

doi:10.13756/j.gtxyj.2019.02.005

光通信系统与网络技术

基于节点选择策略的网络功能虚拟化映射算法

朱国晖,雷兰洁,康潇轩

(西安邮电大学 通信与信息工程学院,西安 710061)

摘要:针对网络功能虚拟化(NFV)中服务功能链的设计和映射问题,提出一种基于节点选择策略的NFV(NS-NFV)映射算法。该算法在服务功能链设计中考虑虚拟节点依赖性,根据节点资源度大小决定虚拟节点映射顺序,根据物理节点重要度为虚拟节点选择最优映射节点,并且以耦合的方式完成服务功能链的设计与映射。仿真结果表明,该算法有效地减少了回溯步长和运行时间,并且在请求接受率方面有明显改善。

关键词:网络功能虚拟化;服务功能链;节点选择策略;节点资源度

中图分类号:TN929

文献标志码:A

文章编号:1005-8788(2019)02-0019-05

Network Function Virtualization Mapping Algorithm based on Node Selection Strategy

ZHU Guo-hui, LEI Lan-jie, KANG Xiao-xuan

(School of Communications and Information Engineering, XUPT, Xi'an 710061, China)

Abstract: Aiming at the design and mapping of service function chain in Network Function Virtualization (NFV), a NFV mapping algorithm based on Node Selection strategy (NS-NFV) is proposed. Considering the virtual node dependency in the design of service function chain, NS-NFV determines the mapping order of virtual nodes according to the node resource degree. It also selects the optimal mapping node for the virtual node according to the importance of the physical node, and completes the design and mapping of service function chain in a coupling manner. Simulation results show that NS-NFV can effectively reduce the backtracking steps and running time. In addition, the request acceptance rate can be improved significantly.

Key words: NFV; service function chain; node selection strategy; node resource degree

0 引言

网络功能虚拟化(Network Function Virtualization, NFV)可将软件从硬件平台分离,实现软硬件解耦^[1-2]。实现NFV最重要的挑战是如何为给出的一组虚拟网络功能(Virtual Network Function, VNF)设计VNF图(VNF Graph, VNFG)并将其映射到物理网络。在传统虚拟网络映射(Virtual Network Embedding, VNE)中^[3-4],需要将虚拟节点和链路以最优的方式映射到物理网络,这被证明是NP-Hard问题^[3]。

目前,虽然VNE得到了广泛的研究^[4-9],但针对VNFG映射问题的研究还很少。文献[1]在生成VNFG后运用混合整数二次约束算法(Mixed Integer Quadratic Constrained Program, MIQC)完成了VNFG映射,其虽节省了资源但却增加了算法的运行时间;文献[10]在VNFG设计中考虑了虚拟节点的依赖性,但主要关注点在VNFG的设计上,并未描述有效的映射算法;文献[11]提出的改进拓扑排序算法(Modified Topological Sort Algorithm, MT-

SA)未考虑网络拓扑对算法实现资源节省目标的影响;文献[12]提出的可协调性VNF(Coordinated VNF, CoordVNF)算法同时实现了VNFG的设计和映射,但在映射过程中随机选择底层节点会导致回溯步长增加。

针对上述算法的不足之处,本文提出了一种基于节点选择策略的NFV(Node Selection NFV, NS-NFV)算法。NS-NFV算法解决了VNFG的设计和映射问题,达到了减少回溯步长和运行时间并提高请求接受率的目标;在VNFG的设计中,采用比较节点资源度大小的方法来确定依赖关系不明确的虚拟节点的映射顺序;在选择映射节点时,加入了节点重要度因素并且结合了物理网络自身的拓扑属性。

1 网络模型与评价指标

1.1 网络模型

(1) 物理网络:采用带权重的无向图 $G_s = (N_s, L_s)$ 来表示物理网络。式中: N_s 为物理节点 n_s 的集合, $n_s \in N_s$,包含两个属性:节点资源能力 $C(n_s)$ 及

收稿日期:2018-09-06

作者简介:朱国晖(1969—),男,北京人。副教授,博士,主要研究方向为移动互联网、对等网络和复杂网络路由算法。

通信作者:雷兰洁(1993—),硕士。E-mail:952633581@qq.com

节点功能 $F(n_s)$; L_s 为物理链路 l_s 的集合, $l_s \in L_s$ 的可用带宽需求为 $B(l_s)$ 。

(2) VNFG: 采用带权重的有向图 $G_v = (N_v, L_v, I)$ 表示 VNFG 请求。式中: N_v 为虚拟节点 n_v 的集合, $n_v \in N_v$, 包含两个属性: 节点资源需求 $C(n_v)$ 及节点功能 $F(n_v)$; L_v 为虚拟链路 l_v 的集合, $l_v \in L_v$ 的可用带宽需求为 $B(l_v)$; I 为初始数据流量。

(3) VNFG 设计: 在 VNFG 中, 用箭头连接的两个节点表示箭头所指向的节点取决于箭头所出的节点, 即后面节点必须在前面节点之后执行; 实线表示节点序列已经确定, 虚线表示节点序列是灵活的。图 1 所示为 VNFG, 图中依赖关系为从节点 a 到 h, 节点 b 和 c 在 a 后面, 节点 d 在 c 后面, 节点 b 和 d 在 e 前面, 节点 e 到 h 是确定的, 节点 b、c 和 d 的序列不确定, 可以是: $b \rightarrow c \rightarrow d$, $c \rightarrow b \rightarrow d$ 或 $c \rightarrow d \rightarrow b$ 。

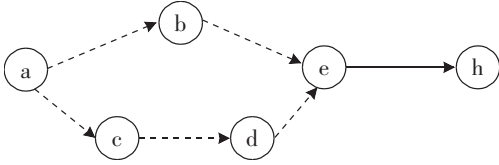


图 1 VNFG

(4) VNFG 映射: 将 VNFG 请求 $G_v = (N_v, L_v, I)$ 映射到物理网络 $G_s = (N_s, L_s)$ 中, 这意味着将每一个虚拟节点映射到物理节点上, 将每条虚拟链路映射到物理网络的无环路径上。

节点映射: 记为 $M_N: N_v \rightarrow N_s$ 。不同 VNFG 中的节点能映射到相同的物理节点上, 需要满足的资源约束条件为

$$F(n_v) \in F(n_s); \quad (1)$$

$$C(n_v) \leq C(n_s)。 \quad (2)$$

对于任意 $n_v \in N_v$, 如果物理节点 $n_s \in N_s$ 满足 n_v 的功能和资源需求, 则节点 n_v 映射到节点 n_s 上。

链路映射: 记为 $M_L: L_v \rightarrow L_s$ 。每条虚拟链路可映射到一条包含多条物理链路的路径上, 每条物理链路的可用带宽需求需满足:

$$B(l_v(p, t)) \leq B(l_s(p', t'))。 \quad (3)$$

对于任意虚拟链路 $l_v(p, t) \in L_v$ 和物理链路 $l_s(p', t') \in L_s$, 虚拟节点 p 和 t 分别映射到物理节点 p' 和 t' 。如果 $l_s(p', t')$ 可以满足 $l_v(p, t)$ 的可用带宽需求, 则虚拟链路 $l_v(p, t)$ 被映射到物理网络的无环链路 $l_s(p', t')$ 上。

1.2 评价指标

物理网络的请求接受率:

$$A_{\text{accept}} = \frac{\sum VNFR_a}{\sum VNFR}, \quad (4)$$

式中: $VNFR_a \in \{0, 1\}$ 为二进制变量, $VNFR_a = 1$ 表示 VNF 请求映射成功, $\sum VNFR_a$ 为映射成功的 VNF 请求总数; $\sum VNFR$ 为 VNF 请求的总数。

2 算法设计

本文主要解决的问题是确定虚拟节点映射顺序及完整的请求图映射。在节点映射算法中选择候选物理节点时, 考虑了物理网络的邻接关系和网络本身的拓扑结构^[13], 引入接近中心度和广度优先搜索算法^[14-15], 并且在链路映射时考虑到运行时间主要取决于转发时间, 所以限制最大路径长度为 k 。

2.1 物理 NS 策略

受文献[13]启发, 算法在进行物理 NS 时, 考虑物理网络自身的拓扑属性并结合网络接近中心度和节点的度, 以达到减少算法运行时间和回溯步骤的目的。

(1) 节点资源度表示节点所能承载的资源能力和可用带宽需求的乘积:

$$H(n_s^i) = C(n_s^i) \sum_{l_s \in L(n_s^i)} B(l_s), \quad (5)$$

式中: $C(n_s^i)$ 为节点资源能力; $B(l_s)$ 为节点可用带宽需求; $L(n_s^i)$ 为物理节点 n_s^i 邻接的链路集合。

(2) 接近中心度被定义为此节点到其他所有节点之间最短路径的总和的倒数:

$$Q(n_s^i) = \frac{1}{\sum_{j=1}^N S(n_s^i, n_s^j)}, \quad (6)$$

式中: $S(n_s^i, n_s^j)$ 为节点 n_s^i 和 n_s^j 之间的最短路径长度, 其限制为 k ; N 为节点的总和。

(3) 节点的度表示此节点与邻节点之间直连的链路数目。节点 n_s^i 的度可表示为

$$D(n_s^i) = \text{dgr}(n_s^i)。 \quad (7)$$

为提高映射效率, 为节点资源度加入接近中心度及节点的度这两个因素, 此时节点资源能力可进一步表示为

$$W(n_s^i) = H(n_s^i) Q(n_s^i) D(n_s^i) = \text{dgr}(n_s^i) \sum_{l_s \in L(n_s^i)} B(l_s) C(n_s^i) \frac{1}{\sum_{j=1}^N S(n_s^i, n_s^j)}。 \quad (8)$$

对于每一个待映射的虚拟节点 n_v^i , 寻找满足约束条件(1)和(2)的物理节点集合 $\varphi(n_v^i)$, 为了使映

射节点距离近和范围小,定义 $E(n_s^i)$ 为被选择的物理节点到与待映射虚拟节点相邻且已映射的虚拟节点的目标物理节点集合 $O(n_v^i)$ 中所有物理节点之间总的路径跳数:

$$E(n_s^i) = \sum_{n_v^j \in O(n_v^i)} S(n_s^i, n_v^j), \quad n_s^i \in \varphi(n_v^i), \quad (9)$$

则物理节点的重要程度可表示为

$$R(n_s^i) = \frac{W(n_s^i)}{E(n_s^i) + \sigma} = \frac{H(n_s^i)Q(n_s^i)D(n_s^i)}{E(n_s^i) + \sigma}, \quad n_s^i \in \varphi(n_v^i), \quad (10)$$

式中, $\sigma = 0.001$ 保证分母不等于零。

物理 NS 策略算法如下:

输入: G_s, G_v ;

输出: Node Mapping Result;

步骤 1: 对于每一个虚拟节点 $n_v^i \in N_v$, 选取满足约束条件(1)和(2)的候选物理节点集合 $\varphi(n_v^i)$ 和 $O(n_v^i)$;

步骤 2: 如果 $\varphi(n_v^i)$ 为空, 则返回“节点映射失败”;

步骤 3: 否则, 对 $\varphi(n_v^i)$ 中的每个物理节点 n_s^i 计算其 $R(n_s^i)$;

步骤 4: 将 n_v^i 映射到物理节点集合 $\varphi(n_v^i)$ 中 $R(n_s^i)$ 值最大的物理节点上并返回“节点映射成功”;

步骤 5: End。

2.2 虚拟 NS 策略

针对一些不能确定映射顺序的节点, 如图 1 中的 b、c 和 d, 提出虚拟 NS 策略。因为请求图是有向的, 并且为链式结构, 所以节点资源度可以用此节点的资源需求与输入的链路的可用带宽资源的乘积表示。虚拟节点 n_v^i 的资源需求为

$$C(n_v^i) = N_c(n_v^i)B(l_v), \quad (11)$$

式中: $N_c(n_v^i)$ 为虚拟节点 n_v^i 的相对处理能力; $B(l_v)$ 为虚拟节点 n_v^i 可用带宽需求, 其为上一节点带宽与其相对流出流量率的乘积。则此节点的节点资源度为

$$H(n_v^i) = C(n_v^i)B(l_v) = N_c(n_v^i)(B(l_v))^2. \quad (12)$$

通过比较两个节点的节点资源度大小来确定节点的映射顺序。

图 2 所示为一个简单的基于图 1 的 VNFG 映射过程。首先, 从节点 a 开始, 由于节点 a 的映射顺序是确定的, 则对其映射处理执行物理 NS 策略算法即可, 如图 2(a)所示; 与节点 a 相连的节点 b 和

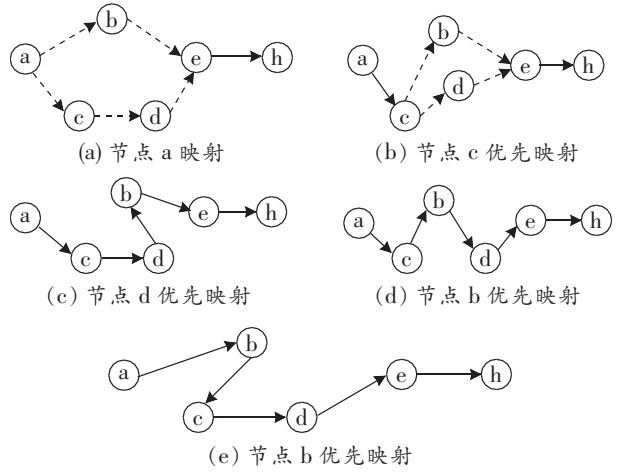


图 2 基于 NS-NFV 映射的简单示例

c, 由于其映射顺序不确定, 则根据式(12)比较节点 b 和 c 的资源度大小, 若 $H(c) > H(b)$, 则先映射节点 c, 对其执行物理 NS 策略算法, 并运用最短路径算法映射节点 a 和 c 间的链路, 如图 2(b)所示; 类似地比较节点 b 和 d, 若 $H(d) > H(b)$, 则如图 2(c)所示, 反之则如图 2(d)所示; 若 $H(b) > H(c)$, 则结果如图 2(e)所示。

2.3 NS-NFV 算法

NS-NFV 算法的具体步骤如下:

输入: 一组 VNF;

输出: VNFG Mapping Result;

步骤 1: 根据 VNF 之间的依赖性关系初步设计 VNFG;

步骤 2: 从端节点开始映射, 如果虚拟节点映射顺序是确定的, 则进入步骤 4;

步骤 3: 若虚拟节点映射顺序不确定, 则根据式(12)比较 $H(n_v^i)$ 大小, $H(n_v^i)$ 大的虚拟节点优先映射, 进入步骤 4;

步骤 4: 对于虚拟节点 $n_v^i \in N_v$, 选取满足约束条件(1)和(2)的候选物理节点集合 $\varphi(n_v^i)$ 和 $O(n_v^i)$;

步骤 5: 如果 $\varphi(n_v^i)$ 为空, 则返回“映射失败”;

步骤 6: 否则, 对 $\varphi(n_v^i)$ 中的每个物理节点 n_s^i 计算其 $R(n_s^i)$;

步骤 7: 将 n_v^i 映射到物理节点集合 $\varphi(n_v^i)$ 中 $R(n_s^i)$ 值最大的物理节点上;

步骤 8: 链路映射, 在满足约束式(3)的条件下, 运用最短路径算法将连接两个虚拟节点之间的虚拟链路映射至对应物理路径上;

步骤 9: 执行此算法直到 VNFG“映射成功”或者“映射失败”;

步骤 10: End。

3 仿真分析

本文将在不同的最大路径长度 k 下评估本文所提算法的回溯步长、运行时间和接受率。本文选择文献[11]和[12]所提出的 MTSA 和 CoordVNF 算法作为比较算法。

3.1 仿真环境

为了验证算法的有效性,本文所提算法的仿真环境与比较算法^[11-12]的基本一致。实验使用 JAVA 语言通过 Myeclipse 工具在 64 位的 Windows7 操作系统平台上进行仿真评估。采用 GT-ITM 工具生成底层物理网络和 VNFG。底层物理网络随机生成 50 个节点,每个节点的中央处理器(Central Processing Unit,CPU)资源在 50~100 units之间随机产生,并支持 4 种不同的 VNF 类型;物理链路以 0.5 的概率链接,每条链路随机分配 50~100 bandwidth units 的带宽资源;随机生成 10 个节点的 VNFG,VNF 之间的依赖性随机选择;初始数据率设置为 50 bandwidth units,虚拟节点的相对处理能力设置为 1 unit,且相对流量率被随机设置在 50%~150%。为了确保仿真的可靠性和稳定性,在不同的参数设置下进行了多次实验。

3.2 仿真结果

由于运行时间主要取决于包的转发次数,所以可以忽略其传输时间。最大路径长度 k 是两个连接的虚拟节点映射到物理节点之间的链路跳数,它会对运行时间和映射成功率造成影响。因此,会以不同的 k 值来评估本文的算法。

本文算法限制 k 值为 1~5。不同 k 值下 NS-NFV、MTSA 和 CoordVNF 算法的回溯步长比较如图 3 所示。由图可知,3 种算法的回溯步长都随着 k 值的增加而增加,但从 $k=4$ 开始,当所有节点

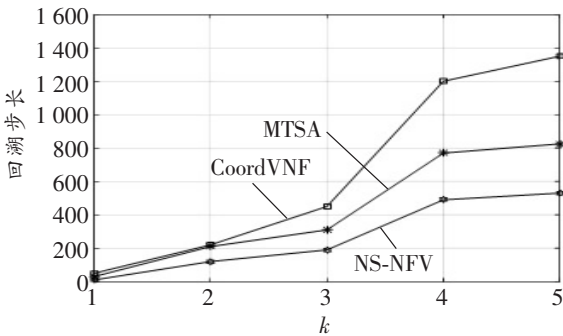


图 3 不同 k 值下,NS-NFV、MTSA 和 CoordVNF 算法的回溯步长比较

和链路在物理网络中被检测到时,回溯步长增长趋势减缓。相比于 MTSA 和 CoordVNF 算法,NS-NFV 算法对虚拟节点和其候选物理节点都做了降序排列处理,因此在很大程度上减少了回溯步长。

运行时间是评估算法的一个重要标准。不同 k 值下 NS-NFV、MTSA 和 CoordVNF 算法的运行时间比较如图 4 所示。由图可知,3 种算法的运行时间也随着 k 值的增加而增加,且增长趋势也从 $k=4$ 时开始减缓,但可明显看出,NS-NFV 比 MTSA 和 CoordVNF 算法的运行时间更短。这是由于 NS-NFV 算法的回溯步长小于 MTSA 和 CoordVNF 算法。由图 3 和 4 可知,回溯步长在很大程度上影响了算法的运行时间。

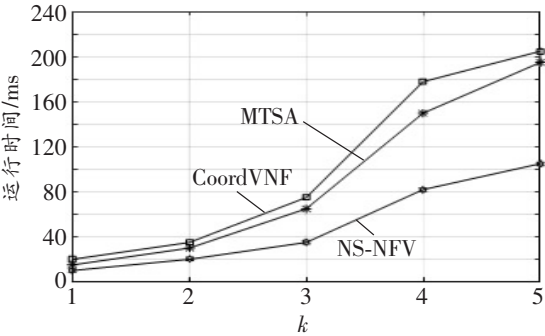


图 4 不同 k 值下,NS-NFV、MTSA 和 CoordVNF 算法的运行时间比较

图 5 比较了不同 k 值下 NS-NFV、MTSA 和 CoordVNF 算法的接受率。总体上,3 种算法的接受率都是随着 k 值的增加而增长的,这是因为随着搜索空间变大,其选择映射节点和链路的可能性变大,增加了算法的接受率。由图可知, $k=1$ 时,3 种算法的接受率不相上下; $k>2$ 时,由于本文算法考虑了物理网络拓扑属性,加入了节点重要度算法,使得 NS-NFV 算法的接受率优于另外两种算法。

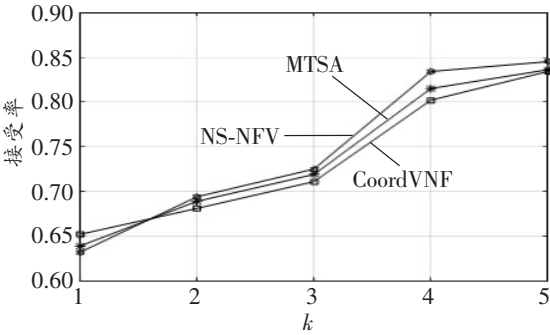


图 5 不同 k 值下,NS-NFV、MTSA 和 CoordVNF 算法的接受率比较

4 结束语

随着 NFV 的快速发展, VNF 代替了配置在物理中间盒子上的网络功能^[16]。为了解决将 NFV 应用于服务功能链时带来的两个问题:对给定的 VNF 请求拓扑进行设计和对设计好的 VNF 拓扑进行映射,本文提出了 NS-NFV 映射算法。为了验证本文所提算法的有效性,选择 MTSA 和 CoordVNF 算法作为比较算法,并在不同 k 值下比较了算法性能。仿真结果表明,所提算法在增加请求接受率的同时有效地减少了回溯步长和运行时间。

参考文献:

- [1] Mehraghdam S, Keller M, Karl H. Specifying and Placing Chains of Virtual Network Functions [C]//International Conference on Cloud Networking. Luxembourg, Luxembourg: IEEE, 2014: 7–13.
- [2] 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.
- [3] Fischer A, Botero F J, Beck M T, et al. Virtual Network Embedding: A Survey [J]. IEEE Communications Surveys & Tutorials, 2013, 15(4): 1888–1906.
- [4] Yu M, Yi Y, Rexford J, et al. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 17–29.
- [5] Cheng X, Su S, Zhang Z, et al. Virtual Network Embedding Through Node and Topology-Aware Node Ranking [J]. ACM SIGCOMM Computer Communication Review, 2011, 41(2): 38–47.
- [6] Chowdhury N, Rahman M R, Boutaba R. Virtual Network Embedding with Coordinated Node and Link Mapping [C]//INFOCOM. Rio de Janeiro, Brazil: IEEE, 2009: 10685483.
- [7] Li Y, Li Y, Shu M, et al. An Efficient VNE Algorithm via Preferentially Mapping Important Nodes [C]//Local Computer Networks Workshops. Dubai, United Arab Emirates: IEEE, 2016: 34–41.
- [8] 孙昱. 虚拟网络的映射技术研究 [D]. 成都: 电子科技大学, 2012.
- [9] Lischka J, Karl H. A Virtual Network Mapping Algorithm based on Subgraph Isomorphism Detection [C]//ACM Workshop on Virtualized Infrastructure Systems and Architectures. New Delhi, India: ACM, 2009: 81–88.
- [10] Jalalitar M, Luo G, Kong C, et al. Service Function Graph Design and Mapping for NFV with Priority Dependence [C]//Global Communications Conference. Washington, DC, USA: IEEE, 2017: 16655040.
- [11] You X M, Wang X X, Chen A G, et al. A Coordinated Algorithm with Resource Evaluation for Service Function Chain Allocation [C]//International Conferences on Big Data & Cloud Computing. Atlanta, GA, USA: IEEE, 2016: 45–49.
- [12] Beck M T, Botero J F. Coordinated Allocation of Service Function Chains [C]//Global Communications Conference (GLOBECOM). San Diego, CA, USA: IEEE, 2015: 15820751.
- [13] Cui H, Gao W, Liu J, et al. A Virtual Network Embedding Algorithm based on Virtual Topology Connection Feature [C]//International Symposium on Wireless Personal Multimedia Communications. Atlantic City, NJ, USA: IEEE, 2013: 13825356.
- [14] Feng M, Liao J, Wang J, et al. Topology-Aware Virtual Network Embedding based on Multiple Characteristics [C]//IEEE International Conference on Communications. Sydney, NSW, Australia: IEEE, 2014: 14546049.
- [15] 张研, 韩露. 用广度优先搜索算法实现路径搜索 [J]. 电脑编程技巧与维护, 2012(19): 78–81.
- [16] Soualah O, Mechtri M, Ghribi C, et al. An Efficient Algorithm for Virtual Network Function Placement and Chaining [C]//Consumer Communications & Networking Conference. Las Vegas, USA: IEEE, 2017: 647–652.