

## 基于最长有效功能序列的服务功能链部署算法

李丹\* 兰巨龙 王鹏 胡宇翔

(国家数字交换系统工程技术研究中心 郑州 450000)

**摘要:** 服务功能链的服务性能取决于功能的部署位置和数据传输路径的选择。针对资源有限的网络中的服务功能链部署问题, 该文设计了一种基于最长有效功能序列(LEFS)的服务功能链部署算法, 以功能复用和带宽需求联合优化为目标, 在控制路径长度的同时根据LEFS逐步搜索中继节点, 直至满足服务请求。仿真结果表明, 该算法能够均衡网络资源, 同时优化网络的功能部署率和带宽利用率, 与其他算法相比, 网络资源利用率降低了10%, 可以支持更多的服务请求, 且算法计算复杂度低, 可以实现对服务请求的快速响应。

**关键词:** 服务功能链; 最长有效功能序列; 功能复用; 带宽需求

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2019)03-0680-07

DOI: 10.11999/JEIT180402

## Service Function Chain Deployment Algorithm Based on Longest Effective Function Sequence

LI Dan LAN Julong WANG Peng HU Yuxiang

(National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450000, China)

**Abstract:** The efficiency of Service Function Chain (SFC) depends closely on where functions are deployed and how to select paths for data transmission. For the problem of SFC deployment in a resource-constrained network, this paper proposes an optimization algorithm for SFC deployment based on the Longest Effective Function Sequence (LEFS). To optimize function deployment and bandwidth requirement jointly, the upper bound of path length is set and relay nodes are searched incrementally on the basis of LEFS until the service request is satisfied. Simulation results show that, the proposed algorithm can balance network resource and optimize the function deployment rate and bandwidth utilization. Compared with other algorithms, the utilization of network resource decreases 10%, so that more service requests can be supported. What is more, the algorithm has a lower computation complexity and can response to service requests quickly.

**Key words:** Service Function Chain (SFC); Longest Effective Function Sequence (LEFS); Function reuse; Bandwidth requirement

### 1 引言

随着互联网的高速发展和广泛应用, 多样化的业务和通信模式需求对网络中的新功能和 new 服务提出了更高的要求, 但是由于网络体系的僵化, 在当前互联网中部署新功能和 new 服务面临着可扩展性不足、灵活性差、建设成本高等弊端。网络功能虚拟化技术<sup>[1]</sup>(Network Function Virtualization, NFV)通过分离逻辑功能与物理资源, 将多样化的网络功

能和服务从底层物理资源中抽象出来, 可以根据需要灵活部署于网络中的任意位置, 使得网络功能和服务的灵活配置、编排和管理成为可能。在NFV的基础上, 服务功能链技术<sup>[2]</sup>(Service Function Chain, SFC)将网络中的定制服务分解为按顺序排列的一组网络功能, 如防火墙、入侵检测、服务代理、网络地址转换等, 这些功能可以部署在网络中的任意节点, 通过路径规划依次连接提供这些功能的节点即可实现相应的服务。在服务功能链部署问题中, 网络的服务性能取决于功能的部署位置和数据传输路径的选择, 如何在资源有限的网络中为每一条服务功能链选择合适的功能部署方式和数据传输路径, 是服务功能链部署算法的研究重点。

为了在网络中提供多样化服务, 需要部署大量收发节点和功能序列不同的服务功能链, 但是网络

收稿日期: 2018-04-28; 改回日期: 2018-10-10; 网络出版: 2018-10-30

\*通信作者: 李丹 pkulidan@foxmail.com

基金项目: 国家863计划(2015AA016102), 国家自然科学基金(61521003, 61702547)

Foundation Items: The National 863 Program of China (2015AA016102), The National Natural Science Foundation of China (61521003, 61702547)

资源却是有限的：一方面，每个节点能够部署的功能数量有限，不可能为网络中的所有服务部署新的虚拟功能；另一方面，节点间的链路带宽有限，过长的传输路径会消耗大量的带宽资源，降低网络的承载能力。因此，设计服务功能链部署算法时需要同时兼顾复用虚拟功能和降低路径带宽。目前针对服务功能链的部署问题已经有了一些相关研究，文献[3]设计了基于SDN和NFV的服务功能链动态部署架构，并对时延、带宽、负载、链路利用率、功耗等优化目标分别进行了归纳和建模。文献[4]把服务控制开销作为优化目标，研究了虚拟网络功能的优化部署问题，但是没有考虑服务中的功能顺序。文献[5,6]以优化传输路径代价和服务请求响应时间为目标，设计了分布式的服务路径选择算法，但是没有研究功能的部署方式。文献[7-9]研究服务功能链的部署问题，分别从功能复用率、节点利用率和控制开销3个角度设计了优化部署算法，但是没有考虑对链路带宽的优化。文献[10,11]研究了在数据中心网络中部署服务功能链的网络流量均衡算法。文献[12]将负载均衡作为优化目标，在部署功能时以较高的概率选择当前负载较低的节点。文献[13]为服务功能链部署问题设计了3个优化目标，分别是链路承载能力、节点使用率和路径延时，但是该文献采用整数线性规划算法，复杂度很高，而且缺乏对优化目标的联合优化分析。文献[14]以传输路径的可信度和时延为优化目标，通过启发式算法降低计算复杂度，在时延较小的路径集合中逐步搜索可信度符合要求的部署方式。

以上服务功能链部署算法大多针对单一优化目标，缺少对功能部署和带宽使用的平衡。针对节点处理能力和链路带宽有限的网络，文献[15]设计了一种图匹配算法，将物理拓扑图和功能拓扑图的连接矩阵进行最优匹配，选择最优的功能部署方式，但是该算法的适用条件要求每个节点只能提供一种功能且每种功能只能部署一次。文献[16]研究了一种功能合并部署算法，该算法统计各功能之间的带宽需求，将带宽需求较大的功能合并部署在同一节点上来降低路径带宽。文献[17]研究了一种启发式的功能部署算法，该算法首先根据服务请求计算收发节点间的一组最短路径，然后选择其中功能复用度最高的路径并在路径上部署剩余功能。虽然这两篇文章平衡了功能复用度和链路带宽需求，但是没有考虑在每个节点上部署的功能数量。文献[18]研究了节点部署功能有限时最佳路径长度和最佳功能复用度的估计算法，逐步搜索功能复用节点逼近最佳路径长度，最后在功能复用节点之间部署剩余功能，但是该算法搜索路径时没有考虑预留剩余功能

的位置，服务请求较多时需要延长路径来部署剩余功能。

基于以上分析，本文提出一种基于最长有效功能序列(LEFS)的服务功能链部署算法，以功能复用和带宽需求联合优化为目标，为每一条服务请求限定路径长度，在该范围内根据最长有效功能序列逐步搜索中继节点，寻找满足服务请求且复用度最高的路径。本算法在搜索路径的过程中预留了剩余功能的部署位置，同步完成路径选择和功能部署，在提高功能复用度的同时避免了部署剩余功能需要的额外带宽，可以有效提高网络服务能力，且算法能够实现对服务请求的快速响应，支持实时服务功能链的部署。

## 2 服务功能链部署问题

考虑一个无向加权网络  $G = (V, E)$ ， $C_E$ 表示每条链路最大带宽， $C_V$ 表示每个节点能够提供的功能个数。网络中有一组服务请求  $R = \{r_1, r_2, \dots\}$ ，每个请求可表示为  $r_i(o_i, S_i, t_i)$ ，其中  $o_i$ 和  $t_i$ 分别表示服务请求的源节点和目的节点， $S_i = \{s_{i1}, s_{i2}, \dots\}$ 表示  $r_i$ 需要经过的服务功能链， $s_{ij}$ 表示  $S_i$ 中的第  $j$ 项功能，每个请求的所需带宽用  $b_i$ 表示。对于每一个服务请求，都要选择一条路径与之对应，并且将服务功能链与路径节点上的功能依次进行复用匹配或新功能部署，在这个过程中应该尽可能降低部署的功能个数和需要的链路带宽。例如，在图1中，需要对一个服务请求  $r_1(A, \{f_1\}, D)$ 进行部署，有两条路径可以选择，其中实线表示的路径只需要2跳就可以完成服务请求，但是需要在路径上部署功能  $f_1$ ，适用于链路带宽受限的情况；相反，虚线表示的路径可以复用节点C上已经部署的功能  $f_1$ ，但是需要经过5跳到达目的节点，适用于功能部署受限的情况。通常，网络中  $C_E$ 和  $C_V$ 均有限，因此应该对功能复用和链路带宽联合优化，来提高网络的服务能力。

为了同时降低部署的功能个数和需要的链路带宽，本文将服务功能链部署问题建模为一个多约束优化问题：

$$\text{Min } \alpha \sum_i \frac{a_i b_i}{|E| C_E} + \beta \sum_i \frac{d_i}{|V| C_V} \quad (1)$$

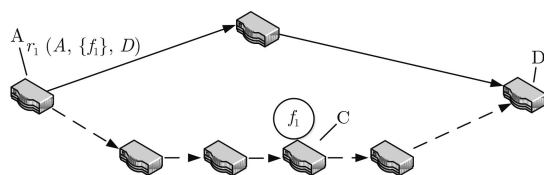


图1 服务功能链部署示意图

$$\text{s.t. } 1 \leq n_{ij} \leq n_{ij+1} \leq a_i + 1, \forall i, \forall j \quad (2)$$

$$d_i \leq C_V, \forall i \quad (3)$$

$$\sum_{i: e_k \in P_i} b_i \leq C_E, \forall e_k \in E \quad (4)$$

其中,  $a_i$ 表示第*i*条路径经过的跳数,  $d_i$ 表示第*i*个节点上部署的功能数,  $n_{ij}$ 表示与功能 $s_{ij}$ 匹配的节点在服务路径上的节点序号,  $e_k \in P_i$ 表示第*i*条传输路径经过链路 $e_k$ ,  $|E|$ 和 $|V|$ 分别表示网络的链路个数和节点个数。式(1)描述网络资源综合利用率, 两项加数分别是全网的链路利用率和功能部署率,  $\alpha$ 和 $\beta$ 是相应的权重参数, 满足 $\alpha + \beta = 1$ , 可以根据实际情况中二者的重要程度进行设定。式(2)要求服务路径提供的功能顺序与服务功能链顺序一致, 式(3)和式(4)分别表示每个节点能提供的功能数限制和每条链路的带宽限制。

文献[18]中已经证明, 求解该多约束优化问题是一个NP-hard问题, 为此本文提出了一种启发式算法, 逐个考虑网络中每一条服务请求的部署方式, 在控制路径长度的同时逐步搜索满足服务请求且功能复用度最高的传输路径, 同步完成路径选择和功能部署, 避免额外的路径带宽, 有效提高网络服务能力。

### 3 基于最长有效功能序列的服务功能链部署算法

本算法逐个考虑网络中每一条服务功能链的部署方式, 在部署第*i*条服务功能链 $S_i$ 时, 已经有*i*-1条服务功能链部署在网络中, 因此网络中的各条链路带宽和各节点功能部署情况也要随时更新, 作为后续服务功能链的部署条件。 $S_i$ 中的任意功能 $s_{ij}$ 既可以复用网络中已经部署的功能, 也可以部署在任意有能力提供功能的节点上。然而, 提高功能复用度和减少传输跳数之间存在着矛盾, 为了联合优化功能部署和带宽需求, 需要在功能复用和路径长度之间找到一个最优的平衡点。因此, 本算法将第*i*条服务功能链的部署问题分解为两个问题: (1)设置传输跳数最大值 $u_i$ , (2)在 $u_i$ 范围内找到复用度最高的有效路径。

传输跳数最大值 $u_i$ 应遵循以下原则:

(1)为了保证服务路径可以连通源节点和目的节点,  $u_i$ 应不小于该收发节点间的最短路径, 即 $u_i \geq |P_{o_i, t_i}|$ , 其中 $P_{o_i, t_i}$ 表示源节点 $o_i$ 和目的节点 $t_i$ 间的最短路径,  $|P_{o_i, t_i}|$ 表示该路径的长度。

(2)为了保证在无复用功能情况下服务路径可以支持 $S_i$ 中的所有功能, 路径上可以部署新功能的功能数应不小于 $S_i$ 中的功能个数。

(3)为了允许功能复用, 应该根据复用度对 $u_i$ 适当修正, 每复用一个功能, 根据式(1)可知优化目标第2项降低了 $\beta/|V|C_V$ , 为了保证优化目标值不变大, 应保证 $\Delta u_i \leq (\beta|E|C_E)/(\alpha b_i|V|C_V)$ 。

综上, 可以得到 $u_i$ 的表达式

$$u_i = |P_{o_i, t_i}| + \max(w_{P_{o_i, t_i}}(|S_i| - m_{P_{o_i, t_i}}), 0) + \Delta u_i \quad (5)$$

其中,  $|S_i|$ 表示 $S_i$ 中的功能个数,  $m_{P_{o_i, t_i}}$ 表示最短路径 $P_{o_i, t_i}$ 上可以部署的功能个数,  $w_{P_{o_i, t_i}}$ 表示 $P_{o_i, t_i}$ 之外可部署功能的节点到 $P_{o_i, t_i}$ 的平均距离,  $\Delta u_i$ 则应根据复用度逐步修正。

在设置传输跳数最大值 $u_i$ 后, 需要搜索 $u_i$ 范围内复用度最高的有效路径, 尽管对传输跳数进行了限制, 逐条路径遍历搜索的复杂度仍然较高。因此, 本算法将路径搜索分割为节点搜索, 从源节点出发搜索第1个中继节点, 再从该节点出发搜索下一个中继节点, 以此类推直至子路径上的功能与服务功能链完全匹配。本文借鉴最长公共子序列<sup>[17]</sup>(Longest Common Sub-sequence, LCS)的思想, 提出了最长有效功能序列(Longest Effective Function Sequence, LEFS)的概念, 将服务功能链与路径上的功能序列(包括空白功能位置)进行匹配, 计算该路径从首项功能开始能够连续支持的最长序列, 作为选择中继节点和子路径的依据。

**定义 1** 最长有效功能序列 给定一个服务功能链 $S$ , 和待匹配路径上的功能序列 $\varphi$ , 对于 $S$ 中从首项元素开始的一个连续子序列, 如果在 $\varphi$ 中的空白功能位置部署适当功能后, 使得该序列成为 $\varphi$ 的子序列, 则称该序列为有效功能序列, 符合此条件的最长的序列称为最长有效功能序列。

为了更加清晰地说明最长有效功能序列和最长公共子序列的区别, 举一个简单的例子: 如图2所示,  $S = \{f_1, f_2, f_3, f_4\}$ 表示服务功能链,  $\varphi = \{f_2, f_3, f_4, f_1, 0, f_3\}$ 表示路径上的功能序列, 其中0表示空白功能位置。虽然 $S$ 和 $\varphi$ 的最长相同子序列是 $\{f_2, f_3, f_4\}$ , 但是该序列无法提供功能 $f_1$ 的部署位置。按照定义1, 该路径的有效功能序列为 $\{f_1, f_2, f_3\}$ , 其中功能 $f_2$ 需要部署在 $\varphi$ 中的空白功能位置上。

根据最长有效功能序列选择中继节点和子路径有两个优点, 一是匹配目标为服务功能链中从首项

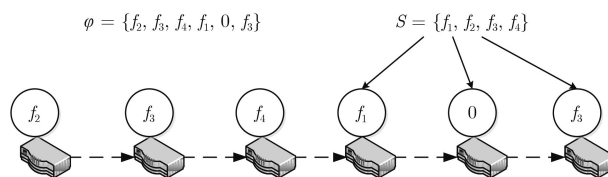


图2 最长有效功能序列示意图

功能开始的连续子序列，在搜索后续中继节点时不需要对已确定路径进行修改，确保每次节点选择的有效性；二是匹配过程中考虑每条备选路径上的尚未部署功能的位置，这些空白功能位置可以与服务功能链中的任意功能匹配，从而在最终传输路径上预留剩余功能部署位置，避免空白功能位置不足带来的额外带宽损耗。

与最长相同子序列类似，最长有效功能序列可以通过动态规划算法实现快速求解。表1中算法1给出了求解路径 $z$ 与服务功能链 $S$ 相匹配的最长有效

功能序列、功能复用度和剩余功能部署方式的函数伪代码，其中第1行提取路径上的功能序列，第2~15行采用动态规划算法计算有效功能序列和相应的功能复用度，第16~32行利用回溯算法计算剩余功能的部署位置。根据动态规划算法流程，该函数的计算复杂度可表示为 $T_1 = O(|S| \times |\varphi|)$ ，由式(5)可知，待匹配的路径长度与服务功能链长度 $|S|$ 线性相关，而路径上的功能序列长度 $|\varphi|$ 是路径长度和每个节点能够提供功能个数 $C_V$ 的乘积，因此LEFS函数的复杂度为 $T_1 = O(C_V |S|^2)$ 。

根据最长有效功能序列选择中继节点的原则如下：

(1)最终路径跳数不超过传输跳数最大值 $u_i$ ，即新生成子路径与中继节点到目的节点的最短路径跳数之和不大于 $u_i$ ，如式(6)所示，其中 $|z_k|$ 表示第 $k$ 次搜索的子路径的跳数， $|P_{v_k, t_i}|$ 表示中继节点 $v_k$ 到目的节点 $t_i$ 间最短路径的跳数。

$$|z_k| + |P_{v_k, t_i}| \leq u_i \quad (6)$$

(2)和已确定路径相比，新生成子路径的最长有效功能序列长度更大，如式(7)所示，其中 $\text{LEFS}(z_k, S_i) \cdot \text{le}$ 表示通过LEFS函数计算的第 $k$ 次搜索的子路径 $z_k$ 与服务功能链 $S_i$ 相匹配的最长有效功能序列长度， $\text{LEFS}(z_{k-1}, S_i) \cdot \text{le}$ 表示已确定子路径 $z_{k-1}$ 对应的最长有效功能序列长度。

$$\text{LEFS}(z_k, S_i) \cdot \text{le} > \text{LEFS}(z_{k-1}, S_i) \cdot \text{le} \quad (7)$$

(3)如果网络中存在不止一个节点满足上述条件，对于多个可选节点，应选择目标函数增益最小的新路径。根据式(1)可知，传输路径每增加一跳，优化目标第1项增大 $\alpha b_i / |E| C_E$ ，每复用一个功能，优化目标第2项降低了 $\beta / |V| C_V$ ，第 $i$ 条服务请求第 $k$ 次中继的目标函数增益 $g_{ik}$ 可表示为式(8)，其中 $\text{LEFS}(z_k, S_i) \cdot \text{re}$ 表示通过LEFS函数计算的第 $k$ 次搜索的子路径 $z_k$ 与服务功能链 $S_i$ 相匹配的功能复用度。

$$g_{ik} = \alpha \frac{|z_k| b_i}{|E| C_E} - \beta \frac{\text{LEFS}(z_k, S_i) \cdot \text{re}}{|V| C_V} \quad (8)$$

综上所述，基于最长有效功能序列的服务功能链部署算法(表2中算法2)流程如下：

第1步 将中继节点初始化为源节点，将备选节点集合初始化为源节点外的全部节点；(第2行)

第2步 若备选节点集合中所有节点均选择过，执行第5步，否则选择备选节点集合中的一个节点，生成该节点到中继节点的最短路径，与子路径相连构成临时路径；(第5、第6行)

第3步 调用LEFS函数计算临时路径与服务功能链相匹配的最长有效功能序列，若该序列长度不

表1 算法1：最长有效功能序列函数

算法1：最长有效功能序列函数(LEFS( $z, S$ ))

```

(1)  $\varphi = \text{functions along } z$ 
(2) for  $i = 1$  to  $|\varphi|$ 
(3)   for  $j = 1$  to  $|\varphi|$ 
(4)     if  $S_i = \varphi_j$ 
(5)        $\theta_{ij} = \theta_{i-1, j-1} + 1, \omega_{ij} = \omega_{i-1, j-1} + 1$ 
(6)     else if  $\varphi_j = 0$ 
(7)        $\theta_{ij} = \theta_{i-1, j-1} + 1, \omega_{ij} = \omega_{ij-1}$ 
(8)     else  $\theta_{ij} = \theta_{ij-1}, \omega_{ij} = \omega_{ij-1}$ 
(9)     end if
(10)  end for
(11)  if  $\theta_{i|\varphi|} < i$ 
(12)     $i = i - 1, \text{break}$ 
(13)  end if
(14) end for
(15)  $\text{le} = \theta_{i|\varphi|}, \text{re} = \omega_{i|\varphi|} // \text{le}$ : 最长有效功能序列长度;  $\text{re}$ : 功能复用度
(16)  $\text{flag} = |\varphi|$ 
(17) for  $k = i$  to 1
(18)   for  $j = \text{flag}$  to 1
(19)     if  $S_k = \varphi_j$  and  $\theta_{kj} = k$  then
(20)        $\text{flag} = j, \text{break}$ 
(21)     end if
(22)  end for
(23) if  $j < k$ 
(24)   for  $j = \text{flag}$  to 1
(25)     if  $\varphi_{2j} = 0$ 
(26)       break
(27)     end if
(28)  end for
(29)   $\text{flag} = j, \text{de}(S_k) = \varphi_j // \text{de}$ : 剩余功能部署位置
(30) end if
(31) end for
(32) return  $\text{le}, \text{re}$  and  $\text{de}$ 

```

表2 算法2: 基于最长有效功能序列的服务功能链部署

算法2 基于最长有效功能序列的服务功能链部署

- (1) for each service request  $r_i$
- (2)  $v_0 = o_i, z_0 = \emptyset, k = 0$
- (3) while LEFS( $z_k, S_i$ ).le <  $|S_i|$
- (4)  $k = k + 1, h_k = \emptyset$  //  $h_k$ : 备选路径集合
- (5) for  $v_k \in V$  and  $v_k \notin z_{k-1}$
- (6)  $z_k = z_{k-1} + P_{v_{k-1}, v_k}$
- (7) calculate LEFS( $z_k, S_i$ ) according to algorithm 1
- (8) calculate  $u_i$  according to Eq. (5)
- (9) if  $|z_k| + |P_{v_k, t_i}| \leq u_i$  and  
LEFS( $z_k, S_i$ ).le > LEFS( $z_{k-1}, S_i$ ).le
- (10) put  $z_k$  into  $h_k$
- (11) end if
- (12) end for
- (13) select  $z_k$  with the maximum  $g_{ik}$  in  $h_k$  according to Eq. (8)
- (14) update  $v_k$  and  $z_k$
- (15) end while
- (16)  $z = z_k + P_{v_k, t_i}$ , deploy functions according to LEFS( $z_k, S_i$ ).de, and update  $C_E$  and  $C_V$
- (17) end for

大于当前子路径的最长有效功能序列, 执行第2步; (第7、第9行)

第4步 计算传输跳数最大值, 若超过临时路径与所选节点到目的节点的最短路径跳数之和满足此限制条件, 将该路径加入备选路径集合并执行第2步, 否则直接执行第2步; (第8—第10行)

第5步 比较备选路径集合中每条路径的目标函数增益, 选择目标函数增益最小的路径, 更新子路径和中继节点(第13、第14行);

第6步 判断更新后子路径能否与服务功能链完全匹配, 若匹配, 通过最短路径连接中继节点与目的节点完成传输路径, 按照LEFS函数部署未复用功能, 并更新各条链路带宽和各节点功能部署情况, 若不匹配, 将备选节点更新为子路径以外的全部节点, 执行第2步。(第3、第16行)

表2的算法2给出了上述算法流程的伪代码, 本算法至多需要 $|S_i| \times |V|$ 次循环, 每次循环需调用一次LEFS函数, 因此本算法的计算复杂度为  $T_2 = O(C_V |S_i|^3 |V|)$ 。

## 4 性能仿真与分析

### 4.1 仿真设计

为了对本文提出的服务路径算法性能进行验证, 在2.80 GHz CPU, 4 GB内存的PC机上通过Matlab软件进行了仿真实验并对实验结果处理分

析, 仿真场景设计如下: 由20个节点构成随机网络拓扑按照Waxman-Salam模型<sup>[19]</sup>生成, 每个节点最多可以部署2个虚拟功能, 任意相邻节点间的链路带宽为10 Mbps。网络可提供8种功能, 每个需要部署的服务功能链由任意顺序排列的2~8个互不相同的功能组成。服务请求中的源节点和目的节点同样在网络中随机选取, 每条服务请求需要的传输带宽为1 Mbps。在仿真过程中令 $\alpha = \beta = 0.5$ , 使得目标函数中的带宽利用率和功能部署率重要性相同。为了提高算法评估的精确性, 采用蒙特卡洛方法, 在每个测试场景下都进行了100组仿真测试, 取100组仿真测试的平均值作为测试结果。

在仿真过程中, 将本算法(LEFS)与其他两种服务功能链部署算法进行了对比, 分别是文献[17]提出的最长公共子序列算法(LCS Based Algorithm, LBA)和文献[18]提出的链路功能联合优化算法(Link and Server Usage, LSU)。其中最长公共子序列算法预先计算收发节点间的一组最短路径, 计算每条路径上已经部署的功能序列和服务功能链的最长公共子序列, 选择其中功能复用度最高的路径, 最后在路径上部署剩余功能; 链路功能联合优化算法分析功能复用和传输跳数的关系, 估计最优路径的跳数, 逐步选择功能复用节点逼近最优跳数, 最后将剩余功能部署在相应的功能复用节点之间。在仿真的过程中, 每部署一条服务功能链, 3种算法均需要更新网络剩余资源作为下一条服务请求相应的参数, 如果该算法不能为某条服务请求找到合适的部署方式, 则拒绝该服务请求。

### 4.2 相关算法性能比较

为了验证本算法的性能, 仿真对比了本算法和上述两种服务功能链部署算法在功能部署率、带宽利用率、网络资源综合利用率、可支持服务请求数和平均计算时间等方面的性能。图3和图4给出了服务功能链长度为5时, 功能部署率和带宽利用率随服务请求数变化的仿真结果, 可以看出, 3种算法对功能部署率的优化性能相差无几, 但是由于LBA算法和LSU算法在选择路径的过程中都没有考虑剩余功能的部署位置, 可能需要在所选路径的基础上扩展新的路径来部署未匹配功能, 因此带宽利用率较高, 且随着服务请求的增多, 网络中空闲功能位置逐渐减少, LBA算法和LSU算法需要扩展更长的路径连接可部署新功能的节点, 因此带宽利用率快速增大。

网络资源综合利用率随服务请求变化的仿真结果如图5所示, 由于本算法同步完成路径选择和功能部署, 不需要额外的带宽损耗, 在3种算法的功

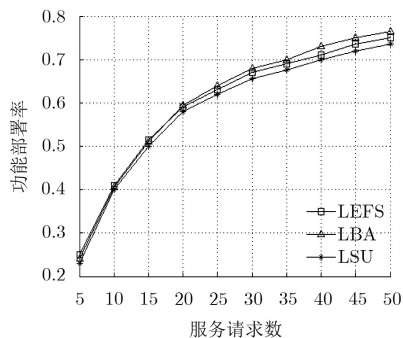


图3 各算法功能部署率

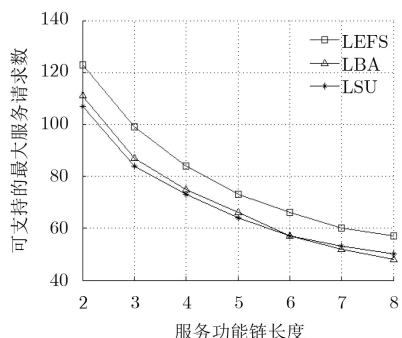


图6 各算法可支持的服务请求数

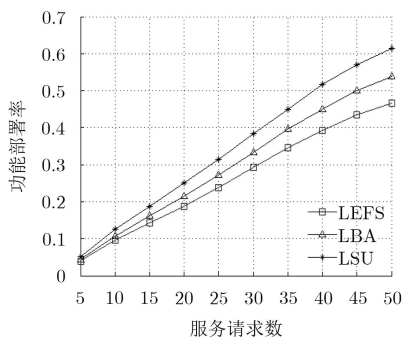


图4 各算法带宽利用率

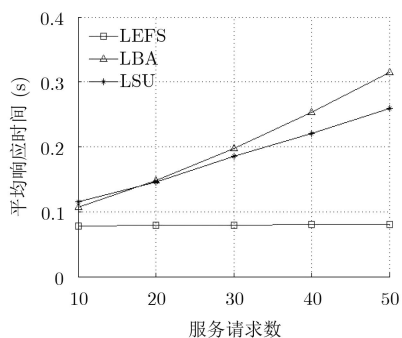


图7 各算法平均响应时间

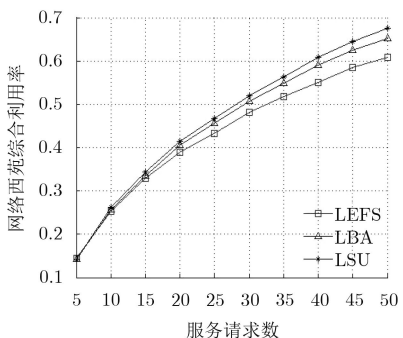


图5 各算法网络资源综合利用率

能部署率非常接近的情况下，本算法的带宽利用率低于LBA算法和LSU算法，根据式(1)，本算法的网络资源综合利用率明显优于其他两种算法，当服务请求数为50时，与性能较好的LBA算法相比，本算法网络资源综合利用率降低了10%。图6给出了3种算法可支持的服务请求数随服务功能链长度变化的仿真结果，LBA和LSU算法可以支持的服务请求数接近，而由于在部署过程中减少了网络资源的利用，在相同的网络资源条件下，本算法可以支持更多的服务请求。

为了验证本算法的计算复杂度，图7给出了服务功能链长度为5时上述各算法平均每条服务请求的响应时间，其中LSU和LBA算法的计算时间随着服务请求数线性增大，因为随着网络中空白功能位置的减少，未复用功能的部署所需的计算时间逐渐

增大，且LBA算法需要逐渐扩大收发节点间的最短路径集合来寻找满足服务请求条件的路径，因此响应时间的增长幅度更大。而LEFS算法的计算时间不随服务请求数变化，符合算法2中给出的计算复杂度只与网络节点数和服务功能链长度相关的特点。与其他两种算法相比，本算法响应时间更短且不随网络状态变化，可以支持实时服务功能链的部署。

### 5 结束语

本文针对资源有限的网络中的服务功能链部署问题，提出了一种启发式的服务功能链部署算法，以功能复用和带宽需求联合优化为目标，逐个考虑网络中每一条服务请求的部署方式。该算法将路径搜索分解为节点搜索，逐步搜索中继节点来寻找满足服务请求的最优路径，并提出了最长有效功能序列的概念，在选择子路径的过程中预留剩余功能的部署位置。该算法可同步完成路径选择和功能部署，同时优化网络的功能部署率和链路利用率，与其他算法相比，降低了网络资源利用率，可以支持更多服务请求，且能够实现对服务请求的快速响应。

### 参考文献

[1] 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](https://doi.org/10.1109/MCOM.2015.7045396).
- [2] QUINN P and GUICHARD J. Service function chaining: Creating a service plane via network service headers[J]. *Computer*, 2014, 47(11): 38–44. doi: [10.1109/MC.2014.328](https://doi.org/10.1109/MC.2014.328).
- [3] LI Yong, ZHENG Feng, CHEN Min, *et al.* A unified control and optimization framework for dynamical service chaining in software-defined NFV system[J]. *IEEE Wireless Communications*, 2015, 22(6): 15–23. doi: [10.1109/MWC.2015.7368820](https://doi.org/10.1109/MWC.2015.7368820).
- [4] COHEN R, LEWIN-EYTAN L, NAOR J S, *et al.* Near optimal placement of virtual network functions[C]. 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 2015: 1346–1354. doi: [10.1109/INFOCOM.2015.7218511](https://doi.org/10.1109/INFOCOM.2015.7218511).
- [5] 李丹, 兰巨龙, 王鹏, 等. 一种集中调控的分布式服务路径选择算法[J]. 电子与信息学报, 2018, 40(4): 785–793. doi: [10.11999/JEIT170600](https://doi.org/10.11999/JEIT170600).
- LI Dan, LAN Julong, WANG Peng, *et al.* Distributed service path selection algorithm under central control[J]. *Journal of Electronics & Information Technology*, 2018, 40(4): 785–793. doi: [10.11999/JEIT170600](https://doi.org/10.11999/JEIT170600).
- [6] 梁宁宇, 兰巨龙, 张岩. 基于分布式选择探测算法的服务路由机制[J]. 电子学报, 2017, 45(7): 1545–1552. doi: [10.3969/j.issn.0372-2112.2017.07.001](https://doi.org/10.3969/j.issn.0372-2112.2017.07.001).
- LIANG Ningning, LAN Julong, and ZHANG Yan. A service routing mechanism based on the distributed selection probing algorithm[J]. *Acta Electronica Sinica*, 2017, 45(7): 1545–1552. doi: [10.3969/j.issn.0372-2112.2017.07.001](https://doi.org/10.3969/j.issn.0372-2112.2017.07.001).
- [7] XIA Ming, SHIRAZIPOUR M, ZHANG Ying, *et al.* Network function placement for NFV chaining in packet/optical datacenters[J]. *Journal of Lightwave Technology*, 2015, 33(8): 1565–1570. doi: [10.1109/JLT.2015.2388585](https://doi.org/10.1109/JLT.2015.2388585).
- [8] BARI M F, CHOWDHURY S R, AHMED R, *et al.* Orchestrating virtualized network functions[J]. *IEEE Transactions on Network & Service Management*, 2016, 13(4): 725–739. doi: [10.1109/TNSM.2016.2569020](https://doi.org/10.1109/TNSM.2016.2569020).
- [9] BARI M F, CHOWDHURY S R, AHMED R, *et al.* On orchestrating virtual network functions[C]. 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain, 2015: 50–56. doi: [10.1109/CNSM.2015.7367338](https://doi.org/10.1109/CNSM.2015.7367338).
- [10] KLINKOWSKI M A and WALKOWIAK K. On the advantages of elastic optical networks for provisioning of cloud computing traffic[J]. *IEEE Network*, 2013, 27(6): 44–51. doi: [10.1109/MNET.2013.6678926](https://doi.org/10.1109/MNET.2013.6678926).
- [11] ZHANG Liang and ZHU Zuqing. Spectrum-efficient anycast in elastic optical inter-datacenter networks[J]. *Optical Switching & Networking*, 2014, 14(4): 250–259. doi: [10.1016/j.osn.2014.05.018](https://doi.org/10.1016/j.osn.2014.05.018).
- [12] XIE Lijun, JIANG Yiming, WANG Binqiang, *et al.* An approach for network function combination based on least busy placement algorithm[J]. *China Communications*, 2016, 13(Sup): 167–176. doi: [10.1109/CC.0.7560888](https://doi.org/10.1109/CC.0.7560888).
- [13] MEHRAGHDAM S, KELLER M, and KARL H. Specifying and placing chains of virtual network functions[C]. 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, 2014: 7–13. doi: [10.1109/CloudNet.2014.6968961](https://doi.org/10.1109/CloudNet.2014.6968961).
- [14] LONG Qu, ASSI C, SHABAN K, *et al.* A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks[J]. *IEEE Transactions on Network & Service Management*, 2017, 14(3): 554–568. doi: [10.1109/TNSM.2017.2723090](https://doi.org/10.1109/TNSM.2017.2723090).
- [15] MECHTRI M, GHRIBI C, and ZEGHLACHE D. A scalable algorithm for the placement of service function chains[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(3): 533–546. doi: [10.1109/TNSM.2016.2598068](https://doi.org/10.1109/TNSM.2016.2598068).
- [16] YANG Ke, ZHANG Hong, and HONG Peilin. Energy-aware service function placement for service function chaining in data centers[C]. 2016 IEEE Global Communications Conference (GLOBECOM), Washington, D.C., USA, 2016: 1–6. doi: [10.1109/GLOCOM.2016.7841805](https://doi.org/10.1109/GLOCOM.2016.7841805).
- [17] FANG Wenjian, ZENG Menglu, LIU Xiahe, *et al.* Joint spectrum and IT resource allocation for efficient vNF service chaining in inter-datacenter elastic optical networks[J]. *IEEE Communications Letters*, 2016, 20(8): 1539–1542. doi: [10.1109/LCOMM.2016.2580151](https://doi.org/10.1109/LCOMM.2016.2580151).
- [18] KUO Tungwei, LIOU Bangheng, LIN Kate Chingju, *et al.* Deploying chains of virtual network functions: On the relation between link and server usage[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(4): 1562–1576. doi: [10.1109/TNET.2018.2842798](https://doi.org/10.1109/TNET.2018.2842798).
- [19] SALAMA H F. Multicast Routing for Real-time Communication of High-speed Networks[M]. Raleigh, USA, North Carolina State University, 1996: 198.
- 李丹: 男, 1989年生, 博士生, 研究方向为新型网络体系结构。  
 兰巨龙: 男, 1962年生, 教授, 博士生导师, 主要研究方向为网络体系结构、信息安全。  
 王鹏: 男, 1985年生, 博士, 主要研究方向为新型网络体系结构、路由技术。  
 胡宇翔: 男, 1982年生, 博士, 主要研究方向为新型网络体系结构、网络安全。