基于拓扑影响度和回溯迁移的虚拟网可靠性映射方案

王志明* 汪斌强 王保进
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：底层物理节点或链路失效将影响虚拟网服务提供的连续性，因此如何实现虚拟网的可靠性映射是当前研究亟待解决的问题。文章建立了虚拟网映射(VNM)的数学模型，量化分析了虚拟网的可靠性，并归结出影响虚拟网可靠性的因素。为了克服这些因素，文章分别提出基于拓扑影响度(TID)的虚拟网映射(VNM-TID)算法和基于回溯机制的迁移算法(MA-Back)。仿真结果表明，VNM-TID & MA-Back 算法在虚拟网请求接受率、迁移成功率和有效承载率上具有优势，提高了虚拟网的可靠性。

关键词：网络虚拟化；虚拟网映射；可靠性；拓扑影响度；回溯迁移
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2012)12-2898-07
DOI: 10.3724/SP.J.1146.2012.00101

Virtual Network Reliable Mapping Scheme Based on Topology Impact Degree and Backtracking Migration

Wang Zhi-ming Wang Bin-qiang Wang Bao-jin
(*National Digital Switching System Engineering & Technology Research Center, Zhengzhou* 450002, *China*)

Abstract: Virtual network service persistence will be impacted by substrate node or link failure, so how to achieve the reliable mapping of virtual network is recently an important problem to be solved. In this paper, a virtual network mapping problem is modeled, and then the reliability of virtual network is quantified and analyzed to sum up the main influence factors. To overcome these factors, an algorithm of Virtual Network Mapping based on Topology Impact Degree (VNM-TID) and a Migration Algorithm based on Backtracking (MA-Back) are separately proposed. Simulation results show that the algorithms have advantages on virtual network Request Accepted Ratio (RAR), Migration Success Ratio (MSR) and Valid Loading Ratio (VLR), and therefore improve the reliability of virtual network.

Key words: Network virtualization; Virtual Network Mapping (VNM); Reliability; Topology Impact Degree (TID); Backtracking migration

1 引言

为解决网络僵化问题[1]，网络虚拟化技术[2]受到学术界的广泛关注。该技术允许在共享的底层网络基础设施之上同时运行多个异构的虚拟网，每个虚拟网相当于底层网络的资源分片，承载特定类型业务。虚拟网映射问题是网络虚拟化研究的关键内容，主要完成将虚拟网请求映射到底层网络空闲资源之上的任务。虚拟网映射问题中的底层网络拓扑结构是处于动态变化的，如何在动态拓扑环境下实现虚拟网的可靠性映射是当前研究亟待解决的问题。

目前，虚拟网映射问题主要关注正常网络环境下的资源分配，并提出一种解决最优化映射(NP-hard 问题)的启发式算法，比如 Zhu 等人[3]提出的实现节点/链路压力均衡的启发式算法，Butt 等人[4]提出的感知关键节点/链路的 TA-RViNE 算法，以及 Cheng 等人[5]利用 PageRank 思想提出的基于节点排序的 RW-BFS 算法等。然而，针对动态拓扑环境下的虚拟网可靠性研究却相对薄弱，主要分为主备切换和在线迁移两种方式。对于前者，Rahman 等人[6]提出了基于混合策略的链路备份生存性算法，Chen 等人[7]提出了备份链路带宽共享的 Pardalis 算法，Yu 等人[8]提出了 l 冗余和 k 冗余虚拟网拓扑增强算法，文献[9]又针对虚拟网区域故障提出了 SOUM 和 IOCM 启发式算法进行最小资源消耗求解。对于后者，Cai 等人[10]提出一种基于代价最小原则的节点迁移与重映射算法，Houidi 等人[11]提出一种分布式容错管理算法；此外，文献[3]提出周期性地检查底层网络的负载情况，将过载的节点链路

2012-02-10 收到，2012-10-22 改回
国家 973 计划项目(2012CB315901, 2012CB315905)和国家 863 计划项目(2011AA01A103)资助课题
*通信作者：王志明 wangzm05@gmail.com

上所有虚拟网按照负载均衡策略进行重映射，文献[12]提出周期性地对持续时间较长的虚拟网链路进行路径分割和迁移，以便提高底层网络资源利用率。

上述研究的不足之处在于：(1)对于动态拓扑环境下的虚拟网可靠性未有明确的量化分析，当前研究仅是按照某种策略恢复受影响的虚拟网，忽略了虚拟网构建时的可靠性评估；(2)对于故障恢复策略，主备切换方式由于需要对链路进行备份，所以存在带宽资源消耗高的问题，而在线迁移方式由于需要依赖当前的资源状况进行路径选择，所以在特殊位置或空闲资源较少时存在迁移成功率低的问题。

为此，本文首先提出一种基于拓扑影响度的虚拟网初始映射算法，根据虚拟网的拓扑结构和映射位置，分析不同节点/链路对虚拟网可靠性的影响，评估虚拟网映射结果的可靠性，并使得虚拟网在初始映射和迁移过程中尽量避开对自身可靠性影响较大的节点/链路。其次，为了改进在线迁移方式成功率低的不足，权衡迁移代价与迁移成功率，本文提出一种回溯迁移机制，该机制从失效位置出发通过回溯方式逐渐增加迁移步数，直至迁移成功或达到迁移上限。该机制所产生的迁移步数不限于失效位置，也非整个虚拟网，而是介于二者之间，从而兼顾迁移成功率和迁移代价。

本文试图通过基于拓扑影响度的映射算法降低故障时的虚拟网受影响范围，并通过回溯迁移机制提高虚拟网的迁移成功率，从而保证虚拟网的可靠性。

2　问题分析

2.1 网络模型

虚拟网映射问题的网络模型描述如下：

底层网络：将底层网络用无向图$G^s=(N^s,L^s,A^n,A^l)$表示，其中N^s表示底层网络的节点集合，L^s表示底层网络的链路集合，A^n和A^l分别表示底层网络的节点属性和链路属性，本文只考虑节点的CPU属性和链路的带宽属性。

虚拟网请求：定义第 i 个到达的虚拟网请求为$\mathrm{VNR}_i=(G_i^v,T_a,T_d)$，其中无向图$G_i^v=(N_i^v,L_i^v,C_i^n,C_i^l)$表示虚拟网拓扑结构，$N_i^v$和$L_i^v$分别为虚拟网的虚拟节点集合和虚拟链路集合，$C_i^n$和$C_i^l$分别为虚拟网的节点约束和链路约束；$T_a$和$T_d$表示虚拟网运行的起止时间。下文将底层网络的节点/链路统称为底层拓扑单元，虚拟节点/链路统称为虚拟拓扑单元。

虚拟网映射：虚拟网映射方案可以描述为从G_i^v到G^s的子图G^{sub}的函数关系，且满足约束条件C_i^n和C_i^l，表达式如下：

$$f:G_i^v(N_i^v,L_i^v)\mapsto G^{\mathrm{sub}}(N^{\mathrm{sub}},P^{\mathrm{sub}}) \tag{1}$$

其中$N^{\mathrm{sub}}\subset N^s$，$P^{\mathrm{sub}}\subset P^s$，$P^{\mathrm{sub}}$为$G^{\mathrm{sub}}$的路径集合，$P^s$为$G^s$的路径集合。映射方案$f$包括节点映射$f_N$和链路映射$f_L$两部分，分别表示为$f_N:N_i^v\mapsto N^{\mathrm{sub}}$和$f_L:L_i^v\mapsto P^{\mathrm{sub}}$。如图 1(a)所示，虚拟节点 a 和 b 分别映射到底层节点 u 和 w 之上，虚拟链路 l 映射到路径$\{uv, vw\}$之上。映射成功后，底层网络将为虚拟网请求分配资源，剩余属性值如图 1(b)所示。

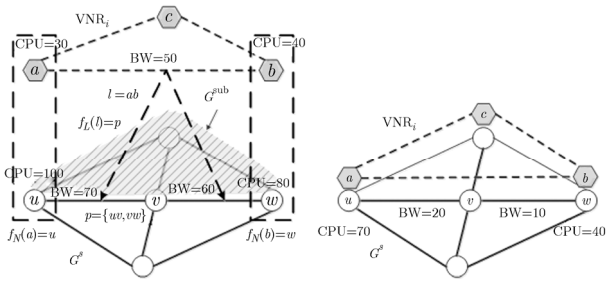

图 1 虚拟网映射模型解析

2.2 可靠性分析

定义 1　虚拟网可靠系数：底层拓扑单元失效时，虚拟网的服务提供能力依然有效的概率。根据文献[13]可知，70%的链路失效是单独出现的，所以本文只考虑底层网络单节点或单链路失效的情况。基于上述假设，虚拟网请求 VNR 的可靠系数γ如式(2)所示。

$$\begin{aligned}\gamma&=1-P(E,F)=1-P(E)P(F\mid E)\\&=1-P(E)\sum_{k=1}^{n+m}P(F_k\mid E)\\&=1-\frac{n+m}{N+M}\left\{\sum_{i=1}^{n}\left[\frac{1}{n+m}(1-(1-\rho)^{x_i})\right]\right.\\&\quad\left.+\sum_{j=1}^{m}\left[\frac{1}{n+m}(1-(1-\rho)^{y_j})\right]\right\}\\&=1-\frac{1}{N+M}\left[\sum_{i=1}^{n}(1-(1-\rho)^{x_i})\right.\\&\quad\left.+\sum_{j=1}^{m}(1-(1-\rho)^{y_j})\right]\end{aligned} \tag{2}$$

式(2)中，事件 E 表示虚拟网请求 VNR 中出现拓扑单元失效，事件 F 表示 VNR 迁移失败，事件F_k表示虚拟网 VNR 的第 k 个拓扑单元失效且迁移失败；N 和 M 分别表示底层网络的节点数和链路数，n 和 m 分别表示 VNR 映射到的底层网络子图G^{sub}的节

点数和链路数；x_i 和 y_j 分别表示 G^{sub} 中第 i 个节点和第 j 条链路失效情况下需要迁移的 VNR 拓扑单元数；$\rho \in [0,1]$ 表示单个虚拟拓扑单元迁移失败的概率。

如果将式(2)中的 $(1-\rho)^{x_i}$ 和 $(1-\rho)^{y_j}$ 进行二项式展开，那么在 ρ 较小时有下式成立：

$$(1-\rho)^{x_i} \approx 1-\rho x_i,\ (1-\rho)^{y_j} \approx 1-\rho y_j$$

近似认为

$$1-\gamma \approx \frac{\rho}{N+M}\left(\sum_{i=1}^{n} x_i + \sum_{j=1}^{m} y_j\right) \tag{3}$$

$$1-\gamma \propto \rho\left(\sum_{i=1}^{n} x_i + \sum_{j=1}^{m} y_j\right) \tag{4}$$

式(4)表明 $1-\gamma$ 正比于以下两个因素：不同失效位置下需要迁移的虚拟拓扑单元总数 $\sum_{i=1}^{n} x_i + \sum_{j=1}^{m} y_j$ 和单个虚拟拓扑单元迁移失败的概率 ρ。基于此结论，本文在第 3 节提出相应的映射算法和迁移算法，分别降低上述因素的影响，从而提高虚拟网可靠系数 γ。

对于特定虚拟网请求，进一步描述虚拟拓扑单元的迁移比例，为此引入如下概念。

定义 2 节点影响度：若虚拟网请求 VNR 映射到底层网络子图 G^{sub}，则 G^{sub} 中的某一节点 n_i 失效时需要迁移的 VNR 拓扑单元比例，称之为节点 n_i 对 VNR 的影响度。如式(5)所示。

$$\pi(n_i) = \frac{x_i}{S} = \begin{cases} \left(\prod_{k=1}^{p}\left(1-\frac{s_k}{S}\right)\right)^{1/p}, & n_i \text{为割点} \\ \frac{\theta_e(n_i)+\theta_n(n_i)}{S}, & n_i \text{非割点} \end{cases} \tag{5}$$

定义 3 链路影响度：若虚拟网请求 VNR 映射到底层网络子图 G^{sub}，则 G^{sub} 中的某一链路 e_j 失效时需要迁移的 VNR 拓扑单元比例，称之为链路 e_j 对 VNR 的影响度。如式(6)所示。

$$\pi(e_j) = \frac{y_j}{S} = \begin{cases} \sqrt{\left(1-\frac{s_1}{S}\right)\left(1-\frac{s_2}{S}\right)}, & e_j \text{为割边} \\ \frac{\varphi_e(e_j)}{S}, & e_j \text{非割边} \end{cases} \tag{6}$$

其中 S 表示 VNR 的虚拟拓扑单元总数，x_i 和 y_j 的含义同式(2)，p 为割点失效时将底层网络分成的子图个数，$s_1, s_2, \cdots, s_p$ 表示每个子图中包含的虚拟拓扑单元个数，由于存在被破坏的拓扑单元，所以 $s_1+s_2+\cdots+s_p < S$。另外，$\theta_n(n_i)$ 表示节点 n_i 承载的 VNR 虚拟节点数，$\theta_e(n_i)$ 表示经过 n_i 的 VNR 虚拟链路数，$\varphi_e(e_j)$ 表示链路 e_j 承载的 VNR 虚拟链路数。

当失效节点 n_i 为普通节点(非割点)时，底层网络的连通性未被破坏，所以 VNR 的影响范围只限于 n_i 承载的虚拟节点数 $\theta_n(n_i)$ 和经过的虚拟链路数 $\theta_e(n_i)$，$\theta_n(n_i)$ 的取值为 0 或 1(即不出现多个虚拟节点映射到同一底层节点)。同理，当失效链路 e_j 为普通链路(非割边)时，影响范围只限于 e_j 承载的虚拟链路数 $\varphi_e(e_j)$。当失效节点 n_i 为割点时，底层网络被分割为若干互不连通的子图，虚拟拓扑单元分散在不同子图中，由于 VNR 是连通图，所以唯一途径是将所有子图的虚拟拓扑单元迁移到同一子图中，若迁移到第 k 个子图，则迁移比例为 $1-s_k/S$，式(5)取所有方式下迁移比例的几何平均值。同理，当失效链路 e_j 为割边时，底层网络被分割为两个子图，取两种方式下迁移比例的几何平均值。

2.3 评价指标

定义 4 虚拟网请求接受率：长期运行情况下映射成功的虚拟网请求所占的比例，如式(7)所示。

$$r_{\text{accept}} = \lim_{T\to+\infty} \sum_{i=1}^{n_a(T)} \delta(\text{VNR}_i) \Big/ n_a(T) \tag{7}$$

其中 $n_a(T)$ 表示 0 至 T 时刻到达的虚拟网请求数，$\delta(\text{VNR}_i)$ 表示虚拟网请求 VNR_i 是否映射成功，若成功则值为 1，反之为 0。

定义 5 有效承载率：长期运行情况下底层网络所承载的虚拟网中能够有效提供服务的虚拟网所占比例的平均值，如式(8)所示。

$$r_{\text{valid}} = \lim_{T\to\infty} \sum_{i=1}^{e(T)} \frac{n_{\text{valid}}(t_i)}{n_a(t_i)-n_d(t_i)} \Big/ e(T) \tag{8}$$

其中 $e(T)$ 表示 0 至 T 时刻底层网络承载状态发生变化(包括拓扑、请求变化)的次数，T_i $(i=1,2,\cdots,e(T))$ 为发生变化的时刻，$n_a(T_i)$ 和 $n_d(T_i)$ 分别表示 0 至 T_i 时刻到达和离开的虚拟网请求数，$n_{\text{valid}}(T_i)$ 表示在 T_i 时刻能够有效提供服务的虚拟网数。

定义 6 迁移成功率：长期运行情况下所有受影响的虚拟网中能够通过迁移恢复服务提供能力的虚拟网所占比例，如式(9)所示。

$$r_{\text{mig}} = \lim_{T\to\infty} \sum_{i=1}^{m(T)} n_{\text{succ}}(t_i) \Big/ \sum_{i=1}^{m(T)} n_{\text{mig}}(t_i) \tag{9}$$

其中 $m(T)$ 表示 0 至 T 时刻底层网络发生的故障次数，t_i $(i=1,2,\cdots,m(T))$ 为发生故障的时刻，$n_{\text{mig}}(t_i)$ 表示 t_i 时刻发生故障时受影响的虚拟网数，$n_{\text{succ}}(t_i)$ 表示 t_i 时刻发生故障时迁移成功的虚拟网数。

3 虚拟网可靠性映射

3.1 基于拓扑影响度的映射算法

本文提出的映射算法以拓扑影响度(节点和链路影响度)指标为前提，同时兼顾了资源因素。有关资源因素的概念如下。

定义 7　节点空闲度：节点空闲 CPU 资源与相邻链路空闲带宽资源和的乘积，经归一化处理后称之为节点空闲度。即

$$\mathrm{FR}(n_s)=\frac{H(n_s)-\min\limits_{u\in N^s}H(u)}{\max\limits_{u\in N^s}H(u)-\min\limits_{u\in N^s}H(u)+\varepsilon} \tag{10}$$

$$H(u)=\mathrm{CPU}(u)\sum_{l\in L(u)}\mathrm{BW}(l) \tag{11}$$

定义 8　链路空闲度：链路空闲带宽资源经过归一化处理后称之为链路空闲度。即

$$\mathrm{FR}(e_s)=\frac{\mathrm{BW}(e_s)-\min\limits_{l\in L^s}\mathrm{BW}(l)}{\max\limits_{l\in L^s}\mathrm{BW}(l)-\min\limits_{l\in L^s}\mathrm{BW}(l)+\varepsilon} \tag{12}$$

上述定义中，N^s 和 L^s 分别为底层网络节点集合和链路集合，$\mathrm{CPU}(u)$ 为节点 u 的空闲 CPU 资源，$\mathrm{BW}(l)$ 为链路 l 的空闲带宽资源，$L(u)$ 为节点 u 的相邻链路集合，ε 为极小正数。

整合式(5)，式(6)与式(10)，式(12)，得到虚拟网映射的目标函数，如式(13)所示。其中 N^{sub} 和 L^{sub} 分别为映射结果 G^{sub} 的节点集合和链路集合。

$$C(G^{\mathrm{sub}})=\sum_{n_s\in N^{\mathrm{sub}}}(1-\mathrm{FR}(n_s))\cdot\pi(n_s)+\sum_{e_s\in L^{\mathrm{sub}}}(1-\mathrm{FR}(e_s))\cdot\pi(e_s) \tag{13}$$

为使目标函数最小，本文提出了基于拓扑影响度的虚拟网映射算法，该算法的特点是节点和链路交替映射，并允许递归回溯，从而避免了传统节点优先映射方式的局部最优问题。算法的基本思路为：通过构造虚拟网拓扑的宽度优先搜索树，确定虚拟节点的映射顺序；每次先进行节点映射，然后检查与该节点相邻的虚拟链路另一端点是否完成映射，若完成则对该链路进行映射；节点映射和链路映射均考虑资源因素和拓扑影响度因素，使得式(13)最小；通过设置搜索空间的上限 MAX_SPACE，将算法的时间复杂度限制为多项式级别。算法的具体流程如表 1 所示。

表 1 基于拓扑影响度的虚拟网映射算法

算法 1　VNM-TID (Virtual Network Mapping algorithm based on Topology Impact Degree)
输入：$G^s(N^s,L^s,A^n,A^l)$，$G^v(N^v,L^v,C^n,C^l)$
输出：f，$G^{\mathrm{sub}}(N^{\mathrm{sub}},L^{\mathrm{sub}})$
(1)构造虚拟网拓扑 G^v 的宽度优先搜索树 T：以资源约束最大的虚拟节点为根，且 T 中每层虚拟节点按照资源约束大小降序排列；
(2)得到 T 的虚拟节点映射顺序：$n_v^i, i=1,2,\cdots,\lvert N^v\rvert$；
(3)map_cnt = 0;　//搜索计数
(4)if SearchMap(n_v^1 ,f, G^{sub} ,map_cnt)== MAP_SUCCESS then return f,G^{sub}; end if
(5)return NULL;

子程序 SearchMap( )负责搜索满足资源约束且使式(13)尽可能小的映射方案如表 2 所示。

表 2 子程序 SearchMap( )的映射方案

PROCEDURE SearchMap(n_v^i, f, G^{sub}, map_cnt):
(1) if $i==\lvert N^v\rvert+1$ then return MAP_SUCCESS; end if
(2) 得到满足 n_v^i 资源约束的底层网络候选节点集合 $S=\{u\mid \mathrm{CPU}(u)\ge C^n(n_v^i), u\in N^s\setminus N^{\mathrm{sub}}\}$；
(3) if $S==\varnothing$ || map_cnt > MAX_SPACE then return MAP_FAILED; end if
(4) 对于 $\forall u\in S$，假设 $f_N(n_v^i)=u$，检查是否存在割点 c 分割了 u 与 $v(v\in N^{\mathrm{sub}})$。若存在，则计算 $G^{\mathrm{sub}}\cup\{u\}\cup\{c\}$ 下的式(13)的值；否则计算 $G^{\mathrm{sub}}\cup\{u\}$ 下的值。根据结果对节点进行降序排列；
(5) $f_{\mathrm{old}}\leftarrow f$; $G_{\mathrm{old}}\leftarrow G^{\mathrm{sub}}$；　//保存当前映射方案
(6) for each $u\in S$ do
(7)　map_cnt++;
(8)　$f_N(n_v^i)=u$; $G^{\mathrm{sub}}=G^{\mathrm{sub}}\cup\{u\}$；　// f_N 为 f 的节点映射
(9)　for each $e_v=(n_v^i,n_v^j)\in\{(n_v^i,n_v^k)\mid(n_v^i,n_v^k)\in L_v,\ k<i,\ n_v^k\in N_v\}$ do
(10)　使用 k-shortest paths 算法得到 u 和 $f_N(n_v^j)$ 间的 k 条最短路径 $P=\{p_1,p_2,\cdots,p_k\}$;
(11)　选取集合 P 中满足 e_v 链路资源约束，且使式(13)最小的路径 p_{best}；
(12)　if p_{best} 不存在 then goto 18; end if
(13)　$f_L(e_v)=p_{\mathrm{best}}$; $G^{\mathrm{sub}}=G^{\mathrm{sub}}\cup p_{\mathrm{best}}$；// f_L 为 f 的链路映射
(14)　end for
(15)　if SearchMap(n_v^{i+1}, f, G^{sub}, map_cnt)==MAP_SUCCESS then
(16)　return MAP_SUCCESS;
(17)　end if
(18)　$f\leftarrow f_{\mathrm{old}}$; $G^{\mathrm{sub}}\leftarrow G_{\mathrm{old}}$；//在搜索新的候选节点之前，恢复原有映射方案
(19)　if map_cnt > MAX_SPACE then
(20)　return MAP_FAILED;
(21)　end if
(22) end for
(23) return MAP_FAILED;

由于割点/边的位置可以预计算，所以子程序 SearchMap( )求解式(13)的复杂度为多项式级别；在求解 k 条最短路径时，采用 k-shortest paths 算法[14]，其时间复杂度为 $O(N\log N+kN+M)$，其中 N 和 M 分别为底层网络的节点个数和链路个数。此外，求解式(13)和 k-shortest paths 算法的次数受 MAX_SPACE 限制，所以 VNM-TID 算法在最坏情况下的复杂度仍为多项式级别。

3.2 基于回溯机制的迁移算法

为权衡迁移成功率与迁移代价，本文提出一种

基于回溯机制的迁移算法。该算法的回溯迁移过程由子程序 BacktrackMig( )完成，基本思路为：首先递归至失效位置，进行直接迁移；若迁移失败，则按照 VNM-TID 算法中的虚拟节点映射顺序，回溯至上一节点继续迁移；失效链路的迁移发生在顺序靠后的端节点的迁移处理中；依此类推，直至迁移成功或迁移步数超出上限 MAX_MIG。算法的具体流程如表 3 所示。

表 3 基于回溯机制的迁移算法

算法 2 MA-Back(Migration Algorithm based on Backtracking)
输入：f, $G^{sub}(N^{sub}, L^{sub})$
(1)构造虚拟网拓扑 G^v 的宽度优先搜索树 T：以资源约束最大的虚拟节点为根，且 T 中每层虚拟节点按照资源约束大小降序排列；
(2)得到 T 的虚拟节点映射顺序： $n_v^i, i = 1, 2, \cdots, \mid N^v \mid$;
(3)mig_cnt = 0;
(4)if BacktrackMig(n_v^1, f, G^{sub}, mig_cnt)== MAP_SUCCESS then return f, G^{sub}; end if
(5)return NULL;

子程序 BacktrackMig( )处理流程如表 4 所示。子程序 BacktrackMig( )的处理流程大致分为两类：对于正常节点或能够成功进行相邻链路迁移的节点，直接递归到下一节点；对于失效节点或相邻链路迁移失败的节点，进行节点迁移。节点迁移过程与 SearchMap( )子程序类似，具体见步骤 14。MA-Back 算法的时间复杂度同样为多项式级别，分析过程与 VNM-TID 算法类似。

表 4 子程序 BacktrackMig( )处理流程

PROCEDURE BacktrackMig(n_v^i, f, G^{sub}, mig_cnt):
(1) if n_v^i 映射失效 then // 节点 $f(n_v^i)$ 失效
(2) goto 15;
(3) else if n_v^i 的相邻边映射失效 then //链路失效或另一端点发生迁移
(4) for each $e_v = (n_v^i, n_v^j) \in \{(n_v^i, n_v^k) \mid (n_v^i, n_v^k) \in L_v, k < i, n_v^k \in N_v\}$ do
(5) if e_v 映射正常 then continue; end if
(6) mig_cnt++;
(7) 使用 k-shortest paths 算法得到 $f_N(n_v^i)$ 和 $f_N(n_v^j)$ 间的 k 条最短路径 $P=\{p_1, p_2, \cdots, p_k\}$;
(8) 选取集合 P 中满足 e_v 链路资源约束，且使式(13)最小的路径 p_{best};
(9) if p_{best} 不存在 then goto 15; end if
(10) $G^{sub} = (G^{sub} \setminus f_L(e_v)) \cup p_{best}$; $f_L(e_v) = p_{best}$; // 链路迁移
(11) end for
(12) end if
(13)if BacktrackMig(n_v^{i+1}, f, G^{sub}, mig_cnt)==MAP_SUCCESS then return MAP_SUCCESS;
(14) end if
(15) 后续步骤同 SearchMap(n_v^i, f, G^{sub}, mig_cnt)，只需将算法 1 子程序中的第(15)函数调用修改为 BacktrackMig(n_v^{i+1}, f, G^{sub}, mig_cnt)，将算法 1 子程序中第(3)，第(19)中的 MAX_SPACE 修改为 MAX_MIG;

4 实验评估

4.1 实验设置

实验环境为 Intel(R) Core(TM) i7 CPU 2.67 GHz, RAM 2 G 的 PC 上，通过 C++编程模拟底层网络的拓扑变化以及虚拟网的映射与迁移。使用 GT-ITM[15]工具生成具有 100 个节点的底层网络初始拓扑，以及 2000 个虚拟网请求。底层网络的节点和链路资源取值在[50, 100]内均匀分布，节点连接概率为 0.2。虚拟网请求的节点个数在[2, 10]内均匀分布，节点和链路资源请求在[0, 50]内均匀分布，节点连接概率为 0.5。仿真过程中，底层网络拓扑结构动态变化，共随机出现 50 次节点增加事件，50 次链路增加事件，节点/链路失效事件服从强度为平均 100 个时间单位发生 4 次的泊松过程，且节点和链路失效事件各占 50%，故障恢复时间服从期望为 50 个时间单位的指数分布。虚拟网请求的到达过程服从强度为平均 100 个时间单位到达 4 个请求的泊松过程，生命周期服从期望为 1000 个时间单位的指数分布。此外，设置 MAX_SPACE 和 MAX_MIG 为 $3n$(n 为虚拟网请求的节点个数)，设置 k-shortest paths 算法中的 k 为 10。

4.2 性能分析

参与比较的算法特点如表 5 所示。

根据实验设置，首先得到不同算法组合下的虚拟网请求接受率，如图 2 所示。由实验结果可知，虚拟网请求接受率呈现递减趋势，并最终达到平稳状态，其中 VNM-TID & MA-Back 算法的虚拟网请

表 5 参与比较的算法特点

名称	初始映射方式	迁移方式
VNM-TID & MA-Back	基于拓扑影响度的 VNM-TID 算法	基于回溯机制的 MA-Back 算法
BAL-COST	节点/链路压力均衡算法[3]	迁移代价最小算法[10]
BAL-BAL	节点/链路压力均衡算法[3]	压力均衡迁移算法[3]
BAL-no	节点/链路压力均衡算法[3]	无迁移
SP-SP	基于贪心策略的节点映射，基于最短路径的链路映射算法[12]	最短路径迁移算法[12]

求接受率最高，稳定在 86.35%, BAL-COST, BAL-BAL 和 BAL-no 算法的请求接受率比较接近，分别稳定在 80.95%, 79.58%和 77.71%，而 SP-SP 算法的请求接受率最低，稳定在 51.88%。虚拟网请求接受率主要与底层网络资源均衡性有关，且故障迁移相比于映射过程对底层网络资源均衡性的影响小，所以迁移算法对虚拟网请求接受率的提高程度也较为有限。基于上述原因，同样采用节点/链路压力均衡映射的 BAL-COST, BAL-BAL 和 BAL-no 算法的请求接受率差距较小，SP-SP 算法由于不考虑资源均衡性，所以请求接受率最低，而 VNM-TID & MA-Back 算法采用节点、链路交替映射方式，比其他算法采用的节点优先映射方式更灵活地调整底层网络资源均衡性，所以请求接受率最高。

虚拟网迁移成功率变化曲线如图 3 所示，其中 BAL-no 算法不进行迁移，所以不予比较。由实验结果可知，SP-SP 算法所获得的迁移成功率处于最低，VNM-TID & MA-Back, BAL-BAL 和 BAL-COST 算法的迁移成功率依次降低。虚拟网迁移成功率主要与迁移算法的迁移步数和底层网络的资源剩余情况有关。VNM-TID & MA-Back 算法采用回溯机制进行迁移，相比 BAL-BAL 和 BAL-COST 算法的一步迁移方式，具有步数多的优势，所以迁移成功率高；BAL-BAL 算法基于底层网络资源的均衡性进行迁移，可以使迁移后的剩余资源分布均衡，避免了 BAL-COST 算法可能出现的部分节点/链路资源拥塞, 从而使迁移成功率高于 BAL-COST 算法；SP-SP 算法由于不考虑资源均衡性，且迁移步数与 BAL-BAL 和 BAL-COST 算法相同, 所以迁移成功率最低。

底层网络有效承载率变化曲线如图 4 所示。由实验结果可知，VNM-TID & MA-Back 算法得到的底层网络有效承载率最高，基本保持在 99.20%以上；BAL-no 算法的有效承载率最低，一度降至 86.66%; BAL-BAL 和 BAL-COST 算法的有效承载率比较接近，且高于 SP-SP 算法。底层网络有效承载率主要与虚拟网请求接受率和迁移成功率有关，VNM-TID & MA-Back 算法由于具有最高的虚拟网请求接受率和迁移成功率，所以有效承载率最高；BAL-no 算法的请求接受率较低，且迁移成功率为 0, 所以有效承载率最低；BAL-BAL 和 BAL-COST 算法的请求接受率和迁移成功率均差距不大，所以二者的有效承载率也比较接近；SP-SP 算法由于请求接受率最低，且迁移成功率最低，所以有效承载率仅优于 BAL-no 算法，而低于其他 3 种算法。

最后，不同算法得到的虚拟网平均可靠系数($\bar{\gamma}$)变化曲线如图 5 所示。$\bar{\gamma}$ 是指 T 时刻底层网络中所有处于运行状态的虚拟网的可靠系数平均值，如式(14)所示。

$$\bar{\gamma}(T)=\sum_{i=1}^{n_{\text{alive}}(T)}\gamma_i(T)\Big/n_{\text{alive}}(T)$$
$$=\sum_{i=1}^{n_{\text{alive}}(T)}\left[1-\frac{|G_i^{\text{sub}}(T)|}{|G^s(T)|}(1-r_{\text{mig}}(T))\right]\Big/n_{\text{alive}}(T) \quad (14)$$

其中 $\gamma_i(T)$ 为单个虚拟网的可靠系数(见式(2)), 为了便于讨论，令式(2)中的虚拟网迁移失败概率 $P(F\mid E)\approx 1-r_{\text{mig}}(T)$，其中 $r_{\text{mig}}(T)$ 表示 T 时刻虚拟网的迁移成功率(如图 3 所示)。此外，$n_{\text{alive}}(T)$ 表示 T 时刻底层网络中处于运行状态的虚拟网个数，$|G_i^{\text{sub}}(T)|$ 表示 T 时刻虚拟网 i 所映射的底层子网节点数和链路数之和，$|G^s(T)|$ 表示 T 时刻底层网络节点数和链路数之和。由图 5 可知，VNM-TID & MA-Back 算法的 $\bar{\gamma}$ 值在 $T>4271$ 时处于最高，且始终保持在 99.20%以上；BAL-COST 和 BAL-BAL 算法的 $\bar{\gamma}$ 值比较接近，保持在 98.40%以上；SP-SP 算法的 $\bar{\gamma}$ 值最低，且存在较大波动性。因此，VNM-TID & MA-Back 算法的 $\bar{\gamma}$ 值最优, 结合迁移成功率(图 3)和有效承载率(图 4)可以印证：利用虚拟网可靠系数量化分析可靠性并提出优化算法具有合理性和有效性。

综上所述，本文对虚拟网的可靠性量化分析是有效的, 且提出的 VNM-TID & MA-Back 在虚拟网请求接受率、迁移成功率和有效承载率 3 项指标上

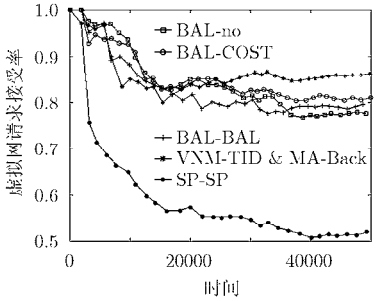

图 2 虚拟网请求接受率

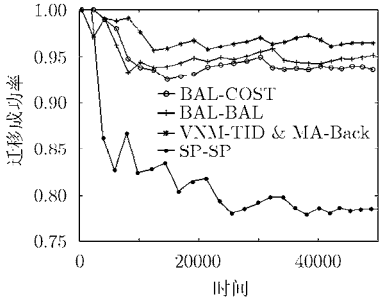

图 3 虚拟网迁移成功率

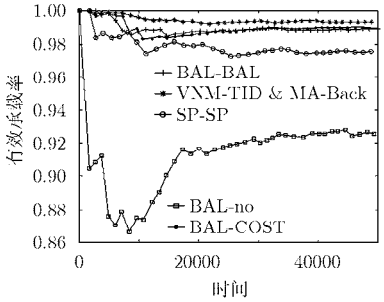

图 4 底层网络有效承载率

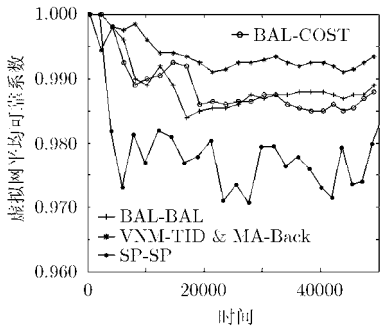

图 5 虚拟网平均可靠系数($\bar{\gamma}$)

具有明显的优势，有利于提高虚拟网服务提供的可靠性。

5 结束语

本文针对底层网络拓扑动态变化下的虚拟网可靠性问题进行了论述，量化分析了虚拟网的可靠性，并归结出影响虚拟网可靠性的因素。针对这些因素，从映射算法和迁移算法两方面出发，分别提出基于拓扑影响度的 VNM-TID 算法和基于回溯机制的 MA-Back 算法。VNM-TID & MA-Back 算法在虚拟网请求接受率、迁移成功率和有效承载率 3 项指标上具有优势，是一种有效的虚拟网可靠性映射方案。

参 考 文 献

[1] Turner J and Taylor D. Diversifying the Internet[C]. Proceedings of IEEE Conference on Global Telecommunications, St. Louis, USA, 2005: 755-760.

[2] Anderson T, Peterson L, Shenker S, *et al*.. Overcoming the internet impasse through virtualization[J]. *IEEE Computer Magazine*, 2005, 38(4): 34-41.

[3] Zhu Y and Ammar M. Algorithms for assigning substrate network resources to virtual network components[C]. Proceedings of IEEE INFOCOM, Barcelona, Catalunya, Spain, 2006: 1-12.

[4] Nabeel B, Chowdhury N M, and Boutaba R. Topology-awareness and reoptimization mechanism for virtual network embedding[C]. Proceedings of the 9th International Networking Conference, Chennai, India, 2010: 27-39.

[5] Cheng Xiang, Su Sen, Zhang Zhong-bao, *et al*.. Virtual network embedding through topology-aware node ranking[J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(2): 39-47.

[6] Rahman M, Issam A, and Boutaba R. Survivable virtual network embedding[C]. Proceedings of the 9th IFIP TC 6 International Conference on Networking, Chennai, India, 2010: 40-52.

[7] Chen Yang, Li Jian-xin, Wo Tian-yu, *et al*.. Resilient virtual network service provision in network virtualization environments[C]. Proceedings of 2010 IEEE 16th International Conference on Parallel and Distributed Systems, Shanghai, China, 2010: 51-58.

[8] Yu Hong-fang, Anand V, Qiao Chun-ming, *et al*.. Enhancing virtual infrastructure to survive facility node failures[C]. Proceedings of Optical Fiber Communication Conference (OFC), Los Angeles, California, USA, 2011: 1-3.

[9] Yu Hong-fang, Qiao Chun-ming, Anand V, *et al*.. Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures[C]. Proceedings of IEEE Global Telecommunications Conference (Globecom), Miami, Florida, USA, 2010: 1-6.

[10] Cai Zhi-ping, Liu Fang, Xiao Nong, *et al*.. Virtual network embedding for evolving networks[C]. Proceedings of IEEE Global Telecommunications Conference (Globecom), Miami, Florida, USA, 2010: 1-5.

[11] Houidi I, Louati W, Zeghlache D, *et al*.. Adaptive virtual network provisioning[C]. Proceedings of the 2nd ACM workshop on Virtualized Infrastructure Systems and Architectures, VISA'10, New York, USA, 2010: 41–48.

[12] 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.

[13] Markopoulou A, Iannaccone G, Bhattacharyya S, *et al*.. Characterization of failures in an IP backbone[C]. Proceedings of IEEE INFOCOM, Hong Kong, China, 2004: 2307-2317.

[14] Eppstein D. Finding the k shortest paths[C]. Proceedings of IEEE Symposium on Foundations of Computer Science, Santa Fe, New Mexico, 1994: 154-165.

[15] Zegura E W, Calvert K L, and Bhattacharjee S. How to model an internetwork[C]. Proceedings of IEEE INFOCOM, San Francisco, California, USA, 1996: 594-602.

王志明： 男，1986 年生，博士生，研究方向为网络虚拟化、下一代互联网.

汪斌强： 男，1963 年生，教授，博士生导师，主要研究领域为宽带信息网络.

王保进： 男，1974 年生，讲师，主要研究领域为网络管控、嵌入式系统.