

基于优化的同构子图搜索的虚拟网络映射算法

魏晓辉, 邹磊, 李洪亮

(吉林大学 计算机科学与技术学院, 长春 130012)

摘 要:针对现有虚拟网络映射算法的不足, 首先提出了一个综合考虑网络中节点资源需求(能力)和拓扑属性的节点资源能力评价方法, 合理地评价节点资源能力, 优化了虚拟节点的映射顺序; 其次改进了虚拟节点映射时备选物理节点的选择过程, 提高了虚拟链路映射质量; 最后通过考虑物理网络子区域内的资源总量, 优化了算法中初始资源分配区域的选择。实验结果表明: 与已有的算法相比, 本文映射算法在映射质量、长期平均收益、长期平均接收率、算法执行时间等方面均有明显提高。

关键词: 计算机应用; 虚拟网络; 映射算法; 同构子图搜索; 网络拓扑结构

中图分类号: TP393 **文献标志码:** A **文章编号:** 1671-5497(2013)01-0165-07

Virtual network embedding algorithm based on improved sub-graph isomorphism search

WEI Xiao-hui, ZOU Lei, LI Hong-liang

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: A new Virtual Network Embedding (VNE) algorithm is proposed, which improves the original sub-graph isomorphism search process, and overcomes the defects in existing VNE algorithm. First, a node resource evaluation method is proposed, which takes both node resource requirement (capability) and topology attribute into account, to improve the mapping order of the virtual nodes. Second, the algorithm improves the selection process of candidate substrate nodes when mapping the virtual nodes, which enhances the quality of virtual link mapping. Third, the algorithm improves the selection of resource allocation sub-area in substrate network by considering the total resource capability in the sub-area. Experiment results show that the proposed algorithm performs better in mapping quality, revenue, acceptance ratio and runtime compared with existing algorithms.

Key words: computer application; virtual network; mapping algorithm; sub-graph isomorphism search; network topology

利用网络虚拟化技术, 可以在现有的物理网络上构建多个异构虚拟网络(Virtual network, VN), 各个 VN 共享同样的底层物理网络资源,

运行不同的应用程序、网络协议, 为不同用户的需求提供定制的服务^[1-2]。每个虚拟网络由运行在不同物理节点上的多个虚拟节点和建立在物理链

收稿日期: 2012-01-20.

基金项目: 国家自然科学基金项目(61170004); 新世纪优秀人才支持计划项目(NCET-09-0428); 深部探测技术与实验研究专项项目(SinoProbe-09-01).

作者简介: 魏晓辉(1972-), 男, 教授, 博士生导师. 研究方向: 分布式系统, 网格系统, 网络安全.

E-mail: weixh@jlu.edu.cn

通信作者: 李洪亮(1983-), 男, 讲师, 博士. 研究方向: 分布式系统, 虚拟化. E-mail: lihongliang@jlu.edu.cn

路上并连接虚拟节点的虚拟链路组成。

由于虚拟节点和虚拟链路均有资源需求,虚拟网络映射是一个 NP 困难问题^[3]。目前的虚拟网络映射算法研究中很多是基于启发式方法^[3-6],这些算法存在以下两个问题:①映射质量较差的问题。它们主要将节点映射与链路映射分阶段进行:先用类似贪心算法将虚拟节点映射到物理节点上,再用最短路径算法或多商品流算法完成虚拟链路的映射。但是,将虚拟节点预先映射的做法限制了虚拟链路映射时的求解域,可能降低算法性能和虚拟网络映射质量;②映射性能存在问题。文献[7]等利用同构子图搜索算法,将节点映射和链路映射在同一阶段完成。但该算法复杂度较高,映射性能受到较大影响。目前多数算法^[3,6-7]在评价网络中节点的资源能力时只是以其自身的资源能力作为评价标准,没有考虑节点在网络中的拓扑属性对节点综合能力的影响。

针对以上问题,本文提出了一个基于同构子图搜索^[8]的虚拟网络映射算法,并进行了实验验证。实验结果表明,本文提出的算法在执行时间、映射质量、映射成功率和资源收益上都明显优于已有的基于相同同构子图搜索算法^[8]的虚拟网络映射算法^[7]。

1 相关工作

虚拟网络映射问题中,节点和链路都带有资源限制,导致其为 NP 难问题^[3]。早期的虚拟网络映射研究都或多或少地减小了问题域来降低问题的求解难度。例如,假设事先知道所有的虚拟网络请求(off-line 模式)^[5-6];只考虑一种限制条件(CPU 或带宽)^[4-5];假设物理资源的能力没有限制,没有考虑任务接纳控制^[4-6];只考虑单一网络拓扑^[5]。

在不减小问题域的情况下,文献[3]提出了物理链路支持虚拟链路分裂和迁移的虚拟网络映射算法。文献[1]重点考虑了虚拟节点的位置限制,使用混合整数规划的方法来解决虚拟网络映射问题。文献[7]将物理网络和虚拟网络描述成一个带权有向图,提出了一个基于同构子图搜索的虚拟网络映射算法。但是,文献[7]中的算法只是初步地将同构子图搜索算法应用在虚拟网络映射中,缺乏对搜索过程的进一步优化;在评价节点资源能力时,文献[7]也仅考虑节点自身的资源能力,没有将节点在网络中的拓扑属性考虑进来。

文献[9]提出了通过拓扑感知来寻找物理网络中的瓶颈节点和链路的方法,优化并调整已有的虚拟网络资源分配方案。

本文假设物理网络资源总是可用的,将研究的重点放在虚拟网络的初次映射中。在不减小问题域的前提下,提出了一个基于优化的同构子图搜索的虚拟网络映射算法。

2 网络模型及问题描述

2.1 网络模型

物理网络可以被描述成一个任意拓扑结构的带权无向图 $G_S = (N_S, E_S, A_S^n, A_S^e)$ 。其中: N_S 和 E_S 分别代表物理节点集合和物理链路集合; A_S^n 代表物理节点的属性信息,如 CPU 能力; A_S^e 代表物理链路的属性信息,如带宽、延时。用 P_S 表示物理网络中所有无环路径的集合。图 1(b)是一个物理网络实例,节点旁方框中的数字代表该节点的可用 CPU 能力,链路上的数字表示该链路的可用带宽。

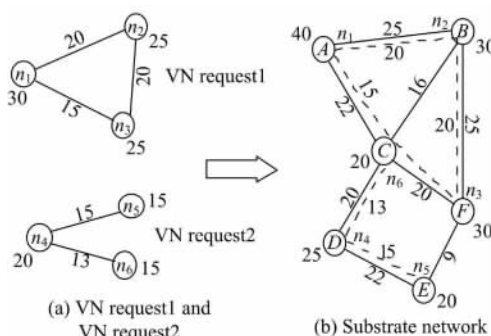


图1 虚拟网络映射实例

Fig. 1 Virtual network embedding instance

与物理网络类似,用一个任意拓扑结构的带权无向图 $G_V = (N_V, E_V, A_V^n, A_V^e)$ 表示虚拟网络。其中 N_V 和 E_V 分别代表虚拟节点集合和虚拟链路集合, A_V^n 和 A_V^e 分别代表虚拟节点和虚拟链路所需的资源信息。本文重点考虑虚拟节点所需的 CPU 能力和虚拟链路所需的网络带宽大小。图 1(a)是两个虚拟网络的实例。

2.2 虚拟网络映射问题描述

虚拟网络映射问题可以描述成将 G_V 映射到 G_S 的一个子图上: $M: G_V(N_V, E_V) \rightarrow G_S(N'_S, P'_S)$, 其中 $N'_S \subset N_S$ 且 $P'_S \subset P_S$ 。虚拟网络映射可以被分解成节点映射和链路映射两个部分,节点映射为 $M_N: N_V \rightarrow N'_S$, 链路映射为 $M_E: E_V \rightarrow P'_S$ 。节点映射中,每个虚拟节点都被映射到一个能够满足其

计算能力需求的物理节点上。链路映射中,每条虚拟链路都被映射到一个能够满足其网络带宽需求的无环物理路径上。图 1 中, VN request1 中节点 n_1, n_2, n_3 分别映射到物理节点 A、B、F,对应的虚拟链路 $(n_1, n_2), (n_1, n_3), (n_2, n_3)$ 分别映射到物理路径 $(A, B), (A, C, F), (B, F)$, 对应的物理节点和物理路径都必须满足虚拟节点和链路的资源需求。

2.3 算法目标

虚拟网络映射算法的主要目标是在保证虚拟网络资源需求的前提下使物理资源得到有效使用。

与文献[1,3,6-7]类似,将成功映射一个虚拟网络到物理网络的收益定义为

$$R(G_V(t)) = \sum_{e_V \in E_V} BW(e_V) + \sum_{n_V \in N_V} CPU(n_V) \quad (1)$$

式中: t 表示该虚拟网络请求到达的时间; $CPU(n_V)$ 表示虚拟节点 n_V 的 CPU 需求; $BW(e_V)$ 表示虚拟链路 e_V 的网络带宽需求。

与文献[7]类似,将成功映射一个虚拟网络的开销定义为分配给该虚拟网络的资源总和:

$$C(G_V(t)) = \sum_{e_V \in E_V} (BW(e_V) \cdot \text{length}(M(e_V)) + \sum_{n_V \in N_V} CPU(n_V)) \quad (2)$$

式中: $M(e_V) \in P_s$ 表示 e_V 被映射到物理网络上的无环路径; $\text{length}(M(e_V))$ 表示这个无环路径的长度,将在后文中对该值的大小做出限制。

与文献[7]类似,用成功映射一个虚拟网络的收益与开销的比值来评价映射方案的质量:

$$R/C = \frac{R(G_V(t))}{C(G_V(t))} \quad (3)$$

该值越大,说明该映射方案的匹配程度越高,浪费的网络带宽资源越少,产生的额外通信延迟越少。

一个有效的 on-line 虚拟网络映射算法能够最大化物理资源的长期收益。与文献[3]类似,将虚拟网络映射算法的长期平均收益定义为

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G_V(t))}{T} \quad (4)$$

3 基于节点拓扑属性的资源评价

现有的虚拟网络映射算法^[3,6-7]在评价节点的资源能力(需求)时,用节点自身的计算能力(需

求)和可用带宽(所需带宽)或同时考虑这两者作为评价条件。与文献[3]类似,将这种评价方式表示为节点自身可用 CPU 能力与其相邻链路的可用网络带宽之和的乘积:

$$CB(n) = CPU(n) \cdot \sum_{e \in E(n)} BW(e) \quad (5)$$

式中: n 表示节点; $E(n)$ 表示 n 邻接链路的集合。

一个节点的资源能力,除了它的自身能力以外,其邻接(直接、间接)节点的能力也会对其产生影响。假设拓扑结构中两个节点的 CB 值(见式(5))相同,第一个节点的邻接节点的 CB 值较大,则其综合资源能力(需求)更强。由于其映射难度更大,在映射过程中该节点需要被优先考虑。对于给定节点 n ,其邻接节点对其资源能力的影响定义为

$$TR(n') = \frac{CB(n')}{\text{Dis}(n, n')} \quad (6)$$

式中: n' 是 n 的邻接节点; $\text{Dis}(n, n')$ 表示 n 与 n' 之间的最短跳数,跳数越大 n' 对 n 的影响越小。节点 n 的综合资源能力定义为

$$\text{sumTR}(n) = \sum_{n' \in N} TR(n') + CB(n) \quad (7)$$

式中: N 表示拓扑结构中节点 n 的邻接节点集合。

为物理节点计算 sumTR 时,考虑到虚拟网络运行时对物理网络资源使用的局部性,仅考虑以该节点为中心,以 G_V 的平均距离的一半为半径之内的节点对其资源能力的影响。为虚拟节点计算 sumTR 时,考虑 G_V 中所有虚拟节点的相互影响,即 N 为 $(N_V - n)$ 。 sumTR 值越大,则节点 n 的资源能力(需求)越强。

4 虚拟网络映射算法

本文提出的虚拟网络映射(Virtual network embedding-VF, VNE-VF)算法,是一种基于同构子图搜索算法的 VF 算法^[8]。

4.1 资源分配区域选择

本文算法是一个回溯型算法,每轮计算为一个虚拟节点分配资源。第一个虚拟节点的宿主物理节点基本决定了该虚拟网络在物理网络中的资源分配区域。优先选择资源相对丰富的子区域进行资源分配,能够有效平衡物理网络的负载,减少瓶颈节点、链路和局部拥塞的产生。所以,对于每个物理节点,计算以该节点为中心,以 G_V 平均距离的一半为半径之内所有节点 sumTR 值的和,

选择和值最大的物理节点作为第一个虚拟节点的宿主物理节点。

4.2 虚拟节点及物理节点选择

本文算法中,虚拟节点的映射顺序和映射位置对算法执行成功率和映射质量有很大影响。算法每轮计算中首先得出该轮的备选虚拟节点集合,然后从中选择 sumTR 值最大的节点进行映射。每轮映射的备选虚拟节点集合可表示为: $\text{Enode}_{G_V^{\text{sub}}}(G_V) = \{n_j \mid (n_i, n_j) \in E_V \wedge (n_i \in N(G_V^{\text{sub}}) \wedge n_j \notin N(G_V^{\text{sub}}))\}$ 。

G_V^{sub} 为 G_V 中已经被映射到 G_S 的部分, $N(G_V^{\text{sub}})$ 为 G_V^{sub} 所包含的节点集合。 $\text{Enode}_{G_V^{\text{sub}}}(G_V)$ 即与已完成映射的虚拟节点直接相连的虚拟节点的集合。算法第一轮中 G_V^{sub} 为空,选择 G_V 中 sumTR 值最大的节点进行映射。

每个虚拟节点在映射时,物理网络中可能存在多个节点满足其资源需求。为了提高虚拟链路映射的质量,对这些备选物理节点排序时,除了考虑该节点的资源能力,对其与 $N_S(G_V^{\text{sub}})$ 中的物理节点的位置关系也应该加以考虑。 $N_S(G_V^{\text{sub}})$ 中包含的是,与当前被选择的虚拟节点有直接通信关系的并且已经完成映射的虚拟节点所映射到的物理节点。给定一个备选物理节点 n ,其优先级定义为

$$R(n) = \frac{\text{sumTR}(n)}{\sum_{n' \in N_S(G_V^{\text{sub}})} \text{Dis}(n, n')} \quad (8)$$

备选物理节点根据其 $R(n)$ 值排序, $R(n)$ 越大,说明该备选物理节点的资源能力越强,并且与 $N_S(G_V^{\text{sub}})$ 中物理节点之间的平均距离越近,潜在的相应虚拟链路映射的成功率越高,映射到的物理链路的长度越小,映射质量越好。

4.3 VNE-VF 映射算法

算法主体框架见算法 1,其中 $M(G_V^{\text{sub}})$ 为当前 G_V^{sub} 在 G_S 上合理的映射方案。每轮映射中 generateCPS(\cdot) 函数(见下文)负责选出本轮要被映射的虚拟节点,选择和排序物理节点,建立备选节点匹配对集合(CPS)。其中,备选物理节点除了必须满足该虚拟节点的所有资源需求,还要保证与 $N_S(G_V^{\text{sub}})$ 中的物理节点之间的最短路径长度小于式(2)中无环物理路径长度的限制值 MaxLength。

算法 1:

VNE-VF($G_V, G_S, G_V^{\text{sub}}, M(G_V^{\text{sub}})$)

前期计算准备:

为每个在 G_V 和 G_S 中的节点计算 sumTR 值
计算每个物理节点周围子区域内的总资源能力(以 G_V 的平均距离的一半为半径),并设置 current_backtrack 为 0

1. if ($G_V^{\text{sub}} = G_V$) then
2. return Embedding_Success
3. end if
4. CPS = generateCPS($G_V, G_S, G_V^{\text{sub}}, M(G_V^{\text{sub}})$)
5. k = 1
6. while (k < MaxLength)
7. for CPS 中的每个匹配对(n_v, n_s) do
8. if 每一条 n_v 和 G_V^{sub} 之间的通信骨架链路,在物理网络 G_S 中都能找到一条路径,满足其网络带宽需求并且满足当前的 k-hop 跳数限制,使其能够被成功映射 then
9. 更新 G_V^{sub} 和 $M(G_V^{\text{sub}})$,将(n_v, n_s)加入其中
10. if VNE-VF ($G_V, G_S, G_V^{\text{sub}}, M(G_V^{\text{sub}})$) == 0 then
11. current_backtrack ++
12. 进行回溯,通过测试 CPS 中 n_v 的下一个匹配对重新映射 n_v
13. end if
14. end if
15. end for
16. k ++
17. end while

然后,检验把备选节点匹配对(n_v, n_s)加入到 $M(G_V^{\text{sub}})$ 后生成的新的部分映射方案是否仍然合理。检验条件为:对于每个在 $M(G_V^{\text{sub}})$ 中并与 n_v 有虚拟链路连接的虚拟节点,其映射到的物理节点与 n_s 之间在物理网络上是否有一条满足当前 k-hop 跳数限制和虚拟链路带宽需求的最短路径。如果(n_v, n_s)通过检验,则算法进入下一轮映射。如果当前虚拟节点的所有备选匹配对都没能通过检验,则算法进行回溯。

决定回溯型算法执行时间上限的因素是算法的回溯次数。本文同文献[7]一样,将算法的回溯次数限制为 $4n$ 。

generateCPS($G_V, G_S, G_V^{\text{sub}}, M(G_V^{\text{sub}})$) 函数:

1. 计算出 $\text{Enode}_{G_V^{\text{sub}}}(G_V)$
2. if $\text{Enode}_{G_V^{\text{sub}}}(G_V)$ 为空 then
3. 从 G_V 中选择本轮要映射的虚拟节点 n_v , 为 n_v 找出备选物理节点
4. else
5. 从 $\text{Enode}_{G_V^{\text{sub}}}(G_V)$ 中选择本轮要映射的虚拟

节点 n_v , 为 n_v 找出备选物理节点

6. end if
7. 将找出的备选物理节点排序
8. 生成备选节点匹配对 CPS
9. 返回 CPS

5 性能测试

5.1 测试设置

与文献[3,7]中类似,用 GT-ITM^[10] 工具生成一个有 100 个节点和约 500 条链路组成的网络拓扑,模拟一个中等程度的物理网络。每个物理节点的 CPU 能力和每条物理链路的可用带宽服从 50~100 的均匀分布。虚拟网络的规模 n 表示其中虚拟节点的数量,任意一对虚拟节点之间的连接概率为 0.5,虚拟节点的 CPU 需求服从 $0 \sim \alpha$ 的均匀分布,虚拟链路的网络带宽需求服从 $0 \sim \beta$ 的均匀分布。 α, β 越大,该虚拟网络的映射难度越大。考虑到试验中物理网络的直径为 5,将算法 1 中 VNE-VF 算法的 MaxLength 值设置为 5。

5.2 对比算法

由于文献[7]已经与文献[3]中的经典两阶段映射算法做了详细对比,所以本文将测试重点放在与文献[7]中的算法进行对比。对比算法如表 1 所示。

表 1 对比算法

Table 1 Compared algorithms

Notation	Algorithm description
VnmFlib_S	文献[7]中的 simple vnmFlib 算法。算法中使用式(5)评价节点的资源能力(需求)。
VnmFlib_A	文献[7]中的 advanced vnmFlib 算法,优化了 simple vnmFlib。
VNE-VF_CB	本文提出的 VNE-VF 映射算法,使用式(5)评价节点的资源能力(需求)。
VNE-VF_sumTR	本文提出的 VNE-VF 映射算法,使用式(7)评价节点的资源能力(需求)。

5.3 单虚拟网络请求测试

比较本文算法和文献[7]中算法在映射单个虚拟网络请求时的映射质量和执行效率。用虚拟网络规模 $n = \{10, 20, 30, 40\}$ 与 $\alpha = \{50, 80\}, \beta = \{50, 80\}$ 进行组合测试。每种组合的实验结果都是 100 次测试的平均值。实验结果见图 2 和图 3。

可以得到以下分析结果:

(1) VNE-VF 算法在映射质量和执行效率上均优于文献[7]中算法。图 2 中 VNE-VF_CB 与

VnmFlib_S、VnmFlib_A 都用式(5)评价节点的

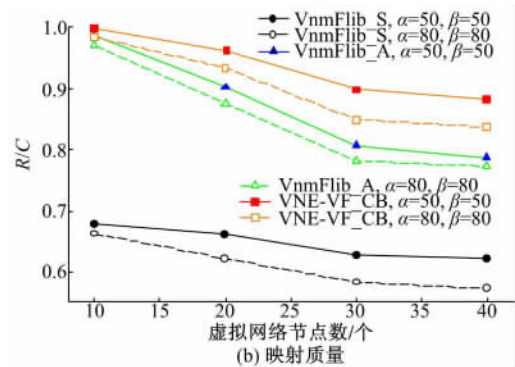
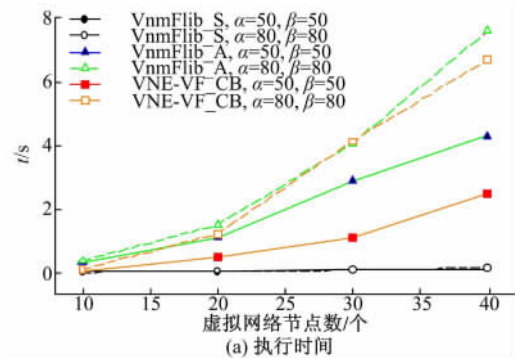


图 2 VnmFlib_S, VnmFlib_A 和 VNE-VF_CB 比较

Fig. 2 Comparison of VnmFlib_S, VnmFlib_A and VNE-VF_CB

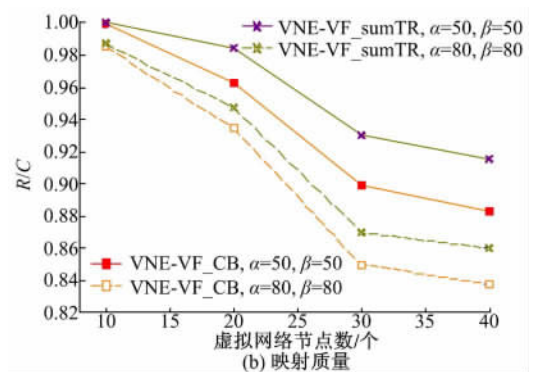
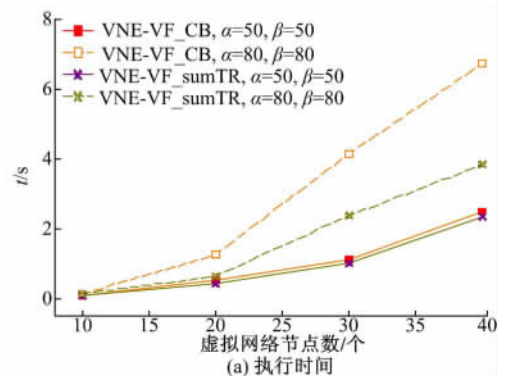


图 3 VNE-VF_CB 和 VNE-VF_sumTR 比较

Fig. 3 Comparison of VNE-VF_CB and VNE-VF_sumTR resource capability. Mapping quality aspect, in different n value and α, β value

下,VNE-VF_CB均明显优于文献[7]中的算法,并且随着 n 的增大,优势有逐渐增大的趋势。在 $n=40, \alpha=50, \beta=50$ 时,VNE-VF_CB的映射质量(R/C 值)比VnmFlib_A高大约10%。执行效率方面,在 n 大于10时,VNE-VF_CB与VnmFlib_A在 $\alpha=50, \beta=50$ 时执行时间相差大约0.8~1.8 s。这是因为VNE-VF算法优化了初始资源分配区域的选择和备选物理节点的选择过程,提高了虚拟链路的映射质量和映射效率。虽然VnmFlib_S在执行效率上有很大优势,但其在映射质量上与VnmFlib_A和VNE-VF_CB算法相差很大。

(2)sumTR方法能够更好地评价节点资源能力(需求)。图3中VNE-VF算法分别使用式(5)和式(7)评价节点的资源能力。映射质量方面,在不同 n 值和 α, β 值下,VNE-VF_sumTR均优于VNE-VF_CB,随着 n 的增大,优势有逐渐增大的趋势。 $n=40, \alpha=50, \beta=50$ 时,VNE-VF_sumTR比VNE-VF_CB的 R/C 值高大约4%。执行效率方面, $n=40, \alpha=80, \beta=80$ 时执行时间相差约3 s。这是因为sumTR方法在评价节点资源能力时除了考虑节点自身的能力外,还考虑了其拓扑属性。sumTR值越大的节点,其资源能力(需求)越强,越被优先选择,从而提高了虚拟网络映射质量和性能。

5.4 多虚拟网络请求测试

比较本文算法与文献[7]中算法处理动态在线虚拟网络请求时的长期平均收益、长期平均接收率、长期平均映射质量和算法执行时间。假设虚拟网络请求的到来符合泊松分布,每100个时间单位有5个虚拟网络请求;每个虚拟网络的生命周期符合指数分布,平均为500个时间单位;虚拟网络请求中节点的数量服从10~30的均匀分布;任意一对虚拟节点之间的连接概率为0.5;虚拟节点的计算能力需求符合0~30的均匀分布;虚拟链路的网络带宽需求符合0~50的均匀分布。每次实验时间为50000个时间单位,共约2500个虚拟网络请求。实验结果如图4所示。

根据图4中的数据可以得到以下分析结果:

(1)VNE-VF算法产生了更好的长期平均收益、长期平均接收率和长期平均映射质量(R/C 值)。图4(a)(b)中,VNE-VF_sumTR均产生了最好的长期平均收益和长期平均接收率。VNE-VF_sumTR和VNE-VF_CB的 R/C 值均明显好

于VnmFlib_A(图4(c))。VnmFlib_S与其他3个算法在以上3个指标上相比均相差很大。这是因为,VNE-VF算法通过对虚拟节点映射顺序、备选物理节点选择过程及初始资源分配区域选择

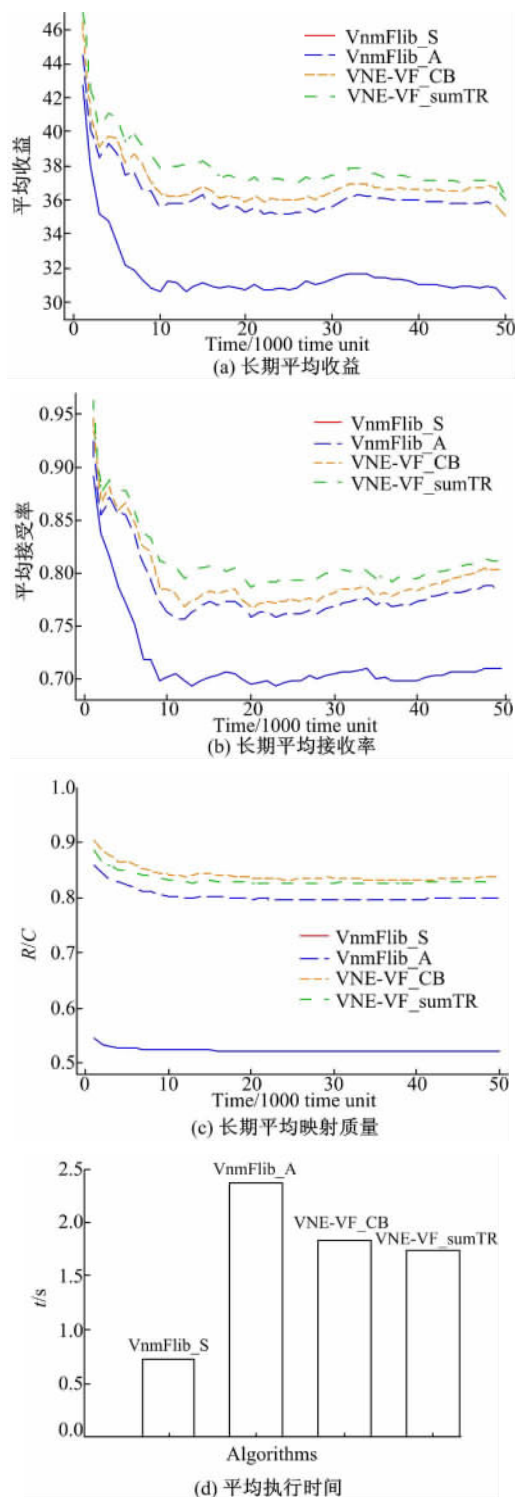


图4 实验结果比较

Fig. 4 Result comparison

进行了优化,有效提高了算法的映射质量和映射

成功率。

(2)物理网络的有效资源利用率。长期平均收益和长期平均接收率能反应物理网络的有效资源利用率。图4(a)(b)中 VNE-VF_sumTR 在这两个指标上都是最好的。这是因为, sumTR 优化了虚拟节点的映射顺序。使用 sumTR 时,第一个被映射的虚拟节点通常处于虚拟集群拓扑结构中心位置附近,且自身资源需求较大,即为虚拟集群中的“重心”。这有效地提高了映射质量。VNE-VF_sumTR 在长期平均收益和长期平均接收率上均明显好于 VNE-VF_CB,但在长期平均映射质量上略低于 VNE-VF_CB。这是由于使用式(5)评价节点资源能力产生了相对较多的资源碎片。

(3)综合考虑算法的总体性能,VNE-VF 算法在执行时间上明显优于文献[7]中的算法。图4(d)中显示了不同算法平均映射一个虚拟网络请求所用的时间。VnmFlib_S 用时最少,VNE-VF_sumTR 比 VNE-VF_CB 要多计算每个节点的 sumTR 值。但由于 sumTR 改善了算法性能,所以它们用时相差不多。VnmFlib_A 在4个算法中用时最多。

6 结束语

提出了一个基于优化的同构子图搜索的虚拟网络映射算法。该算法基于同构子图搜索算法做了改进和优化:①提出了一个新的节点资源能力评价方法,在评价节点的资源能力时着重考虑其拓扑属性;②改进了备选物理节点的选择顺序;③通过考虑物理网络子区域内的总资源能力,优化了初始资源分配区域的选择。实验结果显示,本文算法和已有的算法相比在映射质量、长期平均收益、长期平均接收率、算法执行时间等方面均有明显提高。

参考文献:

- [1] Chowdhury N M M K, Rahman M R, Boutaba R. Virtual network embedding with coordinated node and link mapping[C]// INFOCOM 2009, IEEE, 2009: 783-791.
- [2] Chowdhury N M M K, Boutaba R. Network virtualization: state of the art and research challenges[J]. IEEE Communications Magazine, 2009, 47(7):20-26.
- [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] Fan J, Ammar M. Dynamic topology configuration in service overlay networks: a study of reconfiguration policies[C]// The 25th IEEE International Conference on Computer Communications, 2006:1-12.
- [5] Lu J, Turner J. Efficient mapping of virtual networks onto a shared substrate[R]. Department of Computer Science and Engineering, Washington University in St. Louis, Technical Report WUCSE-2006-35, 2006.
- [6] Zhu Y, Ammar M. Algorithms for assigning substrate network resources to virtual network components[C]// IEEE INFOCOM Proceedings, Barcelona, Spain, 2006:1-12.
- [7] Lischka J, Karl H. A virtual network mapping algorithm based on subgraph isomorphism detection[C]// The 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures Proceedings, Barcelona, Spain, 2009:81-88.
- [8] Cordella L P, Foggia P, Sansone C, et al. A (sub) graph isomorphism algorithm for matching large graphs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(10):1367-1372.
- [9] Butt N, Chowdhury M, Boutaba R. Topology-awareness and re-optimization mechanism for virtual network embedding[C]// The 9th International Ifip Tc 6 Networking Conference, Chennai, India, 2010:27-39.
- [10] Zegura E, Calvert K, Bhattacharjee S. How to model an Internetwork? [C]// IEEE INFOCOM Proceedings, San Francisco, CA, USA, 1996: 594-602.

[1] Chowdhury N M M K, Rahman M R, Boutaba R. Virtual network embedding with coordinated node