

基于最小代价的跨域虚拟网络映射算法^{*}

彭利民

(华南理工大学 自动化科学与工程学院, 广东 广州 510640)

摘要: 针对多个自治域网络环境中的虚拟网络映射问题,提出了基于最小代价的跨域虚拟网络映射(MC-VNE)算法. 首先根据虚拟网络的约束条件,计算每个虚拟节点的可用物理节点集合,然后利用最小权重路由算法,计算出每条虚拟链路的可用映射物理路径集合. 借鉴克鲁斯卡尔最小生成树算法思想,依次在可用映射物理路径集合选择最小权重物理路径,然后将对应的虚拟链路映射到该物理路径上,并协调完成虚拟节点的映射操作. 仿真结果表明,MC-VNE算法有效地降低了虚拟网络映射的资源代价,提高了虚拟网络请求接受率.

关键词: 网络管理; 虚拟网络映射; 自治域

中图分类号: TP393

doi: 10.3969/j.issn.1000-565X.2015.09.011

网络虚拟化技术被视为构建新一代 Internet 体系架构的重要技术. 利用网络虚拟化技术,基础设施提供商(ISP)可在同一个底层物理网络(SN)上创建多个虚拟网络(VN),从而为用户提供多样化、定制的网络服务^[1]. 虚拟网络映射是指将一个具有位置、资源等约束条件的虚拟网络映射到底层物理网络上,其中虚拟节点映射到物理节点上,虚拟链路映射到物理链路或物理路径上^[2-3]. 目前,大部分虚拟网络映射方面的研究主要针对单个基础设施(简称为自治域)网络环境^[4],但随着在线游戏和交互式网络电视(IPTV)等服务的快速发展,单个自治域内提供的虚拟网络很难满足用户的应用需求,用户往往希望将虚拟网络映射到由多个自治域构成的网络环境中,如在不同地理位置的自治域中部署服务器. 因此非常有必要使用跨域虚拟网络映射机制,根据虚拟网络的约束条件,以最小代价映射虚拟网络请求.

近几年已提出的跨域虚拟网络映射算法可分为两类:分布式^[5-8]和集中式^[9-14]. 如 Marquezan 等^[5]针对跨域网络环境中的资源管理问题,提出了分布式资源管理算法; Samuel 等^[6]首先采用位置感知的虚拟网络请求分发机制,将虚拟网络请求分发给各个自治域,各个自治域根据域内的网络资源状态预映射虚拟网络请求,并将不能映射的虚拟网络片段以递归的方式转发给其他的自治域,然后所有的自治域采用分布式竞价方式协调完成整个虚拟网络映射操作; Houidi 等^[7]采用分布式多代理技术,根据自治域内的网络资源状态和最短路径距离参数,协调完成虚拟网络映射操作; Mano 等^[8]将虚拟网络请求分发给所有自治域,各个自治域根据自身的网络资源状态分布式预映射虚拟网络请求片段,然后虚拟网络供应商(VNP)根据各个自治域返回的可映射片段以及相应的代价,使用多方计算(MPC)理论实现虚拟网络映射操作. 虽然分布式跨域虚拟网络映

收稿日期: 2015-04-09

^{*} 基金项目: 国家自然科学基金资助项目(61103037); 广东省自然科学基金资助项目(S2012040007599)

Foundation items: Supported by the National Natural Science Foundation of China(61103037) and the Natural Science Foundation of Guangdong Province(S2012040007599)

作者简介: 彭利民(1976-),男,博士后,副教授,主要从事网络虚拟化、分布式计算研究. E-mail: penglm86@126.com

射算法可有效地避免节点瓶颈问题,但虚拟网络映射过程中自治域之间的频繁通信增加了通信开销,同时虚拟网络映射的动态性以及各个自治域之间交互信息的时效性等不能保证虚拟网络映射方案是最优的问题求解.近几年,学者们提出了一些集中式跨域虚拟网络映射算法,如 Dietrich 等^[9]将各个自治域中的物理节点分为边界节点和内部节点,其中边界节点负责完成自治域间连接,内部节点负责映射虚拟节点,并利用整数规划方法将虚拟网络拓扑划分为多个虚拟网络片段,然后在自治域中分别映射虚拟网络片段; Zhang 等^[10]提出了一种分层线性规划模型,将虚拟网络拓扑划分为多个虚拟网络片段,然后在自治域内分别映射相应的虚拟网络片段; Di 等^[11]根据虚拟网络的服务质量(QoS)需求,首先通过 VNP 将虚拟网络映射请求分发给各个自治域,各个自治域根据域内的网络资源状态将可以在自治域中映射的节点、链路以及相应的代价返回给 VNP,然后 VNP 根据收集到的映射信息以及自治域间的物理链路资源状态,以最小代价求解跨域虚拟网络映射问题; Shen 等^[12]基于最小代价的虚拟网络映射优化目标,对虚拟网络请求拓扑结构进行划分,然后将划分后的虚拟网络片段分发给相应的自治域,并在各个自治域中实施虚拟网络片段映射操作; 肖蔼玲等^[13]为了简化跨域虚拟网络映射问题,仅考虑自治域之间的物理链路代价(忽略自治域内的物理链路代价),利用遗传算法将虚拟网络拓扑划分为多个虚拟网络片段. Houidi 等^[14]根据每个虚拟网络节点的位置、资源等约束条件,在各个自治域中查找满足每个虚拟节点约束的物理节点集合,然后根据自治域的资源信息将虚拟网络请求分割为多个虚拟网络片段,并分别在自治域中完成虚拟网络片段的映射操作.

综上所述,现有的跨域虚拟网络映射算法主要采用正向思维操作方式,将虚拟网络拓扑结构划分为多个虚拟网络片段,然后在自治域中分别映射相应的虚拟网络片段.由于虚拟网络拓扑结构的最优划分以及自治域中的最优映射,未必是最优的虚拟网络映射方案,特别是当自治域内的物理节点被分为边界节点和内部映射节点时,增加了虚拟网络映射操作的复杂性.为此,文中采用反向思维操作方式,利用最小权重路由算法依次在底层物理网络中动态地选择最小代价的物理路径,然后将对应的虚拟链路映射到最小代价的物理路径上,并协调映射相应的虚拟节点,直至完成所有的虚拟节点和虚拟链路的映射操作,最后通过理论分析和实验验

证文中提出的基于最小代价的跨域虚拟网络映射(MC-VNE)算法的网络性能.

1 网络模型与优化目标

VNP 是跨域虚拟网络映射的核心角色,其主要任务是根据虚拟网络映射请求的约束条件以及所有自治域中的网络资源状态,将虚拟网络映射到一个或多个自治域构成的底层物理网络上.

1.1 跨域虚拟网络映射模型

文中采用两层资源管理模式:①每个自治域指派一个控制节点 k ,它负责将自治域内的信息发送给 VNP,这些信息主要包含自治域内节点及链路的性能参数和非性能参数,如节点处理器类型、节点地理位置、链路类型、节点可用资源量、物理链路带宽资源量和传输时延等,节点 k 同时负责接收 VNP 返回的虚拟网络映射方案,并负责自治域中的虚拟网络片段映射操作;②VNP 负责接收虚拟网络用户的虚拟网络映射请求,并根据所有自治域内的资源信息以及自治域间的物理链路带宽资源量,将虚拟网络映射到由多个自治域构成的底层物理网络上,以及将具体的虚拟网络映射方案发送给相应的控制节点 k .

文中使用无向权重图 $G^S = (N^S, L^S, AN^S, AL^S)$ 表示底层物理网络,其中 N^S 为物理节点集合; L^S 为物理链路集合; AN^S 为物理节点的属性矢量,如 $loc(n^S)$ 和 $cpu(n^S)$ 分别为节点 n^S 的地理位置和 CPU 资源量; AL^S 为物理链路的属性矢量,如 $bw(l^S)$ 为节点 n^S 和 m^S 之间物理链路 $l^S(n^S, m^S) \in L^S$ 上的带宽资源量.底层物理网络由 D 个自治域构成,它们之间通过主干物理链路连接.如图 1 所示,底层物理网络由 lnP_1 和 lnP_2 两个自治域构成, lnP_1 和 lnP_2 之间通过主干物理链路 (B, E) 进行连接.类似地,文中使用无向权重图 $G_i^S = (N_i^S, L_i^S, AN_i^S, AL_i^S)$ 表示第 i 个自治域,其中 N_i^S 和 L_i^S 分别为第 i 个自治域中的物理节点集合和物理链路集合.因此,底层物理网络 $G^S = G_1^S \cup G_2^S \cup \dots \cup G_D^S \cup L_{ij}^S$, 其中 $L_{ij}^S (1 \leq i, j \leq D)$ 为自治域 i 和 j 之间的物理链路.文中使用无向权重图 $G^V = (N^V, L^V, AN^V, AL^V)$ 表示虚拟网络请求,其中 N^V 是虚拟节点集合, L^V 是虚拟链路集合, AN^V 是虚拟节点的需求属性, AL^V 是虚拟链路的需求属性.图 1(a) 是一个虚拟网络请求,节点 a 的 CPU 资源需求量为 9 个单位,虚拟链路 $l(a, b)$ 的带宽资源需求量为 12 个单位.

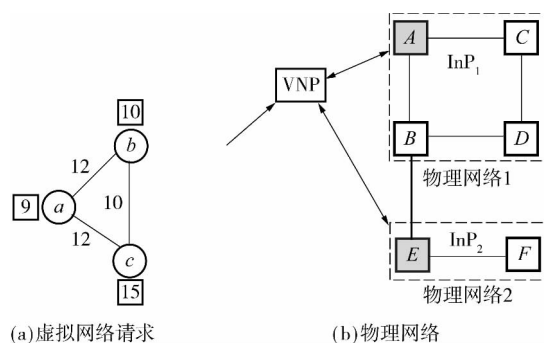


图1 跨域虚拟网络映射模型

Fig.1 Virtual network mapping model in multi-domains

1.2 优化目标

跨域虚拟网络映射是指将虚拟网络请求映射到 G^S 中的自治域上, 它可分解为两个节点映射和链路映射两个操作过程. 由于虚拟节点映射到任何物理节点上消耗的节点资源总是相同的, 因此任何虚拟网络映射算法消耗的节点资源代价总是相等的; 但是, 当虚拟链路映射到不同位置、不同长度的物理路径上时, 其消耗的链路资源代价可能相差较大, 因此不同的虚拟网络映射算法消耗的链路资源代价可能不同.

设图1所示自治域中的链路带宽权重为1, 自治域 InP_1 和 InP_2 之间的主干链路带宽权重为10, 当图1中虚拟节点 a, b 和 c 分别映射到物理节点 A, B 和 C 上, 以及虚拟链路 (a, b) 、 (a, c) 和 (b, c) 分别映射到物理路径 (A, B) 、 (A, C) 和 (B, D, C) 上时, 虚拟网络映射中的节点资源代价为34个单位、链路资源代价为44个单位; 当虚拟节点 a, b 和 c 分别映射到物理节点 B, E 和 F 上, 以及虚拟链路 (a, b) 、 (a, c) 和 (b, c) 分别映射到物理路径 (B, E) 、 (B, E, F) 和 (E, F) 时, 此时虚拟网络映射中的节点资源代价为34个单位、链路资源代价为242个单位. 因此不同的虚拟网络映射算法消耗的资源代价相差较大.

跨域虚拟网络映射算法的主要目标是在满足虚拟网络的约束条件下降低虚拟网络映射的资源代价. 在时刻 t , VNP 接受一个虚拟网络请求的网络收益表示为虚拟网络中的节点 CPU 需求量和链路带宽需求量总和, 它可定义为

$$r(G^V, t) = \sum_{n_v \in N^V} \text{cpu}(n_v) + \sum_{l_v \in L^V} \text{bw}(l_v) \quad (1)$$

类似地, VNP 映射一个虚拟网络请求的代价是指在底层物理网络中为该虚拟网络请求实际分配的资源量总和, 它可定义为

$$c(G^V, t) = \sum_{n_v \in N^V} \text{cpu}(n_v) + \sum_{l(i, j) \in L^V} w_{m, n} \sum_{m \in N^S} f_{m, n}^{i, j} \text{bw}(i, j) \quad (2)$$

式中: $w_{m, n}$ 为物理链路权重, 当物理节点 m 和 n 属于同一个自治域时, $w_{m, n} = \alpha$, 当物理节点 m 和 n 属于不同的自治域时, $w_{m, n} = \beta$. 即

$$w_{m, n} = \begin{cases} \alpha, & m, n \in \text{InP}_h; 1 \leq h \leq D \\ \beta, & m \in \text{InP}_h, n \in \text{InP}_g, g \neq h; 1 \leq h, g \leq D \end{cases};$$

当虚拟链路 $l^V(i, j)$ 映射到物理链路 $l^S(m, n)$ 上且分配所需的链路带宽资源时, $f_{m, n}^{i, j} = 1$, 否则 $f_{m, n}^{i, j} = 0$; $\text{bw}(i, j)$ 为虚拟链路 $l^V(i, j)$ 的带宽需求量. 因此, 在时间 T 内 VNP 的映射代价与网络收益之比可以定义为

$$\frac{c}{r} = \frac{\sum_{t=0}^T c(G^V, t)}{\sum_{t=0}^T r(G^V, t)} \quad (3)$$

跨域虚拟网络映射算法的主要目标是降低虚拟网络映射的资源代价, 因此跨域虚拟网络映射的优化目标可定义为

$$\min \frac{c}{r} \quad (4)$$

当 VNP 接收到虚拟网络请求后, 该虚拟网络的网络收益随之确定, 因此 c/r 越小, VNP 映射虚拟网络请求的资源代价越小, 底层物理网络上可映射的虚拟网络个数越多, VNP 的网络收益越大.

2 跨域虚拟网络映射算法

在跨域网络环境下, 由于自治域内的物理链路带宽代价远小于自治域间的主干物理链路带宽代价, 因此文中假定自治域内的物理链路带宽权重为1, 自治域间的物理链路带宽权重为10.

2.1 跨域虚拟网络映射实例

文中使用 $a.\text{match}[]$ 表示满足虚拟节点 a 约束条件的物理节点集合. 在如图1所示的虚拟网络映射示例中, 假设满足虚拟节点 a, b 和 c 约束条件的物理节点分别是 $a.\text{match}[] = \{B, C\}$ 、 $b.\text{match}[] = \{E\}$ 和 $c.\text{match}[] = \{F\}$. 当虚拟节点的候选物理节点集合确定后, 可用于映射虚拟链路的候选物理路径集合也可随之确定. 在如图1所示的虚拟网络映射示例中, 可用于映射虚拟链路 (a, b) 、 (a, c) 和 (b, c) 的物理路径集合是 $\{(B, \dots, E), (C, \dots, E)\}$ 、 $\{(B, \dots, F), (C, \dots, F)\}$ 和 $\{(E, \dots, F)\}$. 因此, 应用最小权重路由算法可计算出底层物理网络上候选物理节点间的

最小权重路径. 为了增强虚拟网络映射的容错性, 文中约定一个物理节点最多映射同一个虚拟网络请求中的一个虚拟节点, 因此自身物理节点间的路径权重设为无穷大. 如图1所示的虚拟网络映射示例中, 每个虚拟节点的候选物理节点集合以及候选物理路径的最小权重如下:

	B	C	E	F
B	∞	∞	10	11
C	∞	∞	12	13
E	10	12	∞	1
F	11	13	1	∞

2.2 跨域虚拟网络映射算法描述

文中提出的跨域虚拟网络映射算法主要根据底层物理网络的资源状态, 基于最小代价的虚拟网络映射机制, 在多自治域网络环境下实施跨域虚拟网络映射操作. MC-VNE 算法的核心思想可归纳为:

①根据虚拟节点的资源、位置约束等计算每个虚拟节点的可用物理节点集合; ②根据跨域网络环境中的网络资源状态, 采用最小权重路由算法计算底层物理网络中候选物理节点间的最小权重路径; ③借鉴克鲁斯卡尔最小生成树算法思想, 依次在最小权重路径集合中动态地选择最小权重物理路径, 并协调完成节点和链路的映射操作, 直到完成所有虚拟节点和虚拟链路的映射操作. 基于最小代价的跨域虚拟网络映射算法的代码如下:

Procedure of MC-VNE algorithm

for each virtual node vn

vn. MappedState() \leftarrow False

vn. match[] \leftarrow { sn \in N_s, sn meets vn's constraint }

end for

for each substrate node sn

sn. MappedState() \leftarrow False

end for

String minWeightPath() {

for each physical node s_i in vn. match[], List

for each physical node s_j in vn. match[], List

compute minWeightPath(s_i, s_j)

return path

end for

end for }

while |MappedLinkList| < |L_v| do

path = minWeightPath()

m = path.getOneVertex()

n = path.getOtherVertex()

vnList1[] = m.correspondingVirtualNode()

vnList2[] = n.correspondingVirtualNode()

if all link(u, v) have been mapped then

// $u \in$ vnList1[], $v \in$ vnList2[]

path.weight() \leftarrow ∞

else

if (m.UsedState and n.UsedState are false) then

choose one link(u, v) from the above virtual links'

set, which of the end point belongs to the set

vnList1[] and vnList2[] respectively, and

have not been mapped.

map link(u, v) \rightarrow path(m, n)

add link(u, v) in the MappedLinkList

if (u.MappedState is false) then

map $u \rightarrow m$

u.MappedState = True

m.UsedState = True

end if

if (v.MappedState is false) then

map $v \rightarrow n$

v.MappedState = True

n.UsedState = True

end if

path.weight() \leftarrow ∞

update network resource information

end if

end while

end Procedure

MC-VNE 算法首先根据虚拟节点的资源、位置等约束条件, 分别求出每个虚拟节点 vn 的可用物理节点集合 vn.match[], 并根据网络资源状态, 利用最小权重路由算法计算所有 {vn.match[]} 节点间的最小权重路径集合 PathList, 然后借鉴克鲁斯卡尔最小生成树算法思想, 在当前 PathList 中动态地选择权重最小的路径 path, 如果路径 path(m, n) 的两个端点对应的虚拟节点 u 和 v , 以及 u 和 v 之间的虚拟链路 link(u, v) 尚未映射, 其中 $u \in$ vnList1[], $v \in$ vnList2[], vnList1[] 和 vnList2[] 分别为物理节点 m 和 n 对应的虚拟节点集合, 则将虚拟链路 link(u, v) 映射到物理路径 path(m, n) 上, 并将虚拟节点 u, v

分别映射到 m, n 上, 从而协调完成节点和链路的映射操作; 由于每次映射操作后底层物理网络资源会发生变化, 因此需要重新计算候选物理节点间的最小权重路径, 然后按同样的操作方法映射其他的虚拟节点和虚拟链路, 直到所有的虚拟节点和虚拟链路完成映射为止. MC-VNE 算法的时间复杂度为 $O(|L_v| |N_s|^2)$, 其中 $|L_v|$ 为虚拟网络链路数, $|N_s|$ 为物理网络节点数.

定理 1 MC-VNE 算法是最小代价的跨域虚拟网络映射算法.

证明 当虚拟网络中只有两个节点 u 和 v 、一条虚拟链路 $\text{link}(u, v)$ 时, 将虚拟链路 $\text{link}(u, v)$ 映射到路径集合 PathList 中的最小权重路径 $\text{path}(m, n)$ 上, 虚拟节点 u 和 v 分别映射到物理节点 m 和 n , 跨域虚拟网络映射算法即结束. 由于 $\text{path}(m, n)$ 的权重是当前路径集合 PathList 中的最小权重, 故 MC-VNE 算法一定是最小代价的跨域虚拟网络映射算法.

当虚拟网络中有多个虚拟节点和多条虚拟链路时, 则按如下两种情况进行讨论: ①物理节点 m 和 n 之间可用物理路径是否唯一; ②物理节点 m, n 对应的 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中虚拟节点是否唯一.

(1) 物理节点 m 和 n 之间只有一条可用物理路径 $\text{path}(m, n)$, 且 m 和 n 对应的 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中虚拟节点唯一, 令 $u = \text{vnList1}[]$, $v = \text{vnList2}[]$. 由于物理节点 m 和 n 之间只有一条可用物理路径, 故每次求得最小权重路径 $\text{path}(m, n)$ 后, 只需要将虚拟链路 $\text{link}(u, v)$ 映射到路径 $\text{path}(m, n)$ 上, 如果虚拟节点 u, v 尚未映射, 则将 u, v 分别映射到 m 和 n . 按照同样的步骤映射其他的节点和链路, 直至所有的虚拟节点和虚拟链路映射完毕为止. 由于每次映射的 $\text{path}(m, n)$ 是当前网络状态下的最小权重路径, 因此 MC-VNE 算法一定是最小代价的跨域虚拟网络映射算法.

(2) 物理节点 m 和 n 之间可用物理路径不唯一, 但 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中虚拟节点唯一, 令 $u = \text{vnList1}[]$, $v = \text{vnList2}[]$. 当第 1、第 2、第 3 次的最小权重路径都是 $\text{path}(m, n)$ 时, 由于 MC-VNE 算法将虚拟链路 $\text{link}(u, v)$ 映射到第 1 次的 $\text{path}(m, n)$ 路径后, 虚拟链路 $\text{link}(u, v)$ 已经映射, 其状态 MappedState 值为 True , 即使第 2、第 3 次等的最小权重路径仍然是 $\text{path}(m, n)$, MC-VNE 算法通过将 $\text{path}(m, n)$ 权值设置为无穷大后, 就可选择其他的

路径进行映射操作. 由于每次使用的路径 $\text{path}(m, n)$ 是当前网络状态下的最小权重, 故 MC-VNE 算法一定是最小代价的跨域虚拟网络映射算法.

(3) 物理节点 m 和 n 之间的可用物理路径唯一, 但 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中虚拟节点不唯一. 由于 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中可能对应多个虚拟节点, 意味着存在多条虚拟链路 vlinkList , 则 MC-VNE 算法随机地在 vlinkList 中选择一条尚未映射的虚拟链路 $\text{link}(u, v)$ 映射到当前最小权重路径 $\text{path}(m, n)$ 上. 因此 MC-VNE 算法是最小代价的跨域虚拟网络映射算法.

(4) 物理节点 m 和 n 之间的可用物理路径不唯一, 且 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中虚拟节点也不唯一. 由于 $\text{vnList1}[]$ 和 $\text{vnList2}[]$ 中可能对应多个虚拟节点, 意味着存在多条虚拟链路 vlinkList , 则 MC-VNE 算法随机地在 vlinkList 中选择一条尚未映射的虚拟链路 $\text{link}(u, v)$ 映射到当前的最小权重路径 $\text{path}(m, n)$ 上; 如果 vlinkList 中的虚拟链路都已完成映射, 即使物理节点 m 和 n 之间还存在最小权重路径, MC-VNE 算法通过将 $\text{path}(m, n)$ 权值设置为无穷大, 就可以选择其他的可用物理路径进行映射操作. 由于每次映射的物理路径 $\text{path}(m, n)$ 是当前网络状态下的最小权重路径, 因此 MC-VNE 算法一定是最小代价的跨域虚拟网络映射算法.

综上所述, MC-VNE 算法是最小代价的跨域虚拟网络映射算法. 证毕.

3 实验结果与分析

为了测试 MC-VNE 算法的网络性能, 文中通过设计一个离散事件仿真系统, 从映射代价与网络收益之比、虚拟网络请求接受率和网络资源利用率 3 个方面对 MC-VNE 算法进行测试. QoS-VNM 算法^[11]是与文中提出的 MC-VNE 算法最接近的集中式跨域虚拟网络映射算法, 因此, 文中对 QoS-VNM 算法和 MC-VNE 算法的网络性能进行了比较测试.

3.1 仿真环境

仿真实验在 100×100 的网格中使用 GT-ITM 工具^[15]随机生成物理网络拓扑结构, 它包含 9 个自治域, 每个自治域中物理节点的地理位置相同, 并使用自治域在网格中的编号对物理节点的地理位置进行标识; 每个自治域内平均包含 10 个物理节点和 50 条域内链路, 每个物理节点的 CPU 资源量和物理链路的带宽资源量在 100 ~ 150 内均匀分布, 9 个自治域之间使用 50 条主干物理链路互相连接, 主干物

理链路的带宽资源量在 300 ~ 600 内均匀分布; 在相同的网络上使用 GT-ITM 工具^[15] 随机生成虚拟网络请求拓扑结构, 每个虚拟网络请求中虚拟节点数在 5 ~ 12 内均匀分布, 虚拟节点间以 50% 的概率随机相互连接, 虚拟节点的 CPU 资源量和虚拟链路的带宽资源量在 0 ~ 20 内均匀分布, 虚拟节点的地理位置序号服从 1 至 9 的均匀分布; 实验中虚拟网络请求到达模拟泊松过程, $100t'$ (t' 为时间单位) 内虚拟网络请求到达个数服从均值为 10 的泊松分布, 每个虚拟网络的生存周期服从均值为 $1000t'$ 的指数分布, 每次模拟时间为 $50000t'$, 式(2)中的参数 α 和 β 分别设置为 1 和 10; 每次模拟实验运行 $50000t'$.

3.2 实验结果

当虚拟网络请求确定后, 虚拟网络请求产生的网络收益随之确定, 因此较小的映射代价与网络收益之比意味着较小的资源代价, 如图 2(a) 所示, MC-VNM 算法比 QoS-VNM 算法取得了较小的映射代价与网络收益之比. 其主要原因是 MC-VNM 算法通过使用最小权重路由算法, 为每条虚拟链路查找最小资源代价的映射物理路径, 从而有效地降低了虚拟网络映射的链路带宽资源代价. 图 2(a) 说明 MC-VNM 算法可以有效地降低虚拟网络映射的资源代价.

虚拟网络请求接受率表示成功映射的虚拟网络请求个数与所有虚拟网络请求个数之比, 虚拟网络请求接受率越高, VNP 的网络收益越大. 如图 2(b) 所示, MC-VNM 算法比 QoS-VNM 算法取得了较高的虚拟网络请求接受率, 特别是在模拟阶段后期, 两种算法的虚拟网络请求接受率差距越来越大. 其主要原因是 MC-VNM 算法根据底层物理网络资源状态, 动态地为每条虚拟链路在底层物理网络中查找最小代价的映射物理路径, 减少了虚拟链路的带宽资源, 为后续的虚拟网络请求留出更多的网络资源, 从而提高了虚拟网络请求接受率. 图 2(b) 进一步说明 MC-VNM 算法可以有效地降低跨域虚拟网络映射的资源代价.

节点和链路资源的平均利用率表示整个仿真时间内所有节点(链路)的平均资源利用率. 如图 2(c) 所示, MC-VNM 算法比 QoS-VNM 算法有效地提高了底层物理网络中节点的资源利用率. 其主要原因是 MC-VNM 算法的虚拟网络请求接受率较高, 虚拟网络消耗的节点资源量也较多, 因此节点资源利用率也较高. 尽管图 2(b) 中 MC-VNM 算法的虚拟网络请求接受率较高, 但图 2(d) 中 MC-VNM 算法和

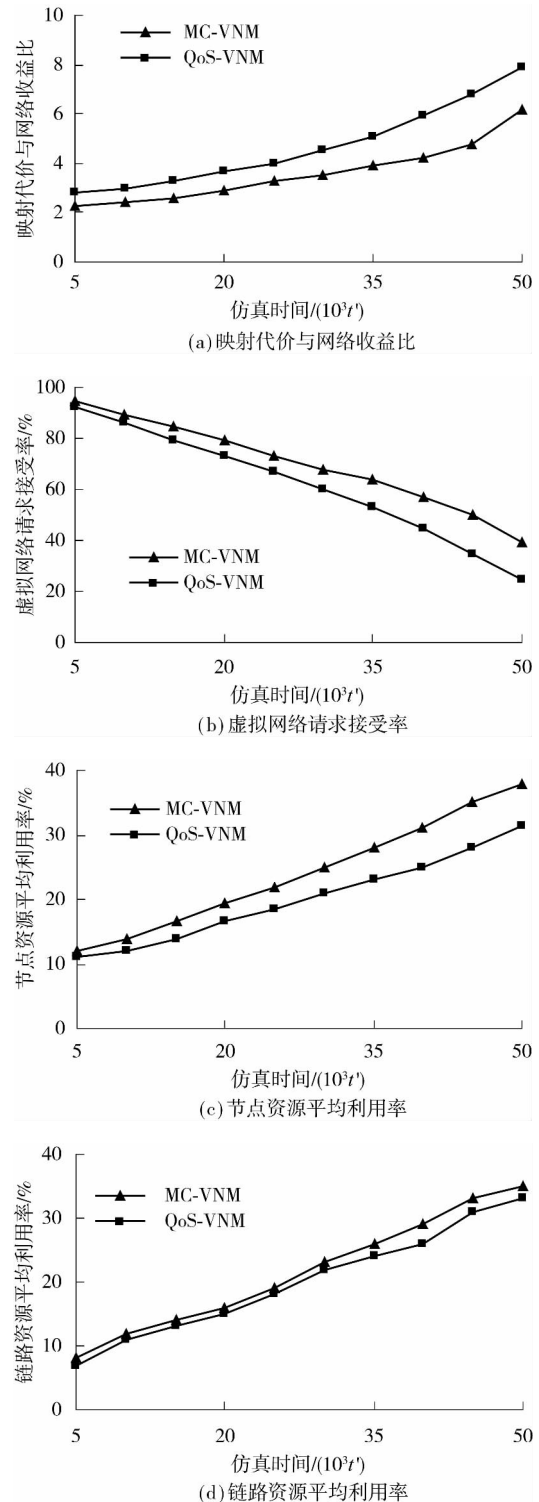


图2 两种算法的仿真实验结果比较

Fig.2 Comparison of simulation results between two algorithms

QoS-VNM 算法的链路资源利用率比较接近, 其主要原因是 MC-VNM 算法使用最小权重路由算法, 为虚拟链路查找最小代价的物理路径进行映射, 有效地降低了虚拟链路的映射路径长度, 因此虚拟链路消耗的链路带宽资源量也随之降低, 图 2(d) 更进一步

地验证了 MC-VNM 算法可以有效地降低虚拟网络映射的资源代价。

4 结论

通过分析现有跨域虚拟网络映射算法存在的不足,文中提出了一种跨域网络环境中的分层资源管理模式,用于动态地管理各个自治域内的网络资源;在充分考虑跨域网络环境、物理链路带宽存在差异化的前提下,为自治域内的链路带宽和自治域间的主干链路带宽设置不同的链路权重,然后根据所有自治域中的物理网络资源状态,基于最小代价的虚拟网络映射优化目标,实施跨域虚拟网络映射操作。理论分析和实验结果表明,文中提出的 MC-VNE 算法有效地降低了虚拟网络映射的资源代价,具有较好的网络性能。

参考文献:

- [1] 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.
Cai Zhi-ping, Liu Qiang, Lü Pin, et al. Virtual network mapping model and optimization algorithms [J]. Journal of Software 2012 23(4): 864-877.
- [2] 彭利民. 基于图的邻接分割的虚拟网络映射算法 [J]. 华南理工大学学报: 自然科学版 2015 43(1): 66-71.
Peng Li-min. Virtual network mapping algorithm based on graph adjacency segmentation [J]. Journal of South China University of Technology: Natural Science Edition 2015, 43(1): 66-71.
- [3] 罗娟,陈磊,李仁发. 一种启发式网络虚拟化资源分配算法 [J]. 中国科学: 信息科学 2012 42(8): 960-973.
Luo Juan, Chen Lei, Li Ren-fa. A heuristic resource allocation algorithm for virtual network embedding [J]. Science China: Information Science 2012 42(8): 960-973.
- [4] 彭利民. 基于广度优先搜索的虚拟网络映射算法 [J]. 四川大学学报: 工程科学版 2015 47(2): 117-122.
Peng Li-min. Virtual network embedding algorithm based on breadth-first search [J]. Journal of Sichuan University: Engineering Science Edition 2015 47(2): 117-122.
- [5] Marquezan C C, Granville L Z, Nunzi G, et al. Distributed autonomic resource management for network virtualization [C]//Proceedings of IEEE Network Operations and Management Symposium. Osaka: IEEE 2010: 463-470.
- [6] Samuel F, Chowdury M, Boutaba R. PolyViNE: policy-based virtual network embedding across multiple domains [J]. Journal of Internet Services and Applications 2013 6(4): 1-23.
- [7] Houidi I, Louati W, Zeghlache D. A distributed virtual network mapping algorithm [C]//Proceedings of IEEE International Conference on Communications. Beijing: IEEE, 2008: 5634-5640.
- [8] Mano Toru, Inoue Takeru, Ikarashi Dai, et al. Efficient virtual network optimization across multiple domains without revealing private information [C]//Proceedings of the 23rd International Conference on Computer Communication and Networks. Shanghai: IEEE 2014: 1-8.
- [9] Dietrich D, Rizk A, Papadimitriou P. Multi-domain virtual network embedding with limited information disclosure [C]//Proceedings of the 2013 IFIP Networking Conference. Brooklyn: IEEE 2013: 1-9.
- [10] Zhang M, Wu C, Wang B, et al. Research on mapping method of logical carrying network across multiple domains [J]. Journal on Communications 2012 33(8): 200-207.
- [11] Di Hao, Anand Vishal, Yu Hongfang, et al. Quality of service aware virtual network mapping across multiple domains [C]//Proceedings of 2013 IEEE Globecom Workshops. Atlanta: IEEE 2013: 476-481.
- [12] Shen Meng, Xu Ke, Yang Kun, et al. Towards efficient virtual network embedding across multiple network domains [C]//Proceedings of the 22nd International Symposium of Quality of Service. Hong Kong: IEEE 2014: 61-70.
- [13] 肖蔼玲,王颖,孟洛明,等. 基于知识描述和遗传算法的跨域虚拟网络映射 [J]. 软件学报 2014 25(10): 2189-2205.
Xiao Ai-ling, Wang Ying, Meng Luo-ming, et al. Knowledge description and genetic algorithm based multi-domain virtual network embedding [J]. Journal of Software 2014 25(10): 2189-2205.
- [14] Houidi I, Louati W, Ameer W B. Virtual network provisioning across multiple substrate networks [J]. Computer Networks 2011 55(5): 1011-1023.
- [15] Zegura E W, Calvert K L, Bhattacharjee S. How to model an inter-network [C]//Proceedings of the 15th Annual Joint Conference of the IEEE Computer and Communications Societies Conference on Computer Communications. San Francisco: IEEE 1996: 594-602.

(下转第 112 页)

Analysis of Critical Load and Width of Plate-Shaped Buckling under Tension

Dai Jie-tao¹ Li Lie-jun¹ Dai Wen-li² Zhang Zu-jiang²

(1. School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, Guangdong, China; 2. Shaoguan Iron and Steel Co. Ltd., Baosteel Group, Shaoguan 512123, Guangdong, China)

Abstract: Based on the actual plate and strip buckling situation in the site and by analyzing the analytic model of plate and strip buckling, which used to be calculated as a whole, this paper takes the compressive stress region as the plate-shaped buckling deformation region to reflect the actual plate shape situation in the site, and then establishes an analytic calculation model of the outer tension affecting the plate-shaped buckling deformation. Thus, the changes of the critical buckling load and critical width of plate and strip are obtained under different outer tensions, that is, with the increase of tension, the critical buckling load increases while the critical width decreases, which accords well with the actual situation in the site. In order to further verify the analytic calculation result, a finite element calculation model of the plate and strip buckling is established under outer tensions, and the simulation result is consistent with the analytic calculation one.

Key words: plate and strip; tension; buckling; finite element

(上接第73页)

A Multi-Domain Virtual Network Embedding Algorithm Based on Minimum Cost

Peng Li-min

(School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, Guangdong, China)

Abstract: Aiming at the problem of the virtual network embedding in the multi-domain network environment, a multi-domain virtual network embedding algorithm (MC-VNE) based on the minimum cost is proposed. First, a feasible substrate node set is calculated for embedding each virtual node based on the constraints of the virtual network. Then, a feasible substrate path set is calculated for embedding each virtual link by using the minimum weight routing algorithm. Finally, on the basis of the Kruskal minimum spanning tree algorithm, the substrate path of the lowest weight is selected from the feasible substrate path set in turn, and the corresponding virtual link is then embedded into the selected substrate path in a proper order. Meanwhile, the corresponding virtual nodes are embedded in a harmonious way. Simulation results show that the MC-VNM algorithm reduces the resource cost of virtual network embedding effectively, and improves the acceptance ratio of virtual network requests.

Key words: network management; virtual network embedding; autonomous domain