

可靠的虚拟网络映射算法研究

刘光远¹, 苏森²

(1. 石家庄铁道大学信息科学与技术学院, 河北石家庄 050043; 2. 北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

摘要: 网络虚拟化技术允许多个异构的虚拟网络共享一个底层物理网络资源, 为目前的网络架构提供了一种有效的扩展手段. 近年来, 底层网络基础设施失效事件频发, 因此如何提高虚拟网络的可靠性成为目前该领域一个研究热点. 本文针对底层节点失效后虚拟拓扑如何最大化连通问题进行研究, 设计了一种基于割集和拥塞感知的虚拟网络映射机制. 实验表明, 该方法在不预留保护资源的情况下, 可获得更好的底层网络长期运行平均收益.

关键词: 网络虚拟化; 虚拟网络映射; 最大化虚拟拓扑连通; 割集和拥塞感知

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2016)08-1820-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.08.007

The Research of Reliable Virtual Network Mapping Algorithm

LIU Guang-yuan¹, SU Sen²

(1. School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang, Hebei 050043, China;

2. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Network virtualization has been proposed as a promising way for running multiple customized virtual networks (VNs) on a shared infrastructure. However, how to provide reliable VN against substrate infrastructure failures has become an increasingly important issue. In this paper, we present a novel VN mapping scheme based on cutset and congestion awareness for the VN topology remain maximizing connected in the event of single substrate node failure. Simulation results show that algorithm can gain more optimal substrate long-term average revenue compared to the previous algorithms without reserving protection resource.

Key words: network virtualization; virtual network mapping; maximizing VN topology connected; cutset and congestion awareness

1 引言

网络虚拟化技术可以有效解决互联网的“僵化”问题. 在网络虚拟化环境下, 传统的 Internet 服务运营商被分为两个角色, 即底层网络运营商和服务提供商. 底层网络运营商负责管理和维护底层网络资源; 服务提供商向底层基础设施运营商租用底层网络资源创建虚拟网络, 并向用户提供端到端的网络服务^[1]. 虚拟网络映射是网络虚拟化研究的关键问题之一, 受到了国内外学者的广泛关注. 每个虚拟网络可看作底层物理网络的一份资源切片, 包含虚拟节点和虚拟链路. 为服务提供商的带有节点和链路资源约束条件的虚拟网络请求分配基础设施提供商的底层网络资源的问题称为虚拟网络映射(嵌入、指派)问题^[2]. 该问题已被证明为 NP-

Hard 问题. 为了提高底层网络运营商的收益, 早期研究主要集中于如何提高虚拟网络映射的成功率和底层网络资源利用率^[3-5].

由于黑客攻击或硬软件错误, 底层物理网络的部分节点和链路可能会失效. 然而, 在网络虚拟化环境中大量的虚拟网络共存于同一底层网络之上, 即使单个底层节点或单条底层链路失效就会导致大量的虚拟网络不可用. 如果基础设施提供商不采取措施应对底层节点或链路故障, 将会使租用虚拟网络的网络服务提供商蒙受巨大经济损失, 并对基础设施提供商所提供的虚拟网络服务丧失信心. 为了解决上述问题, 近年来对虚拟网络可靠性问题的研究逐渐增多, 文献^[6,7]主要基于先验式恢复机制, 在底层节点和链路失效时将其替换为预留保护资源, 该类方法实现简单, 但需要消耗

额外的底层物理计算资源和网络带宽资源,底层资源利用效用不高,同时会增加用户的使用成本.对于轻量级可靠的虚拟网络服务映射请求(在底层节点或链路失效时,需维持除失效节点或链路外的虚拟网络拓扑连通),目前研究还很少.文献^[8]在未分配底层网络保护资源的情况下,对底层单链路失效后如何保持虚拟网络拓扑连通的映射方法进行了研究,但该方法无法解决底层单节点失效问题.

本研究的目的在于设计一种新的面向底层单节点失效的轻量级可靠虚拟网络映射算法,该算法可不预留底层保护资源,在底层物理单节点失效后,虚拟网络拓扑除失效节点外仍保持连通,从而尽可能降低服务运营商的损失.

本文首先对面向节点失效的虚拟网络映射问题进行描述,然后建立该问题的整数线性规划模型,最后提出一种基于割集和拥塞感知的虚拟网络映射算法对该问题进行求解.

2 面向节点失效的虚拟网络映射问题的形式化描述

2.1 虚拟网络请求

虚拟网络拓扑可被标记为带权无向图 $G_v = (N_v, E_v, R_v^n, R_v^l)$, 其中 N_v 为虚拟网络节点的集合, E_v 为虚拟网络链路的集合. R_v^n 表示虚拟节点能力需求, R_v^l 表示虚拟链路带宽资源需求. 当第 i 个虚拟网络请求到达后, 物理网络需要为其分配满足其节点与链路资源需求的相应资源. 当物理网络资源不足时, 应该拒绝或延迟映射该虚拟网络请求. 图 1(a) 描绘了一个带有节点与链路资源约束的虚拟网络请求实例.

2.2 底层网络

与虚拟网络类似, 底层物理网络拓扑也可被标记为带权无向图 $G_s = (N_s, E_s, C_s^n, C_s^l)$, 其中 N_s 表示物理网络节点的集合, E_s 表示物理网络链路的集合, C_s^n 表示物理网络节点的计算能力, C_s^l 表示物理网络链路带宽. 图 1(b) 描绘了一个底层物理网络实例, 其中矩形框里的数字分别表示各个节点的可用计算资源, 链路附近的

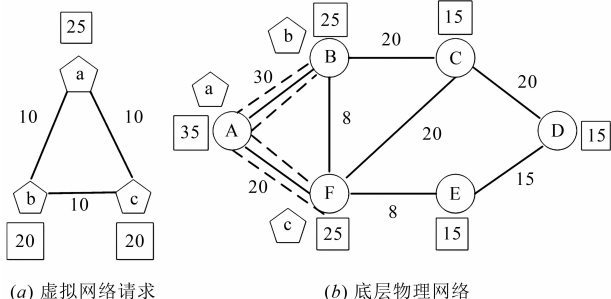


图1 虚拟网络映射实例

数字表示该链路的可用带宽资源.

2.3 面向节点失效的虚拟网络映射问题描述

虚拟网络映射问题被定义为映射: $M: G_v(N_v, E_v) \rightarrow G_s(N_s, E_s)$, 包括节点和链路映射两个方面. 图 1(b) 给出了虚拟网络请求的一种映射方案. 如图所示, 节点映射方案为 $\{a \rightarrow A, b \rightarrow B, c \rightarrow F\}$, 链路映射方案为 $\{(a, b) \rightarrow (A, B), (a, c) \rightarrow (A, F), (b, c) \rightarrow (B, A, F)\}$.

因为每个底层链路的带宽资源可被多条虚拟链路所共享, 所以两个或更多的虚拟链路可能被映射到同一条底层链路上. 当一个底层单节点失效时, 其周围相连的链路也随即失效, 这会导致映射到这些底层链路上的虚拟网络中的多条虚拟链路同时失效, 进而导致已经映射的虚拟网络不再连通, 虚拟网络服务不可用. 如图 1 所示, 底层节点 A 失效, 与其相连的底层链路 (A, B) 和 (A, F) 随即失效, 则映射在其上的虚拟链路 (a, b), (a, c), (b, c) 均失效, 导致虚拟网络拓扑不再连通. 因此, 我们提出面向节点失效的虚拟网络映射问题, 即当底层网络中发生单一节点故障时, 受到影响的虚拟网络仍能够最大限度保持拓扑连通性. 如图 2 所示, 将虚拟网链路 (b, c) 映射到 (B, C, F) 上, 此时底层节点 A 失效, 虚拟拓扑中的 (b, c) 部分仍保持联通.

本文的工作的内容就是要设计高效的映射算法以解决面向节点失效的虚拟网络映射问题. 我们首先对该问题进行数学建模, 然后提出一种基于割集和拥塞感知的虚拟网络映射算法对该问题进行求解.

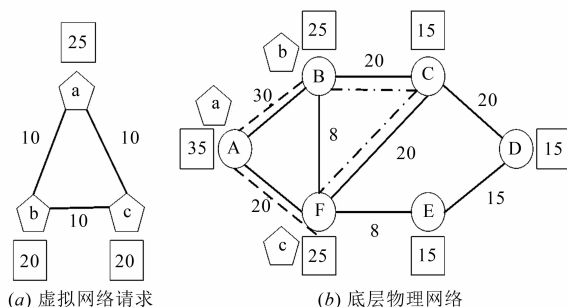


图2 虚拟网络映射实例改进方案

3 面向节点失效的虚拟网络映射问题的整数线性规划模型

本文以降低虚拟网络映射开销为目标, 建立面向节点失效的虚拟网络映射问题的整数线性规划模型.

变量:

f_{ij}^{uv} : 二进制变量. 在映射过程中, 如果物理链路 (i, j) 承载了虚拟链路 (u, v) , 该变量取值 1, 否则取值 0.

x_i^u : 二进制变量. 在映射过程中, 如果虚拟节点 u 被映射到物理节点 i 上, 该变量取值为 1, 否则取值为 0.

目标函数: *Minimize*

$$\sum_{(u,v) \in L_s} \sum_{(i,j) \in L_s} f_{ij}^{uv} \times BW(l_{uv}) \quad (1)$$

资源约束:

$$\forall u \in N_v, \forall i \in N_s, x_i^u \times CPU(u) \leq CPU(i) \quad (2)$$

$$\forall (i,j) \in L_s, \forall (u,v) \in L_v, f_{ij}^{uv} \times BW(l_{uv}) \leq BW(l_{ij}) \quad (3)$$

连接性约束:

$$\forall i \in N_s, \forall (u,v) \in L_v, \sum_{(i,j) \in L_s} f_{ij}^{uv} - \sum_{(j,i) \in L_s} f_{ji}^{uv} = \begin{cases} 1, & \text{if } x_i^u = 1 \\ -1, & \text{if } x_i^v = 1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

节点可靠性约束:

$$\forall (i,j) \in L_s, \forall M_v \subset N_v, \sum_{(u,v) \in CS(M_v, N_v - M_v)} f_{ij}^{uv} + f_{ji}^{uv} < |CS(M_v, N_v - M_v)| \quad (5)$$

变量约束:

$$\forall i \in N_s, \sum_{u \in N_v} x_i^u = 1 \quad (6)$$

$$\forall u \in N_v, \sum_{i \in N_s} x_i^u = 1 \quad (7)$$

$$\forall i \in N_s, \forall u \in N_v, x_i^u \in \{0, 1\}$$

$$\forall (i,j) \in L_s, \forall (u,v) \in L_v, f_{ij}^{uv} \in \{0, 1\}$$

说明:

式(1)为虚拟网络映射的整数规划模型的目标函数,该目标函数只包含了底层链路的带宽资源开销.这是因为不同映射方案的底层节点计算资源开销是恒定的,而底层链路带宽资源开销是不同的.

式(2)和(3)给出了节点和链路的资源约束条件.在节点资源约束方面,底层网络节点的 CPU 能力必须能够满足虚拟网络节点的 CPU 能力需求.在链路资源约束方面,底层链路的带宽能力必须满足虚拟链路的带宽能力需求.

式(4)是连接性约束条件.如果在虚拟节点映射阶段,虚拟节点 u 和 v 分别映射到底层节点 i 和 j ,那么在虚拟链路映射阶段,虚拟链路 (u,v) 将被映射到从节点 i 到节点 j 的一条底层路径 P 上.在源点 i 上,流出的流量为 1,流入的流量为 0,所以 $\sum_{(i,j) \in L_s} f_{ij}^{uv} - \sum_{(j,i) \in L_s} f_{ji}^{uv} = 1$;在汇点 j 上,流出的流量为 0,流入的流量为 1,所以 $\sum_{(i,j) \in L_s} f_{ij}^{uv} - \sum_{(j,i) \in L_s} f_{ji}^{uv} = -1$;而在路径 P 其它节点上,流出的和流入的流量均为 1,所以 $\sum_{(i,j) \in L_s} f_{ij}^{uv} - \sum_{(j,i) \in L_s} f_{ji}^{uv} = 0$.

式(5)是面向节点失效的虚拟网络的可靠性约束,它表示虚拟网络的每一个割集中的所有虚拟链路被映射到同一条底层链路上的数量要小于割集中虚拟链路的总数.这表明属于同一割集的虚拟链路不会被映射到同一条底层链路上.其中, $CS(M_v, N_v - M_v)$ 表示虚

拟网络的一个割集, M_v 和 $\overline{M_v} = N_v - M_v$ 表示被该割集划分的两个不相交的虚拟节点集合. $|CS(M_v, N_v - M_v)|$ 表示割集中虚拟链路的总数.

式(6)表示对于同一个虚拟网络请求,一个底层节点只能承载一个虚拟节点,一个虚拟节点也只能映射到一个底层节点上.

式(7)表示 f_{ij}^{uv} 变量和 x_i^u 的二进制约束.

4 基于割集和拥塞感知的虚拟网络映射算法描述

由于解决整数线性规划问题的复杂性是 NP 难的,因此获得最优的虚拟网络映射方案也是 NP 难的.目前解决该方法的方法从大的方面分为两类:精确算法和启发式算法,分支限界或割平面等传统解决整数规划的方法在解决小规模问题时可得到最优的性能.但当求解问题规模较大时,计算量巨大,无法在合理的时间给出解.所以本设计了一种启发式算法 CCA-RVNM 来解决较大规模虚拟网络映射的整数线性规划问题.算法的目标如下:给定一个虚拟网络请求和底层网络资源,虚拟网络映射方案首先应该满足虚拟网络轻量级可靠需求(在底层节点失效时,需维持除失效节点外的虚拟网络拓扑连通),其次虚拟网络映射消耗的底层网络资源应该尽可能小.

CCA-RVNM 算法为一阶段算法,即映射节点和映射链路在同一阶段进行.在进行虚拟节点映射时,CCA-RVNM 基于广度优先搜索方式映射虚拟节点,目的是方便节点映射回溯处理.算法首先计算所有底层节点可提供的计算能力和虚拟节点的需求能力值,然后基于虚拟网络拓扑构建广度优先搜索树,且树中每一层节点按照其需求能力值降序排列.虚拟节点的需求值越大,则其被映射的优先级越高.每当一个虚拟网络节点映射成功后,便映射与其相连接的虚拟链路.这样可以尽可能地避免虚拟网络链路被映射到一条底层长路径上,从而降低虚拟网络映射引入的底层网络资源开销.所以我们在算法实现上引入了节点之间的距离参数 K ,可以理解为两个节点之间的跳数,根据设置上限值,如果超过了该值就停止该次映射并进行回溯,重新映射上一节点.在虚拟链路映射时,为了满足在底层节点失效后维持虚拟拓扑最大化连通,我们提出了一种割集和拥塞感知的虚拟链路映射策略.割集感知的设计思想是基于前节整数线性规划模型中公式(5)的约束条件,目的是保证虚拟网络拓扑在底层节点失效后最大化连通.拥塞是指当前物理链路负载的大小,因为负载大的物理链路承载的虚拟链路可能多,一旦失效,受影响的范围相对较大,因此拥塞感知的思想就是在满足割集约束的前提下尽量平衡底层链路负载,降低链路失效所带来的风险.具体地来讲,在映射每

一条虚拟链路前,我们首先检查是否存在一个包含此条虚拟链路的割集,在该割集中,其他的虚拟链路是否已经被映射到同一个底层物理节点直接相连的路径上.这是对公式(5)的直接应用.如果存在这样的一个割集,那么在该割集中已经被映射的虚拟链路所共享的与底层同一节点相连的链路将被标记,在使用最小拥塞算法映射当前的虚拟链路时,有标记的底层网络链路将不会被考虑;如果不存在这样一个割集,那么直接使用最小拥塞算法映射虚拟链路.算法 1 和算法 2 分别给出了基于广度优先搜索的虚拟节点映射策略与基于割集和拥塞感知的虚拟链路映射策略的细节. CCA-RVNM 算法以底层网络和虚拟网络请求的拓扑结构和资源能力情况作为输入,以满足虚拟网络轻量级可靠需求的虚拟网络映射方案为输出结果.图 3 给出了 CCA-RVNM 算法的整体流程.

算法 1 (节点映射步骤):

- Step1: 如果待映射节点为虚拟网络根结点,则将其映射到底层网络中具有最大 CPU 能力的节点上,并返回映射成功,否则执行 Step2;
- Step2: 设置距离上限 k , 并置当前距离参数 $h = 1$;
- Step3: 为每个待映射虚拟节点按顺序搜索候选节点集中满足距离约束 h 和满足虚拟网络节点资源请求条件的节点,如果存在,则进行映射,并返回映射成功,否则执行 Step4;
- Step4: 如果 $h < k$, 将 $h + 1$, 执行 Step3; 否则返回映射失败.

算法 2 (链路映射步骤):

- Step1: 映射虚拟网络中的虚拟链路,先判定该链路是否在割集中,如果不在割集中,选取底层链路负载最小的进行映射;
- Step2: 如果要映射的虚拟链路在割集中,则判断割集中其它链路是否都已映射到同一节点周围的链路上;
- Step3: 如果是,则将该虚拟链路映射到除该节点相连链路以外的链路资源上,选取链路负载最小的资源进行映射,输出映射方案;
- Step4: 如果符合要求的链路资源不存在,则映射失败.

5 实验评估与分析

我们在先前工作成果^[10-12]的虚拟网络映射模拟平台上进行了 CCA-RVNM 算法的实现和结果验证.实验结果表明,该算法在满足虚拟网络轻量级可靠需求的同时,节省了底层网络资源,提高了底层网络的长期平均运营收益.

我们首先将 CCA-RVNM 算法与用 GLPK 工具解的该问题的 ILP 结果进行了比较,对于小规模问题,ILP 能在合理的时间给出最优解,是最佳的解决方法.但对于中等规模虚拟网络映射问题,其计算时间无法满足在线虚拟网络映射的需求,如表 1 所示.而我们提出的算法在计算结果上接近最优算法,更重要的是随着问

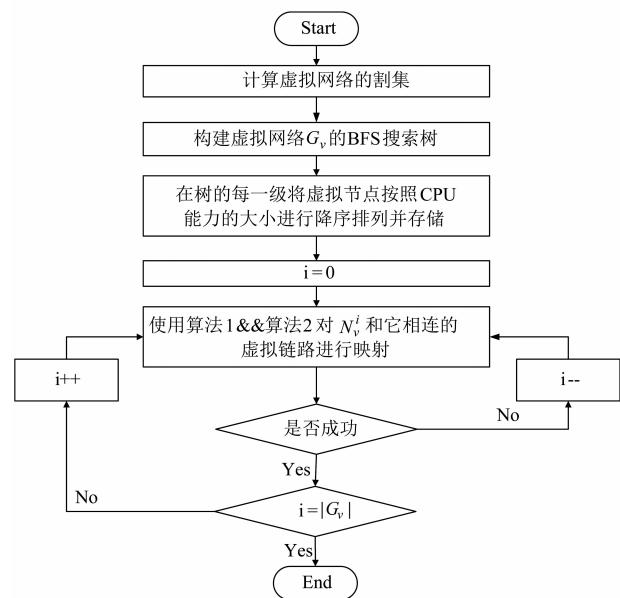


图3 CCA-RVNM算法流程图

题规模的扩大,我们的计算时间是能够满足虚拟网络在线映射需求的.

表 1 ILP vs CCA-RVNM

虚拟拓扑	比较对象	平均值	平均计算时间(s)
3 节点 3 链路	ILP	224	3.3
	CCA-RVNM	233	1.5
6 节点 6 链路	ILP	612	13252
	CCA-RVNM	625	1.9

为了更好地说明 CCA-RVNM 的优点,本文还将 CCA-RVNM 与其它两个在线虚拟网络映射算法(如表 2 所示),从底层网络长期平均运营收益(一段时间内底层网络承载虚拟网络所获收益的平均值)、虚拟网络请求接受率(成功映射的虚拟网络个数/所有虚拟网络请求个数)、底层网络长期平均收益开销比(一段时间内底层网络承载虚拟网络所获收益/承载虚拟网络资源开销)三个方面进行了比较.

表 2 比较算法

CCA-RVNM	本文所提出的基于割集和拥塞感知的启发式虚拟网络映射算法.
Hybrid-SVNE	基于文献 ^[6] 所提出的预先分配保护路径的虚拟网络映射算法的改进算法,使其可以应用于底层节点失效保护.
DP-VNE	简单的将每个虚拟链路映射到不交叉的底层物理链路上,以保证虚拟网络的可靠性.

实验的设置如下:底层物理网络拓扑由 GT-ITM 工具随机产生^[9],其中包含 120 个节点和 600 条链路,与

一个中型规模的 ISP 运营商类似. 物理网络节点计算资源与链路带宽资源服从 30-80 的均匀分布. 虚拟网络节点的度为 2-5, 节点个数随机产生且服从 3-10 的均匀分布, 每一对虚拟网络节点以 0.5 的概率相连. 虚拟网络节点计算资源需求服从 0-15 的均匀分布, 链路带宽资源需求服从 0-30 的均匀分布. 我们假设在每 100 个时间单元内虚拟网络请求的到达服从均值为 5 的泊松分布, 每一个虚拟网络的生存时间服从指数分布, 其平均生存时间为 600 个时间单元. 实验中设置节点失效的间隔时间为 20000 个时间单位, 并假设在下次失效到来前上一次的失效节点已被修复. 每次模拟实验运行 50000 个时间单元, 大约包含 2500 个虚拟网络请求. 算法 1 中使用的距离约束 k , 我们将其设置为 3.

下面是通过模拟实验得出的结果:

(1) CCA-RVNM 算法的长期运行平均收益和虚拟网络请求接受率高于 Hybrid-SVNE 和 DP-VNE 算法.

如图 4 和图 5 所示, 算法 CCA-RVNM 的底层网络长期运行平均收益相比于 Hybrid-SVNE 平均高出 6.9%, 相比于 DP-VNE 平均高出 13.8%. 这是因为 CCA-RVNM 的节点映射策略在虚拟网络映射成功率方面优于 Hybrid-SVNE 中采用的映射策略, 且 CCA-RVNM 不用预留底层保护资源, 使底层网络资源能用来映射更多的虚拟网络, 提高了虚拟网络的接受率, 从而获得了较高的收益. 而 DP-VNE 算法在链路映射阶段很难为所有的虚拟网络链路找到底层不相交的路径, 从而导致较低的虚拟网络接受率和长期平均收益.

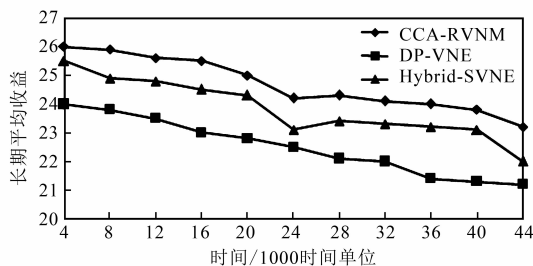


图4 底层网络长期运行平均收益

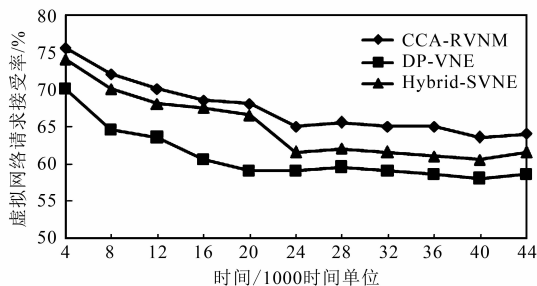


图5 虚拟网络请求接受率

(2) CCA-RVNM 算法的长期收益开销比高于 Hybrid-SVNE 和 DP-VNE 算法.

如图 6 所示, 从长期收益开销比来看, CCA-RVNM

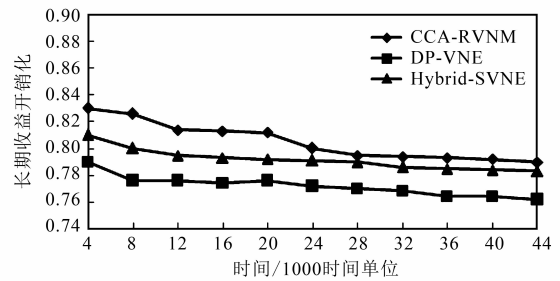


图6 底层网络长期收益开销比R/C

高于其它两种算法, 主要是因为 Hybrid-SVNE 预先分配了保护资源, 使底层可用于映射虚拟链路的资源减少, 从而降低了资源利用率. 而 DP-VNE 算法为了找到不相交的两条路径, 不得不映射到较长的底层物理链路上, 从而增加了映射开销.

6 结论

本文针对虚拟网可靠性问题提出了一种基于割集和拥塞感知的虚拟网络映射算法, 实验表明该算法在满足虚拟网络轻量级可靠需求的前提下, 能最大限度提高底层网络的资源利用效用. 该成果可应用于支持网络虚拟化技术的骨干网络或数据中心网络环境中, 为网络服务的提供商获取更多的运营收益.

参考文献

- [1] Feamster N, Gao L, Rexford J. How to lease the Internet in your spare time [J]. ACM SIGCOMM Computer Communication Review, 2007, 37(1): 61-64.
- [2] 程祥, 张忠宝, 苏森, 杨放春. 虚拟网络映射问题研究综述 [J]. 通信学报, 2011, 32(10): 143-151.
Cheng X, Zhang Z, Su S, et al. Survey of virtual network embedding problem [J]. Journal on Communications, 2011, 32(10): 143-151. (in Chinese)
- [3] 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.
- [4] Chowdhury N, Rahman M, Boutaba R. ViNEYard: virtual network embedding algorithms with coordinated node and link mapping [J]. IEEE/ACM Transactions on Networking, 2012, 20(1): 206-219.
- [5] Lischka J, Karl H. A virtual network mapping algorithm based on subgraph isomorphism detection [A]. Proc the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures [C]. ACM, 2009. 81-88.
- [6] Rahman M, Aib I, Boutaba R. SVNE: Survivable virtual network embedding algorithms for network virtualization [J]. IEEE Transactions on Network and Service Manage-

- ment. 2013, 10 (2): 105 – 118.
- [7] Yeow W L, Westphal C, Kozat U. Designing and embedding reliable virtual infrastructures [J]. ACM SIGCOMM Computer Communication Review, 2012, 41(2): 51 – 64.
- [8] Su S, Cheng X, Zhang Z, et al. Virtual network embedding with survivable routing [J]. Journal of Internet Technology, 2013, 14(5): 741 – 750.
- [9] Zegura E W, Calvert K L, Bhattacharjee S. How to model an Internet network [A]. Proc IEEE INFOCOM [C]. San Francisco, 1996. 594 – 602.
- [10] Cheng X, Su S, Zhang Z, et al. Virtual network embedding through topology-aware node ranking [J]. ACM SIGCOMM Computer Communication Review, 2011, 41(2): 39 – 47.
- [11] Cheng X, Su S, Zhang Z, et al. Virtual network embedding through topology awareness and optimization [J]. Computer Networks, 2012, 56(6): 1797 – 1813.
- [12] 程祥, 张忠宝, 苏森, 杨放春. 基于粒子群优化的虚拟网络映射算法 [J]. 电子学报, 2011, 10: 2240 – 2244.

Cheng X, Zhang Z, Su S, et al. Virtual network embedding based on particle swarm optimization [J]. Acta Electronica Sinica, 2011, 39(10): 2240 – 2244. (in Chinese)

作者简介



刘光远 男, 1981 年出生. 北京邮电大学网络技术研究院博士. 现为石家庄铁道大学信息学院教师. 研究方向为云计算、网络虚拟化.
E-mail: gyuanliu@163.com



苏森 男, 1971 年出生. 北京邮电大学网络技术研究院教授, 博士生导师. 研究方向为下一代网络、服务计算.