

## 拓扑综合评估与权值自适应的虚拟网络映射算法

史朝卫<sup>1</sup> 孟相如<sup>2</sup> 马志强<sup>2</sup> 韩晓阳<sup>1</sup>

1 空军工程大学研究生院 西安 710051

2 空军工程大学信息与导航学院 西安 710077

(cwshi0839@163.com)

**摘要** 针对现有虚拟网络映射算法对节点拓扑特征考虑得不够全面、节点评价方式较为单一且指标权值不能根据网络环境自适应调整等问题,提出一种拓扑综合评估与权值自适应的虚拟网络映射算法。文中在节点映射阶段综合考虑节点中心度、就近度与邻近聚集度等拓扑属性,结合节点CPU与邻接带宽和等资源属性对节点进行多指标重要度排序,根据网络环境的变化利用熵权法自适应调整指标权值。仿真结果表明,相较于最新的和经典的虚拟网络映射算法,所提算法的映射成功率提高了2%~23%,长期平均收益开销比提升了3%~17%,且该算法对不同资源需求类型的虚拟网络请求都能保持良好性能。

**关键词** 虚拟网络映射;拓扑综合评估;权值自适应;邻近聚集;熵权法

**中图分类号** TP393

## Virtual Network Embedding Algorithm Based on Topology Comprehensive Evaluation and Weight Adaptation

SHI Chao-wei<sup>1</sup>, MENG Xiang-ru<sup>2</sup>, MA Zhi-qiang<sup>2</sup> and HAN Xiao-yang<sup>1</sup>

1 School of Graduate, Air Force Engineering University, Xi'an 710051, China

2 School of Information and Navigation, Air Force Engineering University, Xi'an 710077, China

**Abstract** The existing virtual network embedding algorithms do not consider the topological features of nodes comprehensively, the evaluation method of nodes is relative simple and the weights cannot be adaptively adjusted according to the network. To solve these problems, a virtual network embedding algorithm based on topology comprehensive evaluation and weight adaptation is proposed. In the node embedding stage, by considering the centrality, proximity and adjacent aggregation of nodes, this paper establishes a node multi-metric evaluation model combined with the node resource properties such as the node CPU and the sum of adjacent bandwidth. The weights are adjusted adaptively according to the change of network environment by using the entropy weight method. Simulation results show that compared with the latest and classical virtual network embedding algorithms, the acceptance ratio of the proposed algorithm is improved by 2%~23%, and the long-term average revenue-to-cost ratio is increased by 3%~17%. Moreover, the proposed algorithm can maintain good performance for different types of virtual network requests with different resource requirements.

**Keywords** Virtual network embedding, Topology comprehensive evaluation, Weight adaptation, Regional aggregation, Entropy weight method

## 1 引言

随着互联网技术的快速发展,世界正逐步进入5G时代,传统互联网日益僵化<sup>[1]</sup>,已难以应对5G多样化的业务场景。网络虚拟化技术通过屏蔽物理网络架构和协议差异,将物理网络资源池化<sup>[2]</sup>,无差别地为用户提供灵活可定制的网络服务,已成为当前研究热点<sup>[3]</sup>。

虚拟网络映射作为网络虚拟化的核心技术之一,受到学术界的广泛关注<sup>[4-22]</sup>。文献[5]基于节点资源属性评判节点的重要度,采用贪婪算法对虚拟节点进行映射,该算法复杂度

低、运行速度快,但在映射过程中没有考虑节点拓扑特征,导致映射后节点较分散,增加了链路带宽消耗。文献[6]将节点中心度纳入节点排序指标,考虑了节点全局拓扑特征,提高了资源分配效率,但物理网络节点中心度需计算该节点到其他所有物理节点的距离,计算规模较大。文献[7]在进行物理节点排序时,考虑了与已映射虚拟节点对应的物理节点的就近度,避免了盲目计算节点全局中心度。文献[9]借鉴物理学中的场论思想,通过计算网络中节点的相互作用确定各个节点的拓扑重要度。文献[10]在进行节点排序时,综合考虑了节点度数、中心度、邻接带宽和等指标,实现了资源拓扑联合感

收稿日期:2019-06-05 返修日期:2019-09-30 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61873277)

This work was supported by the National Natural Science Foundation of China (61873277).

通信作者:孟相如(xrmeng@126.com)

知。文献[16]将节点中心度和映射后物理节点就近度相结合,充分考虑了节点资源和拓扑特征,进一步提升了算法性能,但亦增加了计算规模。文献[17]借鉴电磁学中的库仑定律和牛顿力学中的万有引力定律,将网络中节点的资源 and 拓扑信息有机结合,统筹考虑节点的拓扑和资源特征,提高了物理网络资源利用率。文献[18]在节点排序过程中考虑邻接节点的资源与拓扑信息,优化资源分配效率,降低了映射开销。由于虚拟网络请求类型具有多样性,在不同网络环境下,这些算法在进行节点排序时只是将节点各项属性指标进行简单的四则运算来评价节点重要度,而忽略了不同网络环境下各指标的权重差异。文献[19]提出了一种逼近理想排序法(Technique for Order Preference by Similarity to Ideal Solution, TOPSIS),根据节点的多个指标与理想节点的相近度进行节点重要度排序,但该算法没有讨论指标赋权问题,指标权重不能根据网络资源与拓扑变化进行自适应调整。文献[20]在传统遗传算法与虚拟网络映射相结合的基础上引入元胞遗传机制,提高了算法的寻优能力。文献[21]在链路映射阶段,采用网络单纯形算法搜寻最佳路径,在降低映射开销的同时缩短了算法的运行时间。

综上所述,现有的虚拟网络映射算法对节点的拓扑特征考虑得不够全面,在进行节点多指标重要度排序时指标权重不能根据网络拓扑类型和资源属性自适应调整。为此,本文在目前研究的基础上提出了一种拓扑综合评估与权值自适应的虚拟网络映射算法(Virtual Network Embedding Algorithm Based on Topology Comprehensive Evaluation and Weight Adaptation, TCEWA-VNE)。首先,基于网络全局和局部角度,综合考虑节点拓扑特征,在节点中心度和就近度的基础上引入节点邻近聚集度,分别采用不同的指标组合方式表示物理网络环境和虚拟网络环境以区分两者规模差异;然后,提出一种熵值赋权的多指标排序方法,对节点进行重要度排序,根据指标集合熵值变化自适应地调整指标权重,实现对不同网络环境的适应能力。在映射过程中,虚拟节点的广度优先搜索<sup>[22]</sup>(Bread First Search, BFS)算法与物理节点的就近聚集度指标配合使用,来对各个方案进行最优化求解。最后,本文设计了两组实验,分别用于验证本文提出的 TCEWA-VNE 算法的性能和对不同网络环境的自适应能力。

## 2 虚拟网络映射模型

在进行虚拟网络映射时需要物理网络和虚拟网络进行抽象建模,以便讨论映射过程中虚拟网络和物理网络的资源与位置约束条件。虚拟网络映射模型如图1所示。

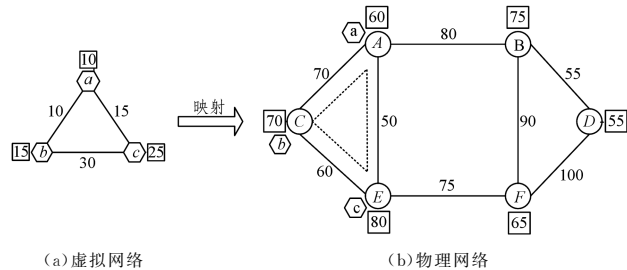


图1 虚拟网络映射示例

Fig. 1 Example of virtual network embedding

## 2.1 虚拟网络

虚拟网络用带权无向图  $G^v = (N^v, L^v, D^v, T^v)$  表示。其中,  $N^v$  为虚拟节点集合,每个虚拟节点  $n^v \in N^v$  包含两个基本属性,即计算资源需求  $cpu(n^v)$  和节点位置坐标  $loc(n^v)$ ;  $L^v$  为虚拟链路集合,每条虚拟链路  $l^v \in L^v$  的带宽需求为  $bw(l^v)$ ;  $D^v$  为节点映射后的距离约束,表示虚拟节点映射至物理网络后允许的最大间距;  $T^v$  表示虚拟网络的持续时间。图1(a)为一个虚拟网络模型,其中  $a, b, c$  为虚拟节点,矩形框内的数字表示节点 CPU 资源需求,链路旁的数字表示链路带宽需求。

## 2.2 物理网络

物理网络用带权无向图  $G^s = (N^s, L^s)$  表示。其中,  $N^s$  为物理节点集合,每个物理节点  $n^s \in N^s$  包含两个基本属性,即可用计算资源  $cpu(n^s)$  和位置坐标  $loc(n^s)$ ;  $L^s$  为物理链路集合,每条物理链路  $l^s \in L^s$  的基本属性为其可用带宽资源  $bw(l^s)$ 。图1(b)为一物理网络模型,其中  $A, B, C, D, E, F$  为物理节点,矩形框内的数字表示节点可用 CPU 资源,链路旁的数字表示链路可用带宽。

## 2.3 虚拟网络映射

进行虚拟网络映射时,基础设施提供商结合自身资源属性,对虚拟网络和物理网络进行综合评价后将物理网络资源高效分配给各个虚拟网络。虚拟网络映射包含两个阶段:

(1) 节点映射  $E_N: N^v \rightarrow N^s$ 。每个物理节点只能承载同一个虚拟网络中的一个节点,不同虚拟网络中的节点可以映射至同一个物理节点上。节点映射需满足的资源约束为  $cpu(n^v) \leq cpu(n^s)$ 。

(2) 链路映射  $E_L: L^v \rightarrow L^s$ 。每条物理链路可以承载不同虚拟网络中的多条虚拟链路。链路映射需满足的带宽约束为  $bw(l^v) \leq bw(l^s)$ 。

图1所示虚拟网络的节点映射结果为:  $\{a \rightarrow A, b \rightarrow C, c \rightarrow E\}$ , 链路映射结果为  $\{(a, b) \rightarrow (A, C), (a, c) \rightarrow (A, E), (b, c) \rightarrow (C, E)\}$ 。

## 3 拓扑综合评估与权值自适应的虚拟网络映射算法

针对现有虚拟网络映射算法节点排序过程中存在的问题,本文提出一种 TCEWA-VNE 算法。首先改进节点评价指标;然后提出一种熵值赋权的多指标排序方法进行节点排序;最后采用 BFS 算法进行虚拟节点映射,采用  $k$ -最短路径算法进行虚拟链路映射。

### 3.1 节点重要度评价指标

(1) 节点 CPU 资源,表示节点可用 CPU 资源或 CPU 资源需求,如式(1)所示:

$$NC(n_i) = cpu(n_i) \quad (1)$$

CPU 资源值越大,节点越重要。

(2) 节点邻接带宽和,如式(2)所示:

$$NB(n_i) = \sum_{l \in L(n_i)} bw(l) \quad (2)$$

其中,  $L(n_i)$  表示网络节点  $n_i$  的邻接链路集合。节点邻接带宽和越大,节点重要度越高。

(3) 节点中心聚集度,本文用节点接近中心度和节点邻近聚集度联合表示虚拟节点拓扑特征。节点中心度表示节点在全局拓扑中的中心性;用节点度数表示节点邻近聚集度,反映

节点邻近聚集特征。本文综合考虑节点局部与全局拓扑特征,来评价虚拟节点重要度,其公式如式(3)所示。

$$C(n_i^v) = \frac{TD(n_i^v)}{\sum_{j=1}^m d(i, j)} \quad (3)$$

其中,  $TD(n_i^v)$  为该节点邻接节点数目, 邻接节点数目越多节点重要度越高;  $\sum_{j=1}^m d(i, j)$  为虚拟节点  $i$  与其他所有节点间的距离和, 表示该节点的全局拓扑特征, 其倒数越大, 则该节点越靠近网络拓扑中心。

(4) 节点就近聚集度, 本文用映射后物理节点就近度和物理节点邻近聚集度联合表示物理节点拓扑特征, 对节点拓扑特征进行综合评估, 并评价节点重要度, 如式(4)所示。

$$C(n_i^p) = \frac{TD(n_i^p)}{\sum_{n_j \in \phi(n)} d(i, j)} \quad (4)$$

其中,  $TD(n_i^p)$  为该物理节点邻接节点数目,  $\phi(n)$  表示虚拟节点  $n_i$  的邻接节点中已映射成功的虚拟节点所对应的物理节点的集合;  $d(i, j)$  为两个物理节点之间最短路径跳数。节点就近聚集度越大, 该节点周围的资源越集中, 越靠近网络中心, 节点越重要。

### 3.2 熵值赋权的多指标节点排序方法

#### 3.2.1 熵值赋权的多指标节点排序方法的原理

熵值赋权的多指标排序方法是对 TOPSIS 排序方法的改进, 它利用熵权法对各项指标进行自适应赋权。传统的 TOPSIS 排序算法在确定评价指标权重时, 一般采用主观赋值法, 根据不同网络环境人工调整指标权重。本文提出的熵权法可根据网络指标参数的变化范围自适应地为各指标赋以合适的权重。熵权法是在熵理论的基础上形成的一种用于评价研究对象重要程度的方法, 它是一种以熵理论核心观点为指导的赋权方法, 若指标集合无序性越高, 该指标提供的信息量就越大, 在综合评价中该指标的权重就应越高。

#### 3.2.2 熵值赋权的多指标节点排序方法的模型

(1) 构建决策矩阵。假设有  $M$  个排序对象, 表示不同的节点, 每个节点有  $N$  个评价指标。记第  $i$  个节点的第  $j$  个评价指标的值为  $X_{ij}$  ( $i=1, 2, \dots, M; j=1, 2, \dots, N$ ), 所有网络节点及其评价指标构成的决策矩阵如式(5)所示:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{pmatrix} \quad (5)$$

(2) 归一化决策矩阵。由于各项指标之间量纲不同, 可能会存在数量级差异, 不便进行直接比较, 为了消除指标间量纲差异, 须对指标进行归一化处理, 如式(6)所示:

$$y_{ij} = \frac{x_{ij}}{\sum_{i=1}^M x_{ij}} \quad (6)$$

由此, 可得标准归一化矩阵为:

$$\mathbf{Y} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & \cdots & y_{2N} \\ \vdots & \vdots & & \vdots \\ y_{M1} & y_{M2} & \cdots & y_{MN} \end{pmatrix} \quad (7)$$

(3) 依据指标熵值计算指标权重, 指标熵值的计算式如式(8)所示:

$$h_j = -k \sum_{i=1}^M Y_{ij} \ln Y_{ij}, j=1, 2, \dots, N \quad (8)$$

其中,  $k = \frac{1}{\ln M}$  为波尔兹曼系数, 表示系统自身属性。

计算信息熵冗余度, 如式(9)所示:

$$r_j = 1 - h_j, j=1, 2, \dots, N \quad (9)$$

计算指标权重, 如式(10)所示:

$$\omega_j = \frac{r_j}{\sum_{j=1}^N r_j}, \sum_{j=1}^N \omega_j = 1 \quad (10)$$

由此得到指标权重矩阵  $\mathbf{W}$ , 如式(11)所示:

$$\mathbf{W} = [\omega_1, \omega_2, \dots, \omega_N] \quad (11)$$

(4) 加权归一化决策矩阵:

$$\mathbf{Z} = \mathbf{Y}\mathbf{W} = \begin{pmatrix} y_{11}\omega_1 & y_{12}\omega_2 & \cdots & y_{1N}\omega_N \\ y_{21}\omega_1 & y_{22}\omega_2 & \cdots & y_{2N}\omega_N \\ \vdots & \vdots & & \vdots \\ y_{M1}\omega_1 & y_{M2}\omega_2 & \cdots & y_{MN}\omega_N \end{pmatrix} \quad (12)$$

(5) 确定最优与最劣理想方案, 如式(13)和式(14)所示:

$$\begin{aligned} \mathbf{A}^+ &= \{\max(z_{i1}, z_{i2}, \dots, z_{iN})\} \\ &= \{z_1^{\max}, z_2^{\max}, \dots, z_N^{\max}\} \end{aligned} \quad (13)$$

$$\begin{aligned} \mathbf{A}^- &= \{\min(z_{i1}, z_{i2}, \dots, z_{iN})\} \\ &= \{z_1^{\min}, z_2^{\min}, \dots, z_N^{\min}\} \end{aligned} \quad (14)$$

(6) 分别计算每个方案到最优与最劣理想方案的欧氏距离, 如式(15)和式(16)所示:

$$D_i^+ = \sqrt{\sum_{j=1}^N (z_{ij} - z_j^{\max})^2} \quad (15)$$

$$D_i^- = \sqrt{\sum_{j=1}^N (z_{ij} - z_j^{\min})^2} \quad (16)$$

(7) 计算各方案与理想方案的相近度, 如式(17)所示:

$$S_i = \frac{D_i^-}{D_i^- + D_i^+}, 0 \leq S_i \leq 1 \quad (17)$$

(8) 根据每个方案与理想方案的相近度  $S_i$  对网络中各个节点进行重要度排序, 相近度越大, 则节点在网络中的重要程度越高。

### 3.3 节点与链路映射流程

在节点映射阶段中, 首先对虚拟节点进行排序, 综合考虑节点的资源属性和拓扑特征, 选取节点 CPU、邻接带宽和以及节点中心聚集度作为排序指标, 对指标进行归一化处理。然后, 采用熵权法为各项指标赋权, 并计算各个节点与理想方案的相近度  $S_i$ , 进而以  $S_i$  最大的虚拟节点为根节点运行 BFS 算法, 按距离根节点的跳数将剩余的虚拟节点划分为不同的集合。之后, 依次选取距离根节点最近的虚拟节点集合中  $S_i$  最大的节点作为待映射节点, 寻找满足虚拟节点资源和位置约束的物理节点集合  $\phi(n_i^v)$ , 选取其 CPU 资源、节点邻接带宽和以及节点就近聚集度作为排序指标, 采用相同方法对各个物理节点进行排序, 并将虚拟节点映射至重要度最大的物理节点上。综合运用虚拟节点的 BFS 算法与物理节点的就近聚集度指标, 增强虚拟网络与物理网络拓扑耦合, 最大程度地减少链路上的带宽资源消耗, 保证虚拟网络的映射成功率与长期平均收益开销比。TCEWA-VNE 算法的节点映射流程如算法 1 所示。

#### 算法 1 TCEWA-VNE 算法的节点映射流程

输入:  $G^v, G^s$

输出: Node\_Embedding\_List

1. for 每一个虚拟节点  $n_i^v \in N^v$  do

2. 选取  $NC(n_i^v), NB(n_i^v)$  和  $C(n_i^v)$  指标组合, 利用熵值赋权的多指标

排序方法计算虚拟节点  $n_i^v$  与理想节点的相近度  $S(n_i^v)$

3. end for
4. 以  $S(n_i^v)$  最大的虚拟节点  $n_i^v$  为根节点运行 BFS 算法,按与  $n_i^v$  的距离将剩余虚拟节点划分为不同集合
5. 依次选取距离根节点最近的虚拟节点集合,根据  $S(n_i^v)$  对集合中的节点进行排序
6. 将排序结果存入 Virtual\_Node\_List
7. for Virtual\_Node\_List 中的每个虚拟节点  $n_i^v$  do
8. 选取满足虚拟节点资源和位置约束的物理节点集合  $\phi(n_i^s)$ ,并排除已映射物理节点集合  $\Theta(n_i^s)$ ,  $\phi(n_i^s) = \phi(n_i^s) - \Theta(n_i^s)$
9. if  $\phi(n_i^s)$  为空 then
10. Return NODE\_EMBEDDING\_FAILED
11. else
12. for  $\phi(n_i^s)$  中每一个待选物理节点  $n_i^s$  do
13. 选取  $NC(n_i^s)$ ,  $NB(n_i^s)$  和  $C(n_i^s)$  指标组合,利用熵值赋权的多指标排序方法计算  $n_i^s$  的重要度  $S(n_i^s)$
14. end for
15. 将  $n_i^v$  映射至  $S(n_i^s)$  最大的物理节点上,并将映射关系存入 Node\_Embedding\_List 中
16. 更新已映射物理节点集合  $\Theta(n_i^s)$
17. end if
18. end for
19. return NODE\_EMBEDDING\_SUCCESS

根据节点映射结果,采用  $k$ -最短路径算法<sup>[23]</sup>进行链路映射。在进行节点映射时,首先计算虚拟节点的重要度  $S(n^v)$ ,这需要计算任意两个节点间的最短路径,对应的时间复杂度为  $O(|N^v|^2)$ ,其中  $|N^v|$  为虚拟节点的数目。对满足待映射虚拟节点资源与位置约束的候选物理节点  $n^s$ ,需要计算其重要度  $S(n^s)$ ,对应的时间复杂度为  $O(|N^v| |N^s|^2)$ ,其中,  $|N^s|$  为物理节点数目,  $O(|N^s|^2)$  表示在计算物理节点重要度时计算任意两个节点间最短路径对应的时间复杂度。在进行链路映射时,采用  $k$ -最短路径算法,对应的时间复杂度为  $O(k |N^v| (|L^s| + |N^s| \log |N^s|))$ 。因此, TCEWA-VNE 算法的总时间复杂度为  $O(|N^v|^2 + |N^v| |N^s|^2 + k |N^v| (|L^s| + |N^s| \log |N^s|))$ ,是多项式时间内可解的启发式算法。

## 4 算法性能评估与分析

本节给出常用算法性能评价指标,并设置两组对比实验验证算法性能。第一组实验将 TCEWA-VNE 算法与其他 4 种经典算法作对比,仿真验证算法的性能提升;第二组实验仿真分析 TCEWA-VNE 算法对不同虚拟网络类型的自适应能力。

### 4.1 算法性能评价指标

本文采用映射成功率和长期平均收益开销比对所提算法性能进行分析评价。

#### (1) 映射成功率

虚拟网络的映射成功率可定义为:

$$\omega_{accept} = \lim_{T \rightarrow \infty} \frac{NUM_{suc}(T)}{NUM_v(T) + \delta} \quad (18)$$

其表示在时间  $T$  内,映射成功的虚拟网络数目与虚拟网络请求总数之比。其中  $NUM_{suc}(T)$  和  $NUM_v(T)$  分别表示在时间  $T$  内成功映射的虚拟网络数目与虚拟网络总数,  $\delta$  为无穷小的正数。在相同条件下,虚拟网络映射成功率越高,

算法性能越好。

#### (2) 长期平均收益开销比

对于基础设施提供商来说,映射的目的是在保证收益的同时,降低物理网络开销。在  $t$  时刻,成功映射一个虚拟网络获得的收益为:

$$R(G^v, t) = \alpha \sum_{n^v \in N^v} cpu(n^v) + \sum_{l^v \in L^v} bw(l^v) \quad (19)$$

$t$  时刻,物理网络承载一个虚拟网络需付出的开销为:

$$C(G^v, t) = \beta \sum_{n^v \in N^v} cpu(n^v) + \sum_{l^v \in L^v} hops(l^v) \cdot bw(l^v) \quad (20)$$

其中,参数  $\alpha$  与  $\beta$  分别表示在收益和开销中节点 CPU 与链路带宽资源的比重,本文将二者均设置为 1;  $hops(l^v)$  表示映射成功的虚拟链路在物理网络中经过的跳数。

通常利用长期平均收益开销比来反映物理网络资源利用效率,其公式如式(21)所示:

$$\eta = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T R(G^v, t)}{\sum_{t=0}^T C(G^v, t)} \quad (21)$$

### 4.2 实验环境设置

本文在 Matlab 环境下对所提算法的性能进行评估与分析。物理网络在长和宽都为 1000 的平面内生成 100 个均匀分布的节点和 500 条链路,节点间以 0.5 的概率相互连接。节点可用 CPU 资源和链路可用带宽资源服从  $[50, 100]$  的均匀分布;虚拟网络节点个数服从  $[2, 10]$  的均匀分布;节点 CPU 和链路带宽需求服从  $[10, 30]$  的均匀分布。设置所有节点映射后位置约束  $D(n_i) = 500$ ;设置虚拟网络请求到达服从  $\lambda = 1/20$  的泊松分布,平均每 100 个时间单位到达 5 个虚拟网络请求;虚拟网络生存周期  $T^v$  服从均值  $\mu = 1000$  的指数分布。

为避免随机因素对实验结果产生扰动,仿真实验共进行 10 次,每次运行 50000 个时间单位,并取实验结果的平均值作为最终结果。实验流程如图 2 所示。

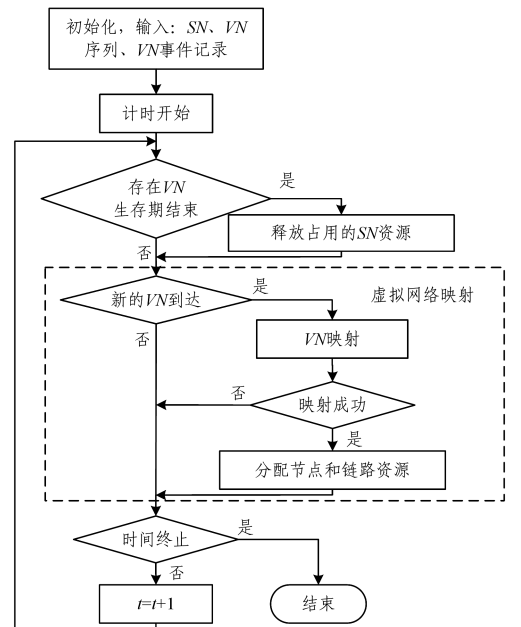


图 2 虚拟网络映射实验的流程图

Fig. 2 Flow chart of virtual network embedding experiment

预先生成物理网络 SN 和虚拟网络请求集合 VN, 包含网络资源拓扑模型和各虚拟网络请求的到达时间、生存时间和结束时间。数据被输入模拟平台后开始计时, 每一时间单元, 依次判断是否存在 VN 生存期结束和新的 VN 请求到达。物理网络在虚拟网络生存期结束时释放占用资源, 在虚拟网络映射成功时分配网络资源。

#### 4.3 TCEWA-VNE 算法的性能分析

本节在相同物理网络拓扑及虚拟网络请求条件下, 从映射成功率和长期平均收益开销比两方面对比分析 TCEWA-VNE 算法与其他 5 种算法(见表 2)的性能, 表 3 列出 6 种算法节点映射复杂度的对比。

表 2 算法描述

Table 2 Description of algorithms

| 算法名称      | 节点映射准则  |
|-----------|---|
| TCEWA-VNE | 本文所提算法, 节点映射基于节点资源度、节点中心度、就近度和邻近聚集度                   |
| RT-VNE    | 文献[16]提出的就近中心性算法, 节点映射基于节点资源度和中心度, 同时考虑映射后节点就近度       |
| TP-VNE    | 文献[9]提出的基于拓扑势的虚拟网络映射算法, 节点映射基于节点的拓扑势和节点资源度            |
| TAS-VNE   | 文献[19]提出的多指标排序算法, 节点映射基于节点资源度和中心度, 采用 TOPSIS 法对节点进行排序 |
| CL-VNE    | 文献[6]提出的中心度算法, 节点映射基于节点中心度                            |
| GS-VNE    | 文献[5]提出的贪婪算法, 节点映射基于节点资源度                             |

表 3 算法复杂度的对比

Table 3 Comparison of algorithm complexity

| 算法名称      | 节点映射复杂度                     |
|-----------|-----------------------------|
| TCEWA-VNE | $O( N^v ^2 +  N^v  N^s ^2)$ |
| RT-VNE    | $O( N^v ^2 +  N^v  N^s ^2)$ |
| TP-VNE    | $O( N^v ^2 +  N^v  N^s ^2)$ |
| TAS-VNE   | $O( N^v ^2 +  N^v  N^s ^2)$ |
| CL-VNE    | $O( N^v ^2 +  N^v  N^s ^2)$ |
| GS-VNE    | $O( N^v  +  N^v  N^s )$     |

图 3 为 6 种不同虚拟网络映射算法对应的映射成功率随时间的变化情况。可以看出, GS-VNE 算法的映射成功率约为 0.63, CL-VNE 算法的映射成功率约为 0.59, 它们在节点映射阶段进行节点排序时仅考虑了节点资源属性或中心度拓扑属性, 未将两者结合考虑, 映射成功率较低。TAS-VNE 算法将节点 CPU 资源、邻接链路带宽以及中心度 3 个指标纳入节点排序指标, 综合评价 3 种属性, 其性能有一定提高, 映射成功率稳定在 0.67 左右, 但指标权重不能随网络环境的变化进行自适应调整, 故其性能提升幅度较小。TP-VNE 算法通过计算节点的拓扑势, 充分考虑节点的拓扑特征, 虚拟网络映射成功率提高至 0.73 左右。RT-VNE 算法在中心度的基础上进一步考虑物理节点的就近度, 结合虚拟节点的 BFS 算法, 使映射后的节点在物理网络拓扑上的距离更近, 减少了链路带宽资源消耗, 使物理网络能够接受更多的虚拟网络请求, 因此, 其算法性能有较大幅度提升, 映射成功率保持在 0.80 左右。TCEWA-VNE 算法综合考虑了节点中心度、就近度和邻近聚集度等拓扑属性, 结合节点 CPU 和邻接链路带宽和资源属性对节点进行排序, 利用熵权法为不同虚拟网络 and 不同物理网络类型的评价指标自适应赋权, 有效缩短了链路映射长度, 节省了带宽资源, 因此, 该算法的性能最优, 稳定在 0.82 附近。

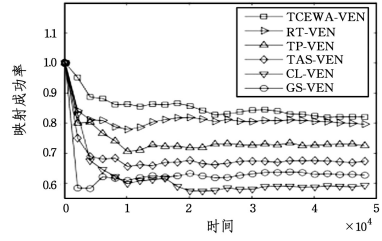


图 3 映射成功率

Fig. 3 Acceptance ratio

图 4 反映了 6 种算法对应的长期平均收益开销比的随时间变化的情况。GS-VNE 算法和 CL-VNE 算法节点评价指标单一, 长期平均收益开销比明显较低。TAS-VNE 综合考虑了节点资源与拓扑特征, 长期平均收益开销比有所提升, 但其没有考虑映射后物理节点间的距离, 增大了链路带宽开销, 故长期平均收益开销比的提升幅度较小。TP-VNE 算法在进行节点映射时考虑了节点的拓扑势, 增强了虚拟网络与物理网络的拓扑关联, 降低了映射开销, 进一步提高了长期平均收益开销比。RT-VNE 算法在选择物理节点时考虑了节点就近度, 有效缩短了链路映射长度, 节省了链路带宽开销, 在保证收益的同时降低了虚拟网络映射开销, 提高了长期平均收益开销比。TCEWA-VNE 算法针对不同网络环境自适应调整节点资源与拓扑各项指标的权重, 高效地利用物理网络 CPU 与带宽资源, 显著提高了长期平均收益开销比。

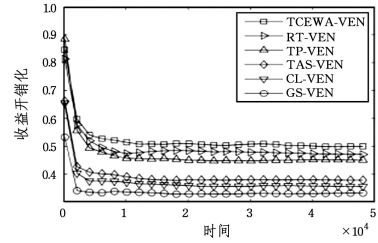


图 4 长期平均收益开销比

Fig. 4 Long-term average revenue to cost ratio

#### 4.4 TCEWA-VNE 算法对不同虚拟网络类型的自适应能力分析

前文对比分析了不同算法在设定的虚拟网络请求下的性能, 但不同用户对网络的需求往往不尽相同, 因此, 映射算法对虚拟网络的适应能力是衡量算法性能的关键因素。本节设置不同带宽需求与不同 CPU 需求两种虚拟网络类型, 仿真分析 TCEWA-VNE 算法对不同网络类型的适应能力。

##### 4.4.1 不同带宽需求

本文在虚拟网络带宽需求不同的情况下, 对比分析不同算法性能。虚拟网络 CPU 需求服从  $[10, 30]$  的均匀分布, 带宽需求服从  $[10, x]$  的均匀分布, 设置虚拟网络带宽需求上限  $x$  为  $[20, 60]$ 。

图 5 和图 6 分别为不同算法的映射成功率和长期平均收益开销比随虚拟网络带宽需求变化的情况。结合图 5 和图 6 可以看出, 随着虚拟网络带宽需求的增大, 每条物理链路能承载的链路数目减少, 所有算法的映射成功率都有所下降, 但在不同环境下 TCEWA-VNE 算法的映射成功率和长期平均收益开销比都明显优于其他算法。当 CPU 需求相对带宽需求

较大时,利用熵权法赋予 CPU 资源指标较大权重,优先映射 CPU 资源需求大的节点,提高映射成功率。随着虚拟网络带宽需求上限的不断变大,带宽指标集合的无序性增加,其权重可得到相应提升,保证了虚拟网络映射成功率。同时, TCEWA-VNE 算法在选择目标物理节点时考虑了物理节点的就近聚集度,缩短了映射后的节点距离,减少了链路映射开销,提高了物理网络长期平均收益开销比。

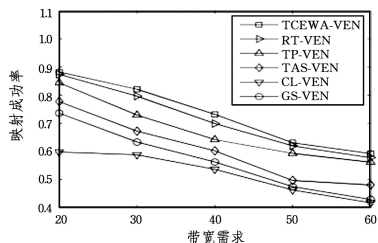


图 5 不同带宽需求时的映射成功率

Fig. 5 Acceptance ratio for different bandwidth requirements

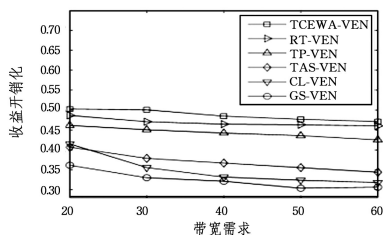


图 6 不同带宽需求时的长期平均收益开销比

Fig. 6 Long-term average revenue to cost ratio for different bandwidth requirements

#### 4.4.2 不同 CPU 需求

本节在虚拟网络 CPU 需求不同的情况下,对比分析不同算法性能。虚拟网络带宽需求服从 $[10, 30]$ 的均匀分布, CPU 需求服从 $[10, x]$ 的均匀分布,设置虚拟网络 CPU 需求上限  $x$  为 $[20, 60]$ 。

图 7 和图 8 分别为不同算法的映射成功率和长期平均收益开销比随虚拟网络 CPU 需求变化的情况。结合图 7 和图 8 可以看出,随着虚拟网络节点 CPU 需求的增大,每个节点能承载的虚拟节点数目减少,所有算法映射成功率随之降低,但在不同环境下 TCEWA-VNE 算法性能都明显优于其他算法。前期节点的带宽需求指标集合无序性较高,利用熵权法赋予带宽指标较大权重,合理利用带宽资源,可保证映射成功率。后期随着节点 CPU 资源需求上限不断增大,物理节点的 CPU 资源限制成为映射失败的主要原因, TCEWA-VNE 算法自适应提升 CPU 资源权重,优先将虚拟节点映射至 CPU 资源充足的物理节点上,可以有效提升算法性能。

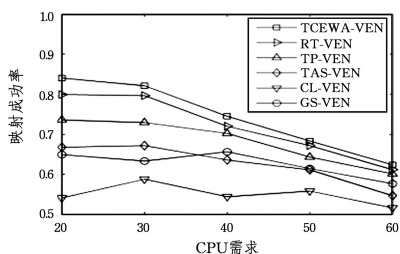


图 7 不同 CPU 需求时映射成功率

Fig. 7 Acceptance ratio for different CPU requirements

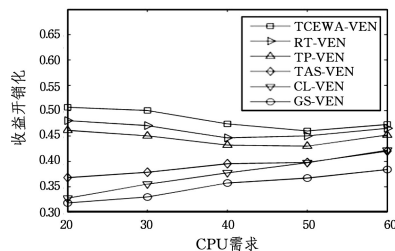


图 8 不同 CPU 需求时长期平均收益开销比

Fig. 8 Long-term average revenue to cost ratio for different CPU requirements

通过以上两组实验可以看出,随着虚拟网络资源需求增大,各算法映射成功率都将受到影响,但 TCEWA-VNE 算法综合考虑节点的全局拓扑特征和邻近聚集度,依据网络资源变化自适应调整相关指标权重,与其他算法相比,其可以有效提高映射成功率与长期平均收益开销比。

**结束语** 本文针对现有虚拟网络映射算法对节点拓扑特征考虑得不够全面、节点评价方式较为单一且指标权重不能根据网络环境自适应调整等问题,提出一种拓扑综合评估与权值自适应的虚拟网络映射算法。首先,基于网络全局和局部角度,综合考虑节点拓扑特征;然后,结合节点资源属性对节点进行多指标排序,利用熵权法对指标变化进行动态感知,根据指标集合的熵值大小,为指标自适应赋权。在节点映射阶段,虚拟节点的 BFS 算法与物理节点的就近聚集度指标配合使用,增强了虚拟网络和物理网络拓扑耦合,减少了链路带宽消耗。链路映射阶段采用  $k$ -最短路径算法,有效提高了映射成功率。实验结果表明, TCEWA-VNE 算法明显提高了虚拟网络映射成功率与长期平均收益开销比,且对不同网络环境有良好的适应能力,性能始终优于其他算法。我们下一步需将物理网络可靠性引入 TCEWA-VNE 算法,研究虚拟网络可靠映射过程中指标选择与最优赋权问题。

#### 参考文献

- [1] LI Z F, LU Z B, DENG S H, et al. Self-Adaptive Virtual Network Embedding Algorithm Based on Software-Defined Networks [J]. IEEE Transactions on Network and Service Management, 2019, 16(1): 362-373.
- [2] CAO H T, HU H, QU Z C, et al. Heuristic Solutions of Virtual Network Embedding: a survey [J]. China Communications, 2018, 15(3): 186-219.
- [3] LI R Z, WU Q B, TAN Y S, et al. On the optimal approach of survivable virtual network embedding in virtualized SDN [J]. IEEE Transactions on Information and Systems, 2018, 101(3): 698-708.
- [4] SU Y Z, MENG X R, MENG Q W, et al. Environment Adaptive and Joint Topology Aware Virtual Network Embedding Algorithm [J]. Journal of Electronics & Information Technology, 2018, 40(1): 79-86.
- [5] 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): 19-29.
- [6] WANG Z H, HAN Y N, LIN T, et al. Resource allocation algo-

- rithms in the reconfigurable network based on network centrality and topology potential[J]. *Journal on Communications*, 2012, 33(8):10-20.
- [7] CUI H Y, GAO W J, LIU J, et al. A virtual network embedding algorithm based on virtual topology connection feature[C]// *International Symposium on Wireless Personal Multimedia Communications*. New Jersey, USA, 2013:1-5.
- [8] FISCHER A, BOTERO J, BECK M, et al. Virtual network embedding: a survey[J]. *IEEE Communications Surveys & Tutorials*, 2013, 15(4):1888-1906.
- [9] LIU X B, WANG B H, YANG Z X, et al. A Virtual Network Embedding Algorithm Based on Topology Potential[J]. *Journal of Electronics & Information Technology*, 2018, 40(7):1684-1690.
- [10] FENG M, LIAO J X, WANG J Y, et al. Topology-aware virtual network embedding based on multiple characteristics[C]// *IEEE ICC 2014-Next-Generation Networking Symposium*. Sydney City, 2014:2956-2962.
- [11] URAYAMA Y, TACHIBANA T. Virtual network construction with K-shortest path algorithm and Prim's MST Algorithm for robust physical networks[J]. *Lecture Notes in Engineering & Computer Science*, 2014, 2209(1):206-210.
- [12] DING J, HUANG T, WANG J, et al. Virtual network embedding through node connectivity[J]. *Journal of China University of posts & Telecommunications*, 2015, 22(1):17-23.
- [13] PENG L M. Virtual Network Embedding Algorithm Based on Breadth-first Search[J]. *Journal of Sichuan University (Engineering Science Edition)*, 2015, 47(2):117-122.
- [14] ESPOSITO F, PAOLA D, MATTA I. On Distributed Virtual Network Embedding With Guarantees[J]. *IEEE/ACM Transactions on Networking*, 2016, 24(1):569-582.
- [15] OLIVEIRA R, MARCON D, BAYS L, et al. Opportunistic resilience embedding (ORE): Toward cost-efficient resilient virtual network[J]. *Computer Networks*, 2015, 89(7):59-77.
- [16] XIE F, MENG X R, ZHAO Z Y, et al. Virtual network mapping algorithm based on adjacent node and topology awareness[J]. *Computer Engineering*, 2018, 44(9):107-112.
- [17] CAO H T, YANG L X, ZHU H B. Novel Node-Ranking Approach and Multiple Topology Attributes-Based Embedding Algorithm for Single-Domain Virtual Network Embedding[J]. *IEEE Internet of Things Journal*, 2018, 5(1):108-120.
- [18] ZHENG Q, LI J, TIAN H L, et al. A 2-Layers Virtual Network Mapping Algorithm Based on Node Attribute and Network Simplex[J]. *IEEE Access*, 2018, 6:77474-77484.
- [19] GONG S Q, CHEN J, HUANG C H, et al. Trust-aware secure virtual network embedding algorithm[J]. *Journal on Communications*, 2015, 36(11):180-189.
- [20] WANG M, ZHUANG L, WANG G Q, et al. Virtual Network Mapping Algorithm Based on Cellular Genetic Mechanism[J]. *Computer Science*, 2018, 45(12):66-70, 103.
- [21] WANG Z Z, ZHENG Q, CHEN C, et al. Virtual Network Mapping Algorithm Based on Network Simplex[J]. *Computer Engineering*, 2019, 45(4):13-17, 24.
- [22] CHENG X, SU S, ZHANG Z B, et al. Virtual network embedding through topology-aware node ranking[J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(2):39-47.
- [23] EPPSTEIN D. Finding the k shortest paths [J]. *SIAM Journal of Computer*, 1994, 28(2):652-763.



**SHI Chao-wei**, born in 1996, postgraduate. His main research interests include network virtualization and so on.



**MENG Xiang-ru**, born in 1963, Ph. D., professor, Ph. D supervisor. His main research interests include next generation internet and cyber security.