

## 基于链路优先的快速协同虚拟网络映射算法

熊文成, 王颖, 邱雪松, 李文璟

(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:** 在网络虚拟化环境中, 为了减少链路映射成本和提高收益, 提出了一种链路优先的协同映射算法。该算法交替映射链路和节点, 并使用贪心思想优先将带宽资源需求较大的虚拟链路映射到跳数较少的物理路径上, 最终达到降低虚拟网络平均消耗和提高虚拟网络接受率的目的。仿真结果表明, 与已有算法相比, 所提算法节约了链路映射的成本, 提高了虚拟网络接受率和物理网络收益, 算法运行时间也有效缩短。

**关键词:** 网络虚拟化; 虚拟网络映射算法; 链路优先; 资源评估; 接受率; 低消耗

**中图分类号:** TP393

**文献标识码:** A

## Link-first based quick coordinated virtual network embedding algorithm

XIONG Wen-cheng, WANG Ying, QIU Xue-song, LI Wen-jing

(State Key Lab of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** In order to reduce costs of link embedding and increase revenue, a coordinated embedding algorithm based on link-first is proposed. The algorithm embeds link and node alternately, and embeds the virtual link that with greater bandwidth requirement first to the physical path that with fewer link hops by using the greedy method, and ultimately aims at reducing average costs of virtual network embedding and improving acceptance rate of virtual network. Simulation results show that, compared with the existing algorithms, this embedding algorithm saves link cost significantly, improves acceptance rate of virtual network and revenue of the physical network, and can effectively shorten the running time.

**Key words:** network virtualization; virtual network embedding algorithm; link first; resource assessment; acceptance rate; low cost

### 1 引言

网络虚拟化是用来解决现有网络僵化问题的新技术<sup>[1-3]</sup>。它允许多个彼此隔离并具有不同拓扑的虚拟网络共存于物理网络设施上。虚拟网络映射是网络虚拟化中的重要问题, 它需要按照某些约束条件将虚拟网络映射到底层物理设施上。在这个问题中, 虚拟网络的网络拓扑由节点集和连接节点的链路集共同构成, 映射时各个虚拟节点要部署在满足需求的不同物理节点上, 虚拟链路要映射到满足

需求的物理路径上。

虚拟网络映射算法面临着诸多挑战<sup>[4]</sup>: 首先, 虚拟网络具有拓扑结构多样化的特点, 虚拟网络映射算法必须能对任意拓扑结构的虚拟网络进行映射; 其次, 虚拟网络的动态性强, 在进行资源分配时无法获知后续到达的虚拟网络信息; 最后, 底层资源有限, 映射算法在满足虚拟网络节点的计算能力和链路的带宽要求的同时, 还要实现映射结果的优化, 即资源消耗最小化和收益最大化。

收稿日期: 2013-11-05; 修回日期: 2014-01-08

基金项目: 国家高技术研究发展计划(“863”计划)基金资助项目(2013AA013502); 国家自然科学基金资助项目(61372108); 中央高校基本科研业务费专项基金资助项目(2013RC1103)

**Foundation Items:** The National High Technology Research and Development Program of China (863 Program) (2013AA013502); The National Natural Science Foundation of China (61372108); Fundamental Research Funds for the Central Universities (2013RC1103)

许多文献指出虚拟网映射是一个 NP-hard 问题<sup>[4-7]</sup>，目前的研究成果都是基于启发式实现的。根据对虚拟节点和虚拟链路的处理方式<sup>[6]</sup>，这些算法主要分为 2 类：1) 先映射节点再映射链路的两阶段映射算法(简称为两阶段映射算法)；2) 节点映射与链路映射两阶段协同的映射算法(简称为协同映射算法)。在两阶段映射算法中，由于在节点映射阶段可能导致虚拟节点过于分散<sup>[4]</sup>，从而造成链路映射的失败或者由于跳数太多而过度消耗链路资源，最终降低虚拟网络映射的接受率。协同映射算法同时处理节点映射和链路映射<sup>[5,7-9]</sup>，映射问题最终用线性规划或者其他复杂数学计算方法解决，这种算法时间复杂度高，且物理资源利用率较低。因此，目前的算法在虚拟网络接受率方面仍然有较大的提升空间，同时底层物理资源的开销需要降低，算法的运行时间也需要缩短。

为了克服虚拟网络映射的难点并弥补上述算法思路的不足，同时高效利用物理网络资源以提高虚拟网络的接受率，本文基于虚拟节点位置不受约束的假设，根据底层物理资源消耗主要体现在不同的链路映射方案上，提出了基于链路优先的快速协同映射算法(LFQ-VNE, link-first based quick coordinated virtual network embedding algorithm)。在本文算法中，链路和节点的映射是交替进行的，在每一次的交替中优先映射带宽需求大的链路，随后映射与此链路关联的节点，以此控制带宽需求大的虚拟链路所映射的物理路径的跳数，进而降低链路资源的消耗。在 LFQ-VNE 的基础上，本文通过引入资源评估的方法对其加以改进，得到的算法简称为 LFQAdvance-VNE。仿真结果表明基本的 LFQ-VNE 算法在链路资源消耗上有缩减，接受率也稳定在一个较高的值上，改进后的算法能在一定程度上提高虚拟网络接受率和物理网络平均收益。

## 2 虚拟网络映射问题描述

### 2.1 物理网络

本文中，物理网络通过一个加权无向图  $G^S = (N^S, L^S, A_N^S, A_L^S)$  表示，相关符号及其含义如表 1 所示。图 1(b)为一个物理网络实例。图中圆圈表示节点，节点旁的数字表示该节点当前可用计算能力；节点间的连线表示链路，链路旁的数字表示该链路当前可用带宽<sup>[4,10]</sup>。

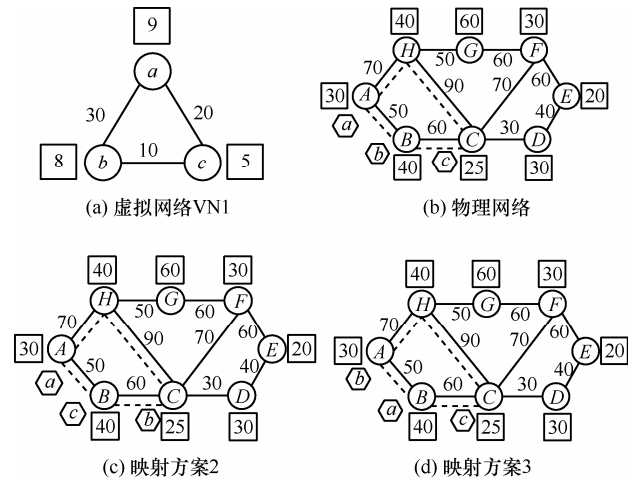


图 1 虚拟网络映射到物理网络示例

### 2.2 虚拟网络

同样地，虚拟网络也通过一个加权无向图  $G^V = (N^V, L^V, C_N^V, C_L^V)$  表示，相关符号及其含义见表 1。图 1(a)为一个虚拟网络实例，图中圆圈表示节点，节点旁的数字表示该节点的计算能力需求；节点间的连线表示链路，链路旁的数字表示该链路的带宽需求。每个虚拟网络都具有生命周期，随着时间的推移，不断有新虚拟网络到达，旧的虚拟网络离开。当虚拟网络终止后，其映射消耗的物理资源会被释放，以便接收其他虚拟网络。

表 1 文中常用符号及其含义

符号	含义
$G^S$	物理网络
$N^S, L^S$	物理节点集合，物理链路集合
$A_N^S$	物理节点属性， $A_N^S = (n^S)$ 表示节点 $n^S$ 的可用计算能力
$A_L^S$	物理链路属性， $A_L^S = (l^S)$ 表示链路 $l^S$ 的可用带宽
$G^V$	虚拟网络
$N^V, L^V$	虚拟节点集合，虚拟链路集合
$C_N^V$	虚拟节点需求， $C_N^V(n^V)$ 表示节点 $n^V$ 的计算能力需求
$C_L^V$	虚拟链路需求， $C_L^V(l^V)$ 表示虚拟链路 $l^V$ 的带宽需求
$N_u, L_u$	未映射的虚拟节点集合和未映射的虚拟链路集合
$N_m, L_m$	已映射的虚拟节点集合和已映射的虚拟链路集合

### 2.3 虚拟网络映射

虚拟网络映射可分为 2 部分：1) 将  $N^V$  中的每个虚拟节点映射到  $N^S$  中的不同物理节点上，并分配计算能力资源；2) 将  $L^V$  中的虚拟链路映射到由  $L^S$  中的物理链路组成的无环路径上，并分配带宽资源，虚拟链路映射到的无环路径两端的物理节点即该虚拟链路两端的虚拟节点所映射到的物理节点。

如图 1(a)和图 1(b)中是一个虚拟网络请求 VN1 成功映射到物理网络上的示例。映射结果包括：虚拟节点映射为  $a \rightarrow A, b \rightarrow B, c \rightarrow C$ ，虚拟链路映射为  $\{(a-b) \rightarrow (A-B), (a-c) \rightarrow (A-H-C), (b-c) \rightarrow (B-C)\}$ 。

### 3 链路优先的协同映射算法

虚拟网络在物理网络上的资源消耗由虚拟节点的消耗和虚拟链路的消耗组成。虚拟节点的资源消耗是一定的，然而虚拟链路的消耗会与其所映射到的物理路径的跳数有直接关系。因此，为了降低映射时的资源消耗，应该从降低虚拟链路的消耗入手，最关键的是在映射时减少虚拟链路映射到物理路径上的跳数。

链路优先算法主要从减少链路映射跳数的角度来降低物理资源的消耗，同时尽量减少带宽需求大的虚拟链路的映射跳数。例如图 1(b)~图 1(d)分别为 3 种将虚拟网络 VN1 映射到物理网络的方案，这 3 种方案的带宽消耗分别是 80、90、70。这 3 种方案映射的总跳数相同(均为 4)，这时总跳数已经达到最优，但方案 3 优先将带宽需求大的虚拟链路映射到跳数较少的物理路径上，带宽需求小的虚拟链路映射到跳数相对较多的物理路径上，从而总体带宽消耗最小。

基于以上考虑，本文设计了优先从链路映射开始的协同虚拟网络映射算法。

#### 3.1 概念定义

下面将一条虚拟链路和链路两端的节点定义为一个组件。

根据节点映射状态，进一步将组件分为 3 类：

1)自由组件：2 个虚拟节点均未被映射，如图 2(a)所示；2)受限组件：2 个虚拟节点仅有一个被映射，如图 2(b)所示；3)固定组件：2 个虚拟节点均被映射，如图 2(c)所示。组件统一表示为  $\{n_i^v, n_j^v\}$ ，本文约定受限组件的未映射节点为  $n_j^v$ 。与自由组件、受限组件和固定组件相对应的分别是自由链路、受限链路和固定链路。

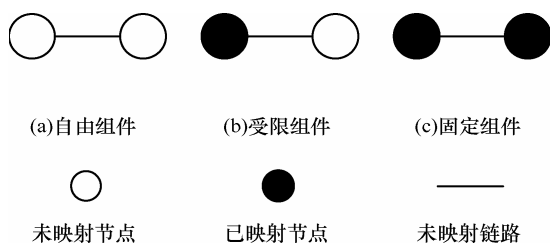


图 2 组件的定义

#### 3.2 算法的流程及说明

虚拟网络的映射过程如图 3 所示。当虚拟网络到达时，算法被触发，此时迭代次数  $vital=0$ ，算法会初始化  $N_u, L_u, N_m, L_m$ ；然后选择并映射自由组件。成功映射自由组件之后，需要映射与这个自由组件相连的所有受限组件，算法循环完成虚拟网络中所有自由组件的映射。最后，当虚拟网络中不存在自由组件时，算法完成剩下的受限组件和固定组件的映射。在映射的过程中，可能会因为物理资源的不足，而导致映射失败，这时采用迭代的方法重新映射虚拟网。如果迭代次数达到上限时仍没有成功，则虚拟网络最终映射失败而被物理网络拒绝。

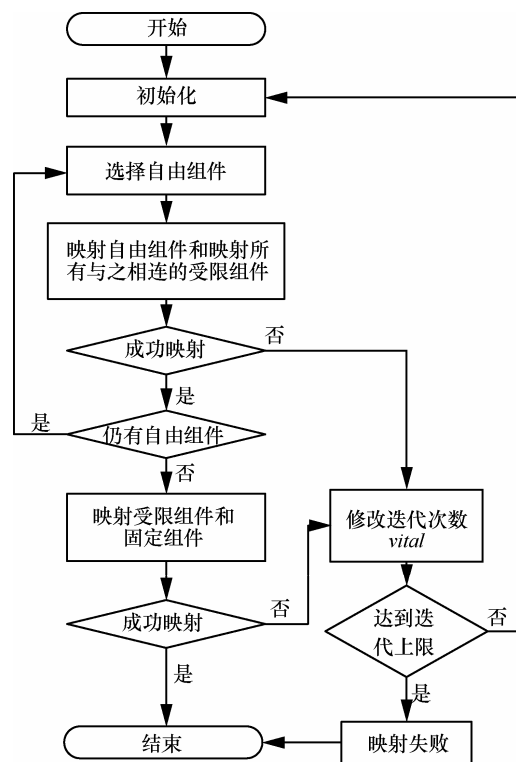


图 3 算法的流程

算法中有一个重要的参数  $vital$ ，它表示迭代次数，同时也决定了虚拟网络将要被映射到的物理网络区域。由于算法是从映射虚拟网络的自由组件开始，然后延伸映射与自由组件相连的受限组件而逐步完成，在算法中物理网络中的所有物理节点按顺序编号为： $0, 1, \dots, |N^S| - 1$ ，当迭代次数为  $vital$  时，令集合  $S = \{vital, \dots, |N^S| - 1\}$ ， $\forall i \in S$  与物理节点  $n_i^S$  一一对应。映射自由组件时的可用物理节点集合为  $S$ ，由于映射自由组件时采用的方式是按顺序遍历所有由  $S$  中的节点对确定的可用最短路径，直到找

到满足自由组件的节点和链路约束的路径为止, 所以编号为 *vital* 的物理节点附近的节点和链路会被优先映射到。因此, *vital* 决定了虚拟网络将要映射到的区域。

当映射失败时, 用迭代继续映射虚拟网络。失败是由于节点 *vital* 附近的资源不足, 所以将 *vital* 更新为 *vital*+1 继续映射虚拟网络。这样做的优点在于: 1) 迭代映射时, 直接避开了编号为  $0, \dots, vital$  的节点决定的区域, 而是从编号为 *vital*+1 的节点所决定的区域开始; 2) 迭代是整体迭代, 而不是部分迭代。当映射失败时, 算法会推翻之前已经映射的所有组件, 而不是采取回溯部分已经映射组件的方式。由于在 *vital* 附近的物理资源有限, 如果采取回溯部分组件的方法, 仍然会由于资源不足而导致失败, 如此只会增加回溯的次数, 降低算法的效率。

当迭代次数达到上限时, 如果虚拟网络仍然没有成功映射, 那么这个虚拟网络就会被物理网络拒绝。

### 3.3 自由组件的选择和映射

在虚拟网络的所有自由组件中, 用贪心算法选择链路带宽需求最大的组件作为即将要映射的自由组件  $\{n_i^V, n_j^V\}$ 。

映射一个自由组件  $\{n_i^V, n_j^V\}$  包括 2 部分: 映射组件的 2 个虚拟节点和虚拟链路。集合 *S* 中的可用节点对表示为  $(n_i^S, n_j^S), i < j, i \in S, j \in S$ , 映射自由组件的算法是按顺序遍历所有由 *S* 中的节点对确定的最短路径, 直到找到满足自由组件的节点和链路约束的路径为止。包括如下 3 个约束。

1) 物理路径的跳数限制  $h_{\text{free}}$

$$\left| sp(n_i^S, n_j^S) \right| \leq h_{\text{free}} \quad (1)$$

其中,  $sp(n_i^S, n_j^S)$  是物理节点  $n_i^S, n_j^S$  之间的最短路径,  $\left| sp(n_i^S, n_j^S) \right|$  是该最短路径的跳数。

2) 虚拟节点计算能力的约束

$$A_N^S(n_i^S) \geq C_N^V(n_i^V), A_N^S(n_j^S) \geq C_N^V(n_j^V) \quad (2)$$

3) 虚拟链路的带宽约束

$$A_p(sp(n_i^S, n_j^S)) \geq C_L^V(l^V(n_i^V, n_j^V)) \quad (3)$$

其中,  $A_p(sp(n_i^S, n_j^S))$  为最短路径  $sp(n_i^S, n_j^S)$  的带宽资源, 它定义为路径上所有链路的最小带宽资源, 即  $A_p(sp(n_i^S, n_j^S)) = \min(A_L^S(l^S)), l^S \in sp(n_i^S, n_j^S)$  (4)

如果成功映射了这个自由组件, 则会完成相应的资源分配, 并且将已经映射的物理节点的编号从 *S* 中移除。如果映射失败, 则需要迭代。

### 3.4 受限组件的映射

虚拟网络映射过程中在 2 种情况下需要映射受限组件: 1) 在一个自由组件完成映射之后, 需要映射与该自由组件相连的所有受限组件; 2) 在虚拟网络中不存在自由组件时, 需要映射虚拟网络中剩余的受限组件。图 4 所给出的为一个虚拟网络示例, 按照映射流程, 组件  $\{e, f\}, \{d, e\}, \{a, b\}, \{b, c\}$  会依次完成映射, 其中,  $\{d, e\}, \{b, c\}$  属于情况 1 的受限组件。此时虚拟网络中不存在自由组件, 但是仍然存在 2 个受限组件  $\{c, g\}, \{d, g\}$  和一个固定组件  $\{c, d\}$ 。

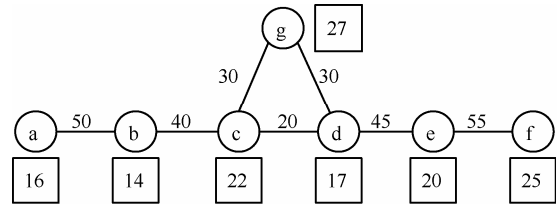


图4 虚拟网络 VN2

在受限组件映射算法中用  $N_c$  表示所有待映射的受限组件中未映射虚拟节点的集合。为了完成受限组件的映射, 需要映射  $N_c$  中的所有虚拟节点, 以及与这些虚拟节点相连的受限链路。其中, 虚拟节点的映射顺序按照贪心的规则确定, 即与虚拟节点相连的所有受限链路的带宽总和  $H(n^V)$  最大的节点首先完成映射, 其定义为

$$H(n^V) = \sum_{l^V \in L_c(n^V)} C_L^V(l^V) \quad (5)$$

其中,  $L_c(n^V)$  表示所有与  $n^V$  相连的受限链路的集合, 下同。

在映射虚拟节点  $n^V \in N_c$  时, 对于候选的物理节点  $n^S$ , 为了在将  $n^V$  映射到  $n^S$  上之后, 在映射所有与  $n^V$  相连的虚拟链路时有充足的带宽资源, 物理节点  $n^S$  需要满足

$$A_p(sp(n^S, M(n_0^V))) \geq C_L^V(l^V), \quad \forall l^V(n^V, n_0^V) \in L(n^V), n_0^V \in N_m \quad (6)$$

映射相连受限组件算法的实现步骤见算法 1。

#### 算法 1 受限组件映射算法

**step1** 将  $N_c$  中虚拟节点按照  $H(n^V)$  减小的顺序进行降序排序。

**step2** 取  $N_c$  中的第一个元素  $n^V$ ，并求所有满足  $n^V$  计算能力约束的物理节点集合，记为  $S_c$ 。

**step3** 如果  $S_c$  为空则转到 **step6**；否则对于  $\forall n^S \in S_c$ ，如果它满足式(6)，则可以将虚拟节点  $n^V$  映射到物理节点  $n^S$ 。若选择将  $n^V$  映射到  $n^S$  上，则映射与  $n^V$  相连的受限链路所需要消耗带宽资源为

$$C_{bw}(n^S) = \sum_{l^V \in L(n^V)} (C_L^V(l^V) \text{hop}(M(l^V))) \quad (7)$$

将  $S_c$  中所有满足式(6)的物理节点按照式(7)定义的  $C_{bw}(n^S)$  进行升序排序得到  $S'_c$ 。

**step4** 若  $S'_c$  为空则转到 **step6**；否则取  $S'_c$  中的第一个元素  $n_0^S$ ，将  $n^V$  映射在  $n_0^S$  上，此时  $L_c(n^V)$  中的元素全部由受限链路变为固定链路，完成  $L_c(n^V)$  中固定链路的映射与 3.5 节中固定组件的映射方法完全一样。

**step5** 更新  $N_u, L_u, N_m, L_m$  中的元素，将  $n^V$  从集合  $N_c$  中移除，如果  $N_c$  非空则转到 **step2** 循环完成映射，直到  $N_u$  为空时算法便成功映射所有的受限组件。

**step6** 映射出现失败，迭代。

### 3.5 固定组件的映射

算法的最后是映射虚拟网络中的固定组件，由于固定组件两端的虚拟节点已经完成映射，完成固定组件的映射仅需完成固定链路的映射。对于任意的固定组件  $\{n_i^V, n_j^V\}$ ，其两端的虚拟节点已经分别映射到  $M(n_i^V)$  与  $M(n_j^V)$  上，因此只需要将固定链路映射到由  $M(n_i^V)$  与  $M(n_j^V)$  确定的最短路径上。

如果这一最短路径的带宽资源不能满足固定链路  $l(n_i^V, n_j^V)$  的带宽资源，则需要迭代。

### 3.6 改进的链路优先映射算法

改进的链路优先映射算法 LFQAdvanced-VNE 是通过在基本的映射算法 LFQ-VNE 中自由组件的映射算法上引入资源评估而得到的。在自由组件映射算法中引入资源评估能增大后续受限组件和固定组件映射的成功率，可避免迭代进而充分发现物理资源，最终可以提高虚拟网络成功映射的概率。

在以上提出的自由组件映射算法中引入资源评估，是通过在式(1)~式(3)的基础上，对候选的物理路径再添加一个新的约束而实现的。这一约束的原理即是在映射自由组件  $\{n_i^V, n_j^V\}$  时，对候选的物

理路径  $\{sp(n_i^V, n_j^V)\}$  周围的带宽资源进行评估。候选路径  $\{sp(n_i^V, n_j^V)\}$  一端的物理节点  $n_i^S$  能提供的最大带宽资源定义为

$$R_{bw}(n_i^S, sp) = \sum_{l^S \in L(n_i^S), l^S \in sp(n_i^S, n_j^S)} A_L^S(l^S) \quad (8)$$

其中， $L(n_i^S)$  表示所有与节点  $n_i^S$  相连的物理链路。按照同样的方式，定义自由链路  $l_0^V(n_i^V, n_j^V)$  的一个端点  $n_i^V$  在后续映射中需要的最小带宽资源为

$$C_{bw}(n_i^S, l_0^V) = \sum_{l^V \in L(n_i^V), l^V \neq l_0^V} C_L^V(l^V) \quad (9)$$

其中， $L(n_i^V)$  表示所有与节点  $n_i^V$  相连的虚拟链路。

为了提高映射受限组件和固定组件的成功率，需要有如下约束

$$\begin{cases} R_{bw}(n_i^S, sp) \geq C_{bw}(n_i^V, l_0^V) \\ R_{bw}(n_j^S, sp) \geq C_{bw}(n_j^V, l_0^V) \end{cases} \quad (10)$$

## 4 仿真实验

本文提出的链路优先映射算法主要从减少链路映射跳数的角度来降低物理资源的消耗，进而在物理网络中留下更多的可用资源以提高后续到达的虚拟网络映射成功率，因此降低物理资源消耗理论上可以提高虚拟网络接受率，同时较高的接受率会带来较大的物理网络收益。为验证本文提出算法的有效性，设计并搭建了仿真实验环境并对实验结果进行了比较和分析。

### 4.1 评价指标和比较方法

1) 物理网络的平均收益。物理网络的平均收益就是所有成功映射的虚拟网络所带来的收益总和与时间之比。其中，成功映射的虚拟网络  $G^V$  在  $t$  时刻带来的收益为

$$R(G^V, t) = \alpha \sum_{n^V \in N^V} C_N^V(n^V) + \sum_{l^V \in L^V} C_L^V(l^V) \quad (11)$$

其中， $\alpha(\alpha \geq 0)$  是计算能力资源在总资源中的权重因子。为方便比较，参照文献[4]，本文统一取  $\alpha=1$ 。

2) 虚拟网络的平均消耗。虚拟网络平均消耗是所有成功映射的虚拟网络消耗总和与时间之比。物理网络接受一个虚拟网络的开销定义为

$$C(G^V, t) = \alpha \sum_{n^V \in N^V} C_N^V(n^V) + \sum_{l^V \in L^V} (C_L^V(l^V) \text{hop}(M(l^V))) \quad (12)$$

其中,  $hop(M(l^v))$  是物理路径的跳数, 参数  $\alpha$  的意义和取值与式(11)中的一致。

3) 虚拟网络的接受率。在  $t$  时刻的接受率为成功映射的虚拟网络个数在到达的虚拟网络个数中的比例。

4) 链路映射的平均跳数。在一段时间内, 链路映射平均跳数定义为所有接受的虚拟网络的平均跳数的平均值。对于一个成功映射的虚拟网络, 其平均跳数可以定义为所有虚拟链路映射在物理路径上跳数的平均值, 它在一定程度上反映虚拟网络在物理网络上的紧凑程度。其定义为

$$\overline{hop}(G^v) = \frac{\sum_{l^v \in L^v} hop(M(l^v))}{|L^v|} \quad (13)$$

其中,  $|L^v|$  表示虚拟链路的个数。

本文针对上述评价指标分别对4种映射方法进行比较, 分别是: 1) 本文提出的链路优先算法 LFQ-VNE; 2) 本文提出的改进映射算法 LFQAdvanced-VNE; 3) 节点映射采用贪心算法且链路映射采用多商品流<sup>[13]</sup>的两阶段虚拟网络映射算法 G-MCF<sup>[4]</sup>; 4) 对物理节点添加元节点和元路径形成局部元图而实现的协同虚拟网络映射的算法 R-ViNE-SP<sup>[7]</sup>。

此外, 实验还分析了网络负载、算法中的参数  $h_{free}$  对虚拟网络接受率的影响。

#### 4.2 实验环境

本文使用 GT-ITM 拓扑生成器生成物理网络拓扑和虚拟网络拓扑。为了方便实验对比, 参照文献[4,11,12], 物理网络拓扑由100个节点和约500条链路组成。物理节点的计算能力服从[40,100]的均匀分布, 物理链路的带宽能力服从[50,100]的均匀分布。虚拟网络的拓扑是随机生成的, 节点之间的连接率为0.5, 每个虚拟网络的节点数服从[2,10]的均匀分布, 虚拟网络中每个虚拟节点的计算能力的资源需求服从[0,20]的均匀分布, 虚拟链路的带宽需求符合[50,100]的均匀分布。

虚拟网络的到达服从参数为  $\lambda = 4$  的泊松过程, 即在一个时间窗口平均会到达  $\lambda$  个虚拟网络。每个虚拟网络的生存时间服从参数为  $\mu$  的指数分布, 因此虚拟网络的平均生命周期为  $1/\mu$  个时间窗口, 在对比实验中取平均生命周期为10个时窗。每个时窗有100个时间单元。物理网络所承受的网络负载就是虚拟网络的到达速度与离开速度之比,

即  $\xi = \lambda/\mu$ 。

每次模拟实验运行到500个虚拟网络到达时为止, 每到达100个虚拟网络做一次数据统计。

#### 4.3 实验结果与分析

1) 不同算法在接受率和平均收益上的比较

图5和图6分别从接受率和物理网络平均收益的角度进行比较。从图5和图6可以观察到, 与现有算法如 G-MCF 和 R-ViNE-SP 相比, 链路优先算法有较高的接受率, 同时也会带来较高的平均收益, 与性能较好的算法 R-ViNE-SP 相比, 映射接受率和平均收益分别提高了13.4%和21.3%。由于 G-MCF 算法和 R-ViNE-SP 算法忽略了资源需求量越大的虚拟链路越容易映射失败的特点, 未能优先将带宽需求大的虚拟链路进行映射, 进而增大了虚拟网络映射失败的概率, 最终导致虚拟网络接受率和物理网络收益较低。另外, 从图5和图6可观察到, 改进的映射算法能在较小程度上提高虚拟网络接受率和物理网络收益。最后, 随着虚拟网络的到达与离开, 会有物理资源被占用与释放, 这便会导致虚拟网络接受率和物理网络平均收益上下波动, 甚至小幅度降低。

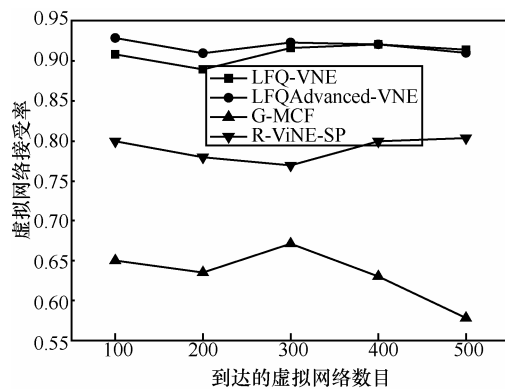


图5 虚拟网络映射算法的接受率

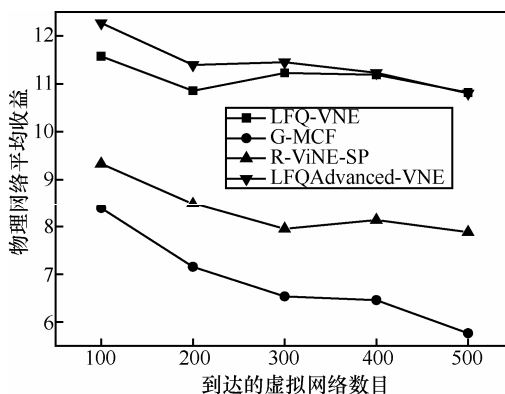


图6 物理网络平均收益

### 2) 不同算法在虚拟网络平均消耗上的比较

图 7 给出虚拟网络平均消耗随着虚拟网络到达数目的变化规律。图 6 和图 7 表明虽然 LFQ-VNE 算法带来了较高的接受率和平均收益,但是算法并没有过度消耗物理网络上的资源,反而降低了平均映射消耗的资源。G-MCF 算法由于在节点映射时未考虑链路映射,而导致链路映射资源消耗过大, R-ViNE-SP 算法虽然在节点映射时考虑了链路映射,但该算法未能有效控制带宽需求较大的虚拟链路映射到物理路径的跳数而导致资源消耗过大。

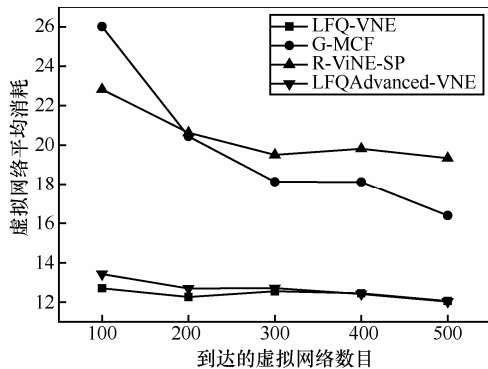


图 7 虚拟网络映射的平均消耗

### 3) 不同算法在平均跳数上的比较

在图 8 中,链路优先算法和改进之后的链路优先算法的平均跳数基本相同,而且比 R-ViNE-SP 算法的平均跳数要小 1.2。本算法的目标就是为了减少虚拟网络映射到物理网络上的平均跳数。在选择自由组件时,按照贪心算法首先将带宽资源较大的链路映射到跳数不大于  $h_{free}$  的物理路径上,接着在映射受限组件时,也是选择了消耗带宽资源最小的方案,由此将虚拟网络的平均跳数控制在较小范围内。

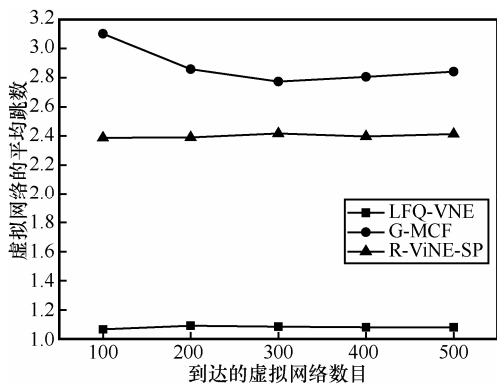


图 8 虚拟网络映射的平均跳数

### 4) 不同算法在运行时间上的比较

图 9 给出成功映射虚拟网络所消耗的平均时间随到达的虚拟网络数目的变化情况, LFQ-VNE 算法能够快速完成虚拟网络的映射。由于算法的迭代方式是整体迭代,避免了采用回溯而过度耗时的方式,其他 2 种算法分别采用的是 glpk<sup>[9]</sup>和 MCF<sup>[8]</sup>,由于其数学计算复杂而消耗了大量的时间。

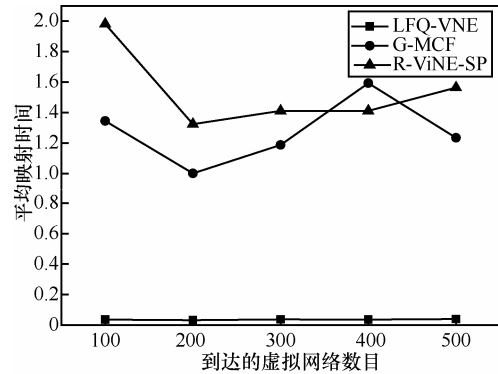


图 9 成功映射虚拟网络的平均映射时间

### 5) 自由链路映射的跳数限制 $h_{free}$ 对接受率的影响

从图 10 可以观察到,当  $h_{free} = 1$  和  $h_{free} = 2$  时,虚拟网络的接受率较大;但是当  $h_{free} \geq 3$  时,接受率有所降低而且各条趋势线重合。在映射自由组件时,由于  $h_{free}$  较大,当选择的物理路径跳数较大时,它仍然能满足式(1)的跳数约束,但是由式(4)可知,物理路径的跳数越多,其可用带宽资源越小,这就会导致映射自由组件时失败。当出现失败时,会由于整体迭代而错过了  $vital$  附近其他的可用资源,导致最终的失败。另外,模拟实验中物理网络的所有最短路径有  $C_{100}^2 = 4950$  条,其中跳数为 1,2,3,4 的最短路径分别有 570, 3087, 1290, 3 条。因此,当  $h_{free} = 3,4, \dots$  时,几乎所有的最短物理路径都能满足式(1),这就是

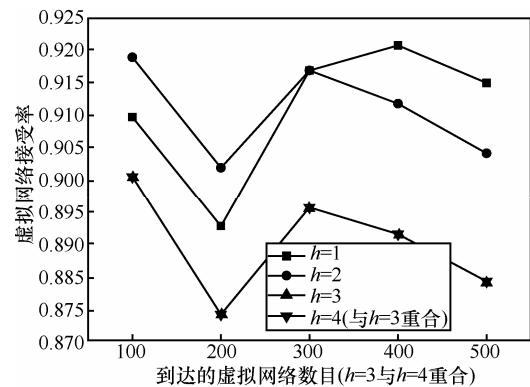


图 10 自由链路映射的跳数限制  $h_{free}$  对接受率的影响

说  $h_{\text{free}} = 3, 4, \dots$  的各个取值是等价的, 所以导致趋势线重合。

对于跳数限制  $h_{\text{free}}$  这一参数的使用, 有另外一种策略。当  $h_{\text{free}}$  取定时, 将式(1)从不等式改为等式, 并让自由链路的映射跳数固定为 1, 如果虚拟网络映射失败, 则让自由链路映射的跳数限制固定为 2, 如果虚拟网络仍然失败, 则如此重复直到跳数限制固定为  $h_{\text{free}}$ 。图 11 是采用这一策略之后  $h_{\text{free}}$  对虚拟网络接受率的影响, 当  $h_{\text{free}}$  较大时, 能在较小的程度上提高接受率, 但是随着时间的推移, 由于在  $h_{\text{free}}$  较大的情况资源消耗较大, 最终会小幅度降低虚拟网络的接受率。

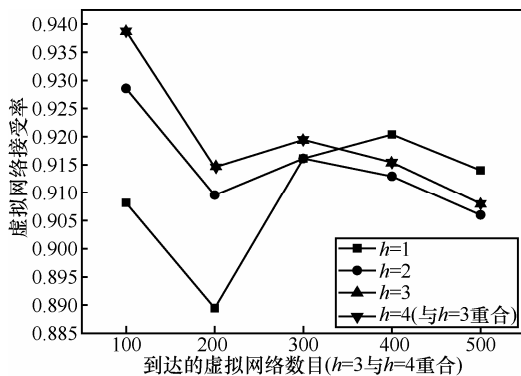


图 11 采用新策略使用  $h_{\text{free}}$  参数对虚拟网络接受率的影响

#### 6) 网络负载 $\xi$ 对虚拟网络接受率的影响

图 12 是每个网络负载的取值进行一次模拟实验, 并在结束时统计实验过程中的最大接受率和最小接受率。当网络负载变大时, 虚拟网络的接受率会降低, 但是 LFQ-VNE 算法中虚拟网络的接受率仍稳定在 75% 之上。

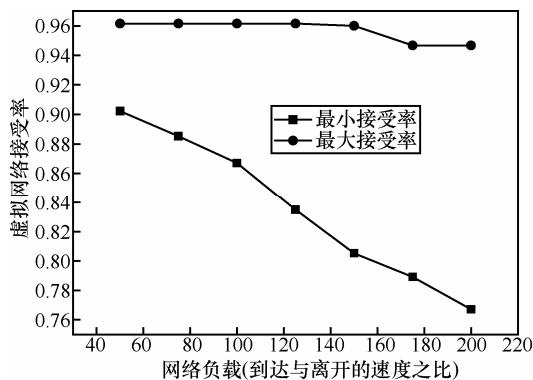


图 12 网络负载  $\xi$  对接受率的影响

## 5 结束语

虚拟网络映射是网络虚拟化中的一个重要问

题。由于物理网络资源的有限性, 在映射虚拟网络时需要高效利用物理网络资源, 以接受更多虚拟网络, 从而带来更多收益。

本文提出的链路优先算法能减少虚拟链路映射时的资源消耗, 以此高效地利用物理网络的资源。链路优先算法用贪心思想优先将带宽资源需求较大的虚拟链路映射到跳数较小的物理路径上, 从而降低其带宽资源的消耗, 最终降低虚拟网络映射的资源消耗。同时, 在此算法的基础上引入了资源评估以实现充分发现资源而提高接受率的目的。实验结果表明, 与现有算法相比, 链路优先算法能降低虚拟网络的平均消耗, 提高虚拟网络的接受率, 引入资源评估之后的算法能在较小程度上提高接受率。

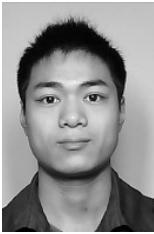
本文提出的算法仍然存在一些需要改进之处。算法在映射过程中, 可能因为链路资源不足而失败, 因为算法只考虑了最短路径, 未来可以考虑支持第  $k (k \geq 2)$  短路径。同时也可以考虑对支持路径分割的虚拟网络进行处理。

#### 参考文献:

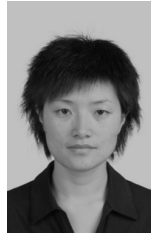
- [1] ANDERSON T, PETERSON L, SHENKER S, *et al.* Overcoming the Internet impasse through virtualization[J]. *Computer*, 2005, 38(4): 34-41.
- [2] FEAMSTER N, GAO L, REXFORD J. How to lease the Internet in your spare time[J]. *ACM SIGCOMM Computer Communication Review*, 2007, 37(1): 61-64.
- [3] CHOWDHURY N M, BOUTABA R. A survey of network virtualization[J]. *Computer Networks*, 2010, 54(5): 862-876.
- [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] CHOWDHURY N M M K, RAHMAN M R, BOUTABA R. Virtual network embedding with coordinated node and link mapping[A]. *IN-FOCOM 2009*[C]. *IEEE*, 2009.783-791.
- [6] FISCHER A, BOTERO J, BECK M, *et al.* Virtual network embedding: a survey[J]. *Browse Journals & Magazines*, 2013, 15(4):1888-1906.
- [7] CHOWDHURY M, RAHMAN M R, BOUTABA R. ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping[J]. *IEEE ACM Transactions on Networking (TON)*, 2012, 20(1): 206-219.
- [8] BUTT N F, CHOWDHURY M, BOUTABA R. *Topology-awareness and reoptimization mechanism for virtual network embedding*[M]. Springer: Berlin Heidelberg, 2010.
- [9] 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): 38-47.

- [10] ZHANG S, QIU X. A novel virtual network mapping algorithm for cost minimizing[J]. *Cyber Journals: Journal of Selected Areas in Telecommunications (JSAT)*, 2011.
- [11] BAVIER A, FEAMSTER N, HUANG M, *et al.* In VINI veritas: realistic and controlled network experimentation[A]. *ACM SIGCOMM Computer Communication Review*[C]. ACM, 2006, 36(4): 3-14.
- [12] ZEGURA E W, CALVERT K L, BHATTACHARJEE S. How to model an internetwork[A]. *Fifteenth Annual Joint Conference of the IEEE Computer Societies Networking the Next Generation*[C]. 1996, 2: 594-602.
- [13] AHUJA R K, MAGNANTI T L, ORLIN J B. *Network flows: theory, algorithms, and applications*[M]. *Library of Congress Cataloging-in-publication Data*, 1993.

#### 作者简介:



熊文成 (1990-), 男, 湖北潜江人, 北京邮电大学硕士生, 主要研究方向为网络虚拟化与资源调度。



王颖 (1976-), 女, 辽宁昌图人, 北京邮电大学副教授、硕士生导师, 主要研究方向为网络管理和通信软件等。



邱雪松 (1973-), 男, 江西上饶人, 北京邮电大学教授、博士生导师, 主要研究方向为网络管理和通信软件等。



李文璟 (1973-), 女, 山西太谷人, 北京邮电大学副教授、硕士生导师, 主要研究方向为网络管理和通信软件等。