

# 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

陈卓<sup>1,2</sup>,冯钢<sup>2</sup>,刘蓓<sup>2</sup>,周杨<sup>3</sup>

- (1. 重庆理工大学计算机科学与工程学院,重庆 200433;  
2. 电子科技大学通信抗干扰技术国家级重点实验室,四川成都 710077;  
3. 奥本大学计算机科学与软件工程学院,美国阿拉巴马州奥本 36849)

**摘要:** 在引入网络功能虚拟化(NFV)后,运营商能基于通用的计算和网络平台提供更具弹性的服务. 本文研究基于NFV的运营商网络中的服务功能链(SFC)的迁移重配置问题. 首先分析了业务流的动态性导致的运营商网络中资源利用率下降和负载不均的问题,然后在将计算资源、网络资源和业务流端到端限制等条件纳入考虑的基础上,建立了面向业务流服务时延优化的数学模型,并证明了SFC迁移重配置问题的NP性. 进一步基于遗传算法提出了一种易于部署的启发式SFC迁移重配置策略. 通过详细的对比实验表明算法在端到端时延、链路带宽资源占用率和通用服务器资源占用率等关键指标上优于其它方案12.3%、10.9%和15.8%以上. 这进一步说明所提出方案通过对运营商网络资源的合理调配,在保证业务流服务质量的同时有效改善了资源利用效率.

**关键词:** 网络功能虚拟化; 服务功能链; 运营商网络; 遗传算法

**中图分类号:** TP393.0 **文献标识码:** A **文章编号:** 0372-2112(2018)09-2229-09

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2018.09.026

## Delay Optimization Oriented Service Function Chain Migration and Re-deployment in Operator Network

CHEN Zhuo<sup>1,2</sup>, FENG Gang<sup>2</sup>, LIU Bei<sup>2</sup>, ZHOU Yang<sup>3</sup>

- (1. College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 200433, China;  
2. National Key Laboratory of Science and Technology on Communications,  
University of Electronic Science and Technology of China, Chengdu, Sichuan 710077, China;  
3. Department of Computer Science and Software Engineering, Auburn University, Auburn 36849, USA)

**Abstract:** With the introduction of network functional virtualization (NFV), operators can provide more flexible services based on general computing and network platforms. This paper investigates the problem of migration and re-deployment of service function chain (SFC) in NFV based operator network. Firstly, this paper analyzes the problem of resource under-utilization and load unbalance in the operator network caused by the dynamic of the flow. Then, based on the consideration of the conditions such as computing resources, network resources and end-to-end restrictions, the mathematical model towards the service delay optimization is formulated, and the NP property of SFC migration and re-deployment problem is proved. Furthermore, a heuristic SFC migration and re-deployment policy is proposed based on genetic algorithm. In-depth experimental result demonstrates that the proposed solution is superior to other solutions in terms of end-to-end delay, the utilization ratio of link bandwidth resource and the utilization ratio of general server by 12.3%, 10.9%, and 15.8%. The results further shows that the resource utilization is effectively improved in the case of guaranteeing the quality of service for flow, through the reasonable deployment of operator network resources.

**Key words:** network function visualization; service function chain; operator network; genetic algorithm

收稿日期:2017-08-10;修回日期:2018-05-02;责任编辑:蓝红杰

基金项目:国家自然科学基金(No. 61631005);中央高校基本科研业务费项目(No. ZYGX2015Z005);重庆市基础与前沿研究计划项目(No. cste2013jcyjA40024)

## 1 引言

网络功能虚拟化 (Network Function Virtualization, NFV) 是一种重新构建网络体系架构和改变网络管理模式的新技术, 近年来已被越来越多的运用于运营商网络中<sup>[1,2]</sup>. 使用 NFV 技术的优势主要体现在 (1) 硬件资源虚拟化为多个虚拟机, 将既有的运营商网络的各种网元功能软件化为虚拟网络功能 (Virtualized Network Function, VNF)<sup>[3]</sup>. 这极大缩短网元功能的资源调配周期, 从而大幅提升业务部署的弹性; (2) 能够使用通用计算平台来构建电信基础设施, 进一步降低了运营商的设备投入成本. 但将 NFV 运用于运营商网络同时也面临一系列挑战, 其关键问题已成为当前产学研界研究关注的热点<sup>[4-6]</sup>.

在基于 NFV 构建的运营商网络中, 业务流的处理需要由不同类型的 VNF 以某种次序组成集合, 构成一条虚拟链路并提供服务, 称为: 服务功能链 (Service Function Chain, SFC)<sup>[6,7]</sup>. 例如: 具有某种特征的业务流需要在访问特定服务器之前先经过防火墙过滤处理, 再经过入侵检测处理, 则虚拟化的防火墙和入侵检测就构成了一条包含两个网络功能的 SFC. 由于部署 VNF 需要占用虚拟机, 因此建立 SFC 需要同时占用网络资源和相关通用服务器的计算资源. 然而, SFC 的引入在实现业务流的弹性化服务的同时也带来了新问题. 一方面, 由于业务流动态进出运营商网络和用户的移动性等因素导致一段时间后大量的 VNF 被部署却很少被使用, 这会严重影响运营商网络的资源利用效率<sup>[8]</sup>. 另一方面, SFC 所服务业务流自身的动态特性导致了通用服务器负载不均的情况出现, 这会影响到对于业务流的服务质量. 本文提出一种针对服务动态业务流的 SFC 迁移重配置策略, 其目的是在满足业务流服务质量的前提下实现运营商对网络资源的优化调配和合理使用. 该策略将网络资源、计算资源的占用以及网络性能纳入考察, 以时延优化为目标. 建立了最优化模型并提出基于遗传算法的 SFC 迁移重配置策略, 并通过详细的仿真实验对所提出算法策略的性能进行深入评估.

## 2 相关研究工作

和本文相关的的研究工作大多集中在针对虚拟网络和虚拟机的迁移重配置问题的探讨. 文献[9]在基于云的边缘网络场景下, 研究了将多台具有关联关系的虚拟机通过在线迁移聚合为功能更强的虚拟机的问题, 其优化的目标是总的服务宕机时间最小和在线迁移时间最短. 针对数据延迟敏感类网络应用的要求, 文献[10]系统性的分析了虚拟机迁移所需网络资源以及在线迁移过程中应用中断的时间, 提出了一种确定性

模型进行形式化描述, 最后给出并通过实验验证了估计虚拟机动态迁移过程中所需网络资源的算法. Xiao 等学者<sup>[11]</sup>研究了一种基于虚拟机资源需求的虚拟机动态迁移和虚拟网络拓扑动态调整的策略, 其目的是虚拟机在迁移过程中所需的资源开销和虚拟机之间的通信开销. 文献[12]在基于 PlanetLab 的网络虚拟化环境下, 设计、实现并评估了负责编排虚拟网络迁移的控制器 PL-VMN, 为有效的进行资源分配和保证虚拟网络运行效率, 提出了虚拟网络迁移调度方案. Ghaznavi 等学者<sup>[13]</sup>提出了一种称为 SLFL 的合并算法, 其目的是优化 VNF 实例 (VNFI) 的部署位置以应对网络的流量动态变化, VNFI 的迁移是基于对当前状态和迁移后状态对网络性能带来的影响进行评估的. 文献[14]中, 研究者们提出了一种贪心策略, 实现在线的 VNF 的映射和调度. 与上述研究工作<sup>[10-15]</sup>相比较, SFC 的迁移重配置涉及的制约条件更多且更复杂. 首先要把 VNF 的类型和构成 SFC 的 VNF 集合作为迁移的基础, 其次 SFC 的迁移需要把虚拟链路所涉及的计算资源和网络资源都纳入考虑, 最后还需要考虑业务流对于 SFC 的请求具有端系统的特性. 目前对于该问题的研究工作还非常缺乏, 亟需深入研究.

## 3 问题描述

在引入 NFV 后, 运营商网络针对业务流的服务模型从逻辑上可划分为自下而上的三层, 即: 物理资源层、VNF 层和业务层. 物理资源层由部署于运营商网络中的各种通用服务器和交换机构成. VNF 层由各种运行在虚拟机之上经软件化的网络功能组成, 虚拟机的运行占用通用服务器的计算资源和相关联的网络资源. 业务层由服务于各类业务流的多条 SFC 组成, 每条 SFC 由不同网络功能 (Network Function, NF) 按某种次序链式构成, 为业务流提供端到端的服务. 某种类型的网络功能实例化并映射到对应类型的 VNF. 当服务一条业务流时, 现有的方案会为该业务流所需的每个 NF 实例化并映射到一个 VNF, 这使得运营商网络中 VNF 的数量随着业务流的增加同步增大, 而完成对业务流的服务后由于缺乏合理的资源重新调配机制导致计算和网络资源的持续消耗. 根据当前业务流对资源的需求量以及业务流自身的服务质量要求, 若能够对 SFC 实施合理的迁移重配置, 让多条 SFC 中相同类型的 NF 共享一个 VNF 实例, 则可合理的降低 VNF 数量, 实现运营商网络资源的优化使用. 该方案的实现不仅涉及到多条 SFC 中相同类型 NF 的迁移合并, 还必须考虑由于业务流在物理资源层的路径改变导致的对计算和网络资源占用的改变. 由于 NF 的迁移可能形成更重的计算和网络负载, 这可能使得迁移后的 SFC 为业务流提供的

服务时延增加,影响服务质量.

## 4 系统模型

### 4.1 形式化定义

将运营商网络服务模型中的物理资源层抽象为一个加权无向图  $G = (N, E)$ ,  $N$  为通用服务器构成的集合,对某一台通用服务器  $v$ ,其资源量  $R(v) \geq 0$  且处理速度为  $proc(v)$ .  $E$  表示物理链路集合,任意一条链路  $(u, v) \in E, (u, v \in N)$  的链路带宽定义为  $b(u, v)$  且  $b(u, v) \geq 0$ ,链路时延表示为  $d(u, v)$ . 变量  $y(u, v), (u, v \in N)$  表示链路  $(u, v)$  是否属于链路集合  $E$ ,若两台通用服务器直接相连  $y(u, v) = 1$ , 否则  $y(u, v) = 0$ . 用集合  $F$  表示服务业务流可能会使用的网络功能(NF)组成的集合. NF 具有类型的区别,某一类  $NF f \in F$  由三元组  $(r(f), c(f), \beta(f))$  确定. 其中,  $f$  实例化并映射到一个 VNF 所需要消耗的资源为  $r(f), r(f) \geq 0$ . 处理单位大小的业务流数据包,  $f$  需使用的资源为  $c(f), c(f) \geq 0$ . 经过  $f$  这类 NF 处理后的网络带宽占用的改变因子为  $\beta(f), \beta(f) \geq 0$ . 运营商网络中需要进行迁移重配置的 SFC 用集合  $C$  表示.  $C$  中的第  $i$  条 SFC 表示为  $SFC_i$ , 用一个六元组  $(I_i, E_i, T_i, D_i, S_i, V_i)$  对其描述,其中  $I_i, E_i \in N$  分别表示第  $i$  个 SFC 的流入/流出网元.  $T_i$  表示  $SFC_i$  从流入网元进入运营商网络所需占用的初始带宽.  $D_i$  为  $SFC_i$  为保证所服务业务流的服务质量,端到端的容忍时延.  $S_i$  表示经过  $SFC_i$  处理后业务流的带宽占用.  $V_i$  表示构成  $SFC_i$  的 NF 集合,用  $\{v_{i,1}, \dots, v_{i,r}, \dots, v_{i,k}\}$  表示. 其中  $v_{i,r}$  表示  $SFC_i$  中第  $r$  个 NF. 对于一个网络功能,为了能表明其所属类型,定义了一个二进制 0-1 变量  $t_{i,r}^f$ :

$$t_{i,r}^f = \begin{cases} 1, & v_{i,r} \text{ 的类型是 } f \in F \\ 0, & \text{其它} \end{cases} \quad (1)$$

$v_{i,r}$  用于处理业务流数据,所需计算资源为:

$$c_i^f = S_i \cdot \sum_{f \in F} t_{i,r}^f \cdot c(f) \quad (2)$$

经  $v_{i,r}$  处理后,业务流对网络带宽的占用为:

$$T_i^{r,r+1} = T_i \cdot \prod_{m=1}^r \left( \sum_{f \in F} t_{i,r}^f \cdot \beta(f) \right) \quad (3)$$

### 4.2 模型的建立

根据问题的描述,本文通过服务功能链合理的迁移和重配置,达到降低业务流的端到端服务时延的目的,该优化问题的目标函数可表述为:

$$\min \max \frac{delay_i}{D_i} \quad (4)$$

其中,  $delay_i$  是服务功能链  $SFC_i$  的实际时延,  $D_i$  是  $SFC_i$  所服务业务流的端到端最长时延限制. 对任意  $SFC_i$ , 系统会将其中的网络功能集合  $V_i$  映射到合适的通用服

器上,用决策变量  $x$  表示  $SFC_i$  中网络功能  $v_{i,r}$  的部署:

$$x_{i,r}^u = \begin{cases} 1, & SFC_i \text{ 中第 } r \text{ 个 NF 部署在服务器 } u \text{ 上} \\ 0, & \text{其它} \end{cases} \quad (5)$$

同时,系统也将为  $SFC_i$  选择一条路径,用决策变量  $\alpha$  如下表示路径:

$$\alpha_{u,v}^{v_{i,r}, v_{i,r+1}} = \begin{cases} 1, & SFC_i \text{ 中 } v_{i,r}, v_{i,r+1} \text{ 的链路经过 } (u, v), (u, v) \in E \\ 0, & \text{其它} \end{cases} \quad (6)$$

继续定义二进制 0-1 变量  $y$ , 表示通用服务器上是否实例化了某类型为  $f \in F$  的 VNF, 下式成立:

$$y_u^f = 1, \text{ if } \sum_{SFC_i \in C} \sum_{r=2}^{k_i-1} x_{i,r}^u t_{i,r}^f \geq 1, \forall u \in N, \forall f \in F \quad (7)$$

运营商网络中,在任意一台通用服务器之上所部署的所有 VNF 所需要的计算资源总和必须小于等于该通用服务器能够提供的最大计算资源量, VNF 所需计算资源不仅包括自身需要的处理资源,还包括将该网络功能实例化运行所需要的资源,即有下面约束,即式(8)成立:

$$\sum_{SFC_i \in C} \sum_{r=2}^{k_i-1} x_{i,r}^u c_i^r + \sum_{f \in F} y_u^f r(f) \leq R(u), \forall u \in N \quad (8)$$

对于每条 SFC 中的每一个网络功能,必须且只能映射到一个 VNF. 同时,每一个 VNF 也只能对应一台通用服务器,因此式(9)成立:

$$\sum_{u \in N} x_{i,r}^u = 1, \forall SFC_i \in C, \forall r \in \{2, \dots, k_i - 1\} \quad (9)$$

对运营商网络中的任意一条实际网络链路,经过该链路的所有 SFC 对带宽资源的占用之和不能超过该链路的最大带宽资源,则式(10)成立:

$$\sum_{SFC_i \in C} \sum_{r=2}^{k_i-1} \alpha_{u,v}^{v_{i,r}, v_{i,r+1}} T_i^{r,r+1} \leq b(u, v), \forall (u, v) \in E \quad (10)$$

继续定义一个变量  $\varphi(u)$  表示和通用服务器  $u$  直接相连的其它通用服务器:

$$\varphi(u) = \{v | y(u, v) = 1\}, \forall u, v \in N \quad (11)$$

在 SFC 之间的虚拟链路映射到实际物理链路后,考虑到网络中流量守恒的限制,对每条 SFC 中的每一段链路,都有下面约束,即式(12)成立:

$$\sum_{v \in \varphi(u)} (\alpha_{u,v}^{v_{i,r}, v_{i,r+1}} - \alpha_{v,u}^{v_{i,r}, v_{i,r+1}}) = x_{i,r}^u - x_{i,r+1}^u, \forall u \in N, \forall SFC_i \in C, \forall r \in \{2, \dots, k_i - 1\} \quad (12)$$

最后,需要保障每条 SFC 的端到端时延要求. 服务功能链  $SFC_i$  为业务流服务产生的时延  $delay_i$  实际包括两部分:链路时延和通用服务器的处理时延. 这两部分时延之和应小于等于业务流的时延限制,我们用  $dp_{i,r}^u$  表示  $SFC_i$  的第  $r$  个网络功能在通用服务器  $u$  上的处理时延,有式(13)成立:

$$\sum_{(u,v) \in E} \sum_{r=1}^{k_i-1} \alpha_{u,v}^{v_i, v_{i,r+1}} d(u,v) + \sum_{r=2}^{k_i-1} \sum_{u \in N} x_{i,r}^u dp_{i,r}^u \leq D_i, \quad \forall SFC_i \in C \quad (13)$$

参考文献[16]对时延部分进行建模,链路时延  $d(u,v)$  包括传输时延  $d_t$ 、传播时延  $d_p^{u,v}$  以及排队时延  $d_q^{u,v}$  三个部分. 对于排队时延,采用一个期望服务时间为  $d_i$  的  $M/M/1$  排队模型加以描述:

$$(1) d_t = \frac{S_i}{T_i^{r,r+1}}, S_i \text{ 表示经过 } SFC_i \text{ 处理的流占用的网络带宽, } T_i^{r,r+1} \text{ 表示 } SFC_i \text{ 的传输带宽.}$$

(2)  $d_p^{u,v} = \frac{l}{c_m}$ , 其中  $l$  表示的是网络中物理链路的长度,  $c_m$  则是电磁波信号在物理介质中的传播速度.

(3)  $d_q^{u,v} = \frac{load_{u,v}}{1 - load_{u,v}} \cdot d_i$ , 其中  $load_{u,v}$  表示链路  $(u,v)$  除功能链  $SFC_i$  所需带宽之外的带宽资源占用和链路总带宽之间的比例, 即有  $load_{u,v} = \frac{b_{SFC_i}^{u,v}}{b(u,v)}$ . 更具体的, 带宽  $b_{SFC_i}^{u,v}$  可以表示为:

$$b_{SFC_i}^{u,v} = \sum_{SFC_i \in C} \sum_{j=2}^{k_i-1} \alpha_{u,v}^{v_i, v_{i,j+1}} T_i^{j,j+1} - T_i^{r,r+1} \quad (14)$$

因此, 排队时延可以表示为:

$$d_q^{u,v} = \frac{\left( \sum_{SFC_i \in C} \sum_{j=2}^{k_i-1} \alpha_{u,v}^{v_i, v_{i,j+1}} T_i^{j,j+1} - T_i^{r,r+1} \right) / b(u,v)}{1 - \left( \sum_{SFC_i \in C} \sum_{j=2}^{k_i-1} \alpha_{u,v}^{v_i, v_{i,j+1}} T_i^{j,j+1} - T_i^{r,r+1} \right) / b(u,v)} \cdot \frac{S_i}{T_i^{r,r+1}} \quad (15)$$

考虑到在链路时延的三个部分中, 传输时延只与  $SFC$  本身有关系, 传播时延的值很小, 可以忽略不计. 因此只考虑排队时延, 即有式(16)成立:

$$d(u,v) = \frac{\left( \sum_{SFC_i \in C} \sum_{j=2}^{k_i-1} \alpha_{u,v}^{v_i, v_{i,j+1}} T_i^{j,j+1} - T_i^{r,r+1} \right) \cdot S_i}{\left( b(u,v) - \sum_{SFC_i \in C} \sum_{j=2}^{k_i-1} \alpha_{u,v}^{v_i, v_{i,j+1}} T_i^{j,j+1} + T_i^{r,r+1} \right) \cdot T_i^{r,r+1}} \quad (16)$$

对于通用服务器的处理时延, 采用基于通用处理器共享<sup>[17]</sup>定义的处理时延模型, 即:

$$dp_{i,r}^u = \frac{load_u}{load_i} \cdot t_{proc} \quad (17)$$

其中, 数据包需占用的处理时间  $t_{proc} = S_i / proc(u)$ ,  $S_i$  表示的是服务功能链  $SFC_i$  的大小,  $proc(u)$  则是物理节点  $u$  的处理速度; 物理节点  $u$  处理器负载百分比为  $load_u = cr_u / R(u)$ ,  $R(u)$  是物理节点  $u$  的最大负载,  $cr_u$  则是物理节点  $u$  当前的负载, 可以用下式表示:

$$cr_u = \sum_{SFC_i \in C} \sum_{r=2}^{k_i-1} x_{i,r}^u c_i^r + \sum_{f \in F} y_u^r r(f) \quad (18)$$

$load_i$  表示的是  $SFC_i$  中  $v_{i,r}$  产生的负载占当前通用服务器  $u$  负载的比重:

$$load_i = c_i^r / cr_u \quad (19)$$

进一步化简后有:

$$dp_{i,r}^u = \frac{\left( \sum_{SFC_i \in C} \sum_{j=2}^{k_i-1} x_{i,j}^u c_i^j + \sum_{f \in F} y_u^r r(f) \right)^2 \cdot S_i}{R(u) \cdot c_i^r \cdot proc(u)} \quad (20)$$

将该问题简单化, 假设运营商网络中只有一条  $SFC$  为业务流提供服务, 该业务流要求  $SFC$  的端到端的数据延迟是  $k$  倍的网络路径最大延迟. 每台通用服务器只能实例化运行一个 VNF, 且一个 VNF 只能对应到一个业务层的 NF. 这时,  $SFC$  的迁移重配置问题就转化为一个虚拟网络的映射问题, 而后者为 NP-hard<sup>[18]</sup>. 因此  $SFC$  迁移问题可以归结为一个 NP-hard 问题.

## 5 基于遗传算法的 SFC 迁移重配置

遗传算法 (Genetic Algorithm, GA) 最早是由 J. Holland 教授在 1975 年提出的一种模拟进化生物学物种进化原则的随机化启发式搜索算法<sup>[19]</sup>. 本文把 GA 算法作为求解服务功能链迁移重配置问题的基础算法, 主要原因在于: (1) 求解问题自身的特性. GA 算法直接对结构对象进行各种遗传操作, 对待搜索的空间并不要求必须连续可导, 而本文是在具有离散性质的搜索空间中实现  $SFC$  迁移的; (2) GA 算法搜索具有并行性的特点; (3) 能够避免局部最优. GA 中的交叉和变异操作具有随机性, 没有确定的规则改变搜索的方向. 通过调整搜索空间避免陷入局部最优.

### 5.1 算法描述

本文采用了一种 0-1 矩阵编码方式, 这种方式能够将矩阵整体作为遗传子代个体, 从而保证子代个体基因的完整性. 基于此, 将矩阵作为群体个体进行遗传运算, 矩阵的行数 (以  $m$  表示) 等于所需要迁移的服务功能链中所需 NFs 的数目之和, 矩阵的列数 (以  $n$  表示) 等于底层物理网络的节点数目. 遗传子代中的第  $k$  代种群中的个体数目为  $N$ , 每个个体又被称之为矩阵染色体, 均为  $m \times n$  阶矩阵, 则  $Q_k = \{A_1, A_2, \dots, A_N\}$  表示该种群, 其中  $A_k^r = (a_{i,j})_{m \times n}$ , 表示第  $k$  代种群中的第  $r$  个个体, 个体中的每个元素  $a_{ij}^r$  是矩阵染色体的基因元素,  $a_{ij}^r$  需满足以下两个约束  $\sum_{j=1}^n a_{ij}^r = 1, \forall i \in \{1, 2, \dots, m\}$  和  $a_{ij}^r \in \{0, 1\}, \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$ , 对染色体的有关的操作描述如下:

(1) 生成初始种群. 在该算法中, 初始种群将包含  $N$  个染色体,  $N$  为常量且与问题规模有关.

(2) 选择复制. 本文在传统轮盘赌选择法的基础上使用精英选择策略 (Elitist Selection Strategy, ESS). ESS

能够使群体收敛到所求解优化问题的最优解。

(3) 交叉. 本文采用多行矩阵杂交, 按杂交概率  $P_c$  ( $0 \leq P_c \leq 1$ ) 交换两个矩阵染色体中对应位置的行基因元素的过程, 交换的行是随机的. 对于种群  $Q_k$  中两个染色体  $A_k^i$  和  $A_k^j$  的交叉运算, 先进行初始化:  $A_k^i = (a_{i,j}^i)_{mn}$ ,  $A_k^j = (a_{i,j}^j)_{mn}$ , 若以第二行交换为例, 则交叉运算后矩阵染色体为:

$$A_k^i = \begin{bmatrix} a_{11}^i & a_{12}^i & \cdots & a_{1n}^i \\ a_{21}^i & a_{22}^i & \cdots & a_{2n}^i \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}^i & a_{m2}^i & \cdots & a_{mn}^i \end{bmatrix}, A_k^j = \begin{bmatrix} a_{11}^j & a_{12}^j & \cdots & a_{1n}^j \\ a_{21}^j & a_{22}^j & \cdots & a_{2n}^j \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}^j & a_{m2}^j & \cdots & a_{mn}^j \end{bmatrix}$$

(4) 变异. 本文采用的是根据适应度变化的自适应值<sup>[20]</sup>, 计算式如下:

$$P_m^i = \begin{cases} \frac{fit_{\max} - fit(A_k^i)}{fit_{\max} - fit_{\min}}, & fit(A_k^i) < \overline{fit} \\ P_m, & fit(A_k^i) \geq \overline{fit} \end{cases} \quad (21)$$

这样的变异概率计算使得适应度较低的个体有更大的概率产生变异, 从而增加了个体转变为适应度较高的个体的可能性. 对于矩阵编码且有约束限制的染色体, 变异操作则是针对每行进行的. 在变异操作中, 根据概率  $P_m^i$  使得一个矩阵染色体中的某行或某些行的基因元素发生改变, 而被改变行的改变方式具有随机性, 但条件是必须满足问题本身的约束.

(5) 可行性检测. 在经过交叉和变异的操作后得到了许多的新的个体, 但这些个体可能并非原问题的可行解. 因此, 在进行遗传操作后加入可行性检测十分必要, 从而确保新生成的每个个体都在原问题可行解的范围内. 每个染色体分别对应着 NFs 到 VNF 的映射及实例化, 由此可以确定目标变量  $x_{i,r}^u$  的值. 此时我们可以对模型中的节点映射的约束是否成立进行判断, 在满足了约束成立的条件, 根据 NFs 的映射方案并结合路径选择算法得到另一个目标变量  $\alpha$  的值. 这样就可以判断模型中链路映射的约束是否满足, 当这些约束均被满足的时候, 就可以得到该问题的可行解.

## 5.2 适应度计算

如果一个染色体对应的适应度函数的数值较大则说明该染色体比较接近最优解, 它被选择用来生成下一代群体的几率就更大. 在这种情况下, 通过逐代地提高种群的平均适应度以及和个体的最佳适应度, 达到求解优化问题的目的. 对每种 NF 映射关系进行编码后可以得到对应的染色体个体. 反之, 若对每个染色体个体进行解码就可以得到网络功能(NF)映射关系. 因此对于一个给定的染色体而言, 其原始问题可以简化为链路映射问题, 通过对该问题进行的路径选择, 最终得到染色体的适应度和相应的最优链路映射方案. 对于

本文中的最小化问题, 采用如下的适应度函数:

$$fit(x) = \begin{cases} c_{\max} - f(x), & c_{\max} - f(x) \geq 0 \\ 0, & \text{其它} \end{cases} \quad (22)$$

其中,  $f(x)$  是系统模型中的优化目标函数.  $c_{\max}$  理论上应该是种群中个体的  $f(x)$  的最大值, 根据问题的物理意义, SFC 端到端的实际时延与最大时延要求的比值在  $(0, 1]$  区间内, 因此  $c_{\max}$  值取为 1, 适应度函数为:

$$fit(x) = 1 - \max \frac{delay_i}{D_i} \quad (23)$$

## 5.3 算法流程

基于 5.2 节的描述, 对于特定染色体实际就是网络功能(NF)到 VNF 的一种映射实例化方案, 也即是确定目标变量  $x_{i,r}^u$  的值. 此时对每个业务流请求而言, 处理时延  $dp_{i,r}^u$  成为定值. 对于每条 SFC 的功能模块之间的路径, 以路径最短作为标准进行路径选择. 这样对于运营商网络的底层网络拓扑, 任意通用服务器之间的最短路径可以提前求出, 从而简化问题的求解. 本文提出的基于 GA 的启发式算法主要步骤如算法 1 所示:

### 算法 1 SFC 迁移重配置算法伪码

输入: 运营商物理网络拓扑  $G = (N, E)$ , SFC 的集合  $C$ , 网络功能集合  $F$

输出: SFC 重配置方案

步骤:

Step1: 初始化相关参数, 计算 SFC<sub>s</sub> 中  $v_{i,r}$  的  $c_i^r$  以及  $v_{i,r}$  与  $v_{i,r+1}$  的  $T_{i,r}^{r+1}$ .

Step2: 令  $fit_{\max} = 0$ ,  $m = 0$ ,  $M_{\text{best}} = \emptyset$ ,  $B_{\text{best}} = \emptyset$ , 最大迭代次数  $T$ .

Step3: 产生初始种群  $Q_0$ , 大小为  $N$ , 交叉概率  $P_c$ , 变异概率  $P_m$ .

Step4: while ( $m < T$ ) {

for  $A_m^i \in Q_m$  do

得到决策变量  $x_{i,r}^u$

由通用服务器之间最短路径确定  $\alpha_{i,r}^{v_i, v_{i,r+1}}$

求解得到适应度  $fit(A_m^i)$

if  $fit(p_i) > fit_{\max}$

$fit_{\max} = fit(p_i)$

记录 VNF 部署方案  $M_{\text{best}}$

记录链路映射方案  $B_{\text{best}}$

end

end

根据 ESS 选择合适的染色体构成父代群  $Q'_m$

for  $i \in |Q'_m|/2$

$A_m^{2i}$  以概率  $P_c$  与  $A_m^{2i+1}$  交叉

对新个体  $A_m^{2i}$  与  $A_m^{2i+1}$  可行性检查

求解新个体适应度  $fit(A_m^{2i})fit(A_m^{2i+1})$

end

for  $A_m^i \in Q'_m$

求解对应的变异概率  $P_m^i$

以概率  $P_m^i$  变异

对变异个体进行可行性检查

end

$$\begin{aligned} Q_{m+1} &\leftarrow Q'_m \\ m &= m + 1 \\ &\vdots \end{aligned}$$

Step5: 输出 VNF 映射策略  $M_{best}$

由于 SFC 迁移重配置算法是建立在 GA 算法基础之上实现的,因此在运营商网络中,当 SFC 和 NF 数量不同的时候,找到一个最优的 SFC 迁移重配置方案的时间复杂度在  $O(n^2)$  以内<sup>[21]</sup>.

## 6 仿真与性能评估

本节通过 Matlab 构建数值仿真环境,对本文提出的策略进行评估.选择了迁移前的部署情况、随机迁移部署方案以及基于贪心策略的迁移部署方案作为对比参考.当网络中有若干个服务业务流的 SFC 需要进行重配置时,随机迁移部署策略是随机的选择具有足够计算资源的通用服务器和网络链路进行节点和链路的映射.基于贪心的迁移部署策略则是对每个业务流的服务请求均以最小化端到端时延为目标,逐一实现部署,最后得到为多条业务流服务的 SFC 迁移后的映射策略.仿真实验中,主要就几个关键指标进行评估,包括:基于 GA 的 SFC 迁移策略在端到端时延、链路带宽利用以及通用服务器的资源利用率.

### 6.1 参数设置

参考文献[20,23],本文选择了两种具有代表性的网络拓扑 NSF 和 USNet,前者网络规模较小而后者网络规模较大.在这两种网络拓扑结构下,各项参数的设置参考文献[22~25],在 NSF 网络,网络节点数:30;物理链路数:45;通用服务器带宽资源:U[60,100];通用服务器处理速度 100;物理链路带宽:U[80,100].对于 USNet 网络,网络节点数:92;物理链路数:152;通用服务器带宽资源:U[60,100];通用服务器处理速度 100;物理链路带宽:U[80,100].对于网络功能参数,网络功能类型数:10;实例化 NF 所需资源:U[5,10];NF 所需的资源:U[2,6];NF 带宽改变因子:[0.5,2].对于虚拟功能链 SFC 的参数,初始带宽 U[5,10];SFC 的大小:[1,3];端到端时延 U{10,15,20};网络功能集合长度:U[2,6];SFC 流入网元节点:随机产生;SFC 流出网元节点:随机产生.对于遗传算法的参数,最大迭代次数:500;种群个数:20;交叉概率:0.6;初始变异概率:0.05.

### 6.2 性能指标

仿真实验中,本文定义了下面三个性能指标评估各种策略的性能.

(1) 最小最大时延比,这是优化目标,具体表述为:

$$\min \max \frac{\text{delay}_i}{D_i}$$

其中,  $\text{delay}_i$  是服务功能链  $SFC_i$  的实际时延,  $D_i$  是

端到端的最高时延要求.

(2) 物理网络中链路带宽的资源占用率  $\varphi_1$ ,如下式所示:

$$\varphi_1 = \frac{\sum_{(u,v) \in ESFC_i \in C} \sum_{r=2}^{k_i-1} \alpha_{u,v}^{v_i, v_{i+1}} T_i^{r,r+1}}{\sum_{(u,v) \in E} b(u,v)}$$

分子部分是网络中实际占用的物理链路带宽资源,分母部分是网络中全部物理链路的带宽资源总量.

(3) 通用服务器的资源占用率  $\varphi_2$ ,如下式所示:

$$\varphi_2 = \frac{\sum_{u \in N} \left( \sum_{SFC_i \in C} \sum_{r=2}^{k_i-1} x_{i,r}^u c_i^r + \sum_{f \in F} y_u^f r(f) \right)}{\sum_{u \in N} R(u)}$$

分子部分是网络中实际占用的通用服务器的资源总量,分母部分是网络中通用服务器的资源总量.

### 6.3 实验结论

对迁移前的部署情况、基于遗传算法的迁移部署方案、随机迁移部署方案以及基于贪心法的迁移部署方案分别以 No Migration、GA、Random 和 Greedy 表示.首先在 NSF 网络环境中,当需要迁移部署的 SFC 数量从 0 增加至 30 的时候,比较不同迁移方案下的端到端时延.

从图 1 可以看到,随着需要迁移的 SFC 数量增多,各种方案的最小最大端到端时延比均呈现增加的趋势,这是由于业务流的服务请求增加,网络中的通用服务器负载和链路负载增大,处理时延和链路时延增加所导致.与迁移前的部署情况、随机迁移方案以及基于贪心的算法方案相比,基于 GA 的迁移方案能够更有效的保证业务流的端到端时延.当服务功能链提供给业务流的服务时延大于等于业务流自身端到端的最高时延要求时,都设定端到端的时延比为 1.当业务流数量大于 14 的时候,基于 GA 的方案时延比优于基于贪心的方案超过 10.5%,在业务流数量达到 30 时候更是达到 12.3%.而若采用随机策略和不进行迁移的时候,当业务流达到 20 和 24 的时候,业务流的端到端延迟就已经达到或超过了服务质量的限制.图 2 对比了不同迁移部署方案下的物理链路资源占用率,可以看到当业务流数量达到 30 的时候,基于 GA 的方案链路资源占用率仅为 66.8%,分别优于基于贪心的方案、不进行迁移的情况以及基于随机方案 10.9%、24.3% 和 33.2%.表明在基于 GA 的方案下,链路带宽资源占用率要明显低于其它几种对比迁移方案,这使运营商网络有更多剩余链路资源服务于其它业务流.

图 3 对比了不同方案对于通用服务器的资源占用率,业务流所需要的最大物理节点资源也就是不进行迁移重配置所需的资源.可以看到当需要迁移部署的

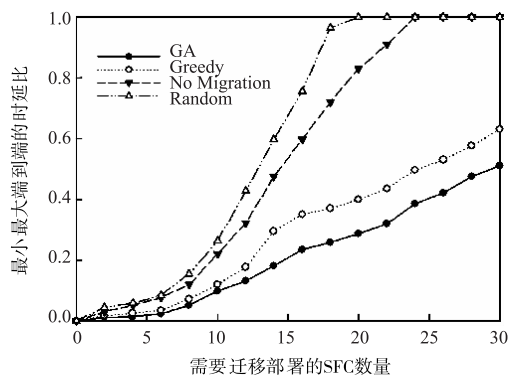


图1 端到端时延对比(NSF)

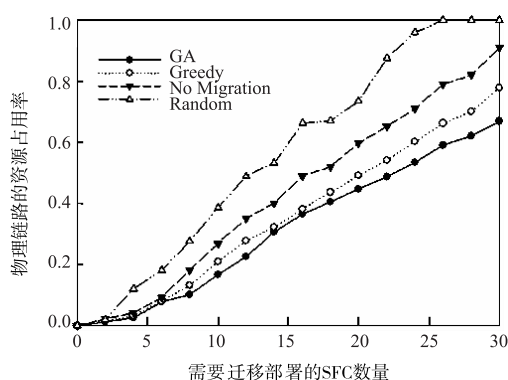


图2 链路带宽资源占用率对比(NSF)

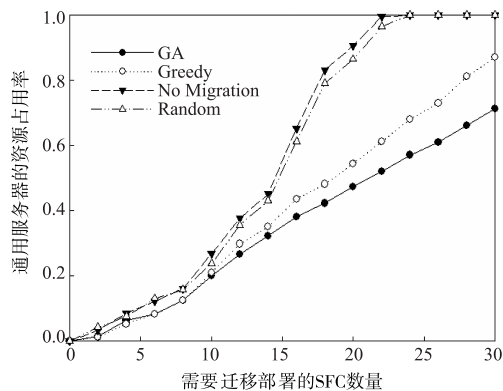


图3 通用服务器资源占用率对比(NSF)

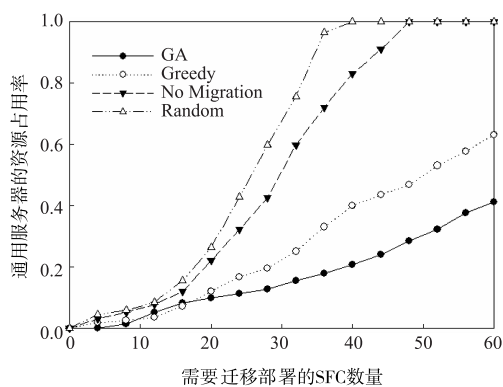


图4 端到端时延对比(USNet)

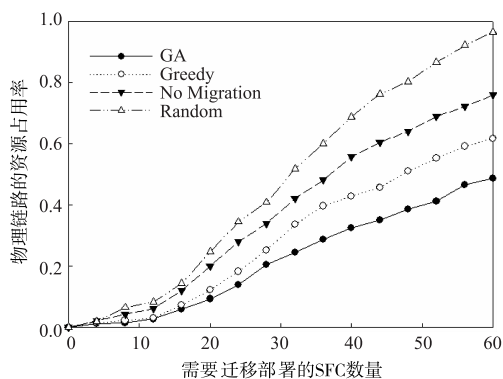


图5 链路带宽资源占用率对比(USNet)

SFC 数量较少,如小于 10 的时候,网络中可以合并的网络功能数量有限,这也使得网络中通过迁移重配置节省的通用服务器资源也较有限.但随着 SFC 数量的增加,则相同的网络功能也增多,这使得迁移重配置之后节约出的通用服务器计算资源也显著增加.当 SFC 的数量达到 30 的时候,不进行迁移重配置则已经耗尽通用服务器资源.而当使用基于 GA 的方案,通过合理的进行迁移实现了通用服务器资源的优化使用,服务相同数量的 30 个 SFC 只占用了 71.3% 的通用服务器资源,比性能较次的基于贪心算法的方案具有 15.8% 的优势.在规模更大的 USNet 网络拓扑下进行评估,需要进行重配置的 SFC 数量最多为 60 个.如图 4 所示,随着需要迁移部署的 SFC 数量增多,最小最大链路时延比呈现增加的趋势,这与在 NSF 网络拓扑下的趋势一致.且当 SFC 的数量越多的情况下,基于 GA 算法的方案的优势越来越明显.在 SFC 数量为 60 的时候,基于 GA 算法的方案的最小最大链路时延比仍控制在 42.2%,优于性能较次的基于贪心算法的方案 21.1%,而不进行迁移和随机方案则已经达到和超过了业务流的端到端时延限制.

从图 5 可以看到在 USNet 网络模型下,基于 GA 算法的迁移方案下链路带宽资源占用率随着迁移重配置的 SFC 数量的增加,优势越明显.其主要原因一方面是

当 SFC 数量较少的时候,网络中可以合并的同类型 NF 模块较少,另一方面若不考虑其它因素合并同类型的 NF 模块则容易在网络中的某些物理节点处形成热点,不利于网络负载均衡的实现进而影响业务流的端到端时延.当 SFC 数量达到 60 的时候,基于 GA 的迁移重配置方案下链路带宽资源占用率对比为 48.7%,而基于贪心算法的方案、不进行迁移,随机迁移方案则分别为 61.7%、76.1% 和 96.6%.图 6 呈现出了和图 5 类似的趋势,当迁移重配置的 SFC 数量增加,通过合理的 NF 模块合并,有效控制了通用服务器资源的快速占用,当 SFC 达到 60,基于 GA 算法的方案将通用服务器资源的

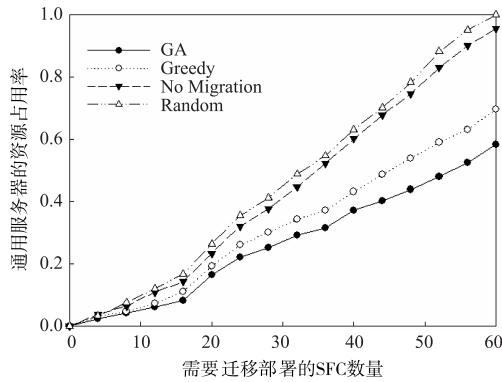


图6 通用服务器资源占用率对比(USNet)

占用率控制在 58.3%，由于性能较次的贪心算法则为 69.7%，不进行迁移和随机迁移均接近占用完所有服务器资源。

相比其它方案,通过上述详细的实验结果说明本文所提出的方案具有两方面的优势,一是具有良好的网络适应性,通过在 NSF 和 USNet 这两种具有代表性的网络拓扑下的实验结果表明基于 GA 算法的方案适应性良好,不会因为网络结构的变化带来性能上的差异,对基于不同网络结构的运营商网络都是适用的。二是基于 GA 算法的方案在网络规模较大的时候性能优势更为明显,随着运营商网络承载的业务越来越多样化和复杂化,网络自身规模在不断扩张,在各类网元、通用服务器和交换机等物理节点越来越多的网络中运用本文提出的方案其性能优势更为明显,能够在性能和资源占用方面取得更好的平衡。

## 7 结语

随着业务流的动态性日益突出,运营商网络的资源利用率下降和通用服务器负载不均的问题也逐渐显现。本文在形式化分析的基础上提出了易于部署的基于 GA 的启发式算法,并在不同拓扑及不同规模的网络场景下进行了详细对比实验,结果表明本文提出的 SFC 迁移重配置策略具有良好的网络适应性,能够在有效的保证业务流服务能力的同时兼顾运营商网络中资源的合理利用。本文提出的方案及算法为改善基于 NFV 的运营商网络服务能力提供了借鉴和理论支持。

### 参考文献

[1] Chatras B, Ozog F F. Network functions virtualization: the portability challenge [J]. *IEEE Network*, 2016, 30(4): 4-8.

[2] Abdelwahab S, Hamdaoui B, Guizani M. Network function virtualization in 5G [J]. *IEEE Communications Magazine*, 2016, 54(4): 84-91.

[3] Eramo V, Ammar M, Lavacca F G. Migration energy aware

reconfigurations of virtual network function instances in NFV architectures [J]. *IEEE Access*, 2017(5): 4927-4938.

- [4] 陈晓华,李春芝,陈良育,等. 虚拟网络映射高效节能运输模型及算法[J]. *电子学报*, 2016, 44(3): 725-731.  
CHEN Xiao-hua, LI Chun-zhi, CHEN Liang-yu, et al. Transportation model and algorithm for energy efficient virtual network embedding [J]. *Acta Electronica Sinica*, 2016, 44(3): 725-731. (in Chinese)
- [5] Blenk A, Basta A, Reisslein M, et al. Survey on network virtualization hypervisors for software defined networking [J]. *IEEE Communications Surveys & Tutorials*, 2016, 18(1): 655-685.
- [6] Ghaznavi M, Shahriar N, Kamali S, et al. Distributed service function chaining [J]. *IEEE Journal on Selected Areas in Communications*, 2017, 35(11): 2479-2489.
- [7] Ma W R, Medina C, Pan D. Traffic-aware placement of NFV middleboxes [A]. *IEEE Global Communications Conference (GLOBECOM)* [C]. San Diego, CA, USA: IEEE, 2015. 1-6.
- [8] Han B, Gopalakrishnan V, Ji L, et al. Network function virtualization: Challenges and opportunities for innovations [J]. *IEEE Communications Magazine*, 2015, 53(2): 90-97.
- [9] Cerroni W, Callegati F. Live migration of virtual network functions in cloud-based edge networks [A]. *IEEE International Conference on Communications (ICC)* [C]. Sydney, Australia: IEEE, 2014. 2963-2968.
- [10] Zhang J, Ren F Y, Shu R, et al. Guaranteeing Delay of live virtual machine migration by determining and provisioning appropriate bandwidth [J]. *IEEE Transactions on Computers*, 2016, 65(9): 2910-2917.
- [11] Xiao S H, Cui Y, Wang X, et al. Traffic-aware virtual machine migration in topology-adaptive DCN [A]. *IEEE International Conference on Network Protocols (ICNP)* [C]. Singapore: IEEE, 2016. 1-10.
- [12] Lo S, Ammar M, Zegura E, et al. Virtual network migration on real infrastructure: A PlanetLab case study [A]. *IF-IP Networking Conference* [C]. Trondheim, Norway: IF-IP, 2014. 1-9.
- [13] Ghaznavi M, Khan A, Shahriar N, et al. Elastic virtual network function placement [A]. *IEEE International Conference on Cloud Networking (CloudNet)* [C]. Niagara Falls, ON, Canada: IEEE, 2015. 255-260.
- [14] Mijumbi R, Serrat J, Gorricho J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions [A]. *IEEE Conference on Network Softwarization (NetSoft)* [C]. Paris, France: IEEE, 2015. 1-9.

- [15] Wen T, Yu H, Sun G, et al. Network function consolidation in service function chaining orchestration [A]. IEEE International Conference on Communications (ICC) [C]. Kuala Lumpur, Malaysia: IEEE, 2016. 1 – 6.
- [16] Dwaraki A, Wolf T. Adaptive service-chain routing for virtual network functions in software defined networks [A]. ACM Workshop on Hot Topics in Middleboxes and Network Function Virtualization [C]. London, UK: ACM, 2016. 32 – 37.
- [17] Li T, Baumberger D, Hahn S. Efficient and scalable multi-processor fair scheduling using distributed weighted round-robin [A]. ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming [C]. Edinburgh, Scotland: ACM, 2009. 65 – 74.
- [18] Fischer A, Botero J F, Till M, et al. Virtual network embedding: A survey [J]. IEEE Communications Surveys & Tutorials, 2013, 15(4): 1888 – 1906.
- [19] 陈国良, 王煦法, 庄镇泉, 等. 遗传算法及其应用 [M]. 北京: 人民邮电出版社, 2001. 35 – 56.
- [20] Lin T, Zhou Z, Tornatore M, et al. Demand-aware network function placement [J]. Journal of Lightwave Technology, 2016, 34(11): 2590 – 2600.
- [21] 何军, 黄厚宽, 遗传算法的平均计算时间分析 [A]. 第五届中国人工智能联合学术会议会议集 [C]. 西安: 西安交通大学出版社, 1998. 108 – 119.
- [22] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms [J]. IEEE Transactions on Systems Man & Cybernetics, 1994, 24(4): 656 – 667.
- [23] 狄浩. 虚拟网络的高效和可靠映射算法研究 [D]. 成都: 电子科技大学通信工程学院, 2013.
- [24] Mijumbi R, Serrat J, Gorricho J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions [A]. IEEE Conference on Network Softwarization [C]. London, UK: IEEE, 2015. 1 – 9.
- [25] 邢文训, 谢金星. 现代优化计算方法-第 2 版 [M]. 北京: 清华大学出版社, 2005. 113 – 142.

#### 作者简介



陈卓 男, 1980 年生, 副教授, 博士. 研究方向为网络虚拟化、数据中心网络和网络性能分析与评估.  
E-mail: chenzhuo@cqu.edu.cn.



冯钢 男, 1964 年生, 教授, 博士生导师, 研究方向为移动通信网络、网络体系结构及网络虚拟化技术.