

# 基于负载均衡的虚拟网映射随机算法

余建军<sup>1</sup> 吴春明<sup>2</sup>

(衢州职业技术学院 衢州 324000)<sup>1</sup> (浙江大学人工智能研究所 杭州 310027)<sup>2</sup>

**摘要** 分析了基于“尽力服务”模式的虚拟网映射算法所存在的问题,并指出了其在资源均衡利用方面的不足,设计了物理网负载均衡代价指标,提出了负载均衡的虚拟网映射随机算法。实验表明,所提出的算法能提高物理网资源的负载均衡度和利用率,从而提高虚拟网构建请求的接受率和物理网提供商的收益。

**关键词** 虚拟网映射,负载均衡,随机算法,强化学习

**中图分类号** TP393 **文献标识码** A

## Randomized Algorithm for Virtual Network Mapping Problem Based on Load Balancing

YU Jian-jun<sup>1</sup> WU Chun-ming<sup>2</sup>

(Quzhou College of Technology, Quzhou 324000, China)<sup>1</sup> (AI Institute, Zhejiang University, Hangzhou 310027, China)<sup>2</sup>

**Abstract** This paper analyzed the existing problems of virtual network mapping (VNM) algorithm based on best-effort service, and pointed out their shortcoming on resource load balancing, and then defined load balancing cost indicator on the physical network, proposed a virtual network mapping randomized algorithm based on load balancing. Experiment shows that the proposed algorithm increases physical network resource load balancing metric and coefficient of utilization, therefore can improve the virtual network construction request acceptance ratio and income profit of physical network service provider.

**Keywords** Virtual network mapping, Load balancing, Randomized algorithm, Reinforcement learning

### 1 概述

网络虚拟化技术被认为是未来互联网架构的发展趋势,该技术通过在底层物理网上构建多个独立的虚拟网,来达到同时支持多种服务和网络体系结构的目的<sup>[1-3]</sup>。虚拟网构建的核心是虚拟网映射问题,即在满足虚拟网构建约束的前提下,把虚拟网的虚拟节点和虚拟链路分别映射到底层物理网的节点和路径上。由于多个虚拟网共享同一物理网资源,为提高物理网资源利用率和物理网提供商的收益,必须设计有效的虚拟网映射算法。虚拟网映射算法必须能处理动态到达的虚拟网构建请求<sup>[4]</sup>,属于在线算法。

虚拟网映射问题是 NP 难问题<sup>[5]</sup>,目前提出的算法主要集中在启发式算法<sup>[4,6-8]</sup>上,这些算法要么假设虚拟节点映射是已知的,要么把虚拟网构建分两个阶段来完成:首先使用贪婪算法完成虚拟节点映射,然后使用最优路径、K 则最优路径、多商品流等算法完成虚拟链路的映射。另外,文献[9]提出了虚拟节点映射与虚拟链路映射相互协调的两阶段算法;文献[10]提出了通过子图同构检测和回溯的方法将虚拟节点和虚拟链路同步映射到物理网的算法;文献[11]基于节点等级,分别设计了虚拟网映射的两阶段算法和虚拟节点映射与虚拟链路映射同步进行的算法;文献[12]提出了基于路径集

多商品流问题模型的映射算法;文献[13,14]提出了负载均衡的虚拟网映射算法。

目前提出的所有虚拟网映射算法都基于“尽力服务”的模式,即当虚拟网建立请求动态到达后,如虚拟网映射算法能够在底层物理网上构建该虚拟网,则一定会完成构建,只有当不能构建时才拒绝。在一般情况下,这种服务模式确实能提高物理网提供商的收益,但当物理网负载较重且完成该虚拟网构建会导致物理网资源较严重的不均衡利用时,接受该虚拟网请求可能会得不偿失,即可能会为得到眼前收益而牺牲将来更多收益。另外,目前所提出的算法的竞争比<sup>[15]</sup>会趋向于无穷大(见定理 1)。

为提高物理网提供商的收益,对动态到达的每个虚拟网请求,要尽量获取该虚拟网映射收益,同时为了提高长期收益和长期虚拟网构建请求接受率,在构建虚拟网时,既要考虑使虚拟网映射所消耗的物理网资源尽量少,也需考虑物理网资源的均衡利用。

目前提出的虚拟网构建算法,要么没有考虑物理网资源利用的均衡性<sup>[4,7,10,11]</sup>,要么仅考虑局部非公平负载均衡<sup>[9]</sup>(针对剩余网络(见定义 3),以当前虚拟网所映射的物理节点和物理链路的相对消耗量之和最小化为目标),要么考虑针对物理网资源的全局公平负载均衡,如文献[6]以物理网资源的

到稿日期:2013-08-17 返修日期:2013-10-29 本文受国家自然科学基金(61070157,61070213),国家高技术研究发展计划(“863”计划)基金(2008AA01A323,2008AA01A325,2009AA01A334),浙江省自然科学基金(LY14F020010),浙江省教育厅科研项目(Y201225106)资助。

余建军(1969—),男,硕士,副教授,主要研究方向为网络虚拟化技术以及算法设计与分析,E-mail:yj691121@126.com;吴春明(1967—),男,博士,教授,主要研究方向为大规模可重构柔性网络构建技术和三网融合体系结构等。

最大消耗量的最小化为目标,文献[8]以物理网瓶颈资源使用的最小化为目标,文献[12]以当前虚拟网所映射的链路的最大相对消耗量最小化为目标,文献[13]以当前虚拟网所映射的链路的平均相对消耗量与当前虚拟网所映射的链路的最大相对消耗量之比的最大化为目标。其不足之处在于,如不考虑负载均衡,会导致物理网的关键路径和节点上的负载过重,从而使后续虚拟网构建请求被拒绝的概率增加;如仅考虑局部非公平负载均衡,虽然叠加的总体资源相对消耗量低,但单个物理节点或物理链路的相对消耗量仍然可能较高,即物理网仍会产生瓶颈节点或链路;而采用物理网全局公平负载均衡,虽然各物理网资源消耗较均衡,但可能造成叠加的总体物理资源消耗较大。另外文献[12]没有考虑物理节点的映射及物理节点的负载均衡。文献[14]提出的针对逻辑承载网(与虚拟网有很多相似之处)构建的负载均衡算法,以当前物理网各物理链路的相对消耗量与剩余带宽之比的和的最小化为目标,该算法虽能实现物理网链路资源的剩余带宽的均衡,但可能造成叠加的总体物理链路消耗较大。

基于以上分析,本文设计了既能体现物理网资源的消耗量、又能体现物理网负载均衡性的物理网负载均衡代价指标,然后基于该指标,设计了负载均衡虚拟网映射算法。另外基于随机的思想,在负载均衡虚拟网映射算法的基础上,设计了负载均衡虚拟网映射随机算法,该算法将按一定概率主动拒绝动态到达的可以完成映射的虚拟网构建请求,拒绝的概率将根据物理网当前状态来确定。

## 2 网络模型和问题描述

**定义 1(物理网)** 物理网表示为无向图  $G^s = (N^s, E^s)$ , 其中  $N^s$  和  $E^s$  分别表示物理节点和物理链路的集合。每个物理节点  $n^s$  具有 CPU 容量  $c(n^s)$  和所在地点  $loc(n^s)$  两个属性, 每条物理链路  $e^s$  具有链路带宽  $b(e^s)$  属性。

**定义 2(虚拟网)** 虚拟网表示为无向图  $G^v = (N^v, E^v)$ , 其中  $N^v$  和  $E^v$  分别表示虚拟节点和虚拟链路的集合。每个虚拟节点  $n^v$  附带 CPU 容量  $c(n^v)$  和虚拟节点所在地点  $loc(n^v)$  两个属性, 每条虚拟链路  $e^v$  具有链路带宽  $b(e^v)$  要求。每个虚拟网请求附带一个非负值  $D^v$ , 表示虚拟节点与所映射的物理节点间的距离必须小于等于  $D^v$ , 距离可以表示物理距离、延迟等。每个虚拟网的请求都有一个开始时间和结束时间。当虚拟网生命周期结束后, 其映射的物理资源将会被释放。

**定义 3(剩余网络)** 把物理网络  $G^s$  的节点的 CPU 容量减去映射到该物理节点的虚拟节点的 CPU 容量, 虚拟链路所映射的物理路径上的物理链路的带宽减去虚拟链路带宽后的网络称为  $G^s$  的剩余网络  $G_{res}^s$ 。物理节点  $n^s$  的 CPU 剩余容量记为  $R_N(n^s)$ , 物理链路  $e^s$  的剩余容量记为  $R_E(e^s)$ 。

**定义 4(虚拟网映射)** 虚拟网映射是指把虚拟网  $G^v = (N^v, E^v)$  映射到物理网  $G^s = (N^s, E^s)$  的一个子图上, 要求每个虚拟节点必须且只能映射到一个物理节点, 每个物理节点只能被一个虚拟节点映射; 每条虚拟链路必须且只能映射到物理网的一条无圈的路径上, 并且物理路径的两个端点分别是虚拟链路的两个虚拟节点所映射的物理节点。虚拟网映射时必须满足由  $c(n^v)$ 、 $D^v$ 、 $b(e^v)$  给出的约束条件<sup>[9]</sup>。

**定义 5(虚拟网映射收益)** 虚拟网映射的收益定义为虚拟网的虚拟节点 CPU 容量之和与虚拟链路带宽之和的线性

加权和, 权重分别是单位 CPU 容量的收费价格和单位链路带宽的收费价格。

**定义 6(虚拟网映射目标)** 从物理网提供商的角度来确定虚拟网映射的目标, 即长期平均收益的最大化。为此, 在虚拟网映射时, 必然希望尽量减少物理网资源消耗量, 并使物理网资源的利用尽量均衡, 以能够保留更多的资源来满足未来虚拟网映射的需要。

## 3 负载均衡虚拟网映射算法 LBVM

### 3.1 物理网负载均衡代价指标的设计

物理网负载均衡代价指标必须既能体现物理网资源的相对消耗量, 又能体现物理网负载均衡性, 下面给出基于多目标效用函数设计思想的具体设计方案。该指标越小, 说明当前物理网资源分配的效用越好, 即物理网资源相对消耗量较低且物理网资源消耗较均衡。

**定义 7(物理网节点使用均衡度)**

$$f_{-n} = \frac{node\_consumptions^2}{|N^s| \times node\_con\_qu} \quad (1)$$

其中,  $node\_con\_qu = \sum_{\omega \in N^s} (\frac{C(\omega) - R_N(\omega)}{C(\omega)})^2$ ;  $node\_consumptions = \sum_{\omega \in N^s} \frac{C(\omega) - R_N(\omega)}{C(\omega)}$  是物理节点的利用率之和。当所有物理节点的利用率相等时,  $f_{-n}$  取最大值 1, 物理节点的利用率的波动越大, 其值越小, 可以无限趋近于 0,  $f_{-n}$  取值范围是  $(0, 1]$ 。

**定义 8(物理网链路使用均衡度)**

$$f_{-e} = \frac{link\_consumptions^2}{|E^s| \times link\_con\_qu} \quad (2)$$

其中,  $link\_con\_qu = \sum_{uv \in E^s} (\frac{b(uv) - R_E(uv)}{b(uv)})^2$ ;  $link\_consumptions = \sum_{uv \in E^s} \frac{b(uv) - R_E(uv)}{b(uv)}$  是物理网链路利用率之和。当物理链路的利用率相等时,  $f_{-e}$  取最大值 1, 物理链路利用率的波动越大, 其值越小, 可以无限趋近于 0,  $f_{-e}$  取值范围是  $(0, 1]$ 。

**定义 9(物理网节点负载均衡代价指标)**

$$u_{-n} = \frac{node\_consumptions / |N^s|}{f_{-n}} \quad (3)$$

$node\_consumptions / |N^s|$  是物理网节点资源相对消耗量。选择  $u_{-n}$  值小的虚拟网节点映射方案, 能够降低物理网节点相对资源消耗量且提高物理网节点资源使用的均衡性。

**定义 10(物理网链路负载均衡代价指标)**

$$u_{-e} = \frac{link\_consumptions / |E^s|}{f_{-e}} \quad (4)$$

$link\_consumptions / |E^s|$  是物理网链路资源相对消耗量。选择  $u_{-e}$  值小的链路映射方案, 能够降低物理网链路资源相对消耗量且提高物理网链路资源使用的均衡性。

**定义 11(物理网负载均衡代价指标)**

$$u = u_{-n} \times \frac{G_n}{G_n + G_e} + u_{-e} \times \frac{G_e}{G_n + G_e} \quad (5)$$

其中,  $G_n$  和  $G_e$  分别表示物理网节点和链路的平均利用率。

物理网负载均衡代价指标的设计涉及  $u_{-n}$  和  $u_{-e}$  两个因素。根据多目标效用函数设计的加法规则, 设计物理网负载均衡代价指标, 如式(5)所示。其中  $u_{-n}$  和  $u_{-e}$  的权重是依据物理网节点和链路的负载确定的, 如某类资源的负载越重, 则

这类资源的代价在物理网负载均衡代价中所占比例越大。这样选择物理网负载均衡代价指标小的虚拟网映射方案,就能提高物理网资源利用均衡性并降低资源相对消耗量。

### 3.2 LBVM 算法流程

基于物理网负载均衡代价指标设计 LBVM 算法,算法分两个过程:(1)求初始可行解:对动态到达的虚拟网构建请求,把虚拟节点和虚拟链路映射到相对资源消耗量最小的资源上;(2)对初始可行解进行迭代优化:计算针对当前可行解的物理网负载均衡代价指标,然后依次改变虚拟节点所映射的符合约束的物理节点,如改变后物理网的负载均衡代价指标能够减少,则接受改变,直到最后一个节点完成。具体步骤如下:

步骤 1 对虚拟网的虚拟节点  $n^v$  按  $c(n^v)$  升序排序。

步骤 2 按排序好的顺序,依次对所有虚拟节点  $n^v$  进行映射:

步骤 2.1 找出符合节点  $n^v$  的  $c(n^v)$  和  $D^v$  约束的物理节点集合  $K$ ;

步骤 2.2 把虚拟节点  $n^v$  映射到集合  $K$  中,映射后与消耗量最小物理节点相对。

步骤 3 对虚拟网的虚拟链路  $e^v$  按  $b(e^v)$  升序排序。

步骤 4 按排序好的顺序,依次对所有虚拟链路  $e^v$  进行映射:

步骤 4.1 求出当前虚拟链路  $e^v$  可以映射的符合  $b(e^v)$  约束的  $K$  短简单路径( $k$  取 4);

步骤 4.2 把当前虚拟链路  $e^v$  映射到  $k$  条路径中平均链路剩余比例最大的路径。

步骤 5 针对当前虚拟网映射方案,计算物理网负载均衡代价指标  $u$  (式(5))。

步骤 6 在当前映射方案中,各虚拟节点  $n^v$  依次完成:

步骤 6.1 找出符合节点  $n^v$  的  $c(n^v)$  和  $D^v$  约束的物理节点集合  $M$  (因物理网部分资源已经按当前映射方案分配给了当前虚拟网,故集合  $M$  和集合  $K$  不一定相同);

步骤 6.2 修改当前虚拟网映射方案:把当前虚拟节点所映射的物理节点,用集合  $M$  中物理节点  $x$  替换。 $x$  是集合  $M$  中使替换后的映射方案所对应的  $u$  指标取值最大的物理节点(计算  $u$  值前,需要把当前虚拟节点的相关虚拟链路按步骤 4.1 和步骤 4.2 重新映射)。

## 4 负载均衡虚拟网映射随机算法 LBRVM

针对前面所述的基于“尽力服务”模式在线算法(包括 LBVM 算法)的问题,本文设计了负载均衡虚拟网映射随机算法,该算法用 LBVM 算法完成虚拟网映射,再根据当前物理网状态,按一定概率拒绝该请求。算法的关键是如何确定不同状态的拒绝概率,LBRVM 算法使用强化学习的方法,动态学习在各状态下的拒绝概率。

### 4.1 基于先验知识的强化学习

强化学习<sup>[16]</sup>是指从环境状态到动作映射的学习,强化学习算法所用模型通常都假设系统参数未知、先验知识未知,这样会导致搜索范围大且收敛速度慢。

针对虚拟网映射问题,我们有以下先验知识:当物理网负载较轻且虚拟网构建不会导致物理网负载严重不均衡时,应以很低的概率拒绝请求;当物理网负载很重且虚拟网构建会导致物理网负载严重不均衡时,应以很高的概率拒绝请求。如抛开此先验知识,必然导致开始时拒绝了较多不该拒绝的请求。

### 4.2 Q 学习算法

虚拟网构建请求是动态到达的,其到达时间和所请求资源量具有不确定性,物理网状态的转移函数很难建立,而 Q 学习算法<sup>[16]</sup>无需已知的精确状态转移概率,适合于动态学习在各物理网状态下的拒绝概率。

Q 学习算法由 Watkins 提出,是一种模型无关的强化学习算法,Q 学习算法通过对状态动作对的值函数  $Q(s,a)$  进行估计以求得最优策略, $Q(s,a)$  为在状态  $s$  执行完动作  $a$  后期望获得的累积回报,它取决于当前的即时回报和期望的延时回报。Q 值的修正公式如下:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a')] \quad (6)$$

式中, $\alpha$  称为学习率, $\gamma$  为折扣率, $r_{t+1}$  是在状态  $s_t$  执行动作  $a_t$  后的即时回报。

### 4.3 状态变量和状态空间

状态变量的设置必须能反映物理网的负载状态和虚拟网请求对物理网资源的要求。为此定义系统的状态为  $s = \langle Gn, Ge, f_{nd}, f_{ed} \rangle$ 。其中, $Gn$  和  $Ge$  状态变量分别表示物理网的节点和链路的当前平均利用率的所在区间; $f_{nd}$  状态变量表示接受当前虚拟网请求前后物理网节点的使用均衡度的差(均衡度计算见式(1))的所在区间; $f_{ed}$  状态变量表示接受当前虚拟网请求前后物理网链路的使用均衡度的差(均衡度计算见式(2))的所在区间。

使用较多系统状态数目可以使  $Q(s,a)$  值的变化更加敏感和更加精确,但是过多的状态数又会使搜索空间增大。根据实验经验,状态变量  $Gn$  和  $Ge$  取值为 1—5,分别对应平均利用率区间  $(0, 0.2]$ 、 $(0.2, 0.3]$ 、 $(0.3, 0.4]$ 、 $(0.4, 0.6]$ 、 $(0.6, 1]$ ;状态变量  $f_{nd}$  和  $f_{ed}$  取值为 1—4,分别对应区间  $(-\infty, 0.06]$ 、 $(0.06, 0.10]$ 、 $(0.10, 0.14]$ 、 $(0.14, 1)$ ,即共有 400 个状态,编号为 1—400。

### 4.4 Q 值的初始化

根据虚拟网映射的先验知识,对  $Q(s,a)$  值初始化如下: $Q(s,1)=5, Q(s,0)=0$ ,当  $s=\langle 1,1,1-3,1-3 \rangle$  时; $Q(s,1)=0, Q(s,0)=2$ ,当  $s=\langle 5,5,4,4 \rangle$  时;在其他情况下, $Q(s,1)=Q(s,0)=0$ 。动作分为“接受”和“拒绝”两种,分别以 1 和 0 表示。

### 4.5 回报函数的计算

在状态  $s_1$  下,如接受虚拟网构建请求,则即时回报值  $r$  为 1,如拒绝,则  $r$  为 0,即所有状态的拒绝动作的即时回报值都暂为 0,但如后继的虚拟网构建需要用到状态  $s_1$  所拒绝的资源,则要对  $Q(s_1,0)$  值重新修整,具体修正方法见 4.6 节。

### 4.6 LBRVM 算法流程

LBRVM 分两个过程:(1)用 LBVM 算法完成虚拟网映射;(2)如用 LBVM 算法能够完成虚拟网构建,则根据当前状态,按一定概率拒绝该请求。

算法中集合  $R_i (1 \leq i \leq 400)$  表示各状态下所拒绝资源的集合,该集合元素是四元组(资源类型、资源编号、资源量、结束时间),如  $R_1$  集合含元素  $\langle 0, 3, 10, 2000 \rangle$ ,则表示状态 1 拒绝了 3 号物理节点(资源类型 0 表示物理节点,1 表示物理链路)10 单位 CPU 容量,该元素将在时间 2000 单位时移出集合。算法具体步骤如下:

步骤 1 初始化  $Q(s,a)$ ;初始化  $R_k (1 \leq k \leq 400)$  为空。

步骤2 接受虚拟网构建请求。  
 步骤3 用LBVM算法完成虚拟网构建。  
 步骤4 如构建失败,则转2。  
 步骤5 计算状态变量(Gn, Ge, f<sub>nd</sub>, f<sub>ed</sub>),确定当前状态 s<sub>i</sub> (i是当前状态的编号)。  
 步骤6 以概率  $\epsilon$  随机选择拒绝还是接受当前请求,以概率  $1-\epsilon$  根据  $Q(s_i, 0)$  和  $Q(s_i, 1)$  的大小选择拒绝 ( $Q(s_i, 0)$  大) 或接受 ( $Q(s_i, 1)$  大)。  
 步骤7 删除  $R_k (1 \leq k \leq 400)$  中结束时间小于当前时间或资源量为零的所有元素。  
 步骤8 如拒绝,则对当前虚拟网所映射的每个物理网节点和每条物理链路,分别构造四元组元素,然后添加到集合  $R_i$  中。  
 步骤9 如选择的动作是接受,则分析所有拒绝状态对本次虚拟网构建的贡献度,根据贡献度修正相应拒绝状态的 Q 值,即  $Q(s_j, 0) = Q(s_j, 0) + \text{贡献度} \times \alpha$ , 其中  $\alpha$  是学习率。并把  $R_k (1 \leq k \leq 400)$  中对当前虚拟网构建有贡献的元素的资源量减去本次贡献的资源量。  
 步骤10 根据式(6),修正当前状态和所选择动作对应的 Q 值,转步骤2。

在步骤8构建四元组元素时,结束时间取虚拟网的生命周期结束时间;资源量取  $\text{Min}\{\text{当前所拒绝资源量, 该资源剩余量} - R_i \text{ 中该资源拒绝量之和}\}$ , Min 指取两个数的相对小值,即本次拒绝资源的资源量要减去属于以前已经拒绝过的部分。

步骤9中贡献度的计算方法是,先计算  $R_i (1 \leq i \leq 400)$  中对当前虚拟网构建有贡献的元素所贡献的“资源量”(如物理节点  $n_i$  的 CPU 容量是 100, 剩余 CPU 容量是 60, 400 个状态中仅有  $s_1$  和  $s_2$  两个状态拒绝  $n_i$  节点,拒绝的 CPU 容量分别是 15 和 30, 当前虚拟网构建需要  $n_i$  的 CPU 容量是 45, 则  $s_1$  和  $s_2$  状态贡献的  $n_i$  节点 CPU 容量是  $45 - (60 - 15 - 30) = 30$ , 按比例计算  $s_1$  和  $s_2$  所贡献的  $n_i$  节点的 CPU 量, 即 10 和 20), 然后累积各  $R_i$  中各元素对虚拟网构建所贡献的资源总量  $CR_i$ ,  $CR_i / \sum CR_i$  即  $i$  状态的贡献度。

在步骤10中,因虚拟网请求是随机到达的,故下一个状态要等到下一个能够完成映射的虚拟网请求到达后才能计算,即真正的修正也要等到下一个能够完成映射的虚拟网构建请求到达后才能进行,即在本算法流程下一次执行到步骤5之后、步骤6之前完成修正。

## 5 算法分析

### 5.1 时间复杂性分析

算法中求解 K 短简单路径,采用 N. Katoh 先生在文献[17]提出的算法,在无向图中,该算法的复杂度是  $O(K(m + n \log n))$ , 其中  $m$  是边数,  $n$  是顶点数。

LBVM 和 LBRVM 算法的时间复杂度由 LBVM 算法的步骤6决定,即其最坏时间复杂度为  $O(K \times |N^v| \times |N^e| \times |E^v| \times (|E^v| + |N^e| \times \log(|N^e|)))$ 。

### 5.2 算法最差性能理论分析

在线算法性能分析的常用方法是竞争分析法<sup>[15,18]</sup>,该方法以相应问题的离线(假设所有决策信息都已知)最优决策作为参照,用在线算法在最坏情况下的性能来衡量该算法的好坏。定量地讲,如果对于一个在线决策问题(最大化问题)的所有实例,某在线算法得到的优化目标函数值都至少是相

应的最优离线决策目标函数值的  $1/p (p \geq 1)$ , 则称该算法是  $p$  竞争的,使该算法为  $p$  竞争的所有  $p$  中的最小值称为该算法的竞争比。尽管竞争分析过于保守,但由于在定量分析上的可行性,它是在线算法分析领域中的主流方法。下面给出以最大化物理网提供商收益为目标的两类虚拟网映射算法的竞争比。

**定理1** 无拒绝虚拟网映射算法的竞争比趋于无穷大。

证:通过构造一个实例来证明。具体实例如下:物理网只有两个物理节点和一条物理链路,物理链路的带宽为 1, 物理节点的 CPU 容量为 1; 虚拟网请求序列是  $(B1, A1)$ , 虚拟网  $B1$  和  $A1$  的节点映射的距离约束为  $D$ ,  $B1$  和  $A1$  都只有两个虚拟节点和一条虚拟链路,  $B1$  的两个虚拟节点的 CPU 容量和虚拟链路的带宽都为  $\epsilon (\epsilon \rightarrow 0)$ ,  $A1$  的两个虚拟节点的 CPU 容量和虚拟链路的带宽都为 1, 虚拟网  $B1$  和  $A1$  的虚拟节点到物理网两个物理节点的距离都小于  $D$ , 虚拟网  $B1$  的生存期长于  $A1$ 。则对无拒绝虚拟网映射算法,必然是接受虚拟网  $B1$  的建立请求而拒绝  $A1$ , 其收益为  $(a+b)\epsilon$  (单位 CPU 容量的收费价格和单位链路带宽的收费价格分别为  $a$  和  $b$ ); 而离线最优算法是拒绝  $B1$  请求而接受  $A1$  请求, 其收益为  $a+b$ , 那么针对该实例,无拒绝虚拟网映射算法的竞争比为:  $(a+b)/((a+b)\epsilon) = 1/\epsilon \rightarrow +\infty$ 。定理1得证。

**定理2** 虚拟网映射随机算法(拒绝概率  $1-P$ )的竞争比小于等于  $1/(p(1-p)^{m-1})$ ,  $m$  是物理网上同时存在的最多虚拟网个数,  $m \geq 1, 0 < p < 1$ 。

证:虚拟网映射随机算法是指先用无拒绝的虚拟网映射在线算法完成映射,然后对能够完成构建的虚拟网构建请求按概率  $1-p$  进行拒绝。下面采用数学归纳法进行证明,其中  $E(\text{ALG}(a_1, a_2, \dots, a_m))$  表示当有  $m$  个虚拟网构建请求且采用虚拟网映射随机算法时物理网提供商收益的数学期望;  $\text{OPT}(a_1, a_2, \dots, a_m)$  表示采用离线最优算法时物理网提供商的收益。另外,使用全部物理网资源都不能满足的单个虚拟网构建请求因没有意义而不予考虑。

(I)  $N=1$  (即只有一个虚拟网请求), 竞争比  $= \text{OPT}(a_1) / E(\text{ALG}(a_1)) = 1/p \leq 1/(p(1-p)^{m-1})$ 。

(II) 设  $N=k$ , 竞争比  $= \text{OPT}(a_1, a_2, \dots, a_k) / E(\text{ALG}(a_1, a_2, \dots, a_k)) \leq 1/(p(1-p)^{m-1})$  成立。

(III) 当  $N=k+1$  时:

• 如离线最优算法拒绝  $a_{k+1}$ , 则  $\text{OPT}(a_1, a_2, \dots, a_k, a_{k+1}) = \text{OPT}(a_1, a_2, \dots, a_k)$ , 那么, 竞争比  $= \text{OPT}(a_1, a_2, \dots, a_{k+1}) / E(\text{ALG}(a_1, a_2, \dots, a_{k+1})) \leq \text{OPT}(a_1, a_2, \dots, a_k, a_{k+1}) / E(\text{ALG}(a_1, a_2, \dots, a_k)) = \text{OPT}(a_1, a_2, \dots, a_k) / E(\text{ALG}(a_1, a_2, \dots, a_k)) \leq 1/p(1-p)^{m-1}$ , 得证。

• 如离线最优算法接受  $a_{k+1}$ , 则  $\text{OPT}(a_1, a_2, \dots, a_k, a_{k+1}) = x + \text{OPT}_{k+1}(a_1, a_2, \dots, a_k)$ ,  $x$  表示接受  $a_{k+1}$  所产生的收益,  $\text{OPT}_{k+1}(a_1, a_2, \dots, a_k)$  表示离线最优算法在剩余物理网络(指物理网中资源减去离线最优算法为  $a_{k+1}$  虚拟网所分配的物理网资源后的网络)上针对虚拟网请求  $(a_1, a_2, \dots, a_k)$  的收益。而  $E(\text{ALG}(a_1, a_2, \dots, a_k, a_{k+1})) = E(\text{ALG}(a_1, a_2, \dots, a_k)) + x_p(1-p)^q$ ,  $q$  表示在线算法为了完成  $a_{k+1}$  请求所需要的物理网资源与当前物理网上  $q$  个虚拟网所占资源有冲突, 则在线随机算法如要能够完成第  $a_{k+1}$  个虚拟网映射, 则在线随机算法必须在之前拒绝这  $q$  个虚拟网构建请求, 显然  $q \leq m$

-1.

$$\begin{aligned}
 & \text{竞争比} = (x + \text{OPT}_{k+1}(a_1, a_2, \dots, a_k)) / (E(\text{ALG}(a_1, a_2, \dots, a_k)) + xp(1-p)^q) \\
 & \leq (x + \text{OPT}_{k+1}(a_1, a_2, \dots, a_k)) / (\text{OPT}(a_1, a_2, \dots, a_k) \times p \times (1-p)^{m-1} + xp(1-p)^q) \\
 & \leq (x + \text{OPT}(a_1, a_2, \dots, a_k)) / (\text{OPT}(a_1, a_2, \dots, a_k) \times p \times (1-p)^{m-1} + xp(1-p)^q) \\
 & \leq (x + \text{OPT}(a_1, a_2, \dots, a_k)) / (\text{OPT}(a_1, a_2, \dots, a_k) \times p \times (1-p)^{m-1} + xp(1-p)^{m-1}) \\
 & = 1 / (p(1-p)^{m-1}), \text{得证。}
 \end{aligned}$$

从定理 1 和定理 2 可以看出,由于 LBRVM 算法对能够完成构造的虚拟网请求保持一定的随机拒绝性,因此 LBRVM 算法和无拒绝虚拟网映射算法相比改善了算法的最差性能。

### 5.3 算法平均性能实验分析

目前在线算法的平均性能分析尚未取得理论上的突破<sup>[15,18]</sup>,故本文采用实验的方法把 LBVM 算法和 LBRVM 算法与启发式算法 G-SP(虚拟链路映射采用最短路径算法,虚拟节点映射采用贪婪算法<sup>[6]</sup>)、均衡的自适应虚拟网构建方法 BACA<sup>[13]</sup>和基于负载均衡的 I-MMCF 算法(不允许分流)<sup>[14]</sup>进行比较分析,从而评价 LBVM 算法和 LBRVM 算法的平均性能。

#### 5.3.1 仿真环境及性能评估指标

对 LBVM 算法和 LBRVM 算法平均性能的评估,通过 Matlab 模拟仿真来进行。对算法性能的评估指标,除了使用虚拟网构建请求的接受率(虚拟网构建成功的个数占构建请求数的百分比)和物理网提供商的平均收益(单位时间虚拟网映射收益,见定义 5)外,还使用物理节点利用率、物理链路利用率、物理节点使用均衡度、物理链路使用均衡度、物理节点最高负载(针对单个物理节点)和物理链路最高负载(针对单条物理链路)等指标来衡量物理网资源的利用情况和资源使用的均衡性。

另外,单位 CPU 容量的收费价格和单位链路带宽的收费价格都取 1;Q 学习算法相关参数设定如下:学习率  $\alpha=0.1$ ,折扣因子  $\gamma=0.9$ ,探索率  $\epsilon=0.05$ 。

#### 5.3.2 实验数据的设定

物理网提供商的物理网采用随机的方式产生,物理网提供商的业务(构建虚拟网)的到达是一个泊松过程。

为了衡量在不同的业务量下各算法的平均性能指标,本实验数据分为两组。第一组数据:每 100 时间单位平均有 3 个虚拟网构建请求( $\lambda=3$ ),对应业务量较轻的情况;第二组数据:每 100 时间单位平均有 6 个虚拟网构建请求( $\lambda=6$ ),对应业务量较重的情况。两组数据的其它部分相同,具体如下:物理网有 50 个物理节点,两两节点之间用 0.5 的概率随机连接,物理节点的 CPU 容量和物理链路的带宽在 50 到 100 整数间均匀分布,表示物理节点所在地的  $x$  和  $y$  值都在 1 到 100 整数间均匀分布。每个虚拟网的生存期符合指数分布,平均每个虚拟网生存期为 1000 个时间单位。每个虚拟网请求的虚拟节点数在 2 和 10 之间等概率随机产生,虚拟网平均连通度是 50%,虚拟节点的 CPU 容量在 1 到 20 整数间均匀分布,虚拟链路的带宽在 1 到 50 整数间均匀分布,表示虚拟节点所在地的  $x$  和  $y$  值都在 1 到 100 整数间均匀分布,所有

虚拟网的虚拟节点所在地与所映射的物理节点所在地间的距离必须小于等于 10。

#### 5.3.3 实验结果及分析

##### (1) 虚拟网构建请求接受率和映射收益分析

从图 1—图 4 的数据中明显观察到,当虚拟网构建请求数不断增多时,随着物理网负载逐渐加重,虚拟网构建请求接受率和平均收益接近线性下降。但当请求数到达一定数量之后,再进一步增加请求数,原来存在的虚拟网到达生存时间不断释放网络资源,虚拟网构建的接收率和平均收益会逐渐达到稳态。从实验结果可以看出,在业务量较轻和业务量较重的两种情况下,采用 LBVM 算法和 LBRVM 算法都有利于提高虚拟网接受率和物理网提供商的平均收益。且在物理网资源负载较重(业务量较重)的情况下,LBVM 算法和 LBRVM 算法的性能更加突出。

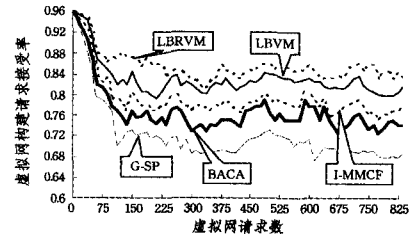


图 1 虚拟网构建请求接受率( $\lambda=3$ )

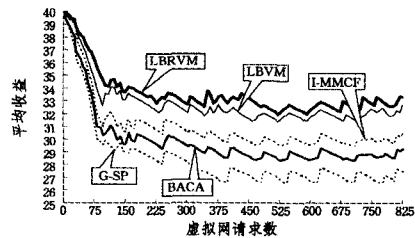


图 2 物理网提供商平均收益( $\lambda=3$ )

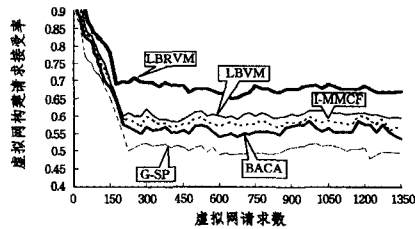


图 3 虚拟网构建请求接受率( $\lambda=6$ )

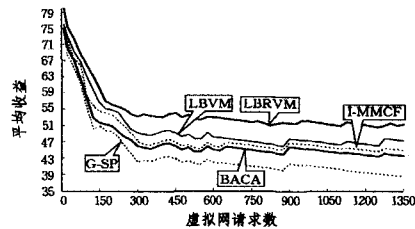


图 4 物理网提供商平均收益( $\lambda=6$ )

##### (2) 物理网负载均衡性和资源利用情况分析

表 1 和表 2 统计了在业务量较重( $\lambda=6$ )的情况下,有关物理网负载均衡的各项指标。其中重负载节点比例和重负载链路比例对应于在虚拟网请求数为 750(此时虚拟网接受率已经较稳定)时,负载超过平均负载 50%的物理节点比例和物理链路比例。表 1 和表 2 表明:①每种算法仍会出现瓶颈

的节点,但采用 LBVM 算法和 LBRVM 算法后,瓶颈节点相对少于其它算法;②采用 LBVM 算法和 LBRVM 算法时,物理网的节点使用均衡度和链路使用均衡度要高于其它算法;③采用 LBVM 算法和 LBRVM 算法时,节点的平均利用率和链路的平均利用率更高。这些数据说明 LBVM 算法和 LBRVM 算法能够将负载更加均衡地分布,既提高了网络资源的利用率,又降低了瓶颈节点的数量。

表 1 资源均衡利用情况(1)

| 算法     | 节点平均<br>利用率 | 链路平均<br>利用率 | 节点使用<br>均衡度 | 链路使用<br>均衡度 |
|--------|-------------|-------------|-------------|-------------|
| G-SP   | 0.220       | 0.483       | 0.645       | 0.631       |
| BACA   | 0.226       | 0.491       | 0.711       | 0.721       |
| I-MMCF | 0.229       | 0.494       | 0.692       | 0.742       |
| LBVM   | 0.233       | 0.501       | 0.765       | 0.771       |
| LBRVM  | 0.245       | 0.512       | 0.793       | 0.789       |

表 2 资源均衡利用情况(2)

| 算法     | 节点最<br>高负载 | 链路最<br>高负载 | 重负载<br>节点比例 | 重负载<br>链路比例 |
|--------|------------|------------|-------------|-------------|
| G-SP   | 0.465      | 0.975      | 10%         | 12%         |
| BACA   | 0.432      | 0.811      | 8%          | 7.5%        |
| I-MMCF | 0.451      | 0.757      | 8%          | 6.4%        |
| LBVM   | 0.403      | 0.756      | 6%          | 6.2%        |
| LBRVM  | 0.399      | 0.735      | 5%          | 5.4%        |

(3) 虚拟网构建的拒绝概率分析

表 3 是在业务量较重( $\lambda=6$ )的情况下, LBRVM 算法仿真结束后  $Q(s,a)$  值的部分数据(选取物理网负载较轻、较重和均值这 3 种情况下的部分数据)。从表 3 可以看出, LBRVM 算法在物理网负载较轻时,拒绝虚拟网构建请求的概率很低;在物理网负载较重时,将以较高概率拒绝会导致物理网较严重的不均衡利用的虚拟网构建请求,而对那些对物理网均衡度影响很小或者能够提高物理网均衡度的虚拟网构建请求的拒绝概率很小。结合表 1 和表 2 数据可知,由于在物理网负载较重时, LBRVM 算法将以较高概率拒绝会导致物理网较严重的不均衡利用的虚拟网构建请求,使得相对于 LBVM 算法,采用 LBRVM 算法时物理网负载分布更加均衡且瓶颈节点更少。从图 3 和图 4 数据可以观察到,当物理网络上运行的虚拟网个数达到一定规模后,采用 LBRVM 算法的虚拟网构建成功率和平均收益要比采用 LBVM 算法高 10% 左右。从表 1 可以看出,由于虚拟网构建成功率的提高, LBRVM 算法的物理网资源利用率更高。

表 3  $Q(s,a)$  值表

| Gn | Ge | $f_{nd}$          | $f_{ed}$          | $Q(s,1)$ | $Q(s,0)$ |
|----|----|-------------------|-------------------|----------|----------|
| 1  | 1  | $(-\infty, 0.06]$ | $(-\infty, 0.06]$ | 8.05     | 0.16     |
| 1  | 1  | $(0.06, 0.10]$    | $(0.06, 0.10]$    | 8.35     | 0.18     |
| 1  | 1  | $(0.10, 0.14]$    | $(0.10, 0.14]$    | 6.84     | 0.17     |
| 1  | 1  | $(0.14, 1)$       | $(0.14, 1)$       | 1.68     | 0.59     |
| 2  | 4  | $(-\infty, 0.06]$ | $(-\infty, 0.06]$ | 4.25     | 2.22     |
| 2  | 4  | $(0.06, 0.10]$    | $(0.06, 0.10]$    | 4.76     | 2.26     |
| 2  | 4  | $(0.10, 0.14]$    | $(0.10, 0.14]$    | 3.54     | 2.18     |
| 2  | 4  | $(0.14, 1)$       | $(0.14, 1)$       | 1.97     | 1.10     |
| 5  | 5  | $(-\infty, 0.06]$ | $(-\infty, 0.06]$ | 1.64     | 0.31     |
| 5  | 5  | $(0.06, 0.10]$    | $(0.06, 0.10]$    | 1.03     | 1.03     |
| 5  | 5  | $(0.10, 0.14]$    | $(0.10, 0.14]$    | 0.14     | 1.21     |
| 5  | 5  | $(0.14, 1)$       | $(0.14, 1)$       | 1.10     | 3.43     |

**结束语** 本文首先介绍虚拟网映射算法的研究现状,并从算法竞争比和资源分配均衡性的角度对当前虚拟网映射算

法存在的问题进行分析;然后针对存在的问题,分别设计了负载均衡虚拟网映射算法 LBVM 和负载均衡虚拟网映射随机算法 LBRVM;最后对相关算法进行了理论分析和实验验证,说明了所设计的算法的有效性和实用性。

参考文献

[1] Chowdhury N M M K, Boutaba R. A survey of network virtualization[J]. Computer Networks, 2010, 54(5): 862-876

[2] 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

[3] Fischer A, Botero J F, Beck M T, et al. Virtual Network Embedding: A Survey[J]. IEEE Communications Surveys and 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 on Computer Communication Review, 2008, 38(2): 17-29

[5] Andersen D. Theoretical approaches to node assignment [EB/OL]. <http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps>, 2002

[6] Zhu Y, Ammar M. Algorithms for assigning substrate network resources to virtual network components[C]// IEEE International Conference on Computer Communications (INFOCOM). Spain, 2006: 1-12

[7] Lu J, Turner J. Efficient mapping of virtual networks onto a shared substrate[R]. Louis: Washington University Washington University, 2006

[8] Ricci R, Alffld C, Lepreau J. A Solver for the Network Testbed Mapping Problem[J]. ACM SIGCOMM Computer Communications Review, 2003, 33(2): 65-81

[9] Mosharaf Kablr Chowdhury N M, Muntasir R R, Raouf B. ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping[J]. IEEE/ACM Transactions on Networking, 2012, 20(1): 206-219

[10] Jens L, Holger K. A virtual network mapping algorithm based on subgraph isomorphism detection[C]// Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures. Spain, 2009: 81-88

[11] Cheng X, Su S, Zhang Z B. Virtual Network Embedding Through Topology-Aware Node Ranking[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(2): 39-47

[12] 姜明, 王保进, 吴春明. 网络虚拟化与虚拟网映射算法研究[J]. 电子学报, 2011, 39(6): 1315-1320

[13] 齐宁, 王保进, 汪斌强. 均衡虚拟网构建算法研究[J]. 电子与信息学报, 2011, 33(6): 1301-1306

[14] 王浩学, 姜明, 付吉. 基于负载均衡的逻辑承载网构建研究[J]. 通信学报, 2012, 33(9): 38-43

[15] Borodin A, El Yaniv R. Online Computation and Competitive Analysis[M]. New York: Cambridge University Press, 1998

[16] 黄炳强. 强化学习方法及其应用研究[D]. 上海: 上海交通大学自动化系, 2007

[17] Katoh N, Ibaraki T, Mine H. An efficient algorithm for K shortest simple paths[J]. Networks, 1982, 12(4): 411-427

[18] 陶继平, 席裕庚. 一种新的在线调度算法竞争比分析方法——基于实例转换的方法[J]. 系统科学与数学, 2009, 29(1): 1381-1389