基于延迟和抖动感知的多播服务功能树嵌入算法

刘 亮*[①] 陈 翔[①] 桂晓菁[②] 徐勇军[①] 杜娅荣[①] 侯泽天[①] 段 洁[①]

[①](重庆邮电大学通信与信息工程学院 重庆 400065)

[②](重庆邮电大学自动化学院 重庆 400065)

摘 要：针对软件定义网络/网络功能虚拟化(SDN/NFV)架构中，多播请求流(MRs)需满足严格时延和抖动约束下遍历由多个虚拟网络功能(VNFs)依序组成的服务功能树(SFT)问题。该文提出一种基于最优链路选择函数进行深度优先搜索构建SFT的路由算法。首先，提出网络资源相对成本函数，以保证网络负载自动均衡。其次，联合考虑网络资源、VNF动态放置及多播流延迟和抖动约束，构建SFT动态嵌入问题的整数线性规划模型(ILP)。最后，针对该NP难问题，设计辅助边权图和最优链路选择函数进行路由路径选择，并以最小化资源消耗成本为目标提出具有延迟和抖动感知的SFT嵌入算法(SFT-EA)。仿真结果表明，SFT-EA在吞吐量，流接受率和网络负载均衡方面具有更好的性能。

关键词：网络功能虚拟化；服务功能树；多播；延迟和抖动

中图分类号：TN915; TP393 **文献标识码**：A **文章编号**：1009-5896(2024)01-0184-11

DOI: 10.11999/JEIT230015

Multicast Service Function Tree Embedding Algorithm Based on Delay and Jitter Awareness

LIU Liang[①] CHEN Xiang[①] GUI Xiaojing[②] XU Yongjun[①] DU Yarong[①]
HOU Zetian[①] DUAN Jie[①]

[①](*School of Communication and Information Engineering*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

[②](*School of Automation*, *Chongqing University of Posts and Telecommunications*, *Chongqing* 400065, *China*)

Abstract: To solve the problem that Multicast Request flows (MRs) need to traverse sequentially a Service Function Tree (SFT) consisting of Virtual Network Functions (VNFs) as well as ensuring stringent delay and jitter constraints of SFT in Network Function Virtualization (NFV)-enabled Software-Defined Networks (SDNs), a routing algorithm for constructing a multicast SFT based on depth-first search with an optimal link selection function is proposed. Firstly, the relative cost functions of network resources are proposed to guarantee the automatic load balancing of the network. Secondly, an Integer Linear Programming model (ILP) for the SFT dynamic embedding is constructed by jointly considering network resources, VNF dynamic placement and delay and jitter constraints of a multicast flow. Finally, for this NP-hard problem, an auxiliary edge-weight graph and optimal link selection function are designed for routing path selection, and a delay and jitter-aware SFT Embedding Algorithm (SFT-EA) is proposed with the objective of minimizing the resource consumption cost. Simulation results demonstrate the SFT-EA has better performance in terms of throughput, traffic acceptance rate, and network load balance.

Key words: Network Function Virtualization (NFV); Service Function Tree (SFT); Multicast; Delay and Jitter

收稿日期：2023-01-12；改回日期：2023-08-07；网络出版：2023-08-19

*通信作者： 刘亮 liuliang@cqupt.edu.cn

基金项目：国家自然科学基金(62171070, 61701058)，重庆邮电大学博士启动基金(A2023007)

Foundation Items: The National Natural Science Foundation of China (62171070, 61701058),Chongqing University of Posts and Telecommunications Doctoral Initiation Foundation (A2023007)

1　引 言

在支持网络功能虚拟化(Network Function Virtualization, NFV)的软件定义网络(Software Defined Networks, SDNs)中，为实现可靠、安全及可扩展传输等目的，用户请求流通常需要依序遍历由多个虚拟网络功能(Virtual Network Functions, VNFs)组成的服务功能链(Service Function Chain, SFC)[1]。对于单播请求流， SFC可以直接沿着单播路由路径依次选择嵌入节点进行VNF部署，目前已有大量文献对这一工作进行了研究，特别是文献[2]利用拉格朗日对偶理论[3]设计松弛算法求解了具有多资源多QoS约束的单播SFC嵌入问题，具有典型的代表性。多播通过共享的多播树进行多路复用，与单播相比，可以减少骨干网50%以上的带宽消耗[4]。2022年至2028年，全球视频流媒体市场预计将以21.0%的复合年增长率扩张[5]，这些视频流大多以多播的形式向用户提供服务。在SDN/NFV架构中具有SFC需求的多播请求流(Multicast Request flows, MRs)是将流量从源点转发到多个目的节点，每个源-目的对之间需要根据SFC的要求依序遍历多个VNFs实例，这将形成一个虚拟网络功能转发图[6]。图1为该文构建的一条多播流从源点s依序经过由防火墙(FW)、入侵检测系统(IDS)、入侵阻止系统(IPS)组成的3条相同SFC后到达各个目的节点的示意。显然这3条SFC共同组成了一颗多播服务功能树(Service Function Tree, SFT)。联合考虑网络中各种资源及QoS约束条件，为多播请求流嵌入SFC是一个巨大的挑战。

目前，在SDN/NFV架构中，研究多播SFT嵌入的文献比较少。文献[7-9]研究了在已经放置了多个VNF实例环境中MRs的路由问题，但文献[7]没有考虑路由过程中VNF的动态部署，文献[8]考虑了VNF的运行和动态部署成本，但没有联合考虑网络中交换节点的转发成本，并且，它们都以多播源-目的对之间的单条SFC进行嵌入。文献[9]重点关注多播业务链的迁移调整，文献[10]则基于预测的方式对多播流的SFC进行动态嵌入，上述工作没有基于多播服务功能树(SFT)对SFC进行灵活整体嵌入，会造成网络资源分配不均衡。文献[11,12]研究了SFT的动态编排和整体嵌入，但文献[11]假设所有VNFs都部署在一个NFV功能节点中，文献[12]则要求多播分发点只能在部署的最后一个VNF节点之后进行，限制了SFT嵌入的灵活性和可扩展性，从而降低了网络资源的使用效率。

此外，对于视频会议、多人游戏等实时性要求较高的多播应用，对时延和抖动有严格要求。文献[13-17]研究了SFT的灵活嵌入和部署问题，但文献[13,14]没有考虑多播延迟和抖动约束，文献[15-17]考虑了端到端的延迟约束，但没有考虑多播目的节点之间的抖动约束，上述工作会影响多播用户的服务体验质量。文献[18]考虑用户的移动性及加入退出机制，研究了NFV网络中多播SFC的嵌入，调整和扩展问题。文献[19]是在无线Mesh网络中，考虑无线信号干扰和资源受限的情况下，以最小化链路成本为目标研究多播服务功能树的嵌入问题。上述工作同样没有考虑多播SFT时延和抖动约束，且和本文的研究场景不同。

综上所述，当前还没有相关工作联合考虑多VNFs实例环境下VNF的动态放置、网络资源约束及网络负载均衡等因素，研究具有严格时延和抖动感知的SFT灵活嵌入和多播路由算法。因此，本文在已有研究基础上，提出一种基于最优链路选择函数进行深度优先搜索构建具有严格时延和抖动约束的多播SFT嵌入算法。具体内容为：

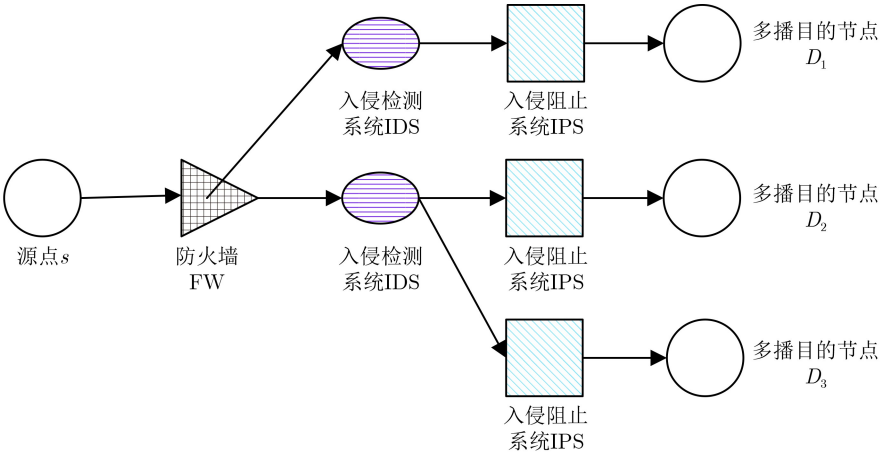

图 1 多播服务功能树

(1)联合考虑VNF动态放置，多资源约束(包括SDN交换机的转发表容量、链路的带宽、功能节点的CPU容量)及网络负载均衡因素正式定义并用ILP模型刻画了延迟和抖动感知的动态SFT嵌入和路由问题(the Delay and Jitter Aware Dynamical SFT embedding and Routing Problem, DJA-DSRP)。(2)设计辅助边权图，将资源消耗及时延指标转化为边权图中的权值降低问题复杂性，提出最优链路选择函数构建SFT嵌入算法(SFT-EA)求解原问题。(3)将提出的SFT-EA算法与现有算法进行性能对比，结果表明SFT-EA不仅较优地解决了多资源及时延和抖动联合约束下的多播SFT嵌入问题，且在流接受率、网络吞吐量和负载平衡方面具有更好的性能。

2 系统模型

2.1 物理网络

将SDN/NFV网络表示为$G=(N,L)$，其中N,L分别表示节点和链路的集合。用$u,v\in N$和$<u,v>\in L$分别表示物理节点和物理链路，为方便起见，后文用$uv\in L$表示物理链路。网络中有两种节点类型，一种是负责转发数据的交换节点$N_s\in N$，另一种是由一台或多台服务器组成的功能节点$N_f\in N$，$N=N_s\cup N_f$。G中有一台SDN控制器，为每条多播请求流执行动态的VNF放置和路由路径选择。图2为网络示例，其中节点v_1,v_3,v_4为功能节点，其余为交换节点。

用MR_i表示第i条具有SFC请求的多播流。其中i为整数。当路由MR_i时，在节点u上，流表容量和剩余流表项的比率用C_u^{ft}和$r_{i,u}^{\mathrm{ft}}$表示。用C_u^{cpu}和$r_{i,u}^{\mathrm{cpu}}$表示节点u上的CPU容量和剩余比率。物理链路$\mathrm{uv}\in L$的带宽容量、剩余带宽比率、传输时延分别用$C_{\mathrm{uv}}^{\mathrm{bw}}$, $r_{i,\mathrm{uv}}^{\mathrm{bw}}$, $\delta_{\mathrm{uv}}^{\mathrm{delay}}$表示。用$\eta(u)$表示节点$u$的邻居节点集合。

$$\eta(u)=\{v|\,\mathrm{uv}\in\mathcal{L}\ \text{或}\ \mathrm{vu}\in\mathcal{L}\} \tag{1}$$

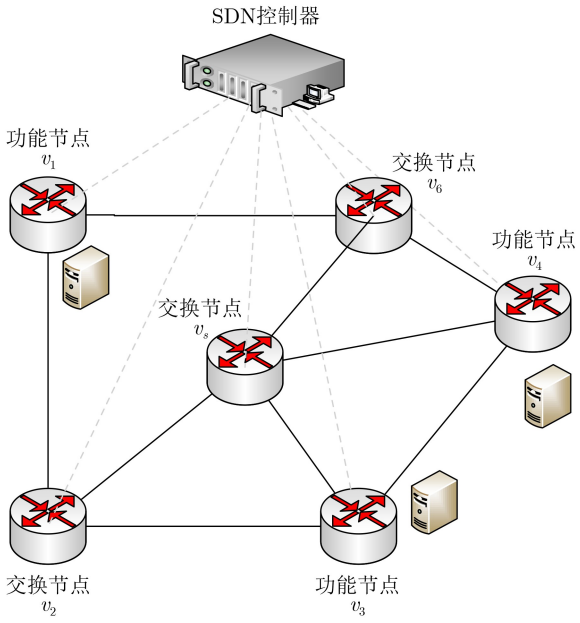

图 2 SDN/NFV网络示例

2.2 虚拟网络功能

网络中可能的VNF类型用集合P表示。每种VNF类型p都有特定的部署成本、CPU需求、处理延迟，分别表示为D_p, k_p^{cpu}, $\delta_p^{\mathrm{delay}}$。具有SFC要求的多播流请求$\mathrm{MR}_i$表示为

$$\mathrm{MR}_i=\left\langle s_i,D_i,\mathrm{SC}_i,R_i^{\mathrm{bw}},R_i^{\mathrm{cpu}},R_i^{\mathrm{delay}},R_i^{\mathrm{Jitter}}\right\rangle \tag{2}$$

s_i和D_i分别表示源节点和目的节点集，$\mathrm{SC}_i=\{\mathrm{SC}_i(1),\mathrm{SC}_i(2),\cdots,\mathrm{SC}_i(l)\}$表示$\mathrm{MR}_i$中每个源目的对必须升序遍历的VNF 集合，即SFC，$l=|\mathrm{SC}_i|$为SFC的长度，即MR_i中每个源目的对需要经过的VNF实例总数。其中R_i^{bw}, R_i^{cpu}, R_i^{Delay}, R_i^{Jitter}分别表示带宽需求、CPU需求、任意源目的对节点的最大容忍延迟和最大容忍抖动。

为处理顺序约束，使用有向无环服务功能图$\bar{G}_i=(\bar{N}_i,\bar{L}_i)$来描述有序的VNF序列$\mathrm{SC}_i$。其中$\bar{N}_i$表示入口节点、VNF节点、出口节点集合，$\bar{L}_i$表示这些节点之间的路径，分别用$\bar{u},\bar{v}\subset\overline{N}_i$, $\bar{u}\ \bar{v}\subset\overline{L}_i$表示$\bar{G}_i$上的两个节点$\bar{u}$和$\bar{v}$以及它们之间的链路。图1展示了一颗多播服务功能树，在到达目的节点之前，多播流必须遍历由3个VNFs 组成的服务链：即SFT 中的FW, IDS 和IPS。

定义$\eta_i(\bar{u}_1)$来表示$\bar{u}_1\in\bar{N}_i$的邻居节点。

$$\eta_i(\bar{u}_1)=\left\{\bar{u}_2|\bar{u}_1\bar{u}_2\in\overline{L}_i\ \text{或}\ \bar{u}_2\bar{u}_1\in\overline{L}_i\right\},\bar{u}_1,\bar{u}_2\in\overline{N}_i \tag{3}$$

2.3 问题定义

在SDN/NFV中，延迟和抖动感知的多播服务功能树动态嵌入和路由问题(DJA-DSRP) 是为MR_i寻找或构建一颗多播树SFT，满足各链路带宽，流表容量和CPU资源约束的同时，任何一条源目的节点对的端到端延迟和任意源目的对路径之间的最大时延抖动分别不大于给定的阈值R_i^{delay}和R_i^{Jitter}，且多播树的实现代价最小并可以自动保证网络的负载均衡。

2.4 问题刻画

2.4.1 物理网络转换

在实际网络中，各功能节点允许部署的VNF类型可能不同。为了方便地表达这个约束并降低问题的复杂性，将通过枚举的方式将原来的物理网络

转换成一个扩展的伪网络。即根据一种类型的VNFp的CPU需求和功能节点的CPU容量，列举所有可以部署在每个功能节点N_{f}上的VNF。这些枚举的VNF称为伪VNF，它们的集合记作M。用$m \in M$表示第m个VNF，用$r_{i,m}^{\mathrm{cpu}}$表示路由MR_i时，第m个VNF上剩余CPU的比率。假如VNF m的类型为p时，定义函数$\tau(m)$返回VNFm的类型

$$\tau(m) = p \tag{4}$$

2.4.2 相对成本的定义

G中资源使用的一个重要特征是，边际成本会随着资源负载的增加而非线性膨胀。比如，与轻负载链路相比，重负载链路将花费更多的成本来处理传入的网络数据包[2]。为了描述这一特性且自动保证网络资源负载均衡，利用相对成本来刻画资源的使用成本并将链路带宽、交换节点和VNF上CPU资源的相对代价分别定义为

$$v_{i,\mathrm{uv}}^{\mathrm{bw}} = (\omega_{\mathrm{bw}})^{1-r_{i,\mathrm{uv}}^{\mathrm{bw}}}, \mathrm{uv} \in \mathcal{L} \tag{5}$$

$$v_{i,u}^{\mathrm{ft}} = (\omega_{\mathrm{ft}})^{1-r_{i,u}^{\mathrm{ft}}}, u \in \mathcal{N}_s \tag{6}$$

$$v_{i,m}^{\mathrm{cpu}} = (\omega_{\mathrm{cpu}})^{1-r_{i,m}^{\mathrm{cpu}}}, m \in \mathcal{M} \tag{7}$$

其中，ω_{bw}, ω_{ft}和ω_{cpu}均为大于1的常数，$r_{\mathrm{uv}}^{\mathrm{bw}}$,$r_u^{\mathrm{ft}}$和$r_m^{\mathrm{cpu}}$分别为链路上带宽资源、交换节点上流表资源和VNF上CPU资源的剩余率。

2.4.3 ILP模型

确保在部署新的VNF实例m时不会违反CPU容量限制，则需满足

$$\sum_{m \in SC_i} x_m^u k_m^{\mathrm{cpu}} \le C_u^{\mathrm{cpu}} r_{i,u}^{\mathrm{cpu}}, \forall u \in N_f \tag{8}$$

$$x_m^u y_m^u = 1, \forall u \in N_f, m \in SC_i \tag{9}$$

其中0,1变量$x_m^u = 1$表示VNFm的实例部署在节点u上，否则$x_m^u = 0$，$y_m^u = 1$表示功能节点u上可以部署VNFm，否则$y_m^u = 0$，k_m^{cpu}表示部署VNFm所需的CPU资源量。

为确保交换节点上的流表容量约束，需满足式(10)。用0,1变量$z_{i,d,u}^{\bar{u}\bar{v}} = 1$来表示$\mathrm{MR}_i$到达目的节点$d \in D_i$时，$\bar{u}\bar{v} \in \overline{L_i}$是经过节点$u$的，否则$z_{i,d,u}^{\bar{u}\bar{v}} = 0$。

$$\sum_{\bar{u}\bar{v} \in \overline{L_i}} z_{i,d,u}^{\bar{u}\bar{v}} \le C_u^{\mathrm{ft}} r_{i,u}^{\mathrm{ft}}, \forall u \in N_s \tag{10}$$

为确保不会违反物理链路上的带宽容量约束，式(11)应被满足。其中$R_{i,\mathrm{uv}}^{\bar{u}\bar{v}}$表示$\bar{u}\bar{v}$通过物理链路$uv$的带宽。

$$\sum_{\bar{u}\bar{v} \in \overline{L_i}} R_{i,\mathrm{uv}}^{\bar{u}\bar{v}} \le C_{\mathrm{uv}}^{\mathrm{bw}} r_{i,\mathrm{uv}}^{\mathrm{bw}}, \forall uv \in L \tag{11}$$

值得注意的是，每个源目的对必须由SFT中的一个SFC服务。由于多播的多路复用特性，流在到达目的节点前可能经过相同的VNF 实例。因此，使用式(12)来保证面向每个目的节点的流只访问SFT中同类型的VNF 1次。

$$\sum_{u \in N_f} \theta_{d,u}^m = 1, \forall m \in SC_i, d \in D_i \tag{12}$$

其中$\theta_{d,u}^m$为0,1变量，$\theta_{d,u}^m = 1$表示流向目的节点d的流是在功能节点u上获得 VNFm的服务，否则$\theta_{d,u}^m = 0$。使用以下约束来确保流向每个目的节点的流只能从源节点流出。

$$\theta_{d,S_i} = 1, \forall d \in D_i \tag{13}$$

为了保证流的守恒和严格的顺序要求，需满足

$$\sum_{v \in \eta(u)} z_{i,d,\mathrm{uv}}^{\bar{u}\bar{v}} - \sum_{v \in \eta(u)} z_{i,d,\mathrm{vu}}^{\bar{u}\bar{v}} \ge \theta_{d,u}^{\bar{u}} - \theta_{d,u}^{\bar{v}},$$
$$\forall \bar{u} \in \overline{N}_i, u \in N, d \in D_i \tag{14}$$

其中，0,1变量$z_{i,d,\mathrm{uv}}^{\bar{u}\bar{v}} = 1$表示$\mathrm{MR}_i$到达目的节点$d \in D_i$时，$\bar{u}\bar{v} \in \overline{L_i}$经过物理链路$uv$,否则$z_{i,d,\mathrm{uv}}^{\bar{u}\bar{v}} = 0$。

使用0,1变量$\varphi_{i,\mathrm{uv}}^{\bar{u}\bar{v}} = 1$表示流向每个目的节点的$\mathrm{MR}_i$通过链路$uv$来获得$\bar{u}$和$\bar{v}$的服务，否则$\varphi_{i,\mathrm{uv}}^{\bar{u}\bar{v}} = 0$。因此，需满足约束

$$\varphi_{i,\mathrm{uv}}^{\bar{u}\bar{v}} \ge z_{i,d,\mathrm{uv}}^{\bar{u}\bar{v}}, \forall \bar{u}, \bar{v} \in \overline{N}_i, u, v \in N, d \in D_i \tag{15}$$

用D_i^t来表示MR_i从s_i经过SFT到$t \in D_i$的延迟

$$D_i^t = \sum_{\bar{u}\bar{v} \in \overline{L}_i} \sum_{\mathrm{uv} \in L} d_{i,d,\mathrm{uv}}^{\mathrm{delay}} z_{i,d,\mathrm{uv}}^{\bar{u}\bar{v}} + \sum_{\bar{u} \in \overline{N}_i} d_{i,d,\tau(\bar{u})}^{\mathrm{delay}}, \forall d \in D_i \tag{16}$$

其中 $d_{i,d,\mathrm{uv}}^{\mathrm{delay}}$和 $d_{i,d,\mathrm{t}(\bar{u})}^{\mathrm{delay}}$分别表示从$s_i$到$d$的流在链路uv上的传播延迟和VNF上的处理延迟，函数$\tau(\bar{u})$负责返回$\bar{u}$的VNF类型。以下约束确保$\mathrm{MR_i}$的任何源目的节点之间端到端的延迟满足阈值

$$D_i^t \le R_i^{\mathrm{delay}}, \forall t \in D_i \tag{17}$$

此外，为确保MR_i中任意源目的地对不违反最大时延抖动，需满足

$$|D_i^{t_1} - D_i^{t_2}| \le R_i^{\mathrm{Jitter}}, \forall t_1, t_2 \in D_i, t_1 \ne t_2 \tag{18}$$

综上，构建多播树的成本计算为

$$\sum_{\bar{u} \in \overline{N}_i} \sum_{u \in N_f} u_{\bar{u},u} v_{i,\tau(\bar{u})}^{\mathrm{cpu}} + \sum_{u \in N_s} \sum_{\bar{u}\bar{v} \in \overline{L}_i} \varphi_{i,u}^{\bar{u}\bar{v}} v_{i,u}^{\mathrm{ft}} + \sum_{\bar{u}\bar{v} \in \overline{L}_i} \sum_{\mathrm{uv} \in L} \varphi_{i,\mathrm{uv}}^{\bar{u}\bar{v}} v_{i,\mathrm{uv}}^{\mathrm{bw}} \tag{19}$$

优化问题为

min 式(19)

s.t. Eq.(8)—(15), (17)—(18)

3　启发式优化算法

由于在网络中嵌入多播服务功能树是NP难的[13]，

为降低问题求解复杂度，将功能节点中的CPU资源消耗和转发节点中流表资源消耗转化为边上的权值进行表示。因此，本节将G转化为辅助边权图G_i'，然后再将G_i'转换为具有相对成本的VNF分裂多阶段边权图$\hat{G}_i$，最后基于$\hat{G}_i$利用启发式算法对问题进行求解，具体过程如下。

3.1 辅助边权图构建

辅助边权图$G_i'=(V_i',E_i';w_{i,e'})$的构造：将每个交换节点$v\in N_s$拆分为两个节点，记作$v'$和$v''$，并将它们添加到$V_i'$中。然后向$E_i'$添加一条有向边$\langle v',v''\rangle$。对于功能节点中的每个VNF$m$，将$m$分裂成两个节点$m'$和$m''$，并将它们添加到$V_i'$中，有向边$\langle m',m''\rangle$被添加到边集$E_i'$。为保证一致性，在$m''$和$m'$之间增加一条高速链路，不考虑这条链路的容量和延迟约束。对于每条链路$\mathrm{uv}\in L$，为E_i'添加一条边$\langle u'',v'\rangle$，图3为构造G_i'的示例。

按式(20)和(21)为$e'\in E_i'$分配成本和延迟权重，特别地，假如$e'=\langle m',m''\rangle\in E_i'$，$\hat{y}_m^a=0,y_m^a=1$，表明VNF$m$是新部署的，则$\omega_{i,e'}^{\mathrm{cost}}=(\omega_{\mathrm{cpu}})^{1-r_{i,m}^{\mathrm{cpu}}}+D_{\tau(m)}$，由于VNF$m$在一个功能节点上，将$\langle m'',m'\rangle$的成本权重设置为0。由于软硬件的性能越来越好，将边$\langle v',v''\rangle$的延迟权重设为0。

$$w_{i,e'}^{\mathrm{cost}}=\begin{cases}(w_{\mathrm{ft}})^{1-r_{i,v}^{\mathrm{ft}}}, & e'=\langle v',v''\rangle\in E_i'\\ (w_{\mathrm{bw}})^{1-r_{i,uv}^{\mathrm{bw}}}, & e'=\langle u'',v'\rangle\in E_i'\\ (w_{\mathrm{cpu}})^{1-r_{i,m}^{\mathrm{cpu}}}, & e'=\langle m',m''\rangle\in E_i',\\ & \hat{y}_m^a=1,y_m^a=1\end{cases}\tag{20}$$

$$w_{i,e'}^{\mathrm{delay}}=\begin{cases}0, & e'=\langle',v''\rangle\in E_i'\\ d_{\mathrm{uv}}^{\mathrm{delay}}, & e'=\langle u'',v'\rangle\in E_i'\\ d_{t(m)}^{\mathrm{delay}}, & e'=\langle m',m''\rangle\in E_i'\end{cases}\tag{21}$$

3.2 多阶段边权图构建

为确保MR_i中每条源目的候选路由路径满足预定义的VNF顺序要求，进一步将G_i'转换为一个由入口节点、满足严格顺序约束的候选VNF实例和出口节点组成的有向无环图$\hat{G}_i=\left(\hat{V}_i,\hat{L}_i;w_{i,\hat{e}}\right)$，其中$\hat{u},\hat{v}\in\hat{V}_i$表示两个节点，$\hat{e}=\langle\hat{u},\hat{v}\rangle\in\hat{L}_i$和$w_{i,\hat{e}}$分别表示链路和边的权重。通过以下两个阶段来构造$\hat{G}_i$。

第1阶段是查找MR_i所需的所有VNF节点，并将它们按照预定义的顺序排列并进行分裂。第2阶段是生成链路。首先，把入口节点s_i与$\mathrm{SC}_i(1)'$列中的所有节点连接起来，然后将$\mathrm{SC}_i(1)'$列中的节点$\hat{u}'$与$\mathrm{SC}_i(1)''$列中的节点$\hat{u}''$一一连接。接着将$\mathrm{SC}_i(1)''$列中的每个节点$\hat{u}''$与$\mathrm{SC}_i(2)'$中所有节点$\hat{u}'$一一连接。接下来，依次执行相同的操作。最后，将$\mathrm{SC}_i(l)''$列中所有节点$\hat{u}''$连接到$t\in D_i$。并且D_i中的目的节点是相互连接的。图4为一个示例。

在$\hat{G}_i$中，将每条链路的成本和延迟权重按以下公式进行设置，其中$p_i(u''v')$表示G_i'中两个节点之间的最短路径。

$$w_{i,\hat{e}}^{\mathrm{cost}}=\begin{cases}(w_{\mathrm{cpu}})^{1-r_{i,m}^{\mathrm{cpu}}},\text{if }\hat{e}=\langle\hat{u}',\hat{u}''\rangle\in\hat{L}_i\\ \displaystyle\sum_{e'\in p_i(u''v')}w_{i,e'}^{\mathrm{cost}},\text{if }\hat{e}=\langle\hat{u}'',\hat{v}'\rangle\in\hat{L}_i\end{cases}\tag{22}$$

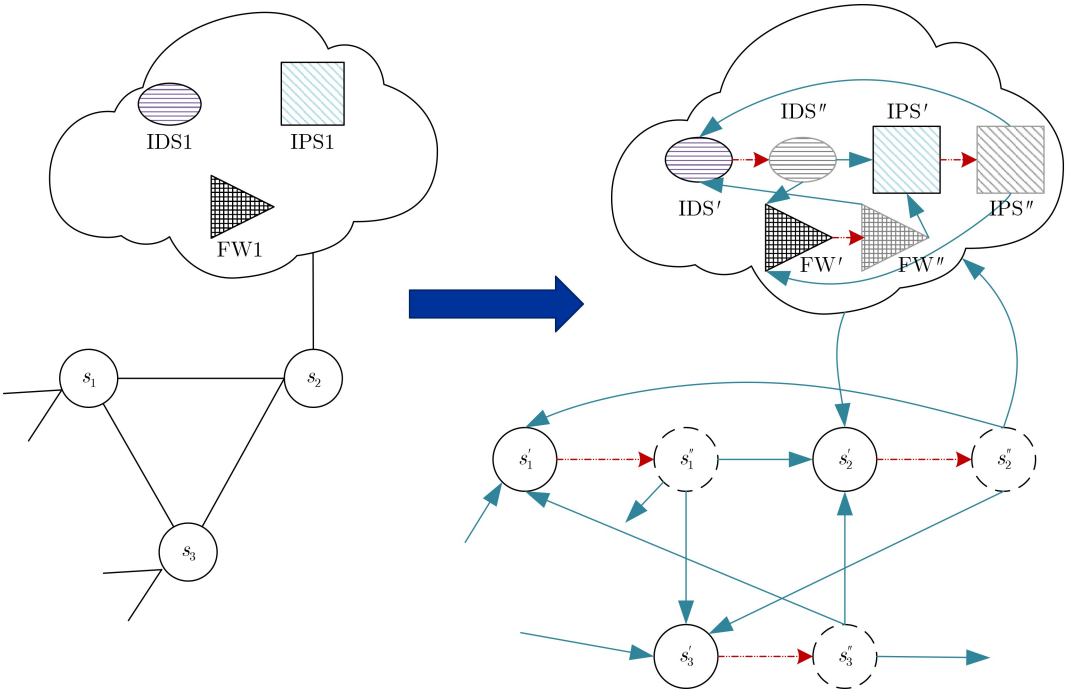

图 3 辅助边权图G_i'的构造

$$w_{i,\hat{e}}^{\text{delay}}=\begin{cases} d_{t(m)}^{\text{delay}},\ \hat{e}=\langle \hat{u}',\hat{u}''\rangle\in \widehat{L}_i \\ \sum\limits_{e'\in p_i(u''v')} w_{i,e'}^{\text{delay}},\ \hat{e}=\langle \hat{u}'',\hat{v}'\rangle\in \widehat{L}_i \end{cases} \tag{23}$$

3.3 启发式算法SFT-EA

基于G,G'和$\hat{G}$之间的关系，网络中资源消耗的相对成本都被转换为$\hat{G}$上的链路进行表示。由于MR_i的所有候选VNF实例都按预定顺序放置在$\hat{G}$上，因此，原问题转化为在$\hat{G}$上找一棵满足式(24)的多播树。用$T_i=\{V_i^T,E_i^T\}$表示该多播树，其中$V_i^T\supseteq\{s_i\}\in D_i$，相邻节点之间的链路记为$\hat{e}=\langle\hat{u},\hat{v}\rangle\in\widehat{L}_i$，$d(\hat{e})=w_{i,\hat{e}}^{\text{delay}}$表示链路$\hat{e}$上的延迟，$\hat{p}_i(s_i,\hat{u})$表示$T_i$上从源节点$s_i$到节点$\hat{u}$的路径。$s_i$到节点$\hat{u}$的延迟用$D(\hat{u})$表示

$$D(\hat{u})=\sum_{\hat{e}\in\hat{p}_i(s_i,\hat{u})} d(\hat{e})$$

$$\min\sum_{\hat{e}\in T_i} w_{i,\hat{e}}^{\text{cost}}$$

$$\text{s.t.}\begin{cases} \text{C1:}\sum\limits_{\hat{e}\in\hat{p}_i(s_i,\hat{u})} d(\hat{e})\le R_i^{\text{delay}},\forall\hat{u}\in D_i,\hat{p}_i\in T_i \\ \text{C2:}\left|\sum\limits_{\hat{e}\in\hat{p}_i(s_i,\hat{u})} d(\hat{e})-\sum\limits_{\hat{e}\in\hat{p}_i(s_i,\hat{v})} d(\hat{e})\right|\le R_i^{\text{Jitter}}, \\ \forall\hat{u},\hat{v}\in D_i,\hat{p}_i\in T_i \end{cases} \tag{24}$$

基于以下思路设计服务功能树嵌入算法(Service Function Tree Embedding Algorithm, SFT-EA)来解决式(24)。定义$\Delta_T\in R^+$为实际多播树T_i中源目的对路径的最大延迟，即

$$\Delta_T=\max\sum_{\hat{e}\in p_i(s_i,\hat{u})} d(\hat{e}),\ \forall\hat{u}\in D_i \tag{25}$$

显然，

$$\Delta_T\le R_i^{\text{delay}} \tag{26}$$

此外，如果$\hat{u}\in D_i$，称$\hat{u}$为目的节点；如果$\hat{u}\notin D_i$，但$\hat{u}\in T_i$，则称$\hat{u}$为中继节点。

将$\sum_{\hat{e}\in\hat{p}_i(s_i,\hat{v})} d(\hat{e})$表示为$D(\hat{u})+d(\hat{u},\hat{v})$，根据式(25)可以得到

$$\Delta_T-(D(\hat{u})+d(\hat{u},\hat{v}))\le R_i^{\text{Jitter}} \tag{27}$$

显然，

$$D(\hat{u})+d(\hat{u},\hat{v})\le\Delta_T \tag{28}$$

根据式(27)、式(28)可以得到

$$\Delta_T-R_i^{\text{Jitter}}\le D(\hat{u})+d(\hat{u},\hat{v})\le\Delta_T \tag{29}$$

首先使用KruskalMST最小生成树算法计算$\hat{G}_i$上的最小时延树(LDT) T_{LDT}，假设T_{LDT}上每个源目的对之间路径的最大延迟为$D_{\max}$，则有$D_{\max}\le\Delta_T$。另外，$R_i^{\text{Jitter}}\le\Delta_T$，否则延迟抖动约束无效。为了保证多播树$T_i$中的源目的对路径满足实际的时延和时延抖动要求，结合式(26)，可以得到Δ_T的取值范围为

$$\Delta_T\in\left[\max\left\{D_{\max},R_i^{\text{Jitter}}\right\},R_i^{\text{Delay}}\right] \tag{30}$$

SFT-EA的主要思想是在Δ_T的范围内从小到大取值，将每一个值作为当前构造的多播树中路径的时延上限。算法首先初始化一棵只包含多播源节点的多播树，然后逐个贪婪地加入$\hat{G}_i$中的节点。在节点加入的过程中，使用式(31)中定义的最佳链路选择函数来选择具有最小值的节点$\hat{v}$，并将该节点对应的边添加到当前多播树T_i中，然后，将$\hat{v}$作为当前节点，重复上述过程。如果找不到符合条件的节点$\hat{v}$，则回溯到当前节点的父节点并继续上述过程。最后，若在Δ_T时延范围内找不到包含所有目的节点的多播树，则多播树构造失败，结束算法。

$$S(\hat{u},\hat{v})=\begin{cases} f(\hat{u},\hat{v}),\hat{v}\notin D_i,D(\hat{u})+d(\hat{u},\hat{v})\le\Delta_T\text{或} \\ \hat{v}\in D_i,\Delta_T-R_i^{\text{Jitter}}\le D(\hat{u})+d(\hat{u},\hat{v})\le\Delta_T \\ \infty,\text{其他} \end{cases} \tag{31}$$

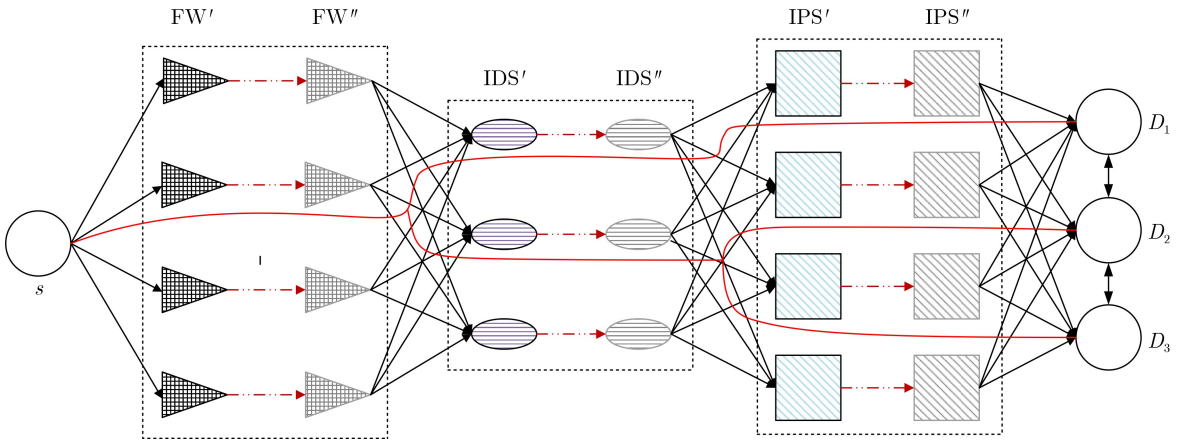

图 4 在图$\hat{G}_i$中解决DJA-DSPR

在式(31)中，$f(\hat{u},\hat{v})=\dfrac{\omega_{i,\hat{u},\hat{v}}^{\text{cost}}}{\Delta_T-D(\hat{u})-d(\hat{u},\hat{v})}+\dfrac{\Delta_T-D(\hat{u})-d(\hat{u},\hat{v})}{\omega_{i,\hat{u},\hat{v}}^{\text{cost}}}$

SFT-EA算法的伪代码如算法1：

定理1 SFT-EA的时间复杂度为$O(K|N|^3)$。

证明 执行SFT-EA时，枚举VNF的时间复杂度为$O(|V_f|)$。计算G'中链路和节点的相对成本的复杂度为$O(|V_s|+|M|+|L|)$。最坏情况下，所有类型的VNF实例都可以部署在每个功能节点上，将执行$|V_f|+(l-1)|V_f|^2+|V_f||D_i|$次SP算法为$\hat{G}$生成链路，通常是$V_f<<D_i$，因此执行SP算法的次数约为$|V_f||D_i|$。$G'$上SP的复杂度为$O(|L|+|V|\log|V|)$，$\hat{G}$最多有$2l|V_f|+1+D_i$个节点和$(l-1)|V_f|^2+l|V_f|+|V_f|+|V_f||D_i|$条链路，因此，建立$\hat{G}$的时间复杂度是：$O(|V_f||D_i|(|(l-1)|V_f|^2+l|V_f|+|V_f|+|V_f||D_i||+|(2l|V_f|+1+D_i)|\log_2|(2l|V_f|+1+D_i)|)\vDash O(|V_f|^2|D_i|^2+|D_i|^2|V_f|\log_2|D_i|)$。KruskalMST计算$T_{\text{LDT}}$的复杂度为$O(V^2)=O((2l|V_f|+1+D_i)^2)=O({D_i}^2)$，在算法中(第10～27行)，当没有回溯时，时间复杂度为$O(V^2)$，当有回溯时，时间复杂度为$O(V^3)$。用于确定是否访问了所有目的地节点的时间复杂度为$O(D_i)$(第30～34行)，在最坏情况下，SFT-EA将迭代K次，路径映射和资源更新的复杂度为$O(1)$。l的值很小。因此，SFT-EA的算法复杂度是$O(K((|V_f|)+(|V_s|+|M|+|L|)+(|V_f|^2|D_i|^2+|D_i|^2|V_f||\log_2|D_i|)\vDash O(K|D_i|^3)$。在最坏的情况下，网络中除源节点外的所有节点都是多播目的节点，所以，SFT-EA的复杂度为$O(K|N|^3)$。 证毕

图4中的红色粗线给出了具有3个目的节点和3个VNF顺序请求SFT的算法运行示例。

4 性能仿真分析

4.1 仿真设置

在Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz, 32.0 GB RAM的计算机上，用NetworkX3.1 Library[20]生成 Palmethone网络[21]对提出的算法进行评估。网络中，选择按节点度降序排序的30%的节点作为功能节点。考虑6种类型的VNFS [2]。D_p的取值为Uniform[0,10]$\times R_i^{\text{cpu}}$。每条链路的带宽(Gbps)为Uniform[1,10]。交换节点的流量表容量设置为800个单元[2]，功能节点上的CPU容量设置为8000MIPS。δ_p^{delay}(ms)设为Uniform[0.002,0.003]$\times R_i^{\text{cpu}}$，链路的延迟(ms)为Uniform[2,5] [2]。w_{bw}, $w_{\text{ft}}, w_{\text{cpu}}$设置为$2|V|$[11]。

对于每条多播流，从目标网络中随机选择源节点和目的节点。为了评估多播规模的影响，将

算法1 SFT-EA算法

输入：$G(N,L)$，流请求MR_i，资源容量，资源剩余率，最大迭代次数K。

输出：多播树T。

1. 根据2.4.1枚举G中功能节点上所有可能的VNFs
2. 根据3.1构造G_i'
3. 基于G_i'构造$\hat{G}_i$
4. 求最小时延树$T_{\text{LDT}}\leftarrow\text{KruskalMST}(\hat{G}_i)$
5. **if** $D_{\max}$ in $T_{\text{LDT}}>R_i^{\text{delay}}$ 则算法结束
6. **else**
7. **while**$((\Delta_T\in[\max\{D_{\max},R_i^{\text{jitter}}\},R_i^{\text{delay}}])$且$k\le K)$//算法最多迭代$K$次
8. $T_i\leftarrow s;V[\hat{u}]=0;L[\hat{u}][\hat{v}]=0;V[s]=1$，//数组$V$和$L$分别用于记录节点和链路是否已被访问

$Pre(s)=\emptyset;\hat{u}=\text{s};\Delta_T=\max\{D_{\max},R_i^{\text{jitter}}\}$//$pre()$函数表示一个节点的前驱

9. **while** $L[\hat{u}][\hat{v}]\neq 1$
10. **if** $\hat{v}\notin D_i\cup\{s\}$ and $D(\hat{u})+d(\hat{u},\hat{v})\leqslant\Delta_T$
11. $N=N\cup\{\hat{v}\}$//N记录候选节点
12. **else if** $\hat{v}\in D_i$且$\Delta_T-R_i^{\text{jitter}}\le D(\hat{u})+d(\hat{u},\hat{v})\le\Delta_T$
13. $N=N\cup\{\hat{v}\}$
14. **end if**
15. **if** $N\neq\emptyset$
16. **for each** $\hat{v}\in N$
17. **if** $D(\hat{u})+d(\hat{u},\hat{v})==\Delta_T$
18. **if** $\hat{v}\in D_i$
19. $V[\hat{v}]=1;L[\hat{u}][\hat{v}]=1;T_i=T_i\cup\{\hat{v}\}$
20. **else** $L[\hat{u}][\hat{v}]=1$
21. **end if**
22. **else if** $D(\hat{u})+d(\hat{u},\hat{v})<\Delta_T$
23. 选择$f(\hat{u},\hat{v})$中最小的节点$\hat{v}$
24. $T_i=T_i\cup\{\hat{v}\};V[\hat{v}]=1;L[\hat{u}][\hat{v}]=1;Pre(\hat{v})=\hat{u};\hat{u}=\hat{v};L[\hat{u}][\hat{v}]=0;$
25. **end if**
26. **end for**
27. **else** 将$L[\hat{u}][\hat{v}]$为0的链路设置为1，$\hat{u}=pre(\hat{u})$
28. **end if**
29. **end while**
30. **for each** $\hat{v}\in D_i$
31. **if** 所有$V[\hat{v}]==1$ 则转到36
32. **else** $\Delta_T++;k++;$break;
33. **end if**
34. **end for**
35. **end while**
36. **if** T_i包含D_i中的所有目的节点
37. 通过将T_i中的每条边替换为其在G中对应的路径来映射多播树T，并更新$r_{i,\text{uv}}^{\text{bw}}$, $r_{i,u}^{\text{ft}}$, $r_{i,u}^{\text{cpu}}$和$r_{i,m}^{\text{cpu}}$
38. **else** 路由失败，MR_i请求被拒绝，Return
39. **end if**
40. **end if**

$|D|/|V|$值设置为0.3，R_i^{bw}(Mbps)为Uniform [10,120][13]。R_i^{cpu}(MIPS)为Uniform[0,10] $\times R_i^{\mathrm{bw}}$，R_i^{delay}(ms)为Uniform [50,100]，R_i^{jitter}(ms)为Uniform[30,50] [2]。此外，将SFC的长度和SFT-EA的迭代最大次数K分别设置为4和10。取30次实验结果的平均值作为结果输出。

4.2 对比算法

选用SDN/NFV网络架构中与本文研究方向紧密相关且最新的对比算法进行仿真对比。

(1)STB[22]：用传统经典的Steiner树算法构造一个覆盖多播所有目的节点的Steiner树，并找到将源节点连接到该树的最小代价路径。然后沿着该路径嵌入所需的SFC。STB只考虑链路和节点资源的成本；

(2)TSA[13]：通过考虑链路和节点资源成本得到SFC嵌入的初始方案，然后通过添加新的VNF实例逐步构建多播服务功能树(SFT)，该算法不考虑流表成本及路径抖动约束；

(3)HAJPR[16]：根据网络中的关键NFV节点构造多径多播树，并且考虑物理资源和延迟约束，不考虑抖动约束，且没有解决VNF的动态嵌入。

4.3 性能比较

图5为平均流接受率的比较。SFT-EA的性能最好。当传入MRs的数量为5 000时，它接受的MR比HAJPR多约6%，比TSA高约13%，比STB高约17%。在SFT-EA中，实现了VNF的动态放置和负载均衡，提供了更好的性能。HAJPR考虑多路径路由，其性能优于TSA和STB。

图6展示了平均吞吐量的比较。当MR的数量为4 000时，SFT-EA的吞吐量比HAJPR高约5 Gbps，比TSA高约14 Gbps，比STB高约20 Gbps。性能和流接受率相符。

图7展示了当4 000条MRs进入网络时链路平均剩余带宽的累积分布。从图中可看出，STB和TSA有大约5%和3%的瓶颈链路。对于SFT-EA，剩余带宽为6G和3G的链路分别约占71%和17%，因此，网络中约有54%的链路剩余带宽在3 Gbps到6 Gbps之间，对于HAJPR, TSA和STB，其剩余带宽在3 Gbps到6 Gbps之间的链路分别约为51%、32%和27%。显然，SFT-EA中使用指数函数来表示链路带宽成本，使其对链路资源的利用比其他算法更加均衡。

图8描述了4 000条MRs进入网络时平均剩余流表条目的累积分布。可以看到，TSA和STB有大约3%和5%的交换节点瓶颈。从图中剩余流表条目为400个单位看，STB大约有60%的交换机节点，其剩余流表条目小于400个单位，TSA为49%，HAJPR为35%，SFT-EA为30%， SFT-EA算法的性能优于其他算法。

图9显示了当4 000条MRs进入网络时，功能节

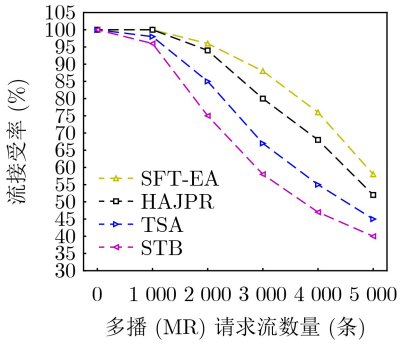

图 5 平均流接受率

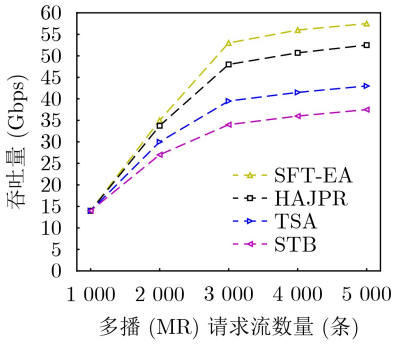

图 6 平均吞吐量

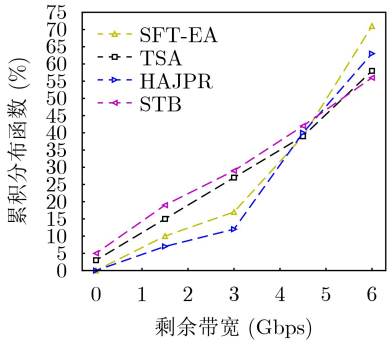

图 7 剩余带宽累积分布

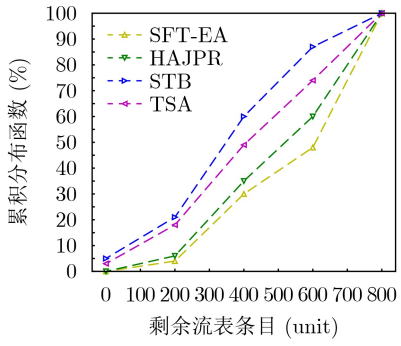

图 8 剩余流表累积分布

点上平均剩余CPU的累积分布。由于STB和TSA不能保证功能节点上CPU的均衡使用，分别有约4%和2%的瓶颈节点。对于SFT-EA，剩余CPU为1600 MIP的功能节点大约占10%，剩余CPU为4000 MIPS的功能节点大约占85%，因此，约75%的功能节点的剩余CPU在1 600～4 000 MIPS，而HAJPR, TSA和STB在这区间的剩余CPU分别约为63%、49%和45%。SFT-EA比其他算法更均衡。

图10展示了不同长度SFC下流接受率的变化。由于MR消耗的网络资源随着SFC长度的增加而增加，因此，流接受率随SFC长度的增加而降低。因STB和TSA无法平衡带宽、流表和CPU资源的利用率，因此它们的性能较差。虽然HAJPR不考虑资源平衡，但由于采用多径路由，且不考虑时延抖动约束，因此性能略高于SFT-EA。

图11显示了不同比例功能节点下网络的平均流接受率。功能节点的比例越高，可以动态部署VNF实例的节点就越多，MR可以选择冗余的VNF实例来满足其预定顺序，因此，这些算法的平均流接受率就越高。同理，虽然HAJPR不考虑资源平衡，但由于采用多径路由，且不考虑时延抖动约束，因此性能略高于SFT-EA。

图12给出了多播目的节点数$|D| \in [5,25]$时，算法的运行时间比较，随着多播目的节点数的增加，

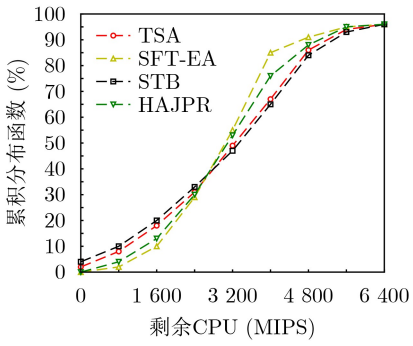

图 9 剩余CPU的累积分布

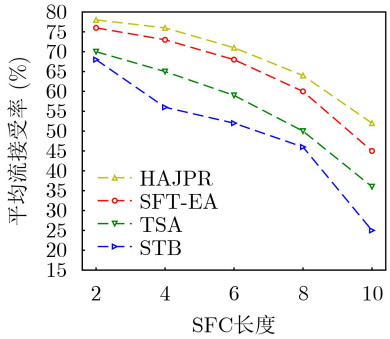

图 10 平均流接受率随SFC长度的变化

4种算法的执行时间都会增加。因SFT-EA需要考虑抖动约束，算法运行时间约高于HAJPR。

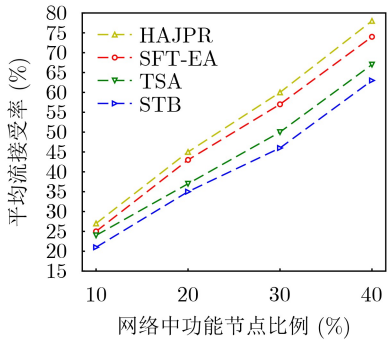

图 11 不同功能节点数下的平均流接受率

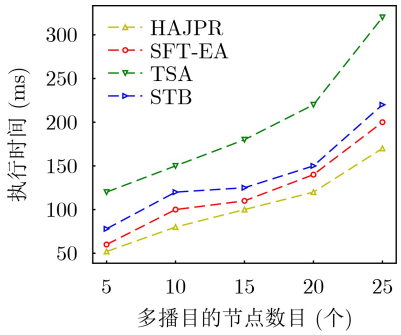

图 12 不同目的节点数下算法的执行时间

5 结束语

本文基于SDN/NFV架构，研究了延迟和抖动感知的多播服务功能树嵌入和路由问题。首先将其建模为ILP模型。然后，利用节点分裂，将节点资源成本转换为边的思想，并设计了一个最优链路选择函数和基于辅助边权图的SFT嵌入算法SFT-EA来解决该问题。最后通过实验仿真对算法的性能进行了评价。结果表明，所提算法在吞吐量、流接受率和网络负载均衡方面比现有算法有更好的性能，为虚拟网络环境下具有延迟和抖动约束的多播流调度提供了有价值的参考。

参 考 文 献

[1] 唐伦, 吴婷, 周鑫隆, 等. 一种基于联邦学习资源需求预测的虚拟网络功能迁移算法[J]. 电子与信息学报, 2022, 44(10): 3532–3540. doi: 10.11999/JEIT210743.

TANG Lun, WU Ting, ZHOU Xinlong, *et al*. A virtual network function migration algorithm based on federated learning prediction of resource requirements[J]. *Journal of Electronics & Information Technology*, 2022, 44(10): 3532–3540. doi: 10.11999/JEIT210743.

[2] LIU Liang, GUO Songtao, LIU Guiyan, *et al*. Joint dynamical VNF placement and SFC routing in NFV-enabled SDNs[J]. *IEEE Transactions on Network and*

Service Management, 2021, 18(4): 4263–4276. doi: 10.1109/TNSM.2021.3091424.

[3] 徐勇军, 谷博文, 谢豪, 等. 全双工中继协作下的移动边缘计算系统能耗优化算法[J]. 电子与信息学报, 2021, 43(12): 3621–3628. doi: 10.11999/JEIT200937.
XU Yongjun, GU Bowen, XIE Hao, *et al*. Energy consumption optimization algorithm for full-duplex relay-assisted mobile edge computing systems[J]. *Journal of Electronics & Information Technology*, 2021, 43(12): 3621–3628. doi: 10.11999/JEIT200937.

[4] MALI R, ZHANG Xijun, and QIAO Chunming. Benefits of multicasting in all-optical networks[C]. SPIE 3531, All-Optical Networking: Architecture, Control, and Management Issues. Boston, United States, 1998: 209–220. doi: 10.1117/12.327060.

[5] Grand View Research. Video streaming market size, share & trends analysis report by streaming type, by solution, by platform, by service, by revenue model, by deployment type, by user, by region, and segment forecasts, 2023 – 2030[EB/OL].https://www.grandviewresearch.com/industry-analysis/video-streaming-market, 2023.

[6] XIE Yanghao, HUANG Lin, KONG Yuyang, *et al*. Virtualized network function forwarding graph placing in SDN and NFV-Enabled IoT networks: A graph neural network assisted deep reinforcement learning method[J]. *IEEE Transactions on Network and Service Management*, 2022, 19(1): 524–537. doi: 10.1109/TNSM.2021.3123460.

[7] XU Zichuan, LIANG Weifa, HUANG Meitian, *et al*. Approximation and online algorithms for NFV-enabled multicasting in SDNs[C]. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, USA, 2017: 625–634. doi: 10.1109/ICDCS.2017.43.

[8] MUHAMMAD A, SORKHOH I, QU Long, *et al*. Delay-sensitive multi-source multicast resource optimization in NFV-enabled networks: A column generation approach[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(1): 286–300. doi: 10.1109/TNSM.2021.3049718.

[9] 孔紫璇, 李航, 向万, 等. 用户动态接入下的多播业务链部署和调整方法[J]. 北京邮电大学学报, 2022, 45(6): 53–59. doi: 10.13190/j.jbupt.2022-116.
KONG Zixuan, LI Hang, XIANG Wan, *et al*. Multicast service chain deployment and adjustment method under user dynamic access[J]. *Journal of Beijing University of Posts and Telecommunications*, 2022, 45(6): 53–59. doi: 10.13190/j.jbupt.2022-116.

[10] WANG Xinhan, XING Huanlai, SONG Fuhong, *et al*. Dynamic multicast-oriented virtual network function placement with SFC request prediction[C]. 2022 14th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 2022: 81–88. doi: 10.1109/ICCSN55126.2022.9817590.

[11] JIA Mike, LIANG Weifa, HUANG Meitian, *et al*. Routing cost minimization and throughput maximization of NFV-enabled unicasting in software-defined networks[J]. *IEEE Transactions on Network and Service Management*, 2018, 15(2): 732–745. doi: 10.1109/TNSM.2018.2810817.

[12] XU Zichuan, LIANG Weifa, HUANG Meitian, *et al*. Efficient NFV-enabled multicasting in SDNs[J]. *IEEE Transactions on Communications*, 2019, 67(3): 2052–2070. doi: 10.1109/TCOMM.2018.2881438.

[13] REN Bangbang, GUO Deke, SHEN Yulong, *et al*. Embedding service function tree with minimum cost for NFV-enabled multicast[J]. *IEEE Journal on Selected Areas in Communications*, 2019, 37(5): 1085–1097. doi: 10.1109/JSAC.2019.2906764.

[14] ASGARIAN M, MIRJALILY G, and LUO Zhiquan. Trade-off between efficiency and complexity in multi-stage embedding of multicast VNF service chains[J]. *IEEE Communications Letters*, 2022, 26(2): 429–433. doi: 10.1109/LCOMM.2021.3132134.

[15] REN Haozhe, XU Zichuan, LIANG Weifa, *et al*. Efficient algorithms for delay-aware NFV-enabled multicasting in mobile edge clouds with resource sharing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2020, 31(9): 2050–2066. doi: 10.1109/TPDS.2020.2983918.

[16] ALHUSSEIN O, DO P T, YE Qiang, *et al*. A virtual network customization framework for multicast services in NFV-enabled core networks[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(6): 1025–1039. doi: 10.1109/JSAC.2020.2986591.

[17] REN Cheng, CHEN Xuxiang, XIANG Haiyun, *et al*. On efficient delay-aware multisource multicasting in NFV-Enabled softwarized networks[J]. *IEEE Transactions on Network and Service Management*, 2022, 19(3): 3371–3386. doi: 10.1109/TNSM.2022.3188777.

[18] LI Hang, WANG Luhan, ZHU Zhenghe, *et al*. Multicast service function chain orchestration in SDN/NFV-Enabled networks: Embedding, readjustment, and expanding[J]. *IEEE Transactions on Network and Service Management*, 2023. doi: 10.1109/TNSM.2023.3257187..

[19] MIRJALILY G, ASGARIAN M, and LUO Zhiquan. Interference-aware NFV-enabled multicast service in resource-constrained wireless mesh networks[J]. *IEEE Transactions on Network and Service Management*, 2022, 19(1): 424–436. doi: 10.1109/TNSM.2021.3083798.

[20] Python Community. Networkx 3.1[EB/OL]. https://pypi.org/project/networkx/, 2023.

[21] Internet topology[EB/OL]. http://topology-zoo.org/maps/,

2022.

[22] CHENG Yulun and YANG Longxiang. VNF deployment and routing for NFV-enabled multicast: A Steiner tree-based approach[C]. 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 2017: 1–4. doi: 10.1109/WCSP.2017.8170947.

刘 亮：男，副教授，博士，研究方向为网络功能虚拟化、端边云资源协同管理、天地一体化网络.

陈 翔：男，硕士生，研究方向为NFV环境中的多播路由、服务功能链部署和重配置.

桂晓菁：女，讲师，研究方向为内容中心网络.

徐勇军：男，副教授，博士，研究方向为智能超表面、资源分配、机器学习算法.

杜娅荣：女，硕士生，研究方向为服务功能链部署迁移、机器学习算法.

侯泽天：男，硕士生，研究方向为移动环境中虚拟网络资源配置.

段 洁：女，博士，副教授，博士，主要研究方向为天地一体化网络、内容中心网络.

责任编辑：马秀强