DOI:10.13471/j.cnki.acta.snus.2015.05.002

高效节能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：针对虚拟网络映射中的能耗问题，根据虚拟网络重构特征以及节点和链路的能耗特性，建立虚拟网络重构优化模型。通过设置底层物理网络的资源利用率阈值，周期性地将资源利用率大于高阈值和小于低阈值物理节点和物理链路上映射的虚拟节点和虚拟链路迁移到能耗增幅较小的物理节点和物理链路上，并采用节点、链路休眠和唤醒机制，动态地调整网络中活动物理节点和物理链路数量。模拟结果表明：EE-VNR 算法有效地均衡了底层物理网络中节点和链路的负载水平，提高了虚拟网络请求接受率，大大地降低了虚拟网络映射的系统能耗。

关键词：网络虚拟化；网络虚拟映射；重构；高效节能；双阈值
中图分类号：TP393 **文献标志码**：A **文章编号**：0529－6579（2015）05－0005－06

An Energy Efficient Virtual Network Reconfiguration Algorithm

PENG Limin
（College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China）

Abstract: Aiming at the problem of energy consumption in the virtual network mapping, according to the characteristics of the virtual network reconfiguration and energy consumption of substrate nodes and links, a reconfiguration optimization model of mapping virtual networks is proposed. By setting resource utilization threshold in the substrate network, virtual nodes that are already mapped onto substrate nodes, which resource utilization is greater than the high threshold or less than low threshold, are remapped onto the substrate nodes consuming smaller energy periodically, the same operation as the virtual links. By using the method of hibernating and waking up substrate nodes and links, the number of active substrate nodes and substrate links in the substrate network are adjusted dynamically. Simulation results show that EE-VNR algorithm balances the substrate nodes and links load effectively, improves the acceptance ratio of virtual network requests and reduces the system energy consumption of mapping virtual networks evidently.

Key words: network virtualization; virtual network mapping; reconfiguration; energy efficient; double threshold

网络虚拟化是指通过整合网络的硬件和软件资源，向用户提供虚拟网络连接的技术。网络虚拟化技术被公认为是解决目前互联网僵化问题的有效手段。通过对公用的底层基础网络设施采用网络虚拟化技术进行抽象并提供统一的可编程接口，可将多个彼此隔离且具有不同拓扑结构的虚拟网络映射到同一个底层物理网络上，从而为用户提供差异化的网络服务。虚拟网络映射是网络虚拟化中的一个关键技术，它是指将虚拟网络（virtual network，VN）映射到底层物理网络（substrate network，SN）上，并根据虚拟网络的资源约束条件，将底层物理网络中的节点和链路资源分配给虚拟网络请求[1]。目

* **收稿日期**：2014－12－22
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）
作者简介：彭利民（1976 年生），男；**研究方向**：网络虚拟化、分布式计算；E-mail：penglm86@126.com

前，大部分的虚拟网络映射算法都是以最小化网络资源代价映射虚拟网络，以此提高虚拟网络请求接受率与系统收益[2]。然而，美国环保署公布的一项数据表明：当前美国数据中心所消耗的电力已占到美国全部电力使用量的 2%，并且其电力需求正在以 12% 的速度递增，2008 年服务于互联网的路由器、服务器、交换机、冷却设施和数据中心等各种设施消耗电量达 8 680 亿度电，占全球总耗电量的 5.3%[3]。因此，如何有效地降低网络设备和计算设备的能耗，已成为一个亟待解决的研究课题。

近几年，学者们提出了一些能耗感知的虚拟网络映射算法。文献［4］通过考虑各个地区以及各个时间段的电价格差异，建立电能消耗模型，将虚拟节点映射到电价较低的节点上，以此降低虚拟网络映射的能耗成本；文献［5－6］通过建立虚拟网络映射能耗模型，将虚拟节点和虚拟链路映射到活动的物理节点和物理链路上，最大限度地关闭或休眠物理节点和物理链路，以此降低虚拟网络映射的系统能耗。文献［2］通过对底层网络资源利用率采用模拟训练的方法，得到不同映射状态下的虚拟网络映射字典库，然后以字典库为蓝本指导虚拟网络映射，以最大限度地关闭或休眠网络节点和网络链路。虽然这些方法可以降低虚拟网络映射的能耗，但由于虚拟网络具有较强的动态特性，当虚拟网络请求随机到达或离开时，底层物理网络中节点和链路的负载随之发生变化，有些节点或链路的负载可能较低，有些节点或链路的负载可能较高甚至可能成为瓶颈资源，不仅浪费了大量的系统能耗，而且降低了后续的虚拟网络请求接受率，因此非常有必要应用虚拟网络重构机制，均衡节点和链路的负载水平，以提高底层物理网络的资源利用率。文献［7］以最小代价为约束条件，将虚拟节点迁移到负载较低的物理节点上，以此均衡物理网络中节点的负载分布。针对虚拟网络的资源需求动态性问题，文献［8］通过提出可增进式重构机制，降低每次虚拟节点迁移个数，以此降低虚拟网络重构的系统开销。文献［9］根据虚拟网络资源需求的历史信息，提出资源需求预测模型，然后基于最小资源代价的虚拟网络映射方法重新映射虚拟网络请求，以此提高虚拟网络请求接受率；文献［10］在现有的虚拟网络映射算法基础上，提出最小能耗链路重构算法，减少活动物理链路使用数量，降低虚拟网络映射的系统能耗。

综上所述，现有的虚拟网络映射方法大部分是基于最小化网络资源代价映射虚拟网络，忽视了虚拟网络映射的能耗问题；另一方面，虽然已提出的能耗感知虚拟网络映射方法可有效地降低虚拟网络映射的能耗开销，但由于虚拟网络随机地到达和离开，底层物理网络中容易出现负载不均衡、系统能耗较高等问题。针对这些问题，本文根据底层物理节点和链路的能耗特性，以及底层物理网络中节点和链路的负载分布状态，采用虚拟网络重构方法，将虚拟节点和虚拟链路动态地迁移到负载较低、能耗增幅较小的物理节点和物理链路上，均衡物理网络的资源分布，降低虚拟网络映射的系统能耗。仿真结果表明，文中提出的 EE-VNR 算法有效地均衡了底层物理网络的负载分布状态，降低了虚拟网络映射的系统能耗，提高了虚拟网络请求接受率。

1 问题描述与优化模型

高效节能的虚拟网络重构是指在保证虚拟网络资源约束的前提下，根据物理节点和物理链路的负载状态，周期地调整虚拟节点和虚拟链路在底层物理网络中的分布位置，并采用休眠和唤醒机制调整活动节点和链路数量，降低虚拟网络映射的系统能耗。本节首先描述虚拟网络映射问题和底层物理网络能耗模型，然后提出虚拟网络重构优化模型。

1.1 虚拟网络映射问题

采用无向带权图 $G_s=(N_s,L_s,A_s^n,A_s^l)$ 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，A_s^n 和 A_s^l 分别表示物理节点和物理链路的属性矢量。与文献［2，4－11］类似，节点属性为 CPU 资源，链路属性为带宽资源。类似地，虚拟网络也采用无向带权图 $G_v=(N_v,L_v,R_v^n,R_v^l)$ 进行表示，其中 N_v 为虚拟节点集合，L_v 为虚拟链路集合，R_v^n 和 R_v^l 分别表示虚拟节点和虚拟链路的资源需求矢量。虚拟网络映射是指将虚拟节点和虚拟链路映射到满足资源需求的物理节点和物理路径上，它可分为节点映射和链路映射两个过程。图 1（a）给出一个具有资源需求的虚拟网络请求，其中方框中的数字表示节点的 CPU 资源需求量，链路旁的数字表示链路的带宽需求量。图 1（b）表示图 1（a）的虚拟网络在图 1（b）物理网络上一个映射方案，其中节点映射为 $\{a\rightarrow A,b\rightarrow C,c\rightarrow F\}$，链路映射为 $\{(a,b)\rightarrow(A,B,C);(a,c)\rightarrow(A,F);(b,c)\rightarrow(C,E,F)\}$。

1.2 底层物理网络能耗模型

底层物理网络的能耗主要包括物理节点能耗和物理链路能耗两个部分。物理节点主要是指服务器，其能耗主要包括处理器、内存、磁盘 I/O 以及

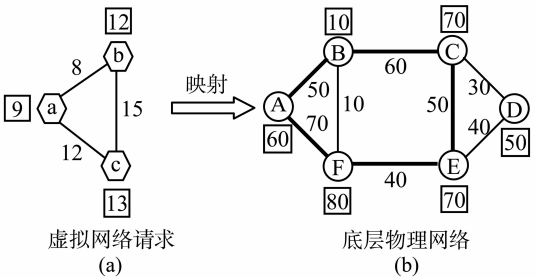

图 1　虚拟网络映射实例

Fig. 1　An example of virtual network embedding

用于冷却的风扇等，其中处理器和内存的能耗占节点能耗的主要部分，目前大部分的处理器，如 Intel 公司的 Speenstep 和 AMD 公司的 PowerNow 技术能够根据负载动态地调节性能[2]。与文献［2，4－7］类似，物理节点的能耗可定义为

$$P_n = \begin{cases} P_b + P_a \cdot u, & \text{当服务器处于开启状态;} \\ 0, & \text{当服务器处于关闭状态} \end{cases} \tag{1}$$

式（1）中 P_b 是服务器的基准能耗，P_m 是服务器最大负荷下的总能耗，$P_a = P_m - P_b$ 是服务器与负载相关的能耗。底层物理网络中物理链路的能耗主要是链路两端网络设备的能耗 P_l，它可以分为静态能耗 p_s 和动态能耗 p_d 两部分，其中，动态能耗与网络设备实际业务负载量相关，静态能耗独立于业务负载量，它取决于网络设备的状态，当网络设备处于开启状态时静态能耗为常量[11]。因此，物理链路的能耗可定义为

$$P_l = \begin{cases} p_s + p_d \cdot \eta, & \text{当物理链路处于开启状态时;} \\ 0, & \text{当物理链路处于关闭状态时} \end{cases} \tag{2}$$

式（2）中，η 为业务负载量因子，它取决于网络设备的业务负载量，如通过链路的数据量。

1.3　虚拟网络重构优化模型

当一个虚拟网络映射到物理网络中后，一个虚拟节点映射到一个物理节点上，一条虚拟链路映射到由一条或多条物理链路组成的无环物理路径上。文中采用 $M(v_i)$ 表示虚拟节点 $v_i \in N_v$ 在物理网络中的映射物理节点，采用二元符号 $x_{v_s}^{v_i}$ 表示虚拟节点 v_i 是否映射到物理节点 v_s 上，当节点 v_i 映射到物理节点 v_s 上时，$x_{v_s}^{v_i}$ 取值为 1，否则为 0；类似地，符号 $f_{l_{mn}}^{l_{uv}}$ 表示虚拟链路 l_{uv} 是否映射到物理链路 l_{mn} 上，当 l_{uv} 映射到物理链路 l_{mn} 上时，$f_{l_{mn}}^{l_{uv}}$ 的值为 1，否则为 0。因此虚拟链路 l_{uv} 映射的能耗可表示为

$$P_{l_{uv}} = \sum_{l_{mn} \in L_S} \Delta P_l^{l_{mn}} \cdot l_{mn} \cdot f_{l_{mn}}^{l_{uv}}, l_{uv} \in L_V \tag{3}$$

一个虚拟网络映射的总能耗可表示为

$$P = \sum_{v_i \in \mathbf{N}_v} \Delta P_n^{v_s} \cdot v_s \cdot x_{v_s}^{v_i} + \sum_{l_{uv} \in L_v} P_{l_{uv}}, v_s \in N_S \tag{4}$$

式（3）中 $\Delta P_l^{l_{mn}}$ 表示当虚拟链路 l_{uv} 映射到物理链路 l_{mn} 时，物理链路 l_{mn} 上能耗的增量，式（4）中 $\Delta P_n^{v_s}$ 表示当虚拟节点 v_i 映射到物理节点 v_s 时，物理节点 v_s 中能耗的增量。

为了刻画底层物理网络中节点和链路负载的分布状态，文中使用资源利用率表示节点和链路的资源状态。物理节点的资源利用率是指物理节点上已被分配的 CPU 资源量与节点 CPU 资源总量之比，物理链路的资源利用率是指分配给虚拟链路的带宽量之和与物理链路带宽总量之比，它们可以定义为

$$Util(v_s) = \frac{\sum_{v_i \to v_s} cpu(v_i)}{cpu(v_s)} \tag{5}$$

$$Util(l_{mn}) = \frac{\sum_{l_{uv} \to l_{mn}} bw(l_{uv})}{bw(l_{mn})} \tag{6}$$

式（5）中 $v_i \to v_s$ 表示虚拟节点 v_i 被映射到物理节点 v_s 上，式（6）中 $l_{uv} \to l_{mn}$ 表示虚拟链路 l_{uv} 被映射到物理链路 l_{mn} 上。

虽然重构虚拟网络可以提高网络性能，降低虚拟网络映射的系统能耗，但是虚拟网络重构也需要付出额外的代价，如虚拟网络提供的网络服务可能暂时中断等。令 RC 表示重新映射虚拟网络 G_v 的重构代价，它主要包含迁移虚拟节点和虚拟链路的代价之和，它可以定义为

$$RC = \alpha \cdot \sum_{v'_i \in \mathbf{N}_v} c(v'_i) + \beta \cdot \sum_{l'_{uv} \in L_v} c(l'_{uv}) \tag{7}$$

式（7）中 v'_i 和 l'_{uv} 分别表示需要重构的虚拟节点和虚拟链路，α 和 β 分别表示重构虚拟节点和链路的相应权重。与文献［7］类似，本文把迁移虚拟节点和虚拟链路数量视为重构代价。

高效节能的虚拟网络重构方法的主要目标是降低虚拟网络映射的系统能耗，同时减少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

$$\min \quad \mu \cdot P + \lambda \cdot RC \tag{8}$$

式（8）中 μ 和 λ 分别表示系统能耗因子 P 和重构代价因子 RC 的相应权重。

2　高效节能的虚拟网络重构算法

虚拟网络重构算法分为两个阶段：① 在物理网络中查找最适合迁移的虚拟节点和虚拟链路；② 根据物理网络资源的分布状态，将需要迁移的虚拟节点和虚拟链路映射到最合适的物理节点和物理链

路上。重构算法的核心思想可归纳为：① 根据底层物理网络的资源分布特性，设置物理节点和物理链路的资源利用率最高阈值和最低阈值；② 根据底层物理网络中节点和链路的负载状态，动态地将资源利用率大于最高阈值以及小于最低阈值的物理节点（和物理链路）上的虚拟节点（和虚拟链路）迁移到能耗增幅较小、资源可用的物理节点（和物理路径）上，均衡底层物理网络的负载强度，减少底层物理网络中活动节点和活动链路数量，降低虚拟网络映射的系统能耗；③ 根据虚拟网络请求的剩余生存时间长短，在重构队列中优先选择生存时间较多的虚拟节点和虚拟链路进行重构，提高虚拟网络重构对网络性能的改善能力，降低重构对网络性能带来的负面影响，优化底层物理网络的资源分布水平。

算法 1：查找迁移虚拟节点算法

```
1：Procedure of getting virtual node migrationList
2：for each substrate node sn in snList do {
3：vnList←sn. getVnList ( )
4：vnList. sortDecreasingLifetime ( )
5：snUtil←sn. getUtil ( )
6：bestFitUtil←Max} //end for
7：for each snUtil < Low_ Threshhold do {
8：migrationList. add (sn. getVnList ( ))
9：vnList. remove (sn. getVnList ( ))
10：snList. remove (sn)} //end for
11：if each snUtil > Up_ threshhold {
12：wake up a hibermating substrate node sn
13：snList. add (sn)}
14：for each sn in snList do {
15：while snUtil > Up_ Threshhold do {
16：for each vn in vnList do {
17：if vn. getUtil ( ) > snUtil - Up_ Threshhold {
18：temp ← vn. getUtil ( ) - snUtil + Up_ Threshhold
19：if temp < bestFitUtil then {
20：bestFitUtil←temp
21：bestFitVn←vn} //end if
22：else if bestFitUtil = = MAX then
23：bestFitVn←vn} //end if
24：break }} //end while
25：snUtil←snUtil - bestFitVn. getUtil ( )
26：migrationList. add (bestFitVn)
27：vnList. remove (bestFitVn)
28：return migrationList} //end for
```

算法 1 根据虚拟节点的剩余生存时间进行降序排列，优先选择剩余生存时间较多的虚拟节点进行重构（步骤 4）；步骤 7 – 10 用于将资源利用率小于低阈值物理节点上映射的虚拟节点直接加入迁移队列中，以便休眠这些节点，从而降低虚拟网络映射的系统能耗；如果底层物理网络中活动节点的资源利用率均大于高阈值，则唤醒一个物理节点（步骤 11 – 13）；步骤 14 – 27 用于将资源利用率大于高阈值的物理节点上映射的虚拟节点自适应地迁出，均衡底层物理网络中的资源分布，提高虚拟网络映射请求接受率。算法 1 的时间复杂度为 $O(|N_s| \cdot |N_v|)$，其中，$|N_s|$ 和 $|N_v|$ 分别为物理节点和虚拟节点的个数。

查找迁移虚拟链路算法与算法 1 基本相同，唯一的差别是每次唤醒物理链路的个数设为 5，其时间复杂度为 $O(|L_s| \cdot |L_v|)$，其中 $|L_s|$ 和 $|L_v|$ 分别为物理链路和虚拟链路的个数。

算法 2：虚拟节点和虚拟链路重映射算法

```
1：Procedure of re-mapping virtual nodes and links
2：for each vn in vnList do {
3：minEnergyCons←MAX
4：mappedSn←NULL
5：for all adjacent virtual link vl connected with vn
6：vlList. add (vl)
7：for each sn in snList do {
8：if sn has enough resource for vn then {
9：energyCons←estimateEnergyCons (sn, vn)
10：if energyCons < minEnergyCons then {
11：mappedSn←sn
12：minEnergyCons←energyCons}}} //end for
13：if mappedSn ≠ NULL then
14：map vn onto the mappedSn
15：for each vl in vlList do
16：map vl using the minimum energy consumption algorithm
17：for each sn in snList and vl in slList {
18：if sn. getVnList ( ) = NULL or sl. getVlList ( ) = NULL
19：hibernate sn or sl }
20：return mapped result }
```

算法 2 首先将与虚拟节点邻接的虚拟链路加入迁移虚拟链路队列中，然后将虚拟节点依次映射到能耗增加最小、资源可用的物理节点上。这里的资

源可用包含两层含义，其一是指该物理节点上的可用 CPU 资源量满足虚拟节点的 CPU 资源需求，并且映射后不会导致该节点的资源利用率大于设置的最高阈值；其二是指物理节点邻接链路的可用累积链路带宽量大于虚拟节点邻接链路的累积链路带宽需求量，并且映射后不会导致物理节点的邻接链路带宽利用率大于设置的最高阈值。步骤 15 – 16 是使用最小能耗路由算法将虚拟链路映射到能耗最小的物理路径上。算法 2 的时间复杂度为 $O(|N_v| \cdot |N_s| + |L_v| \cdot |N_s|^2)$，其中 $|N_v|$ 和 $|L_v|$ 分别为虚拟网络中和虚拟节点和虚假链路的个数，$|L_s|$ 为底层物理网络中物理链路的个数。

3　仿真实验

为了测试 EE-VNR 算法的网络性能，本文通过设计一个离散事件仿真实验，从资源利用率方差、系统能耗和虚拟网络请求接受率等三个方面对 EE-VNR 算法进行性能测试。文中选择文献［12］中的虚拟网络映射算法映射虚拟网络，并用 TA-VNM 表示文献［12］中的虚拟网络映射算法。用 TA-VNM + LC-VNR 表示在文献［12］的基础上采用文献［7］中 LC-VNR 算法重构后的映射过程，用 TA-VNM + EE-VNR 表示在文献［12］的基础上采用本文 EE-VNR 算法重构后的映射过程。

3.1　仿真环境

与文献［12］类似，仿真实验采用 GT-ITM 工具[13]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，每个物理节点的初始状态为关闭状态，物理节点的 CPU 资源量和物理链路的带宽量服从 50 – 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，每个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，每个虚拟网络节点个数服从 2 – 10 的均匀分布，每对虚拟网络节点以 0.5 的概率随机相连，虚拟网络节点 CPU 资源需求量与虚拟链路的带宽需求量服从 1 – 20 的均匀分布。仿真期间内随机选择 10% 的虚拟网络，并使其节点和链路的资源需求量在 1 – 20 随机动态变化。物理节点的基准能耗均设置为 150W，最大能耗为 300W，物理链路的静态能耗设置为 10W，动态能耗设置为 5W。1.3 节式（7）中参数 α 和 β 分别设置为 0.5、式（8）中参数 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，节点资源利用率的最高阈值设置为 75%，最低阈值设置为 25%。每次模拟实验运行约为 50 000 个时间单元，包含 10 000 个虚拟网络请求，共进行 10 次仿真实验，然后取 10 次实验的平均值为实验最终结果。

3.2　实验结果

1）EE-VNR 算法均衡了底层物理网络的负载水平

如图 2 和图 3 可以看出，通过在 TA-VNM 算法的基础上使用 EE-VNR 算法后，底层物理网络中节点和链路的资源利用率方差比 TA-VNM 算法的方差大大地减少。其主要原因是由于 EE-VNR 算法通过利用资源利用率最低阈值，在虚拟网络重构时能有效地休眠资源利用率较低的物理节点和链路。特别是在模拟开始阶段虚拟网络请求数目较小、网络资源充足时，节点和链路的资源利用率方差比 LC-VNR 算法的资源利用率方差大大降低。实验结果表明，EE-VNR 算法能有效地均衡底层物理网络中的负载水平。

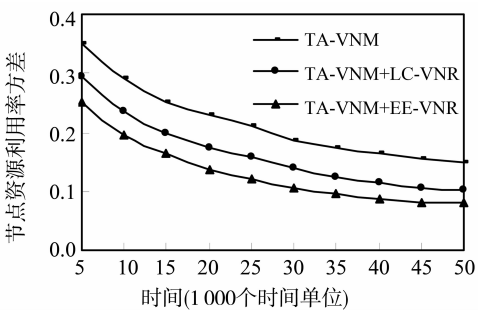

图 2　节点资源利用率

Fig. 2　Node resource utilization ratio

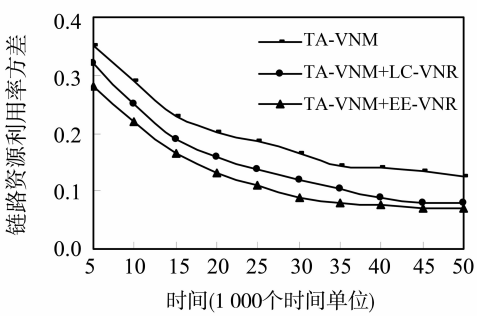

图 3　链路资源利用率

Fig. 3　Link resource utilization ratio

2）EE-VNR 算法降低了虚拟网络映射的系统能耗

如图 4 所示，通过在 TA-VNM 算法的基础上使用 EE-VNR 重构算法，虚拟网络映射的系统能耗比其它两个算法大大地减少。实验结果表明：EE-VNR 算法能有效地利用设定的资源利用率阈值，

动态地休眠资源利用率较低的物理节点和链路；在虚拟节点和虚拟链路迁移时，EE-VNR 算法有效地将虚拟节点和虚拟链路合理地迁移到能耗增幅较小的物理节点和链路上，有效地降低虚拟网络映射的系统能耗。

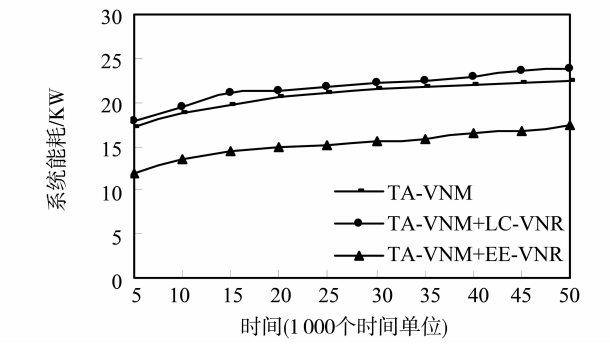

图 4 虚拟网络映射的系统能耗

Fig. 4 The system energy consumption of mapping virtual networks

3）EE-VNR 算法提高了虚拟网络请求接受率

如图 5 所示，通过在 TA-VNM 算法的基础上使用 LC-VNR 和 EE-VNR 重构算法，虚拟网络请求接受率都呈现较大幅度的提升。实验结果表明：EE-VNR 算法能有效地均衡底层物理网络中的负载水平。特别是在模拟阶段后期网络资源相对紧张时，EE-VNR 算法可有效地缓解了资源瓶颈问题，从而显著地提高了虚拟网络请求接受率。

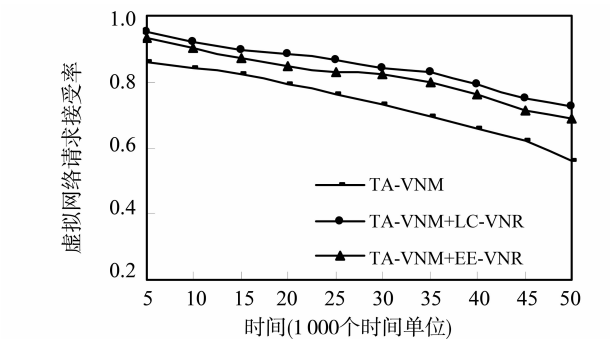

图 5 虚拟网络请求接受率

Fig. 5 The acceptance ratio of virtual network requests

4 结 语

针对网络虚拟化环境下的系统能耗问题，本文通过分析底层物理网络中负载动态变化的主要原因，通过应用资源利用率双阈值方法，周期性地根据节点和链路的负载水平，动态地调整底层物理网络中网络资源的分布状态，并通过采用节点、链路休眠和唤醒机制，自适应地控制底层物理网络中活动节点和活动链路数量。仿真结果表明：文中提出的 EE-VNR 算法可有效地均衡底层物理网络中的负载水平，提高虚拟网络请求接受率，降低虚拟网络映射的系统能耗。

参考文献：

[1] 李小玲，王怀民，丁博，等. 虚拟网络映射问题及其进展[J]. 软件学报，2012，23(11)：3009－3028.

[2] 陈晓华，李春芝，陈良育，等. 主动休眠节点链路的高效节能虚拟网络映射[J]. 软件学报，2014，25(7)：1416－1431.

[3] 张法，ANTA A F，王林，等. 网络能耗系统模型及能效算法[J]. 计算机学报，2012，35(3)：603－615.

[4] AMOKRANE A，ZHANI M F，LANGAR R，et al. Greenhead：Virtual data center embedding across distributed infrastructures [J]. IEEE Transactions on Cloud Computing，2013，1(1)：36－49.

[5] CHANG X L，WANG B，LIU J Q，et al. Green cloud virtual network provisioning based ant colony optimization [C]// Proceeding of the 15th Annual Conference. Companion on Genetic and Evolutionary Computation. New York：ACM Press，2013：1553－1560.

[6] FISCHER A，BECK M T，MEER H D. An approach to energy-efficient virtual network embeddings [C]// Proceeding of the 2013 IFIP/IEEE International Symposium on Integrated Network Management. Ghent：IEEE，2013：1142－1147.

[7] 曲桦，赵季红，郭爽乐，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.

[8] ZHOU Y，YANG X，JIN Y，et al. Incremental re-embedding scheme for evolving virtual network requests [J]. IEEE communications letters，2013，17(5)：1016－1019.

[9] XU Z C，LIANG W F，XIA Q F. Efficient virtual network embedding via exploring periodic resource demands [C]// Proceeding of the 39th Annual IEEE Conference on Local Computer Networks. Edmonton，Canada：IEEE，2014：90－98.

[10] GHAZISAEEDI E，WANG N，TAFAZOLLI R. Link sleeping optimization for green virtual network infrastructures [C]// The 4th IEEE International Workshop on Management of Emerging Networks and Services. Anaheim，CA：IEEE，2012：842－846.

[11] 伍元胜，郭兵，沈艳，等. 面向核心网的多层网络能耗优化方法[J]. 计算机学报，2013，36(7)：1538－1548.

[12] LI X L，WANG H M，GUO C G，et al. Topology awareness algorithm for virtual network mapping [J]. Journal of Zhejiang University-Science C (Computers & Electronic)，2012，13(3)：178－186.

[13] ZEGURA E，CALVERT K，BHATTACHARJEE S. How to model an internetwork [C]// Proceeding of the 15th Annual Joint conference of the IEEE Computer and Communications Society. San Francisco，USA：IEEE，1996：594－602.