

一种拓扑感知的虚拟网络重构算法

彭利民

(华南农业大学 数学与信息学院,广东 广州 510642)

摘要:针对底层物理网络负载不均衡问题,根据节点和链路的负载状态,动态地选择负载过大的物理节点和物理链路进行重构。通过建立拓扑感知的虚拟网络重构模型,将虚拟节点迁移到距离其邻居映射节点最近的物理节点上,优化底层物理网络中虚拟重构节点的映射位置,降低虚拟网络重构的资源开销。模拟结果表明,提出的 TA-VNR 算法有效地均衡了底层物理网络的负载水平,降低了虚拟链路的重构路径长度,提高了虚拟网络请求接受率。

关键词:底层物理网络;虚拟网络;重构;资源利用率;拓扑感知

中图分类号:TP393

文献标志码:A

A Topology-awareness Virtual Network Reconfiguration Algorithm

PENG Limin

(College of Mathematics and Info., South China Agricultural Univ., Guangzhou 510642; China)

Abstract: Aiming at load imbalance problem in the substrate network, the substrate nodes and links with resource utilization exceed the predefined threshold value were selected for reconfiguration based on the load state of the substrate nodes and links. By using topology-awareness reconfiguration model, the virtual node to be migrated was then mapped onto the nearest mapped substrate node, which is the neighbor mapping node of the migration virtual node, mapping location of these reconfiguration virtual nodes in the substrate network was optimized, therefore resource consumption of virtual network reconfiguration was reduced efficiently. Simulation results showed that the load level in the substrate is balanced efficiently, the reconfiguration path distance of virtual links are reduced greatly, and the accept ratio of virtual networks is improved by using the proposed TA-VNR algorithm.

Key words: substrate network; virtual network; reconfiguration; resource utilization; topology awareness

网络虚拟化技术支持多个虚拟网络(virtual network, VN)共享底层物理网络(substrate network, SN)资源,为用户提供多样性的网络服务^[1]。基础设施供应商(infrastructure provider, InP)通过采用高效的虚拟网络映射算法,将物理网络资源分配给虚拟网络请求,提高虚拟网络请求接受率和网络收益。然而,当新的虚拟网络请求到达、过期的虚拟网络离开以及物理网络资源出现故障时,底层物理网络中的节点或链路可能会出现负载不均衡问题,有的节点或链路负载可能较低,有的可能较高,甚至成为资源瓶颈。Fajjari 等^[2]通过模拟实验发现:当底层物理网络的平均链路带宽利用率为 18% 时,虚拟网络请求拒绝率达到 99%。因此,为了提高虚拟网络请

求接受率以及 Inp 的网络收益,非常有必要应用虚拟网络重构机制,动态地迁移虚拟节点和虚拟链路,均衡底层物理网络的负载水平。

近几年,学者们提出了一些虚拟网络重构方法。文献[3]针对虚拟网络的资源需求动态性问题,利用虚拟网络请求的历史信息,建立资源预测模型,估计虚拟网络请求的资源需求量,然后定期重新映射整个虚拟网络请求;文献[4]针对底层物理网络中节点和链路负载动态性问题,以最小化节点和链路负载为优化目标,通过模拟退火方法周期性重新映射整个虚拟网络。虽然这些方法可以改善底层物理网络的负载水平,提高 InP 的网络收益,但是,重构整个虚拟网络同时带来了巨大的系统开销。文献

[5]通过实时监测底层物理网络中节点的负载状态,当节点的资源利用率超过阈值时,则将映射到该物理节点上的所有虚拟节点迁移到另一个物理节点上,虽然该方法可以暂时消除资源瓶颈问题,但是,全部迁移它上面的所有虚拟节点增加了迁移代价,同时,也容易引起新的资源瓶颈,出现所谓“跷跷板”问题;文献[6]针对“跷跷板”问题,以最小化重构虚拟节点个数为优化目标,当节点的资源利用率超过阈值时,则将映射到该物理节点上部分虚拟节点迁移到另一个负载较低的物理节点上,虽然该方法降低了迁移的虚拟节点个数,但是,该重构方法没有考虑虚拟节点在底层物理网络中的迁移位置,容易造成虚拟节点稀疏地分布在底层物理网络中,增加了虚拟网络重构的资源开销(如链路带宽);文献[7]针对底层物理网络中节点故障问题,以最小化节点迁移代价为优化目标,将故障物理节点上的虚拟节点迁移到其它物理节点上,该方法同样没考虑虚拟节点在底层物理网络中的迁移位置。近几年,学者们为了降低虚拟网络映射的资源开销,提出了一些拓扑感知的虚拟网络映射算法。文献[8]通过建立拓扑感知的虚拟网络映射模型,将虚拟节点映射到底层物理网络的局部区域内,降低虚拟链路的映射路径长度;文献[9]通过采用图的广度优先搜索遍历方法和节点同步遍历策略,将虚拟网络中相邻的虚拟节点映射到底层物理网络中邻接的物理节点上,优化虚拟网络在底层物理网络中的映射拓扑结构,降低虚拟网络映射的资源开销;但是,这些拓扑感知的虚拟网络映射算法主要是针对虚拟网络映射操作问题,没有考虑底层物理网络中节点和链路负载动态性问题。

综上所述,现有的虚拟网络重构方法在均衡底层物理网络负载时,忽视了虚拟节点在底层物理网络中的迁移位置,容易导致虚拟网络重构后节点稀疏地分布于底层物理网络中,增加虚拟网络重构的资源开销;部分现有的虚拟网络重构方法将整个虚拟网络作为重构对象,增加了虚拟网络重构的性能代价。针对这些问题,根据底层物理网络中节点和链路的负载状态,将资源利用率超过阈值的节点和链路作为重构对象,通过建立拓扑感知的虚拟网络模型,将虚拟节点迁移到距离其邻居映射节点最近的物理节点上,优化虚拟节点在底层物理网络中的映射位置,降低虚拟网络重构的资源开销,并通过采用自适应机制合理地选择迁移的虚拟节点和虚拟链路,减少重构虚拟节点和虚拟链路数量,降低虚拟网

络重构的性能代价。仿真结果表明,提出的 TA-VNR 算法有效地均衡了底层物理网络的负载水平,取得了较好的网络性能。

1 问题描述与网络模型

拓扑感知的虚拟网络重构方法是指在保证虚拟网络资源约束的前提下,根据物理网络负载状态,动态调整虚拟节点和虚拟链路在底层物理网络中的分布位置,均衡底层物理网络的负载水平,提高虚拟网络请求接受率。下面首先描述虚拟网络重构问题,然后提出虚拟网络重构优化模型。

1.1 问题描述

1) 底层物理网络

采用无向带权图 $G_s = (N_s, L_s, A_s^v, A_s^l)$ 为底层物理网络。其中: G_s 为底层物理网络; N_s 为物理节点集合; L_s 为物理链路集合; A_s^v 为物理节点属性,主要是指节点的存储能力和 CPU 处理能力等; A_s^l 为物理链路属性,主要是指物理链路的传输时延、可用带宽等。与文献[1-9]一样,作者只考虑节点的 CPU 资源属性和链路的带宽资源属性。如图 1 中右图所示,节点 A 的 CPU 资源量为 60 单位,物理链路(A, B)上的带宽量为 50 单位。

2) 虚拟网络请求

类似地,使用无向带权图 $G_v = (N_v, L_v, A_v^v, A_v^l, t)$ 表示虚拟网络请求。其中, N_v 为虚拟节点集合, L_v 为虚拟链路集合, A_v^v 为虚拟节点的资源需求属性, A_v^l 为虚拟链路的资源需求属性, t 为虚拟网络请求的生存时间。如图 1 中左图所示,节点 a 的 CPU 资源需求量为 9 个单位,虚拟链路(a, b)的带宽资源需求为 15 个单位。

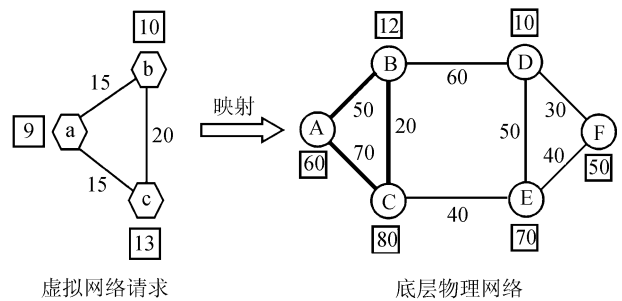


图 1 虚拟网络映射实例

Fig.1 An example of virtual network embedding

3) 虚拟网络映射

虚拟网络映射是指将虚拟网络中的节点映射到底层物理网络中的节点上,以及将虚拟网络中的链路映射到底层物理网络中的物理链路或无环物理路

径上,并满足虚拟节点和虚拟链路的资源约束条件。图 1 中,将节点 a、b 和 c 分别映射到物理节点 A、B 和 C 上,同时将虚拟链路(a,b)、(b,c)和(a,c)映射到物理路径(A,B)、(B,C)和(A,C)上,此时虚拟链路(a,b)、(b,c)和(a,c)的映射路径长度为 1,以节点间的跳步数表示虚拟链路的映射路径长度。

4) 虚拟网络重构

当底层物理网络资源出现故障,或底层物理网络中节点或链路负载过大时,需要将已映射到物理节点(或物理链路)上的虚拟节点(或虚拟链路)重新进行映射。虚拟网络重构包含 3 个基本操作:一是,在物理节点(或物理链路)上选择需要迁移的虚拟节点(或虚拟链路);二是,在目标物理节点(或物理路径)选择需要迁移的虚拟节点(或虚拟链路);三是,将虚拟节点(或虚拟链路)迁移到目标物理节点(或物理链路上),并分配网络资源。如图 2 所示,当节点 B 发生故障或其资源利用率超过阈值时,则需要将已映射到节点 B 上的虚拟节点 b 进行重构,并重构虚拟节点 b 的邻接虚拟链路(b,a)和(b,c),然后重新分配节点和链路资源。

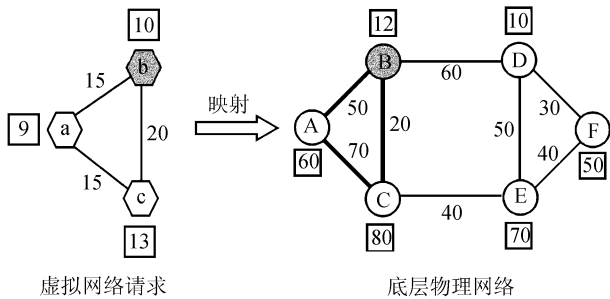


图 2 虚拟网络重构实例

Fig. 2 An example of virtual network reconfiguration

1.2 优化模型

底层物理网络的负载状态主要是指物理节点和物理链路的负载状态,通常使用资源利用率表示节点和链路的负载水平。

1) 资源利用率

节点负载是指映射到该物理节点上的虚拟节点所分配的 CPU 资源量总和,节点的资源利用率表示节点负载与节点 CPU 总资源量的比值,可以反映节点的负载水平。链路负载是指映射到该物理链路上的虚拟链路所分配的带宽资源量总和,链路的资源利用率是指链路负载与链路总带宽的比值。因此节点和链路的资源利用率可定义为:

$$Util(v_s) = \frac{\sum_{v_i \in N_v \rightarrow v_s} cpu(v_i)}{cpu(v_s)} \quad (1)$$

$$Util(l_{mn}) = \frac{\sum_{l_{uv} \in L_v \rightarrow l_{mn}} bw(l_{uv})}{bw(l_{mn})} \quad (2)$$

式(1)中, $v_i \rightarrow v_s$ 表示虚拟节点 v_i 被映射到物理节点 v_s 上。式(2)中, $l_{uv} \rightarrow l_{mn}$ 表示虚拟链路 l_{uv} 被映射到物理链路 l_{mn} 上。

2) 重构资源开销

当底层物理网络中节点或链路的负载过大,或者物理网络资源出现故障时,已映射到这些物理节点或物理链路上的虚拟节点和虚拟链路需要迁移到新的物理节点和物理链路上,并为它们重新分配 CPU 资源和带宽资源。图 2 中,当虚拟节点 b 重新迁移到物理节点 E 上、虚拟链路(a,b)和(b,c)迁移到物理路径(A,B,D,E)和(E,C)上时,虚拟链路(a,b)和(b,c)的重构路径长度分别为 3 和 1。因此,虚拟链路的重构路径长度(rpl)为重构的虚拟链路的映射路径长度;虚拟网络重构的资源开销(rrc)为虚拟网络中重构的虚拟节点和虚拟链路分配的 CPU 资源量和链路带宽量总和。可定义为:

$$rpl(v_i, v_j) = dist(M(v_i), M(v_j)) \quad (3)$$

$$rrc = \sum_{v_i' \in N_v} cpu(v_i') + \sum_{l_{uv}' \in L_v, l_{mn}' \in L_s} bw(l_{uv}', l_{mn}') \quad (4)$$

式(3)中, $M(v_i)$ 和 $M(v_j)$ 为虚拟节点 v_i 和 v_j 的映射物理节点, $dist(M(v_i), M(v_j))$ 为虚拟链路 $link(v_i, v_j)$ 的重构路径长度。式(4)中, $f_{mn}^{l_{uv}}$ 为 1 时,表示虚拟链路 l_{uv} 映射到物理链路 l_{mn} 上且为 l_{mn} 分配所需带宽资源时,否则 $f_{mn}^{l_{uv}}$ 为 0。

3) 重构性能代价

虽然虚拟网络重构可以解决底层物理网络资源动态性问题,但是,虚拟网络重构也需要付出额外的代价,如虚拟网络上提供的网络服务可能暂时中断等。令 $perc$ 表示虚拟网络重构的性能代价,它主要包括迁移虚拟节点和虚拟链路的代价之和,重构性能代价可定义为:

$$perc = \alpha \cdot \sum_{v_i' \in N_v} c(v_i') + \beta \cdot \sum_{l_{uv}' \in L_v} c(l_{uv}') \quad (5)$$

式中, v_i' 和 l_{uv}' 分别为重构虚拟节点和虚拟链路, α 和 β 分别为重构虚拟节点和链路的权重。提出的虚拟网络重构方法的主要目标是降低虚拟网络重构的资源开销,并降低虚拟网络重构对网络服务带来的负面影响。因此虚拟网络重构的优化目标可定义为:

$$\min \mu \cdot rrc + \lambda \cdot perc \quad (6)$$

式中, μ 和 λ 为虚拟网络重构的资源开销因子 rrc 和性能代价因子 $perc$ 的权重。

2 拓扑感知的虚拟网络重构算法

拓扑感知的虚拟网络重构算法主要是指根据底层物理网络资源状况以及节点和链路的负载状态,周期性地对虚拟网络映射方案进行重构,动态调整底层物理网络中的节点和链路资源分配,降低虚拟网络重构的资源开销和性能代价,提高虚拟网络请求接受率和基础设施供应商的网络收益等。

2.1 拓扑感知的虚拟网络重构模型

为了降低虚拟网络重构的资源开销,虚拟节点应迁移到距离其邻居映射节点最近的物理节点上,减少虚拟链路的重构路径长度。因此拓扑感知的虚拟网络重构模型可定义为:

$$TD(v_k, v_s) = \begin{cases} \sum_{v_i \in \text{neig}(v_k)} \text{dist}(v_s, M(v_i)), \\ v_s \in \{v_s \mid \text{cpu}(v_s) > \text{cpu}(v_k), \text{and} \\ \text{bw}(p(v_s, M(v_i))) > \text{bw}(\text{link}(v_i, v_k))\} \end{cases} \quad (7)$$

式中, $\text{neig}(v_k)$ 为虚拟网络中节点 v_k 的邻居节点集合, $M(v_i)$ 为虚拟节点 v_i 在底层物理网络中的映射节点, $\text{dist}(v_s, M(v_i))$ 为底层物理网络中节点 v_s 和 $M(v_i)$ 间的最短路径长度, $\text{cpu}(v_s)$ 为物理节点 v_s 的可用 CPU 资源量, $\text{cpu}(v_k)$ 为待迁移虚拟节点 v_k 的 CPU 需求量, $\text{bw}(p(v_s, M(v_i)))$ 为节点 v_s 到 $M(v_i)$ 最短路径上的可用带宽量, $\text{bw}(\text{link}(v_i, v_k))$ 为链路 $\text{link}(v_i, v_k)$ 的带宽需求量。

通过使用拓扑感知的虚拟网络重构模型,可将虚拟节点 v_k 迁移到其邻居 v_i 的物理映射节点 $M(v_i)$ 的附近节点上,减少虚拟链路的重构路径长度,降低虚拟网络重构的资源开销,其中, $v_i \in \text{neig}(v_k)$, $\text{neig}(v_k)$ 为 v_k 的邻居节点集合, $M(v_i)$ 为 v_i 的物理映射节点。在图 2 中,虚拟节点 b 的邻居节点为 a 和 c,令节点 a 和 c 的物理映射节点分别为 $M(a)$ 和 $M(c)$ (分别对应图 2 中的 A 和 C)。根据虚拟网络重构模型,当物理节点 B 出现故障时,则将虚拟节点 b 迁移到距离节点 A 和 C 最近的物理节点 E 上,虚拟网络重构的带宽资源开销为 65 单位。如果将虚拟节点 b 重构到物理节点 F 上,则虚拟网络重构的带宽资源开销为 100 单位,因此增加了 35 个单位的资源开销。该实例表明了拓扑感知的虚拟网络重构方法可有效地降低虚拟网络重构的资源开销。

2.2 拓扑感知的虚拟网络重构算法

提出的拓扑感知的虚拟网络重构算法 (TA-VNR) 的主要目标是解决底层物理网络资源故障或

节点、链路负载过大问题,同时降低虚拟网络重构的资源开销和性能代价。其核心思想可归纳为:1) 根据拓扑感知的虚拟网络重构模型,将虚拟节点迁移到距离其邻居映射节点最近的物理节点上,优化虚拟节点在底层物理网络中的映射位置,解决虚拟网络重构中的拓扑稀疏问题,降低虚拟网络重构的资源开销;2) 根据重构物理节点的负载状态,通过设置节点和链路的资源利用率阈值,自适应地选择迁移的虚拟节点和虚拟链路,减少迁移虚拟节点和虚拟链路的数量,降低虚拟网络重构的性能代价。

算 法 拓扑感知的虚拟网络重构算法 (TA-VNR)

Procedure of TA-VNR algorithm {

```

1: for each sn in snList do {
2:   snState ← sn.state()
3:   snUtil ← sn.util() }
4: if allnode sn which's snUtil > Thresh then {
5:   return "Reconfiguration Failed" }
6: for each node sn which's snState == Failure {
7:   migrationNodeList.add(sn.getVnList())
8:   vnList.remove(sn.getVnList()) }
9: for each node sn which's snUtil > Thresh {
10:  bestUtil ← MAX
11:  while (snUtil > Thresh) do {
12:    for each node vn in the sn.vnList() do {
13:      if vn.getUtil() > snUtil - Thresh then {
14:        temp ← vn.getUtil - snUtil + Thresh
15:        if temp < bestUtil then {
16:          bestUtil ← temp
17:          migrationVn ← vn } }
18:      else if bestUtil = MAX then
19:        migrationVn ← vn
20:      break } }
21:  migrationNodeList.add(migrationVn)
22:  vnList.remove(sn.migrationVn) }
23: for each node vk in migrationNodeList do {
24:  migrationLinkList.add(vk.getNeigList())
25:  choose maximum TD(vk, vs)
26:  map vk onto vs }
27: for each vlink in migrationLinkList do {
28:  map vlink using the shortest path algorithm }
29: return reconfigurationResult }
```

TA-VNR 算法中 sn 表示物理节点, snList 表示物理节点集合, snUtil 表示节点 sn 的资源利用率, snState 表示节点 sn 的当前状态,如故障或正常等,

sn.vnList()表示sn上已映射的虚拟节点集合, Thresh表示节点、链路的资源利用率阈值, vn表示虚拟节点, vlink表示虚拟链路, migrationNodeList表示迁移的虚拟节点集合, migrationLinkList表示迁移的虚拟链路集合。TA-VNR算法中第6~8行用于将故障物理节点上的虚拟节点全部迁出;第9~22行用于在物理节点sn中自适应地选择迁移的虚拟节点vn;第24行用于将迁移的虚拟节点vk的邻接虚拟链路加入迁移虚拟链路集合migrationLinkList中,第28行采用最短路径算法重新映射虚拟链路。由于链路的重构方法和节点的重构方法基本相似,限于篇幅,该算法中未描述链路的重构操作。TA-VNR算法的时间开销主要是在计算并选择最大的TD(vk,vs)值以及采用最短路径算法重新映射虚拟链路的相关操作,因此,TA-VNR算法的时间复杂度为 $O((|migrNode|+|migrLink|)\cdot|N_s|^2)$,其中,|migrNode|和|migrLink|为迁移的虚拟节点和虚拟链路个数,|N_s|为底层物理网络中节点个数。

3 仿真实验

从资源利用率方差、虚拟链路的重构路径长度和虚拟网络请求接受率3个方面对TA-VNR算法进行性能测试。选择文献[9]中的BFS-VNM算法进行虚拟网络映射操作,并周期性地使用文献[6]中的LC-VNR算法和提出的TA-VNR算法进行虚拟网络重构。用BFS-VNM表示没有采用重构操作的映射过程,用BFS-VNM+LC-VNR表示采用了文献[6]中LC-VNR算法重构后的映射过程,用BFS-VNM+TA-VNR表示采用TA-VNR算法重构后的映射过程。

3.1 仿真环境

仿真实验采用GT-ITM工具^[10]随机生成一个由100个物理节点、560条链路组成的底层物理网络拓扑,物理节点的CPU资源量和物理链路的带宽量服从50到100的均匀分布。虚拟网络请求过程模拟泊松过程,每100个时间单元内虚拟网络请求个数服从均值为20的泊松分布,虚拟网络的生存时间服从均值为500个时间单元的指数分布,虚拟网络请求中节点个数服从3到10的均匀分布,虚拟节点间以0.5的概率随机连接,虚拟节点的CPU资源需求量与虚拟链路的带宽需求量服从5到20的均匀分布。实验中,α和β均设置为0.5,μ和λ分别设置为1和10,重构周期T设置为1000个时间单元,节点和链路的资源利用率阈值均为75%。每次

模拟实验运行50000个时间单元,包含约为10000个虚拟网络请求。共进行了10次仿真实验,实验最终结果为10次仿真结果的平均值。

3.2 实验结果

1)TA-VNR算法均衡了物理网络的负载水平

由图3和4可以看出,使用TA-VNR算法周期性地对虚拟网络进行重构后,物理节点和物理链路的资源利用率方差大大地减少。其主要原因是由于TA-VNR算法通过利用资源利用率阈值,根据底层物理网络负载状态,通过迁移机制动态调整节点和链路的负载水平。实验结果表明,TA-VNR算法有效地均衡了底层物理网络中的负载水平。

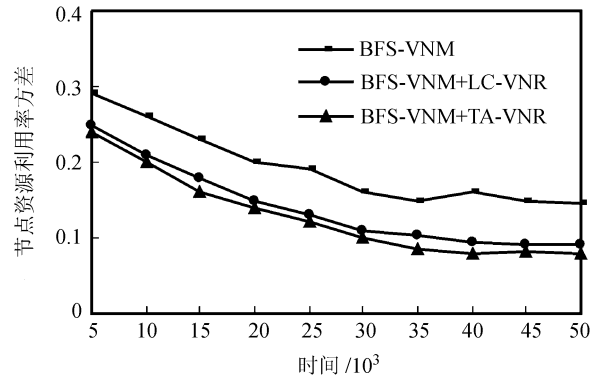


图3 节点资源利用率

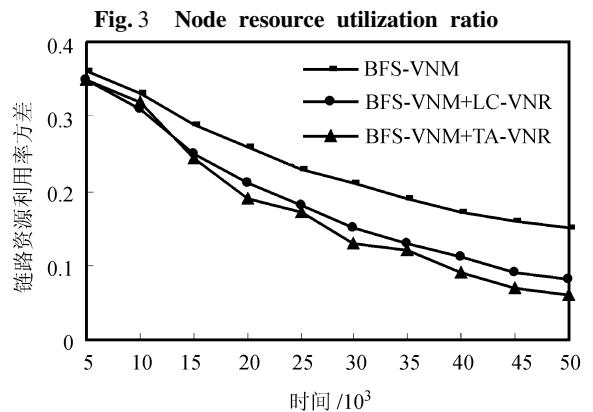


图4 链路资源利用率

Fig.4 Link resource utilization ratio

2)TA-VNR算法降低了虚拟链路的重构路径长度

如图5所示,和LC-VNR算法相比,TA-VNR算法有效地降低了虚拟链路的重构路径长度。从图5可以看出,在整个模拟期间LC-VNR算法的平均重构路径长度呈现不规则变化,表明LC-VNR算法在选择目标物理节点进行迁移时没有考虑网络的拓扑结构,虚拟节点随机地分布在底层物理网络中,而TA-VNR算法根据网络拓扑中节点的邻接关系,将虚拟节点重映射到距离其邻居映射节点最近的物理

节点上,减少了虚拟链路的重构路径长度,降低了虚拟网络重构的资源开销。

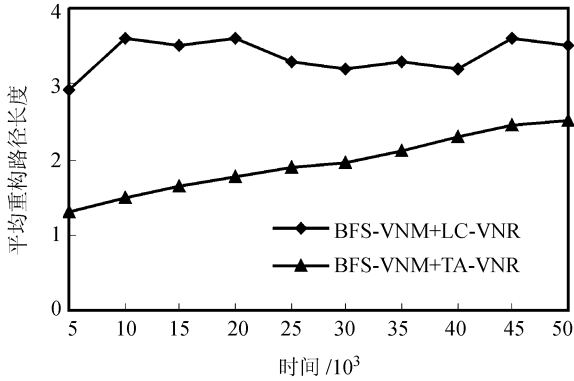


图5 平均重构路径长度

Fig.5 Average reconfiguration path distance

3) TA-VNR 算法提高了虚拟网络请求接受率

如图6所示,LC-VNR算法和TA-VNR算法都提高了虚拟网络请求接受率,特别是在仿真实验中后期阶段链路带宽资源比较稀缺时,TA-VNR算法比LC-VNR算法更有效地提高了虚拟网络请求接受率。其主要原因是TA-VNR算法在重构虚拟节点时,充分地考虑了网络的拓扑结构,优化了底层物理网络中虚拟重构节点的分布位置,减少了虚拟链路的重构路径长度,降低虚拟网络重构的资源开销,因此有效地提高了虚拟网络请求接受率。

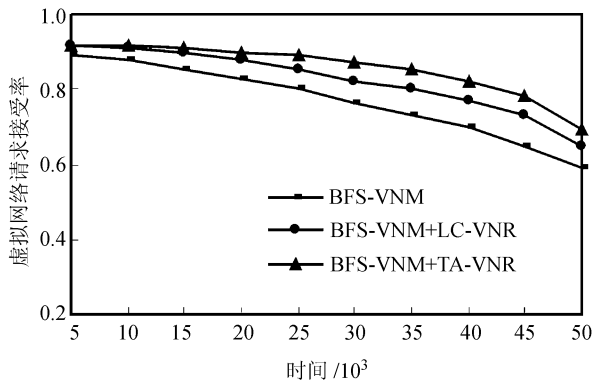


图6 虚拟网络请求接受率

Fig.6 Acceptance ratio of virtual network requests

4 结论

针对底层物理网络中节点和链路负载不均衡问题,通过建立拓扑感知的虚拟网络重构模型,周期性地对虚拟网络映射方案进行重构,将虚拟节点迁移到距离其邻居映射节点最近的物理节点上,动态调整底层物理网络的负载水平,优化虚拟节点在底层物理网络中的映射位置,减少虚拟链路的重构路径长度,降低虚拟网络重构的资源开销。仿真结果表明,TA-VNR算法取得较好的重构性能。

参考文献:

[1] Luo Juan, Chen Lei, Li Renfa. A heuristic resource allocation algorithm for virtual network embedding [J]. Science China: Information Science, 2012, 42(8): 960 - 973. [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.]

[2] Fajjari I, Aitsaadi N, Pujolle G, et al. VNR algorithm: A greedy approach for virtual networks reconfigurations [C]//Proceeding of the IEEE Global Telecommunications Conference. Houston, USA: IEEE, 2011: 1 - 6.

[3] Xu Zichuan, Liang Weifa, Xia Qiufen. Efficient virtual network embedding via exploring periodic resource demands [C]//Proceeding of 39th Annual IEEE Conference on Local Computer Networks. Edmonton, Canada: IEEE, 2014: 90 - 98.

[4] Masi S B, Raghavan S V. Simulated annealing algorithm for virtual network reconfiguration [C]//Proceeding of 2012 8th EURO-NF Conference on Next Generation Internet (NGI). Karlskrona: IEEE, 2012: 95 - 102.

[5] Kansal N J, Chana I. Existing load balancing techniques in cloud computing: A systematic review [J]. Journal of Information Systems and Communication, 2012, 3(1): 87 - 91.

[6] Qu Hua, Zhao Jihong, Guo Shuang, et al. Resource reconfiguration method based on minimum cost for network virtualization [J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37(5): 114 - 118. [曲桦, 赵季红, 郭爽, 等. 基于最小代价的虚拟网络重配置方法 [J]. 北京邮电大学学报, 2014, 37(5): 114 - 118.]

[7] Chang Xiaolin, Muppala J K, Wang Bin, et al. Migration cost aware virtual network re-embedding in presence of resource failures [C]//Proceeding of 2012 18th IEEE International Conference on Networks (ICON). Singapore: IEEE, 2012: 24 - 28.

[8] Li Xiaoling, Wang Huaimin, Guo Changguo, et al. Topology awareness algorithm for virtual network mapping [J]. Journal of Zhejiang University-Science C: Computers & Electronic, 2012, 13(3): 178 - 186.

[9] Peng Limin. Virtual network embedding algorithm based on breadth-first search [J]. Journal of Sichuan University: Engineering Science Edition, 2015, 47(2): 117 - 122. [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.]

[10] Zegura E W, Calvert K L, Bhattacharjee S. How to model an internetwork [C]//Proceeding of the 15th Annual Joint Conference of the IEEE Computer and Communications Societies Conference on Computer Communications. San Francisco: IEEE, 1996: 594 - 602. (编辑 赵婧)