

一种支持节点分割的 vEPC 虚拟网络功能部署模型

汤红波* 袁 泉 卢干强 王晓雷 赵 宇

(国家数字交换系统工程技术研究中心 郑州 450002)

(移动互联网安全技术国家工程实验室 北京 100876)

摘 要: 软件定义网络(SDN)和网络功能虚拟化(NFV)促进了网络的创新, NFV 实现了虚拟网络功能(VNF)的逻辑集中部署。针对 vEPC(virtualized Evolved Packed Core)网络中 VNF 的池组化部署问题, 该文提出一种支持节点分割的 VNF 部署模型, 该模型基于虚拟请求业务流量的感知, 利用节点分割算法动态调整 VNF 与底层网络资源切片的映射关系, 实现 VNF 的跨域组池。与传统的多功能链联合映射算法相比, 该方法能够实现节点资源细粒度化管理和统筹调度, 优化网络视图, 减少资源碎片。在 SNDlib 提供的网络拓扑实例下仿真证明, 该模型可以降低虚拟网络的资源开销, 并提高虚拟网络的请求接收率。

关键词: 网络功能虚拟化; 虚拟网络功能部署; 5G; EPC(Evolved Packed Core); 节点分割

中图分类号: TN915.81

文献标识码: A

文章编号: 1009-5896(2017)03-0546-08

DOI: 10.11999/JEIT160507

A Model for Virtualized Network Function Deployment Based on Node-splitting in vEPC

TANG Hongbo YUAN Quan LU Ganqiang WANG Xiaolei ZHAO Yu

(National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China)

(National Engineering Laboratory for Mobile Network Security, Beijing 100876, China)

Abstract: Software Defined Networking (SDN) and Network Function Virtualization (NFV) promote network innovation. NFV realizes logic centralized deployment of Virtualized Network Function (VNF). This paper proposes a kind of node-splitting VNF deployment model for problem of pooling deployment of VNF in virtualized Evolved Packet Core (vEPC) network. Based on the acknowledgement of virtual request traffic-aware, the model uses node-splitting algorithm to realize the dynamic adjustment of the mapping relation between VNF and physical network slices and organize the same VNFs across different domains as one pool. Compared with the traditional joint mapping algorithm of multi-function chain, the method can realize fine-grained management and overall scheduling of node resources, optimize network view, and reduce resource fragments. It is proven by network topology instance provided by SNDlib that the model can reduce resource overhead of virtual network and improve the ratio of acceptance of virtual network requests.

Key words: Network Function Virtualization (NFV); Virtualized network function placement; 5G; Evolved Packet Core (EPC); Node-splitting

1 引言

当前, 有限的网络资源和僵化的网络结构限制了网络的发展。为了满足未来网络的需求, 提高网络的管理能力和灵活性, IMT-2020(5G)推进组提出

应用软件定义网络(Software-Defined Network, SDN)和网络功能虚拟化(Network Function Virtualization, NFV)技术实现演进分组核心(Evolved Packet Core, EPC)网网元功能基于数据中心网络的集中化部署、编排与管理^[1]。

目前, 多数文献^[2]都是研究互联网服务功能链场景下的虚拟网络功能(Virtualized Network Function, VNF)部署问题。近些年, 随着 5G 的兴起, 国内外学者对 vEPC 网络中 VNF 部署问题进行了相关研究。由于传统的数据中心网络难以满足 5G 移动通信高带宽、低时延和集中管控的要求, 文

收稿日期: 2016-05-17; 改回日期: 2016-09-06; 网络出版: 2016-11-17

*通信作者: 汤红波 thb@mail.ndsc.com.cn

基金项目: 国家 863 计划项目(2014AA01A701), 国家自然科学基金(61521003), 科技部支撑计划(2014BAH30B01)

Foundation Items: The National 863 Program of China (2014AA01A701), The National Natural Science Foundation of China (61521003), The Ministry of Science and Technology Support Plan (2014BAH30B01)

献[3]提出了一种分布式云平台上的虚拟化 EPC (virtualized EPC, vEPC)网络组网方案。该平台通过分布式云数据中心节点提供 vEPC 网络中 VNF 实例化所需的底层资源。云节点之间基于 SDN 技术实现动态组网,所有的云节点由一个统一的管理和编排模块集中管控。文献[4]首次提出了 vEPC 网络中 VNF 的部署问题,并提出了两种部署方式:(1)仅在数据中心内部虚拟化 VNF 的控制面,数据面通过增强的 SDN 转发设备实现。(2)将 VNF 的控制面和数据面全部虚拟化,利用现有骨干网设备完成转发。该文统筹考虑了网络中处理时延和传输时延之间的关系,优化了数据中心网络的总体时延,但该模型必须先对物理网络进行静态子网划分,在每一个子网分别部署相应的 EPC 服务链,模型中各个子网互相独立,网络视图复杂且不利于全网资源统一编排管理。文献[5]提出了多服务链联合映射模型,该模型对域内的 eNodeB 进行动态聚簇,每个 eNodeB 簇连接一条完整的 vEPC 服务链,假设各条服务链之间相互独立,将一条服务链上的 EPC 网元功能尽可能集中部署到相邻的数据中心内部,采用定时更新底层网络状态的方式,完成多条独立 EPC 服务链的联合映射。相比于静态子网划分策略,联合映射思想的优点是对子网进行动态划分,能够相对灵活地调度全网资源而缺点是使网络的全局视图更加复杂,提高了网络功能管理和编排的难度。此外,文献[4]和文献[5]都是以“网元”作为资源分配的粒度,当云节点中剩余的资源不足以实例化一个完整的网元功能时,节点的剩余资源将保持不可达状态,形成资源碎片,降低了物理网络资源的利用率。

现有研究多数假设 vEPC 保留了 EPC 现行架构的主要网元功能,包括移动管理实体(Mobility Management Entity, MME)、服务网关(Serving GateWay, SGW)、PDN 网关(PDN GateWay, PGW)和归属签约用户服务器(Home Subscriber Server, HSS)。文献[4]和文献[5]借鉴了互联网服务功能链的链式结构部署 vEPC 服务链,但是在 LTE 网络中, MME 和 SGW 都采用池组的形式组网。池组技术是指在一定区域内,多套相同的网元设备组成池组,通过通用接口实现资源共享和业务分担,是解决核心网切换、负载均衡和容灾备份的重要方案^[6]。为了支持 vEPC 网络中 VNF 的组池,本文提出一种基于节点分割的 vEPC 虚拟网络功能部署模型。该模型将虚拟网络请求的实时业务流量作为节点分割的决策变量,根据物理网络剩余资源的全局视图,动态调整虚拟节点分割的子节点数量,以及各个子节

点与物理节点的映射关系。相比于传统多条服务链联合映射部署模型,节点分割模型不再对同一 vEPC 管理区域内的物理网络进行子网划分,优化了虚拟网络视图,完成了 EPC 虚拟网络功能的组池,进一步细化资源分配的粒度,提高了资源利用率。

2 数学模型

本文采用虚拟网络映射方法对 vEPC 网络中的 VNF 部署问题建模,将底层网络和虚拟网络请求分别表示为一个无向带权图,根据优化目标设计相应的映射算法寻找最优部署结果。

2.1 底层网络

底层网络表示为一个由云节点和节点间链路组成的无向带权图 $G^S = (N^S, E^S, A^{n^S}, A^{e^S})$,其中 N^S 和 E^S 分别代表底层节点集合和底层链路集合,并把 E^S 中链路组成的路径记为 P^S 。 A^{n^S} 表示底层节点 $n^S \in N^S$ 的属性集合, $A^{n^S} = \{C^{n^S}, S^{n^S}, B^{n^S}, Loc^{n^S}\}$, 包含节点计算资源 C^{n^S} 、节点存储资源 S^{n^S} , 节点最大吞吐量 B^{n^S} 和节点位置 Loc^{n^S} 。 A^{e^S} 表示底层链路属性集合,本文中仅包含链路带宽 B^{e^S} 。

2.2 虚拟网络请求

支持池组技术的 vEPC 虚拟网络视图如图 1 所示,接入网中将同一跟踪区域内的 eNodeB 聚合在一起,对接入云进行分簇,并假设同一个簇内的 eNodeB 在回传网络中都连接到了同一个汇聚节点(Traffic Aggregation Point, TAP)。虚拟网络请求可表示为一个无向带权图 $G^V = (N^V, E^V, A^{VNF}, A^{e^V})$ 。其中, N^V 表示 VNF 的集合, $N^V = \{TAP, MME, HSS, SGW, PGW, IXP\}$ 。 E^V 表示虚拟链路的集合,每一条虚拟链路表示为 $e^V \in E^V$ 。 A^{VNF} 是表示虚拟网络功能属性的集合,与物理网络相同, $A^{VNF} = \{C^{VNF}, S^{VNF}, B^{VNF}, Loc^{VNF}\}$ 。 A^{e^V} 是表示虚拟链路属性的集合,即虚拟链路的带宽 B^{e^V} 。虚拟网络映射可以定义为:虚拟节点映射 $M_N(N^V) \rightarrow N^S$ 和虚拟链路映射 $M_E(E^V) \rightarrow P^S$ 。

2.3 构建一个底层网络的增广图

区别于传统数据中心网络中各节点同质化,各个云节点提供的服务类型具有差异性,因此在进行功能部署之前必须预先判断每一个云节点能够承载的 EPC 网络功能类型。本文参考了 ViNEYard^[7]算法,通过构建基于底层网络和虚拟网络的增广网络,为虚拟网络功能选择候选底层节点。

首先为每一个 VNF 在底层网络中建立一个镜像,用元节点 $\mu(VNF)$ 表示,再为每一个元节点选择一个候选底层节点集合 $\Omega(VNF)$ 。我们引入了二值

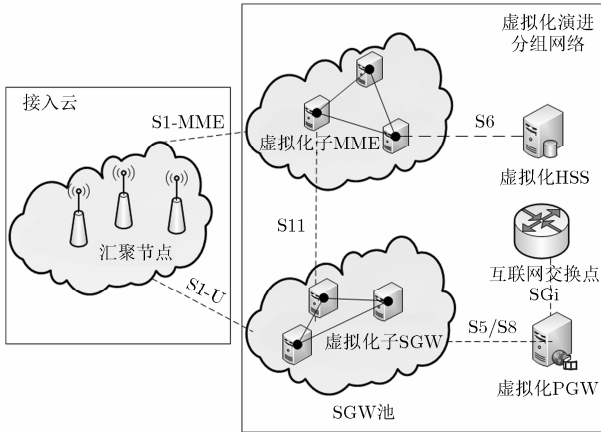


图 1 移动核心网功能链

矩阵 H 来设定每一个云节点可以承载的 VNF，矩阵元素为 $H^{u,VNF}$ ，其中 $u \in N^S, VNF \in N^V$ 。例如，当 $H^{u,MME} = 1$ 时，我们认为底层节点 u 可以承载 MME，反之，则证明底层节点不能承载相关功能。元节点 $\mu(VNF)$ 表示增广网络中的需要映射的 VNF，对于每一个元节点，我们利用 $H^{u,VNF}$ 选择一个候选底层节点集合 $\Omega(VNF)$ 。然后，连接元节点 $\mu(VNF)$ 和候选节点集合 $\Omega(VNF)$ 中的所有节点形成元链路 $\{(\mu(VNF), n^S) | \forall n^S \in \Omega(VNF), \forall VNF \in N^V\}$ ，对每一个元节点重复此操作就可以获得如图 2 所示的增广网络图 $G^{S'} = (N^{S'}, E^{S'}, A^{n^{S'}}, A^{e^{S'}})$ ，图 2 中增广网络部分三角形节点表示元节点，点画线表示元链路，带宽为 $+\infty$ ，圆形节点表示底层节点，实线表示底层链路。增广网络节点集合满足条件 $N^{S'} = \{\mu(VNF) | \forall VNF \in N^V\} \cup N^S$ ，增广网络链路集合满足条件 $E^{S'} = \{(\mu(VNF), n^S) | \forall n^S \in \Omega(VNF), \forall VNF \in N^V\} \cup E^S$ 。

3 节点分割算法

文献[8-10]针对互联网功能链部署提出了多种节点分割算法，文献[8]研究了物理节点可接受同一

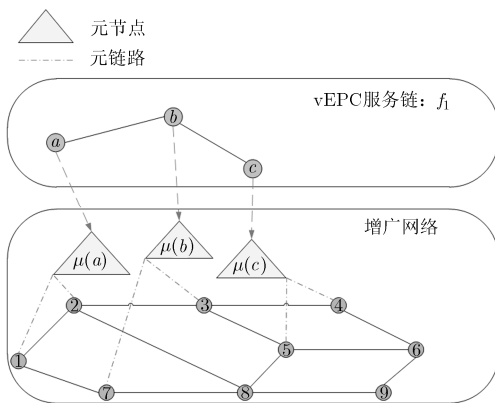


图 2 增广网络示意图

虚拟网络的多个节点映射问题，提高了请求接受率，但是该算法是一个两阶段的映射算法，无法同时获得节点和链路映射的最优解。文献[9]提出了节点和链路协同映射的一阶段映射算法，但是该算法仅支持物理节点承载不在同一个虚拟网络的多个节点映射。文献[10]提出了虚拟机功能链部署模型中，根据底层网络资源分配情况拆分虚拟网络节点的方法。参考以上文献，本文提出了 EPC 网络的节点分割算法，该算法基于 5G 非栈式协议，提出了根据底层网络流量动态拆分 EPC 虚拟网络功能的分割方案。节点分割 EPC 模型用到的符号和含义如表 1 所示。

表 1 节点分割 EPC 模型用到的符号和含义

符号	含义	符号	含义
N^S	底层网络节点 n^S 的集合	$N^{S'}$	增广网络节点
E^S	底层网络链路 e^S 的集合	$E^{S'}$	增广网络链路
A^{n^S}	底层网络节点属性，包含 $C^{n^S}, S^{n^S}, B^{n^S}, Loc^{n^S}$	$A^{n^{S'}}$	增广网络节点属性
A^{e^S}	底层网络链路属性，包含 B^{e^S}	$A^{e^{S'}}$	增广网络链路属性
N^V	虚拟网络功能 VNF 集合	$R(A^{n^S})$	底层网络节点属性剩余资源
E^V	虚拟网络链路 (VNF_2, VNF_1) 的集合	$R(A^{e^S})$	底层网络链路属性剩余资源
A^{VNF}	虚拟网络功能的属性	$\mu(VNF)$	元节点
A^{e^V}	虚拟网络链路的属性	$\Omega(VNF)$	候选节点集合

3.1 决策变量设置和底层网络资源测量

(1)决策变量设置：传统网络协议栈中传输流量不可分割，节点分割算法一般选择虚拟节点和底层节点的映射关系变量作为决策变量，此时，虚拟网络功能到底层网络节点的映射关系是一个模糊集合 $X : \{x^{VNF,i} \in [0,1] | VNF \in N^V, i \in N^S\}$ ，应用混合整数规划的方法不能获得全局最优解。目前，多采用两阶段映射与启发式算法相结合的方式寻找节点分割算法最优解，但是两阶段映射策略无法同时获得节点和链路的最优解而启发式会陷入局部最优^[11]。5G 移动通信提出了“非栈式”协议架构^[12]，对协议栈进行虚拟化，使 5G 网络具备智能处理数据流的能力。基于 5G 非栈式协议的设想以及流分割技术的研究成果，本文提出在映射过程中，将底层网络的流量作为决策变量，利用混合整数规划设计一阶段的节点分割算法。整数规划决策变量和映射关系变量的符号和含义如表 2 所示。

(2)底层网络资源测量：本文提出的节点分割算法是一种动态映射算法，在映射过程中根据底层网络流量不断调整映射关系矩阵。为了降低计算时延，基于已有研究，我们利用了以下两个结论来降低底层网络测量的计算复杂度。

结论 1 根据文献[13]，假设在映射过程中一个

表2 整数规划决策变量和映射关系变量

变量	含义
$f_{(u,v)}^{e^V}$	虚拟链路 e^V 映射到底层链路 (u,v) 在 $u \rightarrow v$ 方向上的流量值
$x_{u,v}$	二值变量, 表示虚拟网络功能与底层节点之间的映射关系, 当 $\sum_{e^s \in E^S} f_{(u,v)}^{e^s} = 0$ 时为 0, 否则为 1

虚拟网络功能(VNF)分割成 m 个子节点, 其与业务量相关的节点属性资源需求按照承担的业务量映射到 m 个子节点上, 如式(1)所示。其中, $i \in [1, m]$ 是子节点序号, A_0^{VNF} 表示部署该虚拟网络功能所必须消耗的基本属性资源, $g_i(\text{VNF})$ 表示第 i 个节点所承担的该虚拟网络功能的业务量, $A_{g_i(\text{VNF})}^{\text{VNF}}$ 表示第 i 个节点承担 VNF 的业务量而消耗的属性资源。同样, 虚拟网络功能的总带宽资源请求等于各个子节点上的数据流量之和, 即 $B^{\text{VNF}} = \sum_i^m g_i(\text{VNF})$ 。

$$A^{\text{VNF}} = \sum_i^m (A_0^{\text{VNF}} + A_{g_i(\text{VNF})}^{\text{VNF}}) \quad (1)$$

结论 2 文献[14]中软件路由器的测量结果表明, 处理单位分组所消耗的平均计算和存储资源近似为常数。因此, 根据式(1)节点计算、存储和带宽资源表达式可以分别近似为式(2)~式(4), 其中 α 和 β 为常数。

$$C^{\text{VNF}} = \sum_i^m (C_0^{\text{VNF}} + C_{g_i(\text{VNF})}^{\text{VNF}}) = mC_0^{\text{VNF}} + \sum_i^m (\alpha g_i(\text{VNF})) = mC_0^{\text{VNF}} + \alpha B^{\text{VNF}} \quad (2)$$

$$S^{\text{VNF}} = mS_0^{\text{VNF}} + \beta B^{\text{VNF}} \quad (3)$$

$$B^{\text{VNF}} = \sum_{(u,v) \in E^S} \sum_{e^V} (f_{(u,v)}^{e^V} + f_{(v,u)}^{e^V} - f_{(v,u),v=u}^{e^V}), \quad \forall (u,v) \in E^S, \forall e^V \in E^V \quad (4)$$

在上述简化的基础上, 我们将在单位时间映射结束后更新底层网络的剩余资源。如式(5), 式(6)所示, $\Delta A(t-1, t)$ 表示在 $[t-1, t]$ 单位时间里, 网络中被占用的资源与被释放的资源之差。

$$R(A^{n^S}, t) = R(A^{n^S}, t-1) - \Delta A^{n^S}(t-1, t) \quad (5)$$

$$R(A^{e^S}, t) = R(A^{e^S}, t-1) - \Delta A^{e^S}(t-1, t) \quad (6)$$

3.2 优化目标和约束条件

优化目标:

$$\begin{aligned} \text{Min}(\text{Cost}) = & \sum_{(u,v) \in E^S} \sum_{i \in E^V} (f_{(u,v)}^i - f_{(u,v),u=v}^i) \\ & + \lambda_1 \sum_{\text{VNF} \in N^V} \left(C_0 \sum_{n^S \in N^S} x_{\text{VNF}, n^S} + \alpha B^{\text{VNF}} \right) \\ & + \lambda_2 \sum_{\text{VNF} \in N^V} \left(S_0 \sum_{n^S \in N^S} x_{\text{VNF}, n^S} + \beta B^{\text{VNF}} \right) \\ & + \lambda_3 B_0 \sum_{u \in N^S} \sum_{i \in E^V} \sum_{\text{VNF} \in N^V} x_{\text{VNF}, u} f_{(u,v)}^i \quad (7) \end{aligned}$$

式(7)中, 第1项表示节点间链路流量占用的带宽资源, 第2项表示虚拟网络请求占用底层网络节点的计算资源, 其中, $C_0 \sum_{n^S \in N^S} x_{\text{VNF}, n^S}$ 表示在底层节点 n^S 上部署某个虚拟网络功能 VNF 所需要的基本资源, αB^{VNF} 则表示在映射某个 VNF 的过程中, 由节点处理的业务量所产生的资源需求。第3项表示虚拟网络请求占用底层网络节点的存储资源, 第4项表示承载虚拟网络功能的底层节点转发业务流所消耗的网络资源。 $\lambda_1, \lambda_2, \lambda_3$ 是表示权重的常数, 满足条件 $\lambda_1, \lambda_2, \lambda_3 \in (0, 1), \lambda_1 + \lambda_2 + \lambda_3 = 1$ 。对于特定网络环境, 可以相应地调整权重参数, 例如要着重考虑计算资源的分配时, 应该适当增大 λ_1 的权重。

约束条件:

容量约束:

$$\begin{aligned} \sum_{i \in E^V} f_{(u,v)}^i & \leq B^{(u,v)}, \forall u, v \in N^S, \forall (u,v) \in E^S \quad (8) \\ C^{n^S} & \geq \sum_{\text{VNF} \in N^V} [C_0 \cdot x_{\text{VNF}, n^S}] + \alpha \\ & \cdot \sum_{i \in N^V} \left(f_{(\text{VNF}, n^S)}^i + f_{(n^S, \text{VNF})}^i \right) x_{\text{VNF}, n^S}, \\ & \quad \forall i \in N^V, \forall n^S \in N^S \quad (9) \end{aligned}$$

$$\begin{aligned} S^{n^S} & \geq \sum_{\text{VNF} \in N^V} [S_0 \cdot x_{\text{VNF}, n^S}] + \beta \\ & \cdot \sum_{i \in N^V} \left(f_{(\text{VNF}, n^S)}^i + f_{(n^S, \text{VNF})}^i \right) x_{\text{VNF}, n^S}, \\ & \quad \forall i \in N^V, \forall n^S \in N^S \quad (10) \end{aligned}$$

$$\begin{aligned} B^{n^S} & \geq \sum_{i \in E^S} (f_{(u,v)}^i + f_{(v,u)}^i - 2f_{(u,v),u=v}^i), \forall u, v \in N^{S'}, \\ & \quad \forall (w, m) \in E^{S'}, \forall i \in N^V, \forall n^S \in N^S \quad (11) \end{aligned}$$

流相关约束:

$$\begin{aligned} \sum_{v \in N^{S'}} f_{(u,v)}^i - \sum_{v \in N^{S'}} f_{(v,u)}^i & = 0, \\ & \quad \forall i \in E^V, \forall u \in N^{S'} \setminus \{s_i, t_i\} \quad (12) \end{aligned}$$

$$\sum_{w \in N^{S'}} f_{(s_i, w)}^i - \sum_{w \in N^{S'}} f_{(w, s_i)}^i = B^{c_i^V}, \quad \forall i \in N^V \quad (13)$$

$$\sum_{w \in N^{S'}} f_{(t_i, w)}^i - \sum_{w \in N^{S'}} f_{(w, t_i)}^i = -B^{c_i^V}, \quad \forall i \in N^V \quad (14)$$

二进制变量约束:

$$x_{u,v} = x_{v,u}, \quad \forall u, v \in N^S \quad (15)$$

$$x_{\text{VNF}, v} = 0, \quad \forall \text{VNF} \in N^{S'} / N^S, \forall v \notin \Omega(\text{VNF}) \quad (16)$$

变量定义域约束:

$$f_{(u,v)}^i \geq 0, \quad \forall i \in N^V, \forall u, v \in N^{S'} \quad (17)$$

$$\sum_i f_{(u,v)}^i + f_{(v,u)}^i \leq B^{(u,v)}, \quad \forall i \in N^V, \forall u, v \in N^{S'} \quad (18)$$

$$x_{u,v} = \{0, 1\}, \quad \forall u, v \in N^{S'} \quad (19)$$

式(8)-式(19)详细说明如下:

式(8)–式(11)表示虚拟网络请求的资源数量与底层网络提供的资源数量之间的约束关系。其中,式(8)是对底层网络链路带宽的约束条件,当虚拟链路*i*映射到底层链路时,所有映射经过同一条底层网络路径的流量总和不能大于该路径的可用带宽值,在这里我们假设物理链路是全双工的。式(9),式(10)是对底层网络节点计算和存储资源的限制条件,当虚拟节点映射到底层时,底层网络节点的资源值必须不小于虚拟网络请求占用的资源。由于式(9),式(10)是非线性约束且 $x_{u,v}$ 是一个二值化变量,所以我们将式(9),式(10)简化为式(20),式(21)的形式。式(11)是对底层网络节点网络资源的约束条件,对于任意一个底层网络节点,流入和流出该节点的流量总和应该小于该节点剩余的网络资源。

$$C^{n^s} \geq \sum_{\text{VNF} \in N^V} \left[C_0^{\text{VNF}} x_{\text{VNF},n^s} \right] + \alpha \cdot \sum_{i \in N^V} \left(f_{(\text{VNF},n^s)}^i + f_{(n^s,\text{VNF})}^i \right), \forall i \in N^V, n^s \in N^S \quad (20)$$

$$S^{n^s} \geq \sum_{\text{VNF} \in N^V} \left[S_0^{\text{VNF}} x_{\text{VNF},n^s} \right] + \beta \cdot \sum_{i \in N^V} \left(f_{(\text{VNF},n^s)}^i + f_{(n^s,\text{VNF})}^i \right), \forall i \in N^V, n^s \in N^S \quad (21)$$

式(12)–式(14)描述的是流量守恒约束。式(12)表示除了在源节点 s_i 和目的节点 t_i 的其他节点(在网络中仅作为转发节点存在)上流入和流出的流量是相等的。式(13),式(14)表示流出(流入)源节点(目的节点)的流量等于相应的虚拟链路的带宽请求。

式(15)表示底层网络是一个无向图,即节点间进行对等通信。式(16)表示候选节点集合约束,考虑到不同的云节点之间存在差异性,对每一个节点所能承载的虚拟网络功能进行了约束。式(17),式(18)表示决策变量定义域的取值范围,式(19)表示映射关系变量 $x_{u,v}$ 是一个二值变量。

3.3 算法描述

本文提出的节点分割算法命名为 EPC-NS 算法(表 3),算法分成 3 个阶段:第 1 阶段是拓扑准备阶段,根据输入算法的底层网络无向图 G^S 和虚拟网络请求无向图 G^V 生成底层网络增广图 $G^{S'}$;第 2 阶段是资源判断阶段,为了降低混合整数规划的计算复杂度,避免不必要的映射,在运行整数规划之前先对网络中的候选节点集合 $\Omega(\text{VNF})$ 的剩余节点资源进行判断;第 3 阶段是算法执行阶段,对增广图运行混合整数规划算法,获取功能部署结果。

(1)拓扑准备:(1)~(4)算法第 1 行表示 EPC-NS 算法以在线的虚拟网络请求作为输入,并把 VNF 嵌入底层网络图生成元节点集合 $\mu(\text{VNF})$ 。第 2,第 3

表 3 EPC-NS 算法

输入:	虚拟网络请求无向图 G^V ; 物理网络无向图 G^S
输出:	节点分割部署方案 $P_{\text{EPC-NS}}$
(1)	在物理网络无向图 G^S 中生成虚拟网络功能 VNFs 镜像节点作为元节点 $\mu(\text{VNF})$
(2)	初始化二值矩阵 \mathbf{H}
(3)	获得候选节点集合 $\Omega(\text{VNF})$
(4)	连接元链路创建增广图 $G^{S'}$
(5)	while traffic
(6)	for VNF in N^V
(7)	for n^s in $\Omega(\text{VNF})$
(8)	if $\left(A^{\text{VNF}} \geq \sum_{n^s \in \Omega(\text{VNF})} R_N(A^{n^s}) \right)$ then
(9)	拒绝 G^V
(10)	do 混合整数规划算法
(11)	if 算法执行成功 then
(12)	更新底层网络资源, 返回部署方案 $P_{\text{EPC-NS}}$
(13)	else
(14)	拒绝 G^V

行根据 SNDlib 中的节点参数初始化变量 $H^{u,\text{VNF}}$,选择每个元节点的候选集合 $\Omega(\text{VNF})$ 。第 4 行将元节点和候选节点通过元链路链接,生成增广网络图 $G^{S'}$ 。

(2)资源判断:(5)~(9)判断 $\Omega(\text{VNF})$ 中节点的剩余资源是否能够完成本次虚拟网络映射。如果节点资源满足本次映射的请求则运行节点分割算法进行映射,如不满足,则直接拒绝请求。通过剩余资源判断可以有效地避免网络对不必要的请求进行映射,缩短了算法的处理时间。

(3)算法执行:(9)~(14)对增广网络图运行混合整数规划算法。如果成功完成映射则更新底层网络资源并获取映射结果,如不成功,则直接拒绝请求。

4 模型仿真与性能分析

4.1 参数设置

EPC-NS 算法仿真所使用的底层网络拓扑结构取自 SNDlib 中的 Polska12 节点测试例^[15],如图 3(a)所示。图中节点位置部署了运营商的云数据中心,假设所有节点都可以承载除了 PGW 之外的任意核心网功能,PGW 作为移动核心网和互联网之间的锚点,位置通常是固定的。在 Polska12 节点测试例中,设定只有 Rzeszow 和 Bydgoszcz 节点可以承载 PGW。在底层网络中,节点网络资源的数量和链路带宽由 SNDlib 给出,底层节点的计算和存储资源根据节点网络资源按比例随机生成。在虚拟网络请求中,用户面的流量是服从泊松分布的随机过程,控制面中节点所需的计算、存储和网络资源请求与用

户面业务流量呈预定的比例,根据 SNDlib 给出的测试数据,下文将采用流量因数(Traffic Demand Scaling Factor, TDSF)表示流量的变化情况, Polska12 节点测试例中 TDSF 的理论取值范围是 [0,120]。为了便于比较且不失一般性,我们假设所有节点上单位计算、存储和网络资源的开销是相等的。为了使部署节点所需的基本开销与该节点上由业务量产生的开销能够保持在同一个数量级,设定目标函数中 $C_0 = 100, S_0 = 100$ 。此外,需要根据网络拓扑中的资源分布情况通过实验确定资源权重参数 λ_1, λ_2 和 λ_3 。本次仿真实验的平台是 CPU 型号为 Intel i7 4790, 内存为 4 GB 的个人电脑。混合整数规划的求解方案选择了 CPLEX version 12.4, 在 MATLAB 环境下使用 YALMIP 接口编写了仿真程序。

EPC-NS 算法通过调节优化目标式(7)中的权重参数 λ_1, λ_2 和 λ_3 实现节点计算、存储、网络资源和链路带宽资源的自定义优化,以适应不同场景下底层网络资源分布的差异性。式(7)中, λ_1 表示节点计算资源的权重, λ_2 表示节点存储资源的权重, λ_3 表示承载 VNF 的底层节点转发业务流所消耗的网络资源权重。分析式(7)可知,对于相同的虚拟网络请求,影响链路带宽资源消耗的主要因素是底层网络中承载 VNF 的底层节点之间底层链路的跳数;影响节点计算和存储资源消耗的主要因素都是承载 VNF 的底层节点的数量,影响节点网络资源消耗的主要因素是承载 VNF 的底层节点的业务吞吐量。由于部署过程中,影响节点计算和存储资源消耗的主要因素一致,本节将不再区别讨论 λ_1 和 λ_2 的权重对部署结果的影响。

根据实验结果,本节选择 TDSF = 80 的场景下,不同权重参数的部署结果进行分析。在图 3(b)中, $\lambda_3 = 0.1, \lambda_1 = \lambda_2 = 0.45$, 此时 VNF 承载节点网络资源所占比重较小,物理节点 Poznan 在承载 MME 和 HSS 的同时还作为 TAP ↔ SGW 链路的转发节点,网络中 VNF 承载节点的吞吐量趋于饱和,链路

复用程度高, VNF 部署相对集中。当 TDSF 发生变化时,部署策略随流量调节的灵敏度过高,部署策略表现稳定性低,提高了网络管理的复杂度。在图 3(c)中, $\lambda_3 = 0.3, \lambda_1 = \lambda_2 = 0.35$, VNF 承载节点网络资源所占比重提高,链路复用减少,此时权重参数的设定不符合底层网络资源分布的特点,导致网络中底层链路跳数增多,链路带宽资源浪费。当 $\lambda_3 \geq 0.5$ 时, VNF 承载节点占用的网络资源成为影响部署结果的首要因素,此时网络中不再复用底层链路,部署策略随流量调节的灵敏度明显降低,部署策略趋于稳定。实验表明权重参数设定为 $\lambda_1 = \lambda_2 = 0.25, \lambda_3 = 0.5$ 时,目标函数能够适应 Polska12 节点测试例中资源分布的特性,适中地根据流量变化调整 VNF 和底层节点之间的映射关系,统筹优化节点和链路资源分配。

4.2 性能评估

图 4 显示了 TDSF=20 时, EPC-NS 算法的部署结果。图中, T 表示流量汇合节点的控制面功能, M, H, S, P 分别代表 MME, HSS, SGW, PGW。此时网络流量较小, EPC-NS 算法会尽量减少部署 VNF 的底层节点数量,减少 VNF 部署的基本开销。由于 PGW 的位置是固定的,算法会从候选集合 $\Omega(\text{VNF})$ 中尽可能选择距离 PGW 较近的节点进行映射以减少网络带宽资源的开销。为了充分利用 Rzeszow 节点碎片资源, EPC-NS 算法分割了 TAP 的控制面功能。图 5 显示了 TDSF=20 时, Andreas 提出的算法的部署结果。该算法同样遵循集中部署原则,但是,由于算法不支持节点分割,无法利用节点的 Bydgoszcz 节点和 Rzeszow 节点的碎片资源,只能将 MME 映射到了 Bialystok 节点上。通过对比 EPC-NS 算法更充分地利用节点资源碎片,部署业务的底层节点更少,降低了节点部署的基本开销和节点间通信的链路带宽开销。

图 6 显示了 TDSF=80 时 EPC-NS 算法的部署结果。此时网络中流量较大,单节点难以满足多个 VNF 部署所请求的资源,算法对 SGW 进行了节点

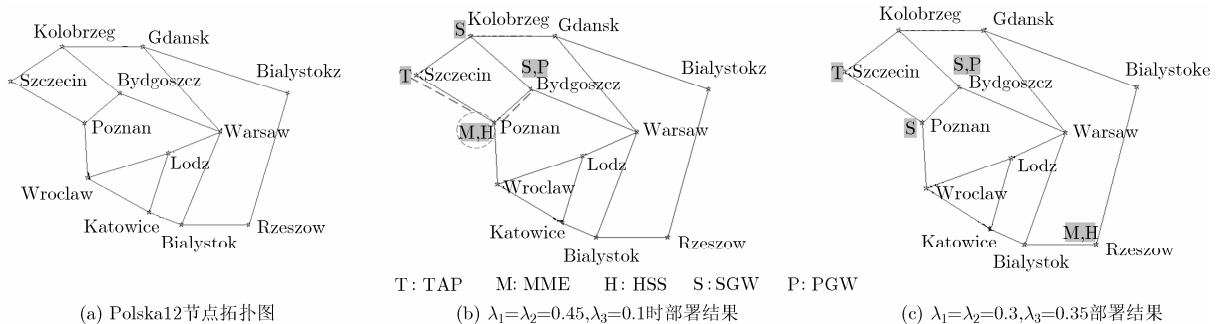


图 3 $\lambda_1, \lambda_2, \lambda_3$ 对部署结果的影响

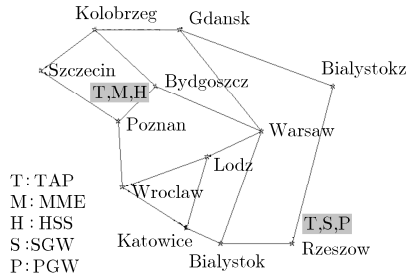


图 4 EPC-NS 算法在 TDSF=20 时的部署结果

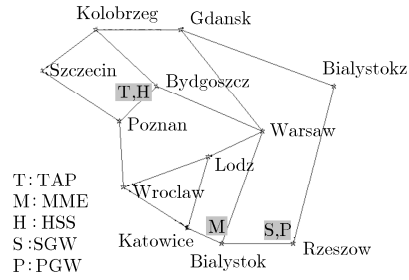


图 5 Andreas 算法在 TDSF=20 时的部署结果

分割，为了充分利用资源碎片，首先将 SGW 请求映射到 PGW 所在的 Rzeszow 节点，然后将剩余请求映射到相邻节点，最后将 Bialystok 和 Rzeszow 节点上的子 SGW 组成 SGW 池。此时，文献[5]算法已经无法利用一条服务链完成映射，如图 7 所示，Polska12 测试例被分割为两个子网，随着业务量增多该算法会增加核心网服务链的数量，为了支持不同服务链之间的跨域协同，需要提高底层网络视图的更新频率，然而频繁上报底层网络状态信息将不可避免地造成全局视图获取开销增大。综合上述两个部署结果，EPC-NS 算法能够简化虚拟网络视图，高效地利用节点碎片资源。

图 8 显示了两种算法在不同流量因子条件下的虚拟网络映射开销。映射开销由式(7)所示的目标函数定义，采用虚拟网络请求在映射过程中消耗的节点计算、存储、网络资源以及链路带宽资源的加权和表示。文献[5]算法以“网元”为颗粒度进行资源分配，当节点的剩余资源不足以承载任何 EPC“网元”时，剩余资源将成为资源碎片，在底层网络视图更新前保持不可达状态。EPC-NS 算法在混合整数规划寻优过程中选择底层链路的流量值作为决策变量，因此底层网络流量必须是最小分组的整数倍。根据文献[14]，底层节点处理单位分组所消耗的平均计算和存储资源近似为常数。综上，EPC-NS 算法将资源分配的颗粒度由“网元级”降低为处理单位数据流占用的资源片，实现了资源的细粒度分配。在相同的虚拟网络请求下，EPC-NS 部署的节点数量更少，减少了网络中非业务因素占用的基本

资源开销。当网络流量因子 $TDSF > 110$ 时，因网络中资源碎片过多文献[5]算法无法生成有效解空间，虚拟网络请求被拒绝。

图 9 给出了当虚拟网络请求服从强度 $\lambda = 6$ 的泊松分布时，EPC-NS 算法和文献[5]算法部署的请求接受率。EPC-NS 请求接受率曲线的均值为 0.9579，方差为 0.0042；文献[5]算法的均值为 0.7488，方差为 0.0184，EPC-NS 算法在请求接受率及其稳定性上都表现得更好。一方面，EPC-NS 算法颗粒度更细化，资源池中资源碎片更少。另一方面，文献[5]算法需要进行子网划分，各个子网之间相互独立，不能实现全网资源的统一调度，获得全局最优解。通过对比，EPC-NS 算法可以更加灵活地对网络切片进行调度，并能获得全网资源分配的全局最优解，在相同的底层网络状态更新频率下，映射更多的虚拟网络请求。

5 结束语

本文从 vEPC 中 VNF 的池组化部署问题入手，兼顾 5G 移动通信更高的资源需求，提出了一种支持节点分割的 VNF 部署模型。该模型借助 EPC-NS 算法将资源分配的颗粒度从传统模型的“网元级”降低到处理单位流量所占用的网络切片，提高了网络的灵活性和硬件资源利用率。仿真结果和性能评估验证了其有效性。后续将针对 vEPC 池组模型的可靠性问题进行研究，以满足 5G 移动业务的高可靠性要求。

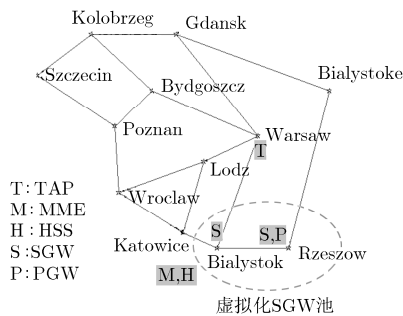


图 6 EPC-NS 算法在 TDSF=80 时的部署结果

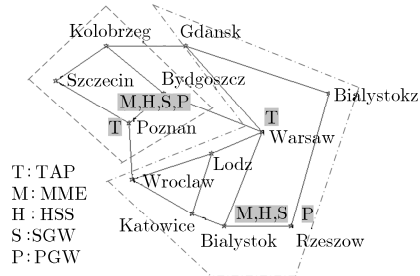


图 7 文献[5]算法在 TDSF=80 时的部署结果

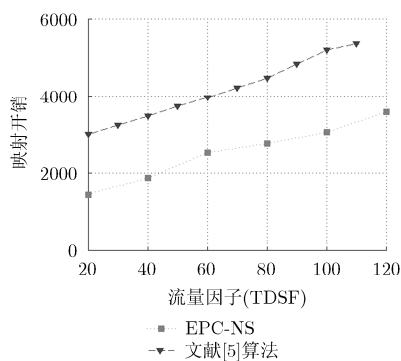


图8 网络开销对比图

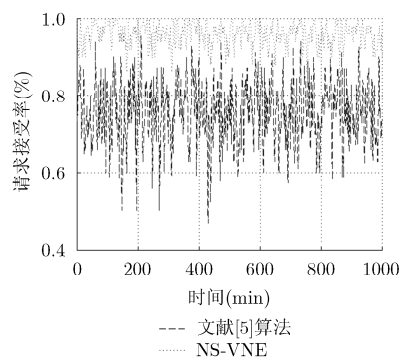


图9 请求接受率对比图

参考文献

- [1] IMT-2020(5G). 5G concept whitepaper[OL]. <http://www.imt-2020.cn/zh/documents/download/23>. 2015.
 - [2] FISCHER A, BOTERO J F, TILL BECK M, *et al*. Virtual network embedding: a survey[J]. *IEEE Communications Surveys & Tutorials*, 2013, 15(4): 1888–1906. doi: 10.1109/SURV.2013.013013.00155.
 - [3] SAMA M R, CONTRERAS L M, KAIPPALLIMALIL J, *et al*. Software-defined control of the virtualized mobile packet core[J]. *IEEE Communications Magazine*, 2015, 53(2): 107–115. doi: 10.1109/MCOM.2015.7045398.
 - [4] BASTA A, KELLERER W, HOFFMANN M, *et al*. Applying NFV and SDN to LTE mobile core gateways, the functions placement problem[C]. *ACM Workshop on All Things Cellular: Operations, Applications, & Challenges*, Chicago, USA, 2014: 33–38. doi: 10.1145/2627585.2627592.
 - [5] BAUMGARTNER A, REDDY V S, and BAUSCHERT T. Mobile core network virtualization: a model for combined virtual core network function placement and topology optimization[C]. *IEEE Conference on Network Softwarization*, London, United Kingdom, 2015: 1–9. doi: 10.1109/NETSOFT.2015.7116162.
 - [6] TALEB T, CORICI M, PARADA C, *et al*. EASE: EPC as a service to ease mobile core network deployment over cloud[J]. *IEEE Network*, 2015, 29(2): 78–88. doi: 10.1109/MNET.2015.7064907.
 - [7] CHOWDHURY M, RAHMAN M R, and BOUTABA R. Vineyard: virtual network embedding algorithms with coordinated node and link mapping[J]. *IEEE/ACM Transactions on Networking*, 2012, 20(1): 206–219. doi: 10.1109/TNET.2011.2159308.
 - [8] 李文, 吴春明, 陈健, 等. 物理节点可重复映射的虚拟网映射算法[J]. *电子与信息学报*, 2011, 33(4): 908–914. doi: 10.3724/SP.J.1146.2010.00735.
LI Wen, WU Chunming, CHEN Jian, *et al*. Virtual network mapping algorithm with repeatable mapping over substrate nodes[J]. *Journal of Electronics & Information Technology*, 2011, 33(4): 908–914. doi: 10.3724/SP.J.1146.2010.00735.
 - [9] JARRAY A and KARMOUCH A. Decomposition approaches for virtual network embedding with one-shot node and link mapping[J]. *IEEE/ACM Transactions on Networking*, 2015, 23(3): 1012–1025. doi: 10.1109/TNET.2014.2312928.
 - [10] LIU L, XU J, YU H, *et al*. A novel performance preserving VM splitting and assignment scheme[C]. *IEEE International Conference on Communications*, Sydney, Australia, 2014: 4215–4220. doi: 10.1109/ICC.2014.6883982.
 - [11] ZHOU Y, LI Y, JIN D, *et al*. A virtual network embedding scheme with two-stage node mapping based on physical resource migration[C]. *IEEE International Conference on Communication Systems*, Amsterdam, Netherlands, 2010: 761–766. doi: 10.1109/ICCS.2010.5686504.
 - [12] CHIH-LIN I, HAN S, XU Z, *et al*. 5G: rethink mobile communications for 2020+[J]. *Philosophical Transactions of the Royal Society A*, 2016, 374(2062): 20140432. doi: 10.1098/rsta.2014.0432.
 - [13] 蔡志平, 刘强, 吕品, 等. 虚拟网络映射模型及其优化算法[J]. *软件学报*, 2012, 23(4): 864–877. doi: 10.3724/SP.J.1001.2012.04063.
CAI Zhiping, LIU Qiang, LÜ Pin, *et al*. Virtual network mapping and optimization algorithm[J]. *Journal of Software*, 2012, 23(4): 864–877. doi: 10.3724/SP.J.1001.2012.04063.
 - [14] DOBRESCU M, EGI N, ARGYRAKI K, *et al*. RouteBricks: exploiting parallelism to scale software routers[C]. *ACM SIGOPS Symposium on Operating systems principles*, Big Sky, USA, 2009: 15–28. doi: 10.1145/1629575.1629578.
 - [15] ORLOWSKI S, WESSÁLY R, PIÓRO M, *et al*. SNDlib 1.0-survivable network design library[OL]. <http://sndlib.zib.de/home.action>. 2015.
- 汤红波：男，1968年生，教授，研究方向为移动通信网络、新型网络体系结构。
- 袁泉：男，1991年生，硕士生，研究方向为移动通信网络、网络功能虚拟化。
- 卢干强：男，1990年生，硕士生，研究方向为移动通信网络、网络功能虚拟化。