

· 通信与信息工程 ·

点边联合协调的高效虚拟网络映射

虞红芳, 高秀娇

(电子科技大学通信与信息工程学院 成都 611731)

【摘要】把虚拟网络高效地映射到一个公共的物理网络对网络虚拟化非常重要。该文提出了一种基于线性规划松弛的点边联合协调的高效虚拟网络映射算法。该算法首先松弛MILP并添加约束条件得到加强的线性松弛模型STRICT_LP, 然后运用渐进启发式算法求解, 得到映射解。仿真结果显示, 该算法能够很好地实现虚拟网络点边协调映射, 在成功率、网络开销、收益和网络收益开销比值4个方面优于D-ViNE和R-ViNE算法。

关键词 混合整数线性规划; 网络虚拟化; 松弛; 虚拟网络映射

中图分类号 TN915

文献标识码 A

doi:10.3969/j.issn.1001-0548.2012.06.003

Efficient Virtual Network Embedding with Jointly Coordinated Node and Link Mapping

YU Hong-fang and GAO Xiu-jiao

(School of Communication Information and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract The problem of efficiently mapping a virtual network (VN) over a common physical network is very important for the network virtualization. This paper proposes a VN mapping algorithm based on the relaxed mixed integer linear program (MILP) with close coordination between the node and link mapping. The integer constraints in the MILP formulation of VN mapping problem are relaxed and new linear constraints are added to obtain a strict LP relaxation model, called STRICT_LP. And then a heuristic algorithm is devised to progressively solve the STRICT_LP to get the whole mapping solution. The simulation results show that the proposed algorithm can realize a better coordination between node and link mapping stages and the performances are better than D-ViNE and R-ViNE in term of the acceptance ratio, revenue, cost, and the ratio of revenue to cost.

Key words mixed integer linear program; network virtualization; relaxation; virtual network mapping

目前, 因特网已成为全球商业运行、多媒体业务、信息交换等的重要基础设施。因特网普遍化和集中化的趋势要求网络具有灵活性, 能够进行网络结构的变化, 但是因特网现有网络构架和协议难以满足这一要求。另外, 因特网是由不同的运营商共同经营的, 不论是建立新的网络架构还是在原有结构上进行大的改变, 都必须在相互竞争的运营商之间达成共识。显然, 因特网架构的改变是非常有限的。网络虚拟化是打破因特网这一瓶颈的有效方式^[1]。各种各样的网络结构、实验和服务能够以虚拟网络的形式共享物理网络资源^[2-4]。由于多个虚拟网络共享同一个物理网络, 如何有效利用物理资源变得至关重要, 这需要将虚拟网络节点和边映射到物理网络中的有效的虚拟网络映射(virtual network embedding,

VNE)算法。

虚拟网络由虚拟节点集合和连接虚拟节点的虚拟边集合组成。虚拟网络映射算法通过在物理网络中找到最优的虚拟节点和虚拟边的映射, 其中每个虚拟节点对应于不同的物理节点, 虚拟边对应于物理拓扑上的一条路径。目前主要有两类虚拟网络映射算法: 一类是基于启发式的算法^[5-8], 该类算法没有考虑虚拟网络节点映射和边映射之间的紧耦合关系, 将两个映射阶段独立进行。在运用贪婪算法选择节点映射后, 再利用最短路(shortest path)、多商品流(multi-commodity flow)以及K最短路(K-shortest paths)等算法进行虚拟网络的边映射。第二类算法是考虑虚拟节点映射和虚拟边映射之间的紧耦合关系, 联合优化这两项映射。文献[9]提出了一种同时

收稿日期: 2010-12-24; 修改日期: 2012-06-28

项目基金: 国家自然科学基金(60872032)

作者简介: 虞红芳 (1975-), 女, 博士, 副教授, 主要从事下一代网络方面的研究。

进行虚拟节点映射和边映射的分布式协同映射算法, 但是该算法的稳定性和性能差于集中式算法。文献[10-11]通过将虚拟网络映射问题描述成混合整数规划模型MILP, 为降低求解的复杂度, 松弛MILP模型的整数变量, 提出了基于确定性的和随机性的四舍五入技术的D-ViNE和R-ViNE两种算法。虽然松弛MILP使得问题在多项式时间内可解, 但是松弛削弱了节点和边映射的协调性, 导致算法性能较差。

本文提出基于MILP松弛的渐进贪婪式算法用于解决虚拟网络映射问题。该算法能够协同进行节点和边映射, 在网络收益, 网络开销和虚拟网络映射成功率等方面有很好的表现。算法基本思想是先将虚拟网络映射问题描述为MILP模型。由于直接求解MILP是NP-hard问题, 将MILP的整数约束条件松弛得到能在多项式时间内求解的线性模型。松弛后的模型允许一个虚拟节点映射到多个物理节点, 破坏了虚拟节点和物理节点的一一映射关系。因此, 添加线性约束条件得到加强的松弛模型STRICT_LP以保证节点和边联合优化映射。同时, 将所有未映射虚拟节点按所需资源从大到小排列, 渐进地进行虚拟节点映射, 即每解一次加强的松弛模型完成一个虚拟节点的映射。当所有虚拟节点映射完毕, 运用多商品流模型将虚拟边映射到物理网络中。

1 网络模型和问题描述

在虚拟网络和物理网络的网络模型基础上, 给出虚拟网络映射问题的MILP模型。

1.1 虚拟网络请求

一个虚拟网络请求包含一组分布在不同位置的具有计算资源(如CPU、内存等)需求的虚拟节点。虚拟节点之间通信以进行中间结果、数据文件等信息的传递和交换。因此, 虚拟节点之间建立有拓扑、带宽和时延等限制的连接以实现网络服务等级协议。

虚拟网络请求可用带有权重的无向图 $G_v=(N_v, L_v)$ 表示, 其中, N_v 是虚拟节点的集合, L_v 是虚拟边的集合。定义 $c(n_v)$ 表示每一个虚拟节点 $n_v \in N_v$ 的CPU资源需求; $loc(n_v)$ 表示地理位置; $LL(n_v)$ 表示虚拟节点到物理节点之间的物理距离或时延限制; $b(l_v)$ 表示虚拟节点 i 和 j 之间的虚拟边 $l_v(i, j) \in L_v$ 的带宽资源请求。

分别由3个和2个虚拟节点组成的虚拟网络请求如图1a所示。图中虚拟边上的数字代表边的带宽资源请求, 矩形框里面的数字代表虚拟节点的CPU资源请求。

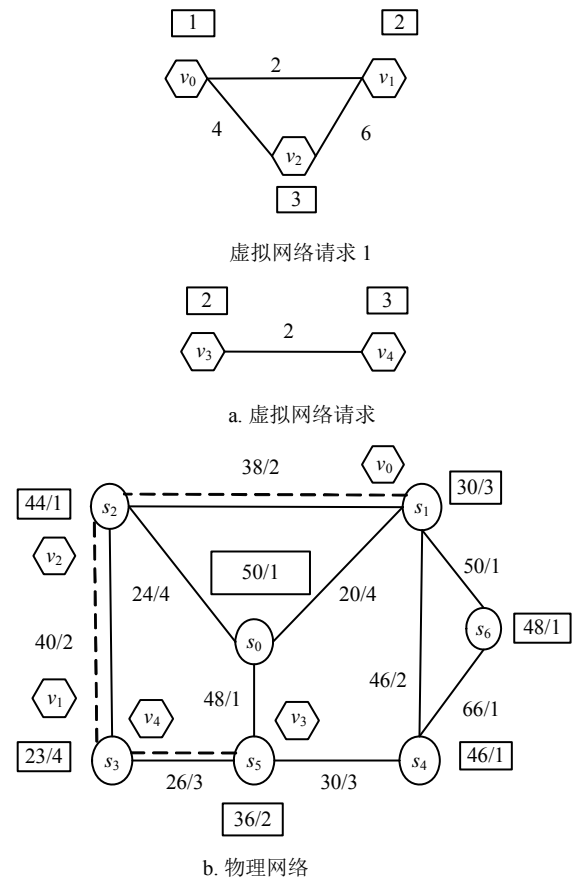


图1 虚拟网络请求和物理网络

1.2 物理网络

物理网络可用带有权重的无向图 $G_s=(N_s, L_s)$ 表示, 其中, N_s 是物理节点的集合, L_s 是物理链路的集合。每一个物理节点的地理位置、CPU资源总量以及单位CPU资源的开销分别用 $loc(n_s)$ 、 $c(n_s)$ 和 $uc(n_s)$ 表示。物理节点 i 和 j 之间的物理链路 $l_s(i, j) \in L_s$ 的可用带宽资源以及单位带宽资源的开销分别用 $b(l_s)$ 和 $uc(l_s)$ 表示。 P_s 表示所有物理路径的集合; $P_s(s, t)$ 表示源节点 s 到目的节点 t 之间的路径。

物理节点 $n_s \in N_s$ 剩余CPU资源定义为节点的可用资源:

$$R(n_s) = c(n_s) - \sum_{\forall n_v \uparrow n_s} c(n_v)$$

其中, $\forall n_v \uparrow n_s$ 表示虚拟节点 n_v 映射到物理节点 n_s , 因此, 物理节点的可用资源是总的资源减去映射到该物理节点的虚拟节点所请求资源之后的结果。

同理, 物理链路 $l_s \in L_s$ 可用资源定义为:

$$R(l_s) = b(l_s) - \sum_{\forall l_v \uparrow l_s} b(l_v)$$

其中, $\forall l_v \uparrow l_s$ 表示虚拟边 l_v 映射的物理路径经过物理链 l_s 。

一条物理路径 $P \in \mathcal{P}_s$ 的可用带宽资源定义为:

$$R(P) = \min_{l_s \in P} R(l_s)$$

物理网络图如1b所示,链路上以符号“/”作为分隔的数字分别代表链路剩余可用带宽和单位带宽开销;矩形框中以符号“/”作为分隔的数字分别代表物理节点可用CPU资源和单位CPU资源开销。

1.3 虚拟网络映射

给定一个物理网络拓扑 G_s 和一个虚拟网络请求,将该虚拟网络映射到物理拓扑中,同时实现对1.4节中描述的4个特性(虚拟网络请求接收比例、总代价,总收益和收益代价比)的优化。虚拟网络映射可以分成两部分:

1) 节点映射。不同虚拟网络的虚拟节点可以映射到同一个物理节点,但是同一虚拟网络中不同虚拟节点必须映射到不同的物理节点,其虚拟节点映射过程 $M_N: N_v \rightarrow N_s$ 对任意 $n_v, m_v \in N_v$ 满足:

$$M_N(n_v) \in N_s$$

$$M_N(n_v) = M_N(m_v) \quad m_v = n_v$$

2) 约束条件。

$$c(n_v) \leq R(M_N(n_v)) \quad (1a)$$

$$\text{dis}(\text{loc}(n_v), \text{loc}(M_N(n_v))) \leq LL(n_v) \quad (1b)$$

式中, $\text{dis}(\text{loc}(n_v), \text{loc}(M_N(n_v)))$ 表示物理节点 $M_N(n_v)$ 和虚拟节点 n_v 之间的距离。

图1b给出了两个虚拟网络在物理网络中的映射结果,分别是 $\{v_0 \rightarrow s_1, v_1 \rightarrow s_3, v_2 \rightarrow s_2\}$ 和 $\{v_3 \rightarrow s_5, v_4 \rightarrow s_3\}$ 。

对边映射而言,有两种方法:一种是不可分流的虚拟边映射,即每条虚拟边只能映射到一条物理路径上;另一种是可分流的虚拟边映射,每条虚拟边能映射多条物理路径上。本文采用第二种可分流的虚拟边映射方法。

虚拟边映射 $M_L: L_v \rightarrow L_s$ 过程,对任意 $l_v = (m_v, n_v) \in L_v$ 满足:

$$M_L(m_v, n_v) \subseteq \mathcal{P}_s(M_N(m_v), M_N(n_v))$$

同时,所有物理路径上的流量之和不小于虚拟边的带宽请求:

$$\sum_P R(P) \geq b(l_v) \quad \forall P \in M_L(l_v) \quad (2)$$

图1b展示了图1a两个虚拟网络到物理网络的边映射的结果,分别是 $\{(v_0, v_1) \rightarrow (s_1, s_2, s_3), (v_0, v_2) \rightarrow (s_1, s_2), (v_1, v_2) \rightarrow (s_3, s_2)\}$ 和 $\{(v_3, v_4) \rightarrow (s_5, s_3)\}$ 。

1.4 优化目标

本文研究动态虚拟网络请求下的映射算法,目的是在增加网络收益和业务接收比例的同时,降低

网络开销,充分利用物理网络资源。

与文献[10]类似,虚拟网络请求的收益定义为:

$$R_{\text{venue}}(G_v) = \sum_{l_v \in L_v} b(l_v) + \sum_{n_v \in N_v} c(n_v) \quad (3)$$

同样,虚拟网络请求的开销定义为:

$$C_{\text{ost}}(G_v) = \sum_{l_v \in L_v} \sum_{l_s \in L_s} f_{l_s}^{l_v} + \sum_{n_v \in N_v} c(n_v) \quad (4)$$

式中, $f_{l_s}^{l_v}$ 表示虚拟边 l_v 映射到物理链路 l_s 上的总流量。文中将式(4)的变形作为MILP的目标函数式。

1.5 MILP描述

参照文献[10]的方法,用扩展边 $\{(n_v, n_s) \mid n_v \in N_v, n_s \in S(n_v)\} \in L_{vs}$ 将虚拟节点 $n_v \in N_v$ 与相应的可选物理节点集合中的点 $S(n_v) \subset N_s$ 建立连接,得到扩展拓扑 $G_s = (N_s, L_s)$,其中 $N_s = N_s \cup N_v$, $L_s = L_s \cup L_{vs}$ 。每一个虚拟节点 $n_v \in N_v$ 对应的可选映射物理节点集合 $S(n_v)$ 是根据其物理位置 $\text{loc}(n_v)$ 以及距离限制 $LL(n_v)$ 信息得到的:

$$S(n_v) = \{n_s \in N_s \mid \text{dis}(\text{loc}(n_v), \text{loc}(n_s)) \leq LL(n_v)\}$$

扩展拓扑后,可将虚拟网络映射问题描述成一个混合整数 $|L_v|$ 个商品流问题。每一条虚拟边 $l_v^i (1 \leq i \leq |L_v|)$ 可看成以 s_i 为源节点,以 t_i 为目的节点的商品流。每一个虚拟节点只能选择一个物理节点用扩展边进行连接,同时一个物理节点最多与同一个虚拟网络中的一个虚拟节点建立映射关系。所有虚拟链路将以流分配的方式在物理网络内部进行映射。VNE问题的MILP模型VNE-MILP描述如下:

1) 变量。

f_{mn}^i : 变量。表示第 i 条虚拟边经过物理链路 (m, n) 的流量。

x_{mn} : 布尔变量。若链路 (m, n) 上有流量则置为1;否则为0。

2) 目标函数:

$$\min = \sum_{mn \in L_s} \sum_i f_{mn}^i \text{uc}(l_{mn}) / 2 + \sum_{n \in N_s} \sum_{m \in N_v} x_{mn} c(m) \text{uc}(n) \quad (5)$$

VNE-MILP的目标函数是最小化虚拟网络的映射开销。虚拟网络和物理网络均采用无向图进行表示,当一条物理链路双向上有相同的流量才认为此条链路被占用,因此目标函数中关于链路开销的部分除以2。

3) 约束条件。

① 容量约束:

$$\sum_i (f_{mn}^i + f_{nm}^i) \leq R(m, n) x_{mn} \quad \forall m, n \in N_s \quad (6)$$

$$R(n) \geq x_{mn} c(m) \quad \forall m \in N_v, \forall n \in N_s \quad (7)$$

式(6)为链路容量约束, 式(7)为节点容量约束。链路容量约束保证物理链路任一方向的流量总和在可用链路带宽容量之内。节点容量约束保证映射节点被占用容量在可用容量范围之内。

② 流守恒约束:

$$\sum_{n \in N_s} f_{mn}^i - \sum_{n \in N_s} f_{nm}^i = 0 \quad \forall i, \forall m \in N_{s'} \setminus \{s_i, t_i\} \quad (8)$$

$$\sum_{n \in N_{s'}} f_{s_i n}^i - \sum_{n \in N_{s'}} f_{n s_i}^i = b(l_v^i) \quad \forall i \quad (9)$$

$$\sum_{n \in N_{s'}} f_{t_i n}^i - \sum_{n \in N_{s'}} f_{n t_i}^i = -b(l_v^i) \quad \forall i \quad (10)$$

对任意虚拟边请求而言, 中间节点的流入流出流量相等, 差值为0。源节点 s_i 的出流量与入流量之差等于业务的大小, 即 $b(l_v^i)$, 目的节点 t_i 的流入流量与流出流量之差等于业务的大小 $b(l_v^i)$ 。

③ 扩展及布尔约束:

$$\sum_{n \in \Omega(m)} x_{mn} = 1 \quad \forall m \in N_v \quad (11)$$

$$\sum_{m \in N_{s'} \setminus N_s} x_{mn} \leq 1 \quad \forall n \in N_s \quad (12)$$

$$x_{mn} \leq R(m, n) \quad \forall m, n \in N_{s'} \quad (13)$$

$$x_{mn} = x_{nm} \quad \forall m, n \in N_{s'} \quad (14)$$

约束式(11)保证每个虚拟节点有且仅有一个物理节点与之对应, 约束式(12)保证同一个虚拟网络中不同虚拟节点必须映射到不同的物理节点。当物理链路 (m, n) 上有流量时, 布尔变量 x_{mn} 则设为1。

④ 变量域约束:

$$f_{mn}^i \geq 0, \forall m, n \in N_{s'} \quad (15)$$

$$x_{mn} \in \{0, 1\} \quad \forall m, n \in N_{s'} \quad (16)$$

该约束是对VNE-MILP模型中变量范围的约束, 其中, f_{mn}^i 是非负变量, x_{mn} 满足整数约束条件式(17), 导致VNE-MILP成为NP-hard问题:

$$x_{mn} \in \{0, 1\} \quad \forall m, n \in N_{s'} \quad (17)$$

2 虚拟网络映射算法

由于VNE-MILP是NP-hard问题, 在松弛该模型的基础上提出渐进贪婪算法, 算法核心思想主要有以下3点:

1) STRICT_LP模型。松弛VNE-MILP的整数约束条件使其在多项式时间内可解, 但是松弛破坏了虚拟节点和物理节点的一一映射关系。为了进一步实现这种一一映射关系, 在扩展边上添加约束条件的STRICT_LP模型的目标函数仍如式(5)所示, 除了约束条件式(17)被松弛, VNE-MILP的其他约束条件仍然起作用。松弛的约束条件和新增的约束条件为:

$$1 \geq x_{mn} \geq 0 \quad \forall m, n \in N_{s'} \quad (18)$$

$$f_{s_i n}^i = b(l_{s_i t_i}^i) x_{s_i n} \quad \forall i, \forall s_i, t_i \in N_{s'}; \forall n \in N_s \quad (19)$$

$$f_{m t_i}^i = b(l_{s_i t_i}^i) x_{m t_i} \quad \forall i, s_i, t_i \in N_{s'}; \forall m \in N_s \quad (20)$$

$$\sum_n f_{mn}^i = b(l_{s_i t_i}^i) x_{s_i m} \quad \forall i, \forall s_i, t_i \in N_{s'}; \forall m, n \in N_s \quad (21)$$

$$\sum_m f_{mn}^i = b(l_{s_i t_i}^i) x_{m t_i} \quad \forall i, s_i, t_i \in N_{s'}; \forall m, n \in N_s \quad (22)$$

约束条件式(18)通过松弛约束条件式(17)得到。约束条件式(19)和式(20)通过在扩展边上增加约束避免同一个虚拟节点映射到不同的物理节点, 实现包含该虚拟节点的不同虚拟边的通信, 约束条件式(21)和式(22)确保虚拟边选择的路由经过了物理网络。

2) 渐进式映射虚拟节点。每解一次STRICT_LP, 只实现一个虚拟节点的映射。经过多次求解STRICT_LP, 渐进实现所有虚拟节点的映射。

3) 运用贪婪的思想逐个确定虚拟节点的映射。首先, 在所有未映射的虚拟节点中, 优先映射CPU资源请求和带宽资源请求(与此虚拟节点连接的所有虚拟边的带宽请求资源和)总和最大的虚拟节点; 其次, 对每一个虚拟节点 $n \in N_v$, 选择具有最大概率的物理节点 $m \in N_s$ 进行映射。选择所依据的概率是变量 x_{mn} 与扩展边 (m, n) 双向流量总和的乘积, 由此使得那些拥有较大 f 和 x 变量值的物理节点被选择作为次虚拟节点的映射节点。

当一个虚拟网络请求到达时根据1.5节描述的方法生成扩展图, 然后初始化 $\phi(n_s)=0, n_s \in N_s$, 代表所有物理节点都未被使用。每求解一次STRICT_LP, 就可确定一个虚拟节点 $n_v \in N_v$ 到某个物理节点 n_s 的映射, 并令 $\phi(n_s)=1$, 以确保每个物理节点最多被映射一次。将所有虚拟节点按照CPU资源和带宽请求资源总和降序排列, 然后按此顺序进行虚拟节点的映射。进行每一个虚拟节点 $n_v \in N_v$ 的映射都要求解 STRICT_LP, 然后查看对应的 $S(n_v)$ 中是否有满足条件的未映射物理节点, 若不存在这样的物理节点, 拒绝当前虚拟网络请求, 返回步骤1) 以等待处理下一个虚拟网络请求; 若存在满足条件的未映射物理节点, 则对每一个 $n_s \in S(n_v)$ 计算概率值。算法将选择具有最大概率值的物理节点 $n_{s'} \in S(n_v)$ 作为映射节点, 同时令 $x_{m, n_{s'}}=1$ 和 $\phi(n_{s'})=1$, 表示虚拟节点 n_v 映射到物理节点 $n_{s'}$ 。

当虚拟网络中所有虚拟节点映射完毕, 渐进贪婪算法运用多商品流算法完成虚拟边的映射。最后, 算法更新网络剩余可用资源, 为下一个虚拟网络请

求的到达做准备。当虚拟网络请求的生命期结束时，算法释放相应虚拟网络请求所占用的网络资源。

由于STRICT_LP是线性规划模型，多商品流算法在多项式时间内可解，因此渐进贪婪算法的时间复杂度在可接受的范围，也是多项式时间内可解的。同时，相比于只求解一次多商品流模型的D-ViNE和R-ViNE算法，渐进贪婪算法由于要求解多次基于STRICT_LP的多商品流模型，时间复杂度更高。求解STRICT_LP的时间与未映射的虚拟网络节点数目密切相关，随着未映射虚拟节点数目的减少而降低，而算法中每求解一次STRICT_LP未映射的虚拟节点数目减少一个，故在整个求解过程中，每次求解STRICT_LP的时间逐渐减小。因此，渐进贪婪算法的复杂度是可接受的。

3 仿真结果与分析

下面给出渐进贪婪算法、D-ViNE 和 R-ViNE 算法的仿真结果与分析。

3.1 仿真环境

随机产生20个节点的物理网络拓扑，每对节点连接的概率是0.5，每个节点可用的CPU资源是在20~25之间均匀分布的整数，每条物理链路可用带宽资源是在20~30之间均匀分布的整数。CPU资源和带宽资源的单位开销均是1~3之间均匀分布的整数，物理节点的位置信息用1~10之间的整数表示，随机生成。

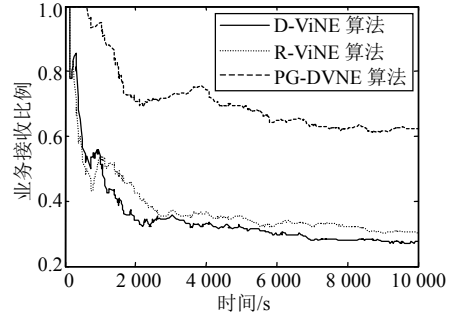
虚拟网络请求以每一百秒平均4个请求的泊松过程到达，每个虚拟网络请求的生存期服从均值为1000 s的指数分布。每一个虚拟网络请求的虚拟节点数目服从2~10之间整数随机分布，每对节点之间的连接概率是0.5。虚拟节点的CPU资源请求和虚拟边的带宽请求都服从1~20之间的整数均匀分布。每一个虚拟节点 $n_v \in N_v$ ，其位置信息服从 $1 \sim |N_v|$ 的整数均匀分布，物理节点到虚拟节点 $n_v \in N_v$ 之间的距离限制 $LL(n_v)$ 服从1~10之间的整数均匀分布。对任何虚拟节点 $n_v \in N_v$ ，根据其物理节点位置信息以及虚拟节点到物理节点的距离限制信息可以得到虚拟节点 n_v 的可选映射物理节点的集合 $S(n_v)$ 。

本文采用ILOG CPLEX 10.0^[12]求解STRICT_LP。

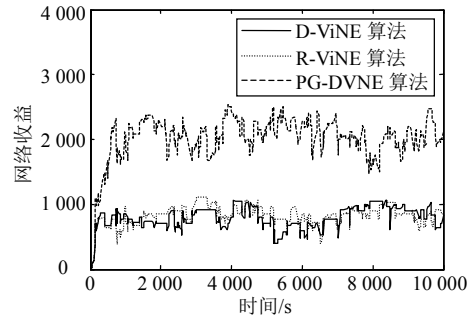
3.2 仿真结果与分析

本文从虚拟请求业务接受比例、网络收益、网络开销以及网络收益和网络开销之比4个方面对渐进贪婪算法、D-ViNE和R-ViNE算法进行了性能对

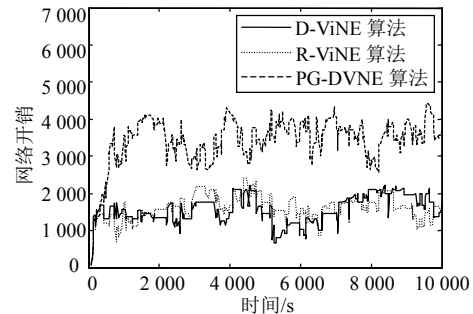
比。当一个虚拟网络请求映射成功后，在虚拟网络生命期内接受该映射业务的开销和收益将分别累加到网络开销和网络收益中；当虚拟网络生命期结束，将其所产生的开销和收益从网络开销和收益中减掉。仿真结果表明渐进贪婪算法，具有以下优势。



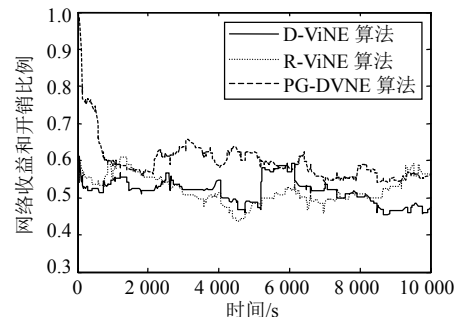
a. 业务接收比例比较



b. 网络收益比较



c. 网络开销比较



d. 网络收益和开销比例比较

图2 3种算法的仿真结果对比

1) 虚拟网络业务请求接受率高和网络收益大：图2a和图2b显示渐进贪婪算法比D-ViNE和R-ViNE

算法有较高的业务接受率和网络收益, 在同样的物理网络资源基础上能够映射更多的虚拟网络请求, 产生更大的网络收益。渐进贪婪算法能够映射收益较大的虚拟网络请求, 而非只映射小的虚拟网络请求仅仅提高业务接收比例。

2) 网络开销大, 收益和开销比例大。图2c显示渐进贪婪算法的网络开销较大, 但是大的网络开销是由于渐进贪婪算法所接纳的虚拟网络请求数多。图2d显示本文算法拥有较高的网络收益和网络开销比例, 结合较高的业务接受比例这一性能, 充分说明渐进贪婪算法能够更充分地利用物理网络资源, 映射同样的单个虚拟网络到同一物理网络上时所需开销较小。

3) 时间复杂度稍高。尽管渐进贪婪算法的时间复杂度高于D-ViNE和R-ViNE算法, 如表1所示(表中给出了3种算法的求解时间), 但是渐进贪婪算法仍然是多项式时间内可解的, 其高收益低开销的性能优势充分弥补且超过了时间复杂度上的弱势。

表1 D-ViNE、R-ViNE和渐进贪婪算法的运行时间对比表

| $ N_v $ | 运行时间 | | |
|---------|--------|--------|---------|
| | D-ViNE | R-ViNE | PG-VNE |
| 3 | 0.141 | 0.157 | 0.360 |
| 4 | 0.156 | 0.187 | 0.515 |
| 6 | 1.641 | 1.750 | 6.359 |
| 8 | 2.985 | 3.063 | 16.547 |
| 10 | 6.281 | 6.360 | 41.703 |
| 15 | 74.678 | 75.468 | 537.625 |

4 结 论

不同虚拟网络共享同一物理网络资源的网络虚拟化过程使得因特网更加灵活和多样化。为有效利用物理网络资源, 需要高效的虚拟网络映射算法。本文基于松弛的线性模型提出的渐进贪婪算法解决虚拟网络映射问题。在虚拟网络映射过程中, 渐进贪婪算法实现了虚拟节点和虚拟边的协调映射。通过对比, 本文算法在虚拟请求业务接收比例、网络收益、网络开销以及网络收益和网络开销之比4个方面的性能优于D-ViNE和R-ViNE算法。虽然渐进贪婪算法的时间复杂度稍高, 但是仍然是多项式时间内可求解的, 并且其高收益低开销的良好性能使得时间上的牺牲是值得的。

参 考 文 献

- [1] ANDERSON T, PETERSON L, SBENKER S, et al. Overcoming the internet impasse through virtualization[J]. IEEE Computer Magazine, 2005, 38(4): 34-41.
- [2] TURNER J, TAYLOR D. Diversifying the internet[C]// Global Telecommunications Conference. St. Louis, MO: IEEE, 2005, 756-760.
- [3] CHOWDHURY N, BOUTABA R. Network virtualization: state of the art and research challenges (topics in network and service management)[J]. IEEE Communications Magazine, 2009, 47(7): 20-26.
- [4] CHOWDHURY N M K, ZAHEER F, BOUTABA R. iMark: An identity management framework for network virtualization environment[C]//IFIP /IEEE International Symposium on Integrated Network Management (IM). Long Island, NY: IEEE, 2009: 335-342.
- [5] ZHU Y, AMMAR M. Algorithms for assigning substrate network resources to virtual network components[C]// INFOCOM 2006, 25th IEEE International Conference on Computer Communications Proceedings. Barcelona, Spain: IEEE, 2006: 1-12.
- [6] HONG Y, CHUNMING Q. Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures[C]// IEEE Globecom. Miami, USA: IEEE, 2010: 236-241.
- [7] LU J, TURNER J. Efficient mapping of virtual networks onto a shared substrate[R]. USA: Washington University, 2006.
- [8] 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.
- [9] HOUIDI I, LOUATI W, ZEGHLACHE D. A distributed virtual network mapping algorithm[C]// IEEE International Conference on Communications. Beijing: IEEE, 2008: 5634-5640.
- [10] CHOWDHURY N M M K, RAHMAN M R, BOUTABA R. Virtual network embedding with coordinated node and link mapping[C]// IEEE INFOCOM. Riode Janeiro: IEEE, 2009: 783-791.
- [11] MUNTASIR R, ISSAM A, RAOUF B. Survivable virtual network embedding[J]. Lecture Notes in Computer Science, 2010(4): 40-52.
- [12] IBM. ILOG CPLEX 10.0[EB/OL] (2010-11-12). <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>.
- [13] ERGIN F, YAYIMLI A, UYAR S. An evolutionary algorithm for survivable virtual topology mapping in optical WDM networks[C]//EvoWorkshops09. Berlin Heidelberg: Springer-Verlag, 2009: 31-40.