 $\overline{G}=(\overline{V},\overline{L})$。逻辑功能链图是一个带权有向图，其中 $\overline{V}$ 和 $\overline{L}$ 分别表示为逻辑节点集和逻辑链路集，$\overline{u}$，$\overline{v}\in\overline{V}$ 表示两个逻辑节点，$\overline{uv}\in\overline{L}$ 表示连接 $\overline{u}$，$\overline{v}$ 节点的逻辑链路。图 2 为一个逻辑功能链图，在 SFC 请求流到达节点 t_{r_i} 之前，必须先按顺序遍历完节点 s_{r_i}，$SC_{r_i,1}$，$SC_{r_i,2}$ 以及 $SC_{r_i,3}$。如下所示：

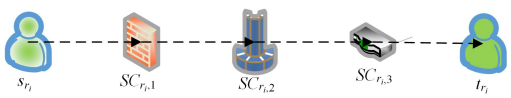

图 2 服务功能链图

Fig. 2 Service function chain diagram

当 SFC 请求到来时，SDN 控制器中的编排模块会根据服务功能链和底层物理网络的资源状态，按照某种部署方法确定 VNF 的优化部署节点，最终找到一条满足全局网络资源约束的优化服务路径。

3.4 资源成本定义

启用 NFV 的 SDN 网络的全局资源消耗包括 3 个主要部分：1)功能节点的计算资源消耗；2)物理链路带宽资源的消耗；3)交换机节点流表条目资源的消耗。

对于 G 中的物理节点和链路上的资源的使用有一个重要的特性：资源使用的边际成本随着资源负载的增加而膨胀。与负载较小的节点和链路相比，负载较大的节点和链路成本开销将会更大。因此，在接收请求流时，应该使用负载更小的节点和链路。当第 r_i 条请求流到来时，交换机节点的可用流表条目为 $C_u^{ft}r_{i,u}^{ft}$，链路 $uv\in L$ 的剩余带宽资源为 $C_{uv}^{bw}r_{i,uv}^{bw}$。我们使用如下函数分别刻画流表资源成本 $Cost_{r_i,u}^{ft}$、链路带宽资源成本 $Cost_{r_i,uv}^{bw}(uv)$、计算资源成本 $Cost_{r_i,u}^{cpu}$。

$$Cost_{r_i,u}^{ft}(u)=\frac{\max\limits_{u\in V_s}C_u^{ft}}{C_u^{ft}r_{i,u}^{ft}+\alpha} \tag{3}$$

$$Cost_{r_i,uv}^{bw}(uv)=\frac{\max\limits_{uv\in L}C_{uv}^{bw}}{C_{uv}^{bw}r_{i,uv}^{bw}+\alpha} \tag{4}$$

$$Cost_{r_i,u}^{cpu}(u)=\frac{\max\limits_{u\in V_m}C_u^{cpu}}{C_u^{cpu}r_{i,u}^{cpu}+\alpha} \tag{5}$$

其中，α 为一个极小值，以保证分母大于 0。$Cost_{r_i,u}^{ft}(u)$，$Cost_{r_i,uv}^{bw}(uv)$，$Cost_{r_i,u}^{cpu}(u)$ 的取值范围均为 $(1,\infty)$，并且随着资源的消耗以非线性的形式越来越大，其可以充分表征资源使用的边际成本随着资源负载的增加而膨胀。

3.5 基于整数规划的优化模型

优化模型是综合考虑服务资源需求、网络节点资源、网络链路资源等情况，按照指定目标函数找到 VNF 节点部署的最优位置。本节以最小化资源代价为目标，将问题刻画为整数规划模型(Integer Liner Programming，ILP)。

3.5.1 变量

二进制变量 x_u^m 表示 VNF m 是否被部署在物理节点 u 上：

$$x_u^m=\begin{cases}1, & \text{VNF } m \text{ 部署在节点 } u \text{ 上}\\ 0, & \text{VNF } m \text{ 没有部署在节点 } u \text{ 上}\end{cases} \tag{6}$$

二进制变量 $y_{r_i,m}^{\overline{u}}$ 表示 VNF 请求是否使用了节点 $\overline{u}$ 上对应的 VNF：

$$y_{r_i,m}^{\overline{u}}=\begin{cases}1, & \text{VNF 请求使用了节点}\overline{u}\text{上对应的 VNF}\\ 0, & \text{其他}\end{cases} \tag{7}$$

式(8)、式(9)分别表示 $\overline{u}$ $\overline{v}\in\overline{L}$ 是否通过链路 $uv\in L$ 或者物理节点 $u\in V$：

$$z_{r_i,uv}^{\overline{u}\ \overline{v}}=\begin{cases}1, & \overline{uv}\text{通过链路 } uv\\ 0, & \text{其他}\end{cases} \tag{8}$$

$$z_{r_i,u}^{\overline{u}\ \overline{v}}=\begin{cases}1, & \overline{uv}\text{通过节点 } u\\ 0, & \text{其他}\end{cases} \tag{9}$$

一旦某条链路被选中，则这条链路的两个端点也必须被选中：

$$z_{r_i,u}^{\overline{u}\ \overline{v}}z_{r_i,v}^{\overline{u}\ \overline{v}}=\begin{cases}1, & z_{r_i,uv}^{\overline{u}\ \overline{v}}=1,\forall u,v\in V,\forall uv\in L,\forall\overline{uv}\in\overline{L}\\ 0, & \text{其他}\end{cases} \tag{10}$$

式(11)为连接性约束，确保 SFC 请求流的路由路径是连续的：

$$\sum_{u,v\in V}\sum_{\overline{u}\ \overline{v}\in\overline{L}}(z_{r_i,uv}^{\overline{u}\ \overline{v}}-z_{r_i,vu}^{\overline{u}\ \overline{v}})=\begin{cases}1, & u=s_{r_i}\\ -1, & u=t_{r_i}\\ 0, & \text{其他}\end{cases} \tag{11}$$

3.5.2 资源约束

对于链路 $uv\in L$，SFC 请求 p_{ri} 的带宽资源消耗不能超过物理链路上剩余的带宽资源：

$$\sum_{\overline{u}\ \overline{v}\in\overline{L}}z_{r_i,uv}^{\overline{u}\ \overline{v}}bw_{ri}\leqslant C_{uv}^{bw}r_{r_i,uv}^{bw},\forall uv\in L \tag{12}$$

SFC 请求 p_{ri} 的流表条目资源消耗不能超过交换机节点上剩余的流表条目资源，必须满足以下约束：

$$\sum_{\overline{u}\ \overline{v}\in\overline{L}}r_{i,u}^{ft}z_{r_i,u}^{\overline{u}\ \overline{v}}\leqslant C_u^{ft},\forall u\in V_s \tag{13}$$

此外，SFC 请求 p_{ri} 的功能节点上所有 CPU 消耗不能超过所选功能节点上剩余的 CPU：

$$\sum_{\overline{u}\in\overline{V}}\sum_{m\in M}CP_{r_i}y_{r_i,m}^{\overline{u}}x_u^m\leqslant C_u^{cpu}r_{r_i,u}^{cpu},\forall u\in V_m \tag{14}$$

3.5.3 SFC 请求流路由路径消耗的资源总成本

1)SFC 请求流路由路径消耗的链路带宽总成本为：

$$\mathbb{Z}=\sum_{uv\in L}\sum_{\overline{u}\ \overline{v}\in\overline{L}}z_{r_i,uv}^{\overline{u}\ \overline{v}}Cost_{r_i,uv}^{bw}(uv) \tag{15}$$

2)SFC 请求流路由路径消耗的交换机流表条目资源总成本为：

$$\mathbb{Q}=\sum_{u\in V_s}\sum_{\bar{u}\bar{v}\in\bar{L}} z_{r_i,u}^{\bar{u}\bar{v}} Cost_{r_i,u}^{ft}(u) \tag{16}$$

3）SFC 请求流路由路径消耗的功能节点的计算资源消耗总成本：

$$\mathbb{N}=\sum_{u\in V_m}\sum_{\bar{u}\in\bar{V}}\sum_{m\in M} Cost_{r_i,u}^{cpu}(u) y_{r_i,m}^{\bar{u}} x_u^m \tag{17}$$

3.5.4 目标函数

$$\min \mathbb{Z}+\mathbb{Q}+\mathbb{B} \quad \text{s.t. 式(6)}-\text{式(17)} \tag{18}$$

由于服务功能链部署问题被证明是 NP-Hard 问题[6]，因而无法使用多项式时间算法来求解该问题，我们提出一个高效的启发式搜索算法（Heuristic Search Algorithm，HSA）进行求解。该算法综合考虑了链路带宽资源、交换机节点流表条目资源以及功能节点的计算资源，在此类资源的约束下实现了 VNF 的部署以及 SFC 请求的路径选择，最终实现网络全局的资源分配。

4 算法描述

HSA 算法首先需要输入 SFC 请求流 $p_{r_i}=(s_{r_i},t_{r_i};SC_{r_i},bw_{r_i},CP_{r_i})$ 以及物理网络 $G=(V,L)$，并根据到达的 SFC 请求流构建出逻辑功能链路图 $\bar{G}=(\bar{V},\bar{L})$。

然后，HSA 算法先将 VNF 按照计算资源需求量从小到大排列，并依次添加至队列 Q 中；然后将所有满足约束(14)的物理节点添加至一个可映射集合 N 中；接着根据 $u=\{u_1 \mid u_1=\arg\max C_u^{cpu} r_{r_i,u}^{cpu}, \forall u\in V\}$ 将 VNF 成功映射到物理节点 u 上，将映射成功的物理节点 u 添加到 VNF 部署最优位置集合 $Loc_{opt}(u)$ 中；最后以 $Cost_{r_i,uw}^{bw}(uv)+Cost_{r_i,u}^{ft}(u)+Cost_{r_i,u}^{cpu}(u)$（即带宽资源代价、交换机节点流表条目资源代价以及功能节点计算资源代价之和）为权重，利用 K 条最短路径算法（K-Dijkstra）计算从源点到目的点的 K 条最短路径集合 $Rout=\{r_1,r_2,\cdots,r_k\}$。其中，对于功能节点，其流表条目资源代价等价为 0；对于交换机节点，其计算资源代价也等价为 0。最终从这 K 条最短路径集合中选择链路带宽资源剩余量最多的路径记录到集合 $Rout_{opt}$ 中，作为最终的 SFC 请求路由路径。

算法 HSA 具体描述步骤如算法 1 所示。

算法 1 HSA

输入：SFC 请求流 $p_{r_i}=(s_{r_i},t_{r_i};SC_{r_i},bw_{r_i},CP_{r_i})$，物理网络 $G=(V,L)$

输出：$Loc_{opt}(u)$，$Rout_{opt}$

1. 根据到达的 SFC 请求流构建逻辑功能链路图 $\bar{G}=(\bar{V},\bar{L})$。
2. 初始化一个 VNF 队列 Q。
3. 初始化一个可映射节点集合 N。
4. 初始化一个 VNF 部署最优位置集合 $Loc_{opt}(u)$。
5. 将集合 $\bar{V}$ 中 VNF 的计算资源需求量 CP_{r_i} 按从小到大的顺序排序，并将排序好的 CP_{r_i} 结果放入队列 Q 中。
6. 构建可映射节点集合 N。搜索物理节点集合 V，如果物理节点的剩余计算资源量 $C_u^{cpu} r_{r_i,u}^{cpu}$ 满足约束(14)，则将此物理节点添加至集合 N 中。
7. 选择 VNF 最优部署位置 $Loc_{opt}(u)$。取出队列 Q 中的元素，并根据 $u=\{u_1 \mid u_1=\arg\max C_u^{cpu} r_{r_i,u}^{cpu}, \forall u\in V\}$ 搜索可映射集合 N 中的物理节点，将 VNF 映射到物理节点 u 上，如果 VNF 成功映射到物理节点 u 上，则 $x_u^m=1$，并将节点 u 添加到集合 $Loc_{opt}(u)$ 中；否则，返回步骤 7 求得 $u=\{u_1 \mid u_1=\arg\max C_u^{cpu} r_{r_i,u}^{cpu}, \forall u\in V\}$ 的次优解，并更新队列 Queue。
8. 如果队列 Q 中有 VNF 没有完成映射，则返回步骤 7；否则，完成 VNF 映射，并得到最终 $Loc_{opt}(u)$ 集合。
9. 根据集合 $Loc_{opt}(u)$、服务功能链 SC_{r_i}，并且以 $Cost_{r_i,uv}^{bw}(uv)+Cost_{r_i,u}^{ft}(u)+Cost_{r_i,u}^{cpu}(u)$ 为权重，利用 K-Dijkstra 算法计算从源点到目的点的 K 条最短路径集合 $Rout=\{r_1,r_2,\cdots,r_k\}$。
10. 将路径集合 Rout 中的 K 条最短路径按照链路带宽资源剩余量从小到大排序，并首先选取资源剩余量最大的路径，记录到集合 $Rout_{opt}$ 中，得到 SFC 请求流的路由路径。否则，退回到最短路径集合 $Rout=\{r_1,r_2,\cdots,r_k\}$ 中求得次优解。

该算法有两处采取回溯的机制，即：1）在算法的步骤 7 中，假如 VNF 没有成功映射到计算资源剩余量最多的物理节点，则需要计算 $u=\{u_1 \mid u_1=\arg\max C_u^{cpu} r_{r_i,u}^{cpu}, \forall u\in V\}$ 的次优解，然后重新映射；2）算法执行到步骤 8 时，假如队列中还有 VNF 没有映射，则需要回溯到步骤 7，从 VNF 队列中再次取出一个元素，完成后续的步骤。

5 仿真实验

本节详细描述了 HSA 算法与现有两种典型部署算法对比的仿真实验，并给出性能比较。两种典型的部署算法分别为：1）仅仅优化节点资源[7]的部署算法（TS 算法）；2）仅优化链路带宽资源[10]的部署算法（Greedy 算法）。本文实验采取负载均衡度和请求流的接收率这两个重要评价指标来体现 HSA 算法的有效性。

5.1 实验环境和参数设置

本实验在配置为 Intel Core i7-3770 3.60 GHz、8 GB 内存的 Windows 系统 PC 机上运行。网络拓扑结构是 mininet 的仿真环境下生成的 fat-tree 结构。功能节点计算资源剩余量、交换机节点流表条目资源剩余量以及链路带宽资源剩余量均服从[1000，2000]的随机分布，每条 SFC 请求流由不同类型的服务功能组成，并且每条 SFC 请求流所需功能节点计算机资源以及链路带宽资源的需求量都服从[1，2]均匀分布，交换机节点资源需求量均为 1，即 1 条流表条目。实验中为了避免资源分配不均匀导致的对实验指标的影响，将服务功能链的长度统一设置为 3。

5.2 算法性能对比

5.2.1 负载均衡度

负载均衡度表示一条请求流被接收时整条通路的负载累积情况。该指标反映算法在整个网络中所有资源利用率的优劣程度，负载均衡度的取值越小越好。本文定义的负载均衡度的公式为：

$$Load=\psi_{uv}\sum_{r_i\in R,uv\in L}\frac{bw_{r_i}}{C_{uw}^{bw} r_{i,uw}^{bw}}+\psi_{v_s}\sum_{r_i\in R,v_s\in V}\frac{ft_{r_i}}{C_{v_s}^{ft} r_{i,v_s}^{ft}}+\psi_{v_f}\sum_{r_i\in R,v_f\in V}\frac{cp_{r_i}}{C_{v_f}^{cpu} r_{i,v_f}^{cpu}} \tag{19}$$

其中，ψ_{uv}，ψ_{v_s}，以及 ψ_{v_f} 用于调整链路负载、交换机节点负载以及功能节点负载的比率。

实验中分别生成 1000～10000 条请求流，并重复进行 10 次取平均值，分别统计 3 种算法的负载均衡度，如图 3 所示。可知，随着请求流的增加，网络的负载均衡度逐渐增大，曲线逐步趋于平缓并接近于数值 1。即随着请求流的增加，网络中剩余的可用资源逐渐减少，并且随着负载均衡度趋于 1，网

络中请求接收率逐渐降低。其次，随着请求流的增加，HSA算法的部署策略能最有效地平衡负载，由于TS算法和Greedy算法都仅考虑一种网络资源的分配，从而使得整个网络的负载处于相对不平衡的状态。因此，HSA算法能更有效地平衡负载。

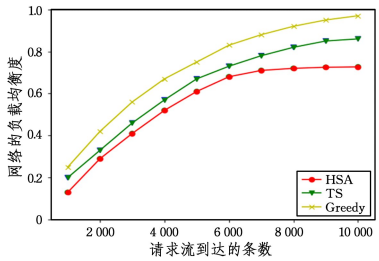

图3 负载均衡度比较

Fig. 3 Comparison of load balance

5.2.2 请求流的接收率

请求接收率表示满足各项资源约束的请求接收比例，它能很好地体现算法的效用性，其取值越大越好。请求接收率可表示为成功接收并成功部署到网络中的请求流条数与请求流总条数的比值，用 r_{accept} 来表示，其计算公式如下：

$$r_{\text{accept}}=\lim_{t\to\infty}\frac{\sum_{t=0}^{t} r_{i_{\text{accpet}}}}{\sum_{t=0}^{t} r_{i_{\text{total}}}} \tag{20}$$

其中，$\sum_{x=0}^{x} r_{i_{\text{accpet}}}$ 表示从 $t=0$ 时刻到 T 时刻成功接收并成功部署到网络中的请求流条数；$\sum_{t=0}^{t} r_{i_{\text{total}}}$ 表示网络中到达的请求流总条数。

实验中同样随机产生1 000～10 000条请求流，并重复进行10次取平均值，分别统计这3种部署算法的请求接收率，如图4所示。由图4可知，HSA算法的最大请求接收率、最小请求接收率以及平均请求接收率比其他两种部署算法要高。并且到达网络中的请求流条数相同时，HSA算法的请求接收率比其他两种部署算法要高。因为HSA算法考虑了全局资源的分配，降低了整个服务路径的负载均衡度，从而间接地提高了网络中资源的利用率，最终提高了请求接收率。

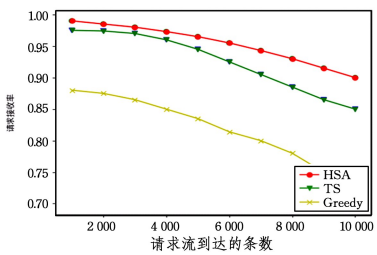

图4 请求接收率比较

Fig. 4 Request reception rate comparison

结束语 本文提出了在交换机节点流表资源、功能节点计算资源以及链路带宽资源这3种网络资源约束下进行VNF部署的部署策略。首先，根据虚拟网络功能部署问题构建一个网络模型，然后将网络模型建模成整型线性规划数学模型，最终通过考虑全局网络资源的部署算法确定虚拟网络功能的部署位置以及请求流的最终服务路径，并与TS算法和Greedy算法进行对比。实验结果表明，HSA算法在负载均衡度、请求接收率性能指标上均优于TS算法和Greedy算法。

参 考 文 献

[1] MIJUMBI R, SERRAT J, et al. Network function virtualization: State of theart and research challenges[J]. IEEE Commun. Surveys Tuts., 2016, 18(1): 236-262.

[2] PHAM C, TRAN N H, REN S. Traffic-aware and Energy-efficient VNF Placement for Service Chaining: Joint Sampling and Matching Approach[J]. IEEE Trans. Serv. Comput., 2017, 13(1): 172-185.

[3] OpenFlow Switch Specification: Version 1.5.1.[OL]. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switchv1.5.1.pdf.

[4] BHAMARE D, JAIN R, SAMAKA M, et al. A survey on service function chaining[J]. J. Netw. Comput. Appl., 2016, 75: 138-155.

[5] JALALI F, HINTON K, AYRE R. Fog computing may help to save energy in cloud computing[J]. IEEE J. Sel. Areas Commun., 2016, 34(5): 1728-1739.

[6] 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.

[7] HERRERA J G, BOTERO J F. Resource allocation in NFV: A comprehensive survey[J]. IEEE Transactions on Network and Service Management, 2016, 13(3): 518-532.

[8] SHI J G, XU H L, LU L P. Research on themigration queue of data center's virtual machine in softwaredefined networks[J]. Journal of Electronics &InformationTechnology, 2017, 39(5): 1193-1199.

[9] MIJUMBI R, SERRAT J, GORRICHO J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions[C]// Network Softwarization. 2015: 1-9.

[10] LUKOVSZKI T, ROST M, SCHMID S. It's a match!: near-optimal andincremental middlebox deployment[J]. ACM SIGCOMM ComputerCommunication Review, 2016, 46(1): 30-36.

[11] DWARAKI A, WOLF T. Adaptive service-chain routing for virtual networkfunctions in software-defined networks[C]// Proc of Workshop on HotTopics in Middleboxes and Network Function Virtualization. New York: ACM Press, 2016: 32-37.

[12] 刘益岑，卢昱，王珊，等. 一种基于软件定义网络的服务功能链优化部署机制[J]. 计算机应用研究，2019(10)：1-3.

HUANG Mei-gen, born in 1963, senior engineer. His research interests include software definition network, data center network, and machine learning.

LIU Liang, born in 1979, Ph.D candidate, associate professor. His research interests include data center network, software-defined network and mobile edge computing.