

# 具有高可靠特征的无线虚拟网络映射方法研究

高航航, 赵尚弘, 王翔, 彭聪

空军工程大学 信息与导航学院, 西安 710077

**摘要:**针对虚拟网络映射过程中链路资源受限和网络生存性问题, 分别提出一种基于节点扩展资源的节点映射方案和一种虚拟网络重映射方案。为克服节点映射后链路映射阶段的资源不足问题, 通过将节点扩展资源大的节点作为承载节点以确保链路映射的可靠性和高效性, 链路映射阶段采用一种基于最小链路代价的映射方案以获得高可靠映射, 在此基础上针对网络故障采取一种重映射策略, 将失效节点迅速重映射至候选节点集中以确保虚拟网络服务的连续性。通过仿真对几种算法性能进行对比, 结果表明所提算法在虚拟网络映射成功率、虚拟网络恢复成功率和资源负载利用率等方面均取得较好结果。

**关键词:**虚拟网络映射; 节点扩展资源; 链路代价; 网络故障; 重映射

**文献标志码:**A **中图分类号:**TP393 **doi:**10.3778/j.issn.1002-8331.1902-0225

高航航, 赵尚弘, 王翔, 等. 具有高可靠特征的无线虚拟网络映射方法研究. 计算机工程与应用, 2020, 56(12): 73-80.

GAO Hanghang, ZHAO Shanghong, WANG Xiang, et al. Research on wireless virtual network mapping method with high reliability features. Computer Engineering and Applications, 2020, 56(12): 73-80.

## Research on Wireless Virtual Network Mapping Method with High Reliability Features

GAO Hanghang, ZHAO Shanghong, WANG Xiang, PENG Cong

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

**Abstract:** Aiming at the problem of link resource limitation and network survivability in virtual network mapping process, a node mapping scheme based on node extended resources and a virtual network remapping scheme are proposed respectively. In order to solve the problem of insufficient resources in the link mapping phase after node mapping, this article selects a physical node with a large node expansion resource as the bearer node to ensure the reliability and efficiency of the link mapping, and the link mapping phase adopts a mapping scheme based on the minimum link cost to obtain a highly reliable mapping. Based on this, a remapping strategy is adopted for network failure, and the failed nodes are quickly remapped to the candidate node set to ensure the continuity of the virtual network service. The performance of several algorithms are compared by simulation and the results show that the proposed algorithm achieves good results in virtual network mapping success rate, virtual network recovery success rate and resource load utilization.

**Key words:** virtual network mapping; node extension resources; link cost; network failure; remapping

## 1 引言

网络虚拟化<sup>[1-2]</sup>被认为是未来网络中的一项关键技术, 是一种应对当前互联网僵化问题的重要方法, 其有效地增强了网络的灵活性, 可以使不同的虚拟网络请求 (Virtual Network Request, VNR) 共存于同一个底层物理网络中, 共享底层物理网络资源, 并且这些虚拟网络

请求之间相互隔离, 在安全性方面也有一定的保证。网络虚拟化改变了传统的网络服务与资源紧密耦合的特性, 实现了网络服务与资源灵活且有效的利用。虚拟网络映射<sup>[3-5]</sup>作为网络虚拟化当中的一项关键技术, 是指将底层物理网络资源分配给虚拟网络中带有约束条件的虚拟节点以及虚拟链路的过程, 目前已被证明是NP难

**基金项目:**国家自然科学基金(No.91638101, No.61571461)。

**作者简介:**高航航(1994—), 男, 硕士研究生, 研究领域为航空信息网络, E-mail: gaohh321@126.com; 赵尚弘(1964—), 男, 博士, 教授, 研究领域为空天信息网络、空间光通信技术; 王翔(1984—), 男, 博士, 讲师, 研究领域为空间信息网络、软件定义网络; 彭聪(1995—), 男, 博士研究生, 研究领域为空间光电混合交换技术。

**收稿日期:**2019-02-28 **修回日期:**2019-07-10 **文章编号:**1002-8331(2020)12-0073-08

**CNKI网络出版:**2019-07-17, <http://kns.cnki.net/kcms/detail/11.2127.TP.20190717.1026.002.html>

问题<sup>[6-7]</sup>。虚拟网络映射目前主要面临着资源约束、准入控制、在线请求和拓扑多样等问题<sup>[8]</sup>,且由于VNR的到达时间、持续时间和服务需求不可预知,当底层网络拓扑较大时,采用传统方法求解将会产生庞大的搜索空间,难以在有限时间内满足VNR的各种需求。

传统虚拟网络映射算法将节点映射阶段与链路映射阶段完全分离,忽略了链路映射结果对物理节点排序的影响,导致高映射成本和虚拟节点在底层物理网络上的分布式分布,针对这些问题,文献[9]提出了一种基于节点属性和网络单纯形的双层虚拟网络映射算法Simplex-VNM,在节点映射层中采用一种考虑拓扑属性、全局网络资源和映射物理节点信息的新节点排序方法以此为链路映射层提供候选节点和节点映射成本,在链路映射阶段中,通过引入与传统最短路径算法相比具有较小时间复杂度的网络单纯形算法,以在节点映射层的结果下找到最优链路映射解决方案,最后根据两层结果以最小的总映射成本映射虚拟节点和相关链路,仿真表明本文算法在考虑节点属性和使用最短路径算法下的总映射成本和映射时间性能方面优于其他四种算法;当前大多数虚拟网络映射算法主要面向底层网络无故障环境下如何提高网络资源利用率、 $V_n$ 请求接受率等指标,如文献[10]在马尔科夫随机模型中采用 $K$ 分解技术将VNR分割为核心网络和边缘网络,并提出一种ViNEYard算法,算法通过协调节点映射和链路映射两个过程并优先映射生存时间短的VNR,结果表明所提算法有效地提高了 $V_n$ 接受率和网络收益;针对物理节点负载不均衡问题,文献[11]采用一种基于最小节点负载优先的映射算法进行求解,算法通过计算底层网络中各节点的负载强度,得到具有剩余资源大和负载小的物理节点作为承载节点进行映射,并采用 $k$ 最短路径法完成链路映射,结果表明文中算法在资源利用率、链路均衡度方面均得到有效提升;文献[12]为解决底层网络在映射中所存在的能耗过高问题,提出一种基于链路可分割的链路休眠算法,在网络低负载阶段中由于部分链路资源不能得到充分利用,从而将低资源利用率的虚拟链路重映射到高资源利用率的链路上,并将资源利用率低的链路进行休眠操作,结果表明该算法在提升链路资源利用率的同时也增大了 $V_n$ 映射成功率,但也带来了通信延迟等问题;文献[13]主要讨论了分布式和混合云环境中的网络资源映射问题,为解决数千个节点的大型虚拟和物理网络的复杂性和可扩展性,文中提出了一种依靠拓扑模式和二分匹配的高效启发式改进算法以获得最优映射方案,通过将虚拟资源分离到不同的物理节点并在相同的物理节点中同定位,最终算法采用虚拟分解技术和物理模式图的二分匹配法以在线性时间内解决该虚拟映射问题;在动态网络拓扑中,由于节点的移动性导致无线链路易发生中断,造成映射在该链路上的 $V_n$ 失效,使得相应的网络服务无法完成,此外,底层节点发生故障也会造成 $V_n$ 失效,文献[14]针对底层网络

单节点失效问题提出一种拓扑感知的节点映射策略和一种基于禁忌搜索算法的链路映射策略,仿真表明虚拟请求失效时所提算法在获得较高映射成功率的同时也能达到较高的网络收益和资源利用率;文献[15]针对底层多节点故障问题提出一种基于拓扑感知的虚拟映射方法,文中首先在 $V_n$ 到来之前为底层节点构建候选集,当 $V_n$ 到来时利用可恢复性的映射策略对 $V_n$ 的主要资源进行分配,将重要度较大的虚拟节点映射到可恢复性较好的物理节点上,当物理节点发生故障时采用利益驱动的重映射算法对受影响的 $V_n$ 进行恢复;文献[16]针对无线链路可靠性差提出一种基于链路可靠性的虚拟网络分配算法,算法允许同一虚拟请求中不同节点映射到同一物理节点,并利用 $Q$ 因子对稀疏拓扑分配时 $V_n$ 映射成功率低进行改进,仿真表明该算法可获得较高的 $V_n$ 构建成功率和资源利用率;文献[17]针对底层链路故障问题分别提出链路恢复和路径恢复两种方法,文中采用一种混合策略的启发式算法用来解决可生存性虚拟网络映射问题,算法预先计算每条物理链路的备份绕道路径,并对到达的 $V_n$ 执行节点映射和链路映射操作,当底层网络出现故障时,直接在备份绕道路径中为失效的链路重新路由,仿真表明该算法在 $V_n$ 接受率、网络收益和算法复杂度等方面要优于一般算法;针对可生存性的虚拟网络映射问题,文献[18]在文中将网络链路资源分为主要流量资源和次要流量资源以满足资源保护要求,并引入了全局资源容量这一指标,该指标可用于测量节点映射容量以改善网络的负载平衡,最后提出一种基于全局资源容量的启发式生存虚拟网络映射算法GRC-SVNE,算法在节点映射阶段通过计算所有节点的映射容量,并选择部分节点作为虚拟网络映射的候选节点以提高映射成功率,之后使用Dijkstra算法进行链路映射,结果表明文中算法与传统GREEDY、R-ViNE和D-ViNE算法相比不仅实现了虚拟网络的可生存性,在网络负载均衡和收益方面也获得了较优解;文献[19]在计算容量约束和网络容量约束下针对单个云请求的可生存性映射问题提出一种VSMF启发式算法,文中算法共分为两个阶段,在第一阶段中将虚拟网络划分为一组星类,每个星类都使用一个虚拟节点作为中心,第二阶段中通过允许一个虚拟节点映射一个或多个物理节点以放宽虚拟节点映射约束,使得共享相同虚拟节点的虚拟链路可共享相同的物理节点,结果表明所提VSMF算法在小规模网络 and 大规模网络中均可获得比传统算法更低的成本和阻塞概率;文献[20]针对无线网络移动性提出一种近似算法BIRD-VNE,由于底层节点的移动性导致节点间的连接关系发生改变,使得已经被虚拟链路使用的物理路径资源不足或发生故障,文中算法可以有效捕获到节点移动和拓扑变化,并能够减小由于底层节点的移动性所导致的 $V_n$ 重映射所产生的开销,结果表明该算法在获得最佳虚拟映射收益的同时能够减少由于节点移动所造成的 $V_n$



迁移个数,有效地减小了网络拓扑移动性对虚拟网络迁移的影响。

上述研究内容主要针对底层节点或链路发生故障时VNR的可生存性映射问题,主要以虚拟节点或虚拟链路重映射为主,而未考虑底层网络中节点资源与链路资源间的关系。本文分别在节点映射和链路映射阶段引入节点扩展资源和物理链路代价这一概念,建立一种高可靠性的虚拟网络映射方案,并针对网络故障设计一种可恢复的重映射方案,结果表明所提算法能够获得较优结果,其性能要优于一般算法。

## 2 模型分析

### 2.1 网络模型描述

用加权无向图  $G^S=(N^S, E^S)$  表示底层物理网络拓扑,其中  $n^s \in N^S$  表示物理节点,其属性包括节点计算能力  $cpu(n^s)$ 、节点位置信息  $loc(n^s)$ ;  $l^s \in E^S$  表示物理链路,其属性包括带宽资源  $b(l^s)$ ,用  $P$  表示底层物理网络无环路径集合,对于路径  $p^s \in P^S$ ,其可用带宽资源  $b(p^s)$  大小由该路径所经过的各链路最小带宽资源所决定。底层物理网络承载多个虚拟请求,在一段时间内,新虚拟请求到达的同时也存在某虚拟请求完成释放物理网络资源,故底层物理网络中的资源是动态变化的,其数学表达式如下:

$$cpu(n^s) = \sum_{n^s \in N^S} cpu(n^s) - \sum_{n^v \rightarrow n^s} cpu(n^v) + \sum_{n^v \rightarrow n^s} cpu_{rel}(n^v) \quad (1)$$

$$b(l^s) = \sum_{l^s \in E^S} b(l^s) - \sum_{l^v \rightarrow l^s} b(l^v) + \sum_{l^v \rightarrow l^s} b_{rel}(l^v) \quad (2)$$

$$b(p^s) = \min_{l^s \in p^s} b(l^s) \quad (3)$$

其中,式(1)表示物理节点资源能力,式(2)和式(3)分别表示物理链路带宽和物理路径带宽能力,且路径带宽能力由该路径所经过各链路的最小带宽能力所决定。

用加权无向图  $G^V=(N^V, E^V, T)$  表示虚拟网络请求VNR,其中  $n^v \in N^V$  为虚拟网络节点,其属性包括节点计算能力需求  $cpu(n^v)$ 、位置需求  $loc(n^v)$ ;  $l^v \in E^V$  为虚拟网络链路,其属性包括带宽需求  $b(l^v)$ ,  $T$  表示该虚拟网络请求的生存时间。

### 2.2 虚拟映射约束条件

将节点间映射关系用矩阵  $F^N$  表示,其中行向量表示虚拟节点,列向量表示物理节点,当虚拟节点  $i$  映射到物理节点  $m$ ,即  $\forall n_i^v \rightarrow n_m^s, F^N(i, m)=1$ , 否则为0,节点资源约束条件如下:

$$cpu(n^s) \geq cpu(n^v) F^N(i, m) \quad (4)$$

$$\sum_i F^N(i, m) \leq 1 \quad (5)$$

$$\sum_m F^N(i, m) = 1 \quad (6)$$

其中,式(4)表示物理节点资源能力大于VNR节点资源请求能力;式(5)表示同一个虚拟请求中的不同节点不

能被映射到同一个物理节点上;式(6)表示若一个虚拟节点属于不同虚拟网络请求,则该节点只能被映射到同一物理节点上。

与节点映射矩阵类似,链路间的映射关系用矩阵  $F^L$  表示,其中行向量代表虚拟链路,列向量代表物理链路,当虚拟节点  $i$  和  $j$  分别映射到物理节点  $m$  和  $n$  上,  $(i, j)$  间的链路映射到  $(m, n)$  间的路径时,即  $\forall l_{ij}^v \rightarrow l_{mn}^s, F^L(ij, mn)=1$ , 否则为0。此外,虚拟链路  $l_{ij}^v$  映射的所有物理链路  $l_{mn}^s$  组成了物理节点  $m, n$  之间的一条无环路径  $p(m, n)$ , 链路资源约束条件如下:

$$b(p(i, j)) = \min_{l^s \in p(i, j)} b(l^s) \quad (7)$$

$$b(l^s) \geq b(p(m, n)) F^L(ij, mn) \quad (8)$$

其中,式(7)表示物理路径可用资源大小为该路径所经过的物理链路中最小可用资源值;式(8)表示虚拟链路资源需求小于物理路径所提供的可用带宽资源。在考虑链路资源约束的同时也需要考虑链路的连通性约束,如式(9)所示:

$$\forall i \in N^s, \forall l_{ij} \in E^v,$$

$$\sum_{l_{mn} \in E^s} F^L(l_{ij}, l_{mn}) - \sum_{l_{mn} \in E^s} F^L(l_{ji}, l_{mn}) = \begin{cases} 1, F^N(i, m) \\ -1, F^N(i, n) \\ 0, \text{其他} \end{cases} \quad (9)$$

若某物理链路包含虚拟链路的源节点,式(9)值为1时表示源节点流出流量为1、流入流量为0;若某物理链路包含虚拟链路的节点,式(9)值为-1时表示目的节点流出流量为0、流入流量为1;若某物理链路不包含虚拟链路的源节点和目的节点,则流入流量和流出流量平衡,且均为0。

### 2.3 物理节点故障分析

当某虚拟网络请求VNR到达后,底层物理网络判断是否接受该VNR,若当前条件不允许,则拒绝该VNR,不提供物理资源进行映射;若当前条件允许接受,则物理网络要从自身中选择满足虚拟网络需求的物理节点和链路,映射问题定义为  $F: (G^V \rightarrow G^{S*}, G^{S*} \subseteq G^S)$ , 包括节点映射和链路映射两方面。在节点映射中通常优先选择可用资源较多的物理节点作为映射节点,但与该节点相连的物理链路资源不一定满足虚拟链路映射所需求的资源,如图(1)所示,图1(a)中虚拟网络请求包含两个虚拟节点和一条虚拟链路,图中数字表示节点和链路资源需求,图1(b)所示为物理网络,图中数字为节点和链路可提供资源。一种节点映射方案为  $\{a \rightarrow C, b \rightarrow A\}$ , 但此时物理链路  $(A, C)$  间带宽资源不足以提供给虚拟链路  $(a, b)$ , 导致该虚拟链路  $(a, b)$  推迟映射,当其余虚拟请求完成后,其占用的物理链路资源将被释放掉,释放的资源再用于对该链路进行映射。

在链路映射中,每条物理链路的带宽资源可以被多个虚拟链路所共享,因而会存在多条虚拟链路被映射到同一条物理链路这种情况。假如此时某个物理节点发

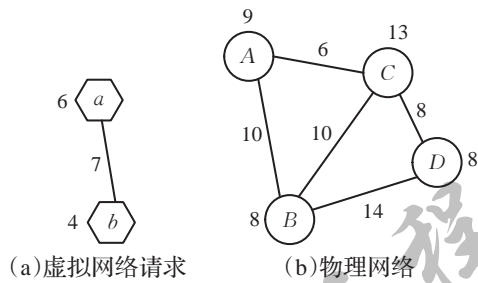


图1 链路映射请求推迟

生故障,则会导致与该物理节点相连的周边物理链路失效,使得这些物理链路所承载的虚拟链路随之失效,进而导致该虚拟网络不再连通。如图2所示为一种虚拟网络映射案例,图2(a)表示某虚拟网络请求VNR,图2(b)代表一种虚拟网络映射方案,其中节点映射方案为 $\{a \rightarrow A, b \rightarrow D, c \rightarrow C\}$ ,链路映射方案为 $\{(a, b) \rightarrow (A, D), (b, c) \rightarrow (D, C), (a, c) \rightarrow (A, D, C)\}$ ,假如当物理节点D失效,则导致与其相连的物理链路 $(A, D), (D, C)$ 均失效,进而使得映射在其上的虚拟链路 $(a, b), (a, c), (b, c)$ 全部失效,造成虚拟网络失去连通性;若使用图2(c)所示的映射方案,将虚拟链路 $(a, c)$ 映射到 $(A, B, E, C)$ ,当物理节点D失效时,虚拟网络中 $a, c$ 之间仍保持连通;如果采用图2(d)所示的映射方案将虚拟链路 $(a, c)$ 映射到 $(A, E, C)$ ,相比较方案2而言,虚拟网络中 $a, c$ 之间在保持连通的基础上占用了较小的开销,该方案可以取得较理想的结果。

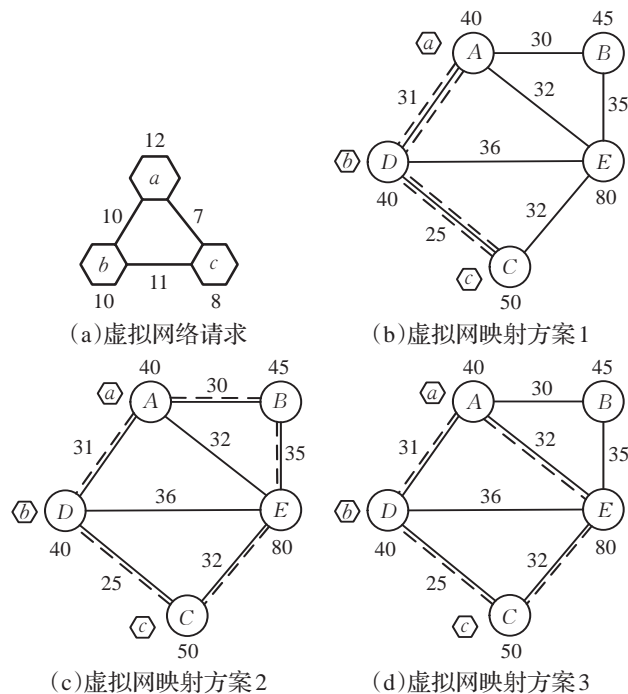


图2 虚拟网络映射案例

## 2.4 节点与链路指标分析

### (1) 物理节点扩展资源

在虚拟网络节点映射过程中通常优先选择可用资源较多的物理节点作为承载节点,再进行链路映射,然

而这种节点映射和链路映射相互独立的方法,会出现与已映射节点相连的物理链路剩余资源小于虚拟链路请求资源的情况,进而造成某链路映射推迟甚至无法完成链路映射。针对这一问题,文中引入物理节点扩展资源,如式(10)所示:

$$cpu_{ex}(n^s) = \mu_1 cpu(n^s) + \mu_2 \sum_{l^s \in E^s \oplus n^s} b(l^s) \quad (10)$$

其中,  $\mu_1$  和  $\mu_2$  为常数,表示节点和链路资源的重要度,  $E^s \oplus n^s$  表示与节点  $n^s$  相连的链路集。在映射节点时,选择物理节点扩展资源较大的节点能够很大程度上避免当虚拟节点映射成功后,由于与映射节点相连的物理链路资源不足而导致的链路映射请求被推迟问题。

### (2) 物理链路代价 $L(l^s)$

在无线网络中链路存在频繁失效,故需要重点考虑链路中断这一因素,文中综合考虑链路中断概率、链路剩余带宽和链路重要度,并定义如下物理链路代价:

$$L(l^s) = \frac{b_{\max}(l^s)}{b(l^s)} + \lambda_1 \frac{p(l^s)}{p_{\min}(l^s)} + \lambda_2 \delta(l^s) \quad (11)$$

$$\delta(l^s) = \sum_{h \in p} x_p^h \quad (12)$$

式(11)中  $b_{\max}(l^s)$  为剩余链路中最大带宽,  $p_{\min}(l^s)$  为剩余链路最小中断概率,  $\delta(l^s)$  为链路  $l^s$  重要度,  $\lambda_1$  和  $\lambda_2$  为系数。式(12)中  $x_p^h$  定义为链路标识符,表示链路  $l^s$  是否在已映射的物理路径  $h$  中,若在则为1,否则为0。在映射路径时,要选择那些代价小的物理链路构成物理路径,能够最大程度上保证该路径性能最佳。

### (3) 候选节点集合 $Ran(m^s)$

物理节点  $m^s$  发生故障后,映射在其上的虚拟节点随之失效,为保证虚拟请求的正常工作,需要对失效的虚拟节点重新映射。在重映射阶段,对故障节点  $m^s$  构建候选节点集合可以有效地减少映射时间,进而使虚拟请求得到及时响应,候选节点集合如式(13)所示,表示与节点  $m^s$  间距离不超过  $\delta$  跳的物理节点集合:

$$\forall n^s, m^s \in N^s, \delta > 0, Ran(m^s) = \{loc(m^s) - loc(n^s) \leq \delta\} \quad (13)$$

### (4) 节点备用系数 $Alter(n^s)$

虚拟节点重映射时,候选节点集合  $Ran(m^s)$  中往往存在多个节点,而发生故障次数多、承载较多虚拟节点的物理节点很有可能再次发生故障,为避免重映射后的节点发生故障,文中使用节点备用系数  $Alter(n^s)$  作为选取重映射节点的依据,其表达式如下所示:

$$Alter(n^s) = \omega_1 num_{fa}(n^s) + \omega_2 num_{em}(n^s) \quad (14)$$

式中,  $num_{fa}(n^s)$  表示节点发生故障次数,  $num_{em}(n^s)$  表示映射在该节点上的虚拟节点个数,  $\omega_1$  和  $\omega_2$  分别为二者的比例系数。在节点重映射时,应当选择那些故障次数少、承载较少虚拟节点的备用节点进行映射。在实际映射过程中,由于节点故障次数和承载节点个数二者在映射过程中均会存在,且对于节点选择的影响均不可忽略,其重要性程度相差较小。

## 2.5 评价指标

### (1) 虚网请求接受率

在虚拟网络请求 VNR 到来时,底层网络将为其提供物理网络资源以供映射,同时也存在某节点或链路资源受限使得 VNR 推迟映射或映射失败问题,文中用虚网请求接受率  $\xi_{acc}$  衡量 VNR 成功映射的情况,其表达式如下:

$$\xi_{acc} = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T VNR_{acc}}{\sum_{t=0}^T VNR_{arr}} \quad (15)$$

其中,  $VNR_{arr}$  表示在一定时间内到达的 VNR 个数,  $VNR_{acc}$  表示成功映射的 VNR 个数,  $T$  表示 VNR 的生存时间。

### (2) 虚拟网络恢复成功率

当底层网络节点发生故障时,映射在该节点上的 VNR 将会中断,为了保证 VNR 服务的连续性,该请求需要重新映射至正常的物理节点和路径上,文中用虚拟网络恢复成功率  $\xi_{res}$  表示故障 VNR 成功恢复映射的情况:

$$\xi_{res} = \lim_{T \rightarrow \infty} \frac{\sum_{t=0}^T VNR_{res}}{\sum_{t=0}^T VNR_{err}} \quad (16)$$

其中,  $VNR_{err}$  表示因物理节点发生故障而失效的 VNR 个数,  $VNR_{res}$  表示发生故障后成功重映射的 VNR 个数。

### (3) 物理网络资源负载利用率

虚拟网络请求映射成功后,底层网络的资源被暂时分配出去从而造成剩余资源减少,若此时下一个 VNR 到达,则可能发生某节点或链路的物理剩余资源不足,进而导致该 VNR 映射的时间变长。文中定义节点资源负载利用率和带宽资源负载利用率,若资源负载利用率越大则代表当前底层网络资源大部分被占用, VNR 映射需要等待的可能性越大。

$$cpu_{load}(n^s) = \frac{1}{k_n} \sum_{n_a^s \in N^s} \frac{cpu(n^v)}{cpu(n_a^s)} \quad (17)$$

$$b_{load}(l^s) = \frac{1}{k_l} \sum_{l_\beta^s \in E^s} \frac{b(l^v)}{b(l_\beta^s)} \quad (18)$$

其中,  $cpu(n_a^s)$  表示  $n_a^s$  的节点能力,  $cpu(n^v)$  表示虚拟节点所占用的节点能力,  $k_n$  为底层物理节点个数;  $b(l_\beta^s)$  表示物理链路  $l_\beta^s$  的带宽资源,  $b(l^v)$  表示虚拟链路所占用的带宽资源,  $k_l$  为底层物理链路个数。

## 3 算法设计

### 3.1 虚拟网络映射方案

在无线网络中,节点间利用激光或微波链路传输数据,当物理网络拓扑在一定区域时,由于激光或微波的传输特性,一方面可忽略数据在链路中的传输时延,另一方面需要考虑链路中断在链路映射中所带来的影

响<sup>[21]</sup>。因此文中在虚拟链路映射阶段不考虑在网络链路中所产生的时延,即路径最短距离由该路径中所包含的节点跳数决定,节点跳数越多,则认为该路径距离越长;此外,无线网络链路由于受外界干扰或节点间距离远而发生链路中断,若某链路中断概率越小,则认为该链路越可靠。

由上述分析,本文提出一种高可靠性的无线虚拟网络映射算法(WVNEA-HR),其中该算法包括基于扩展资源的节点映射阶段和最小链路代价的链路映射阶段。针对无线虚拟网络映射问题中的节点映射和链路映射两个阶段,算法综合考虑了节点扩展资源以及链路中断概率,WVNEA-HR 通过对虚拟网络请求 VNR 和物理网络资源进行预处理,使两个映射阶段时间减少,进而在一定时间内虚拟网络映射的个数增多。该虚拟网络映射算法主要包括 3 个模块,分别为预处理模块、节点映射模块和链路映射模块,具体 WVNEA-HR 算法伪代码如下所示:

```

1. for all  $n^v, l^v \in VNR$  do
2. Sort  $\{n^v \in VnNodeList, l^v \in VnLinkList\}$  //根据虚拟资源排序
3. end for
4. for all  $n^s, l^s \in G^S$  do
5. if  $Resource(n^s) > Resource_{min}(n^v \in VnNodeList)$  then
6.  $UtilisablePhysicalNodeList \leftarrow n^s$  //节点加入该集合
7. Sort  $\{n^s \in UtilisablePhysicalNodeList\}$  //按物理资源排序
8. else  $VNRList.Last \leftarrow VNR$  //推迟映射
9. if  $Resource(l^s) > Resource_{min}(l^v \in VnLinkList)$  then
10.  $UtilisablePhysicalLinkList \leftarrow l^s$ 
11. Sort  $\{l^s \in UtilisablePhysicalLinkList\}$ 
12. else  $VNRList.Last \leftarrow VNR$ 
13. end for
14. for all  $n^v \in VnNodeList \setminus Node(Failure)$  do
15. if  $n^v = n_{max}^v$  then //最大资源需求的虚拟节点
16.  $k = 0$  //设置虚拟节点个数变量
17. if  $k < Num.VnNodeList$  then
18. for all  $n^s \in UtilisablePhysicalNodeList$  do
19. if  $n^s > n_{max}^v$  then //节点资源是否满足映射条件
20.  $NodeSet \leftarrow n^s$ 
21. if  $NodeSet \neq \emptyset$  then
22. Compute  $cpu_{ex}(n^s \in NodeSet)$  //节点扩展资源
23. if  $cpu_{ex}(n^s) = cpu_{ex}(n^s(max))$  then
24.  $n^s \rightarrow n_{max}^v$  //节点映射成功
25. delete  $n_{max}^v \in VnNodeList$ 
26.  $k = k + 1$ 
27. update //更新物理节点资源
28. end if
29. else  $n_{max}^v \leftarrow failed$  //节点映射失败

```



```

30.          VnNodeList.Last  $\leftarrow n_{\max}^v$ 
31.      end if
32.  end for
33.  else EmbeddedNodeSet  $\leftarrow$  Node(Success) //节点
映射完毕
34.  end if
35.  end for
36.  for all  $l^v \in VnLinkList \setminus Link(Failure)$  do
37.  if  $l^v = l_{\max}^v$  then
38.       $t = 0$ 
39.      for all  $t < Num.VnLinkList$  do
40.  if  $l_{\max}^v \rightarrow l_{m,n}^s$  then
41.  if  $Bandwidth(l_{m,n}^s) > Bandwidth(l_{\max}^v)$  then
42.      LinkSet  $\leftarrow l_{m,n}^s$ 
43.  if  $\{l_{m,n}^s \in LinkSet\} \cap \{EmbeddedNodeSet(m, n) = 1\}$ 
then
44.      delete  $l_{m,n}^s \in LinkSet$ 
45.  end if
46.  if  $LinkSet \neq \emptyset$  then
47.      Compute  $L(l_{m,n}^s \in LinkSet)$  //物理链路代价
48.       $l_{\max}^v \rightarrow l_{m,n}^s(shorst)$  //映射最短物理路径
49.      delete  $l_{\max}^v \in VnLinkSet$ 
50.       $t = t + 1$ 
51.      update //更新物理链路资源
52.  else  $l_{\max}^v \rightarrow \emptyset$ 
53.  VnLinkSet.Last  $\leftarrow l_{\max}^v$ 
54.  end if
55.  end if
56.  end for
57.  end if
58.  end for

```

### 3.2 失效节点和链路恢复策略

当底层网络节点发生故障时,映射在该节点上的虚拟节点失效,与其相连的虚拟链路也将失效,为了保证虚拟请求的正常工作,需要对失效节点以及相连链路进行重映射。在节点重映射时,考虑到虚拟请求的恢复时间,因此在故障节点的临近范围内选取备用节点进行映射,备用节点范围如式(13)所示,其含义为节点  $m, n$  间距离不超过  $\delta$  跳。节点恢复策略如下所示:

输入:物理网络拓扑  $G^S$ , 故障节点  $m$ , 节点跳数  $\delta$

输出:备用节点  $n$

(1)构造备用节点集合  $Ran(m)$

(2)判断集合  $Ran(m)$  中节点资源是否满足虚拟节点资源需求,将不符合条件的节点删除;若集合  $Ran(m)$  中的节点均不满足,则转向步骤(6)

(3)计算集合  $Ran(m)$  中节点的备用系数  $Alter(n^s)$ , 并将结果按照从小到大排列,选取节点备用系数最小的节点作为虚拟重映射节点

(4)更新节点资源

(5)虚拟节点映射成功

(6)虚拟节点映射请求被推迟

重映射的节点在备用节点集合  $Ran(m)$  中产生,这样在恢复虚拟网络请求的同时可以减少映射时间,构造备用节点集合  $Ran(m)$  可通过设置合适的跳数实现。虚拟链路重映射与链路映射方法类似,当失效节点重映射后,针对失效链路,同样采取基于链路代价的方法对其进行映射,若当前资源不允许,则该虚拟链路重映射被推迟,等待有足够资源时再进行修复。

可以看出,在虚拟节点初始映射阶段对于节点的选取是基于节点扩展资源的,而在故障生存重映射阶段对于节点的选取则是基于节点备用系数的,这是因为在初始阶段底层网络主要是以接受 VNR 为主,为提高 VNR 映射成功率,需要重点考虑底层网络资源;而在重映射阶段由于网络故障使得需要重点考虑节点可靠性,以此确保映射的成功率。

## 4 仿真结果分析

### 4.1 实验设置

本文中虚拟网络拓扑和物理网络拓扑均采用 GT-ITM 工具生成,其中物理网络中包含 100 个节点和 400 条链路,每对节点之间相连的概率在区间  $[0.3, 0.6]$  内,物理节点的计算资源和链路带宽资源分别服从  $[30, 70]$  和  $[30, 100]$  之间的均匀分布。每个虚拟请求都是随机生成的,本文假设虚拟请求的到达率满足每 100 个时间单位到达 4 个的泊松分布,每个虚拟请求的生存时间满足 1 000 个时间单位的指数分布。虚拟网络请求中节点的计算资源需求和链路带宽资源需求分别服从  $[1, 10]$  和  $[1, 20]$  之间的均匀分布,虚拟请求节点个数服从  $[2, 6]$  之间的均匀分布,每一对虚拟节点之间以 0.5 的概率相连。在每次仿真实验中运行 50 000 个时间单位,大约包含 2 000 个虚拟请求。节点的失效间隔设为 20 000 个单位,并假定在下一个节点失效到来之前上一个失效节点已经被修复。将物理节点扩展资源中  $\mu_1$  和  $\mu_2$  分别设置为 0.4 和 0.6,物理链路代价中  $\lambda_1$  和  $\lambda_2$  均设置为 0.5,候选节点集合中跳数  $\delta$  设置为 10,为简化计算,节点备用系数中  $\omega_1$  和  $\omega_2$  设置为 0.5。

### 4.2 实验结果分析

本文以虚拟请求接受率、虚拟网络恢复成功率、虚拟请求重映射接受率、节点和链路负载利用率作为算法性能的评价指标,分别与算法 DP-VNE<sup>[22]</sup>、SVNE-Hybrid<sup>[23]</sup>、VNM-SM<sup>[24]</sup> 进行对比分析,表 1 所示为几种算法对比。

图 3 所示为虚拟映射接受率曲线,由于初始阶段虚拟网络请求少,因而几种算法都获得了较高的映射接受率,随着时间的推移,几种算法的性能逐渐趋于平稳。可以看出,SVNE-Hybrid 算法初始时性能略优于 VNM-SM,这是由于 VNM-SM 针对不同服务采取不同策略,而初始时刻由于  $Vn$  个数较少,故性能略低于 SVNE-Hybrid,而随着  $Vn$  个数的增多,SVNE-Hybrid 算法由于采用备

表1 算法比较

算法名称	算法简介
DP-VNE	将 $V_n$ 中每条虚拟链路简单映射到底层不交叉的物理链路中,确保虚拟映射的可靠性
SVNE-Hybrid	采取备份链路策略,针对网络故障预先为每条链路计算备份路径
VNM-SM	一种区分服务的故障恢复算法,针对不同优先级分别采用预留保护路径机制和节点链路迁移策略
WVNEA-HR	优先选取物理节点扩展资源大的节点进行映射,并采用最短路径法映射虚拟链路

份链路策略,因而一定程度上占用了剩余资源,导致其性能低于VNM-SM算法,本文算法WVNEA-HR性能总体上优于其余三种算法,是由于在虚拟节点映射过程中除节点资源外还考虑与之相连物理链路的资源,一定程度上确保了链路映射的可靠性,如当仿真时间为10 min时,DP-VNE算法的接受率降到56%左右,本文所提算法的接受率逐渐稳定在67%左右。

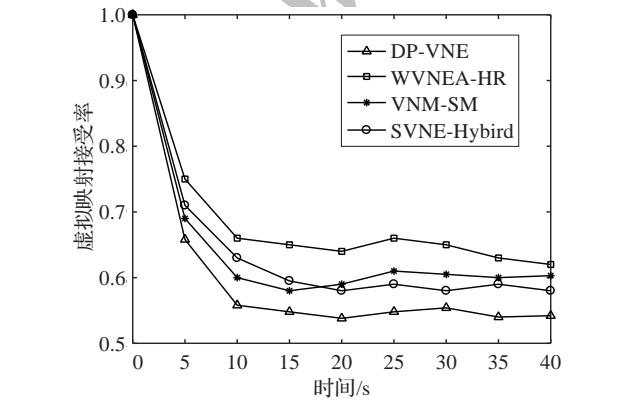


图3 虚拟请求接受率

图4所示为虚拟网络恢复成功率随时间的变化情况,初始映射阶段到达的  $V_n$  较少,发生故障后底层网络有足够多的资源供生存恢复使用,因而初始阶段的恢复成功率较高,随着时间的推移,底层网络的资源逐渐被使用,能够提供的资源减少,因而恢复成功率变小最终趋于稳定。DP-VNE算法的恢复成功率最低,VNM-SM和本文算法的恢复成功率相近,文中算法采取了节点候选集合,当物理节点发生故障后能够快速找到其迁移的

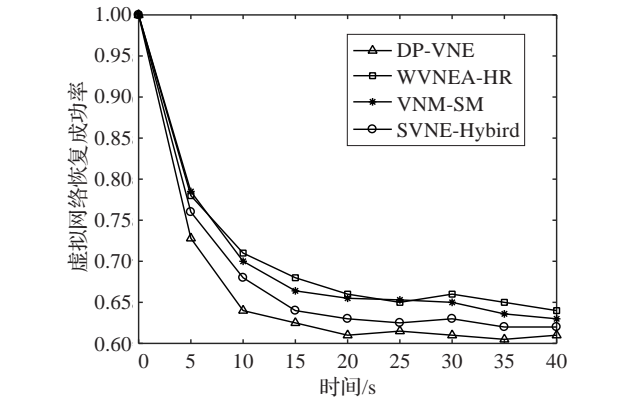


图4 虚拟请求恢复成功率

物理节点,因而使得发生失效的虚拟请求能够迅速恢复,可以看出,本文算法的恢复成功率最终稳定在66%左右。

图5所示为随着候选集合节点跳数的变化虚拟请求重映射接受率的情况,可以看出,虚拟网络重映射能否成功在很大程度上取决于网络中节点跳数的选取。仿真显示出随着节点跳数的增加 ( $h=5, 8, 11$ ),虚拟请求重映射接受率呈现出增大的趋势,当节点跳数  $h$  分别为5、8、11时,对应的请求接受率依次在64%、59%和49%左右,这是由于候选集合中的节点个数增多,使得故障节点有更多候选节点资源使用,一定程度上提高了请求接受率。

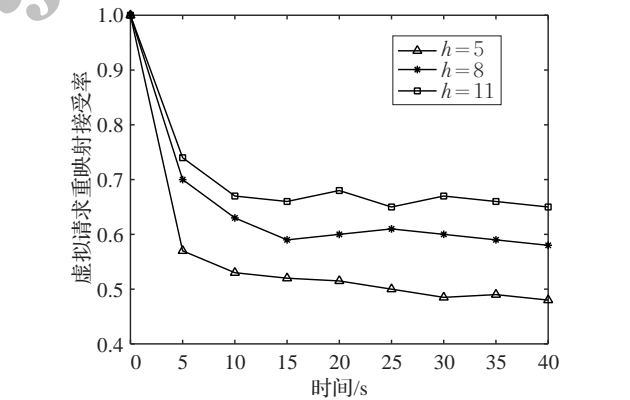


图5 虚拟请求重映射接受率

节点资源负载利用率和链路资源负载利用率仿真如图6和图7所示,可以看出,在三种算法的负载利用率

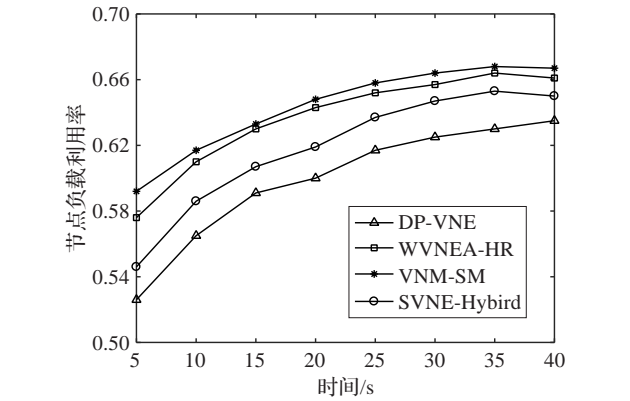


图6 节点负载利用率

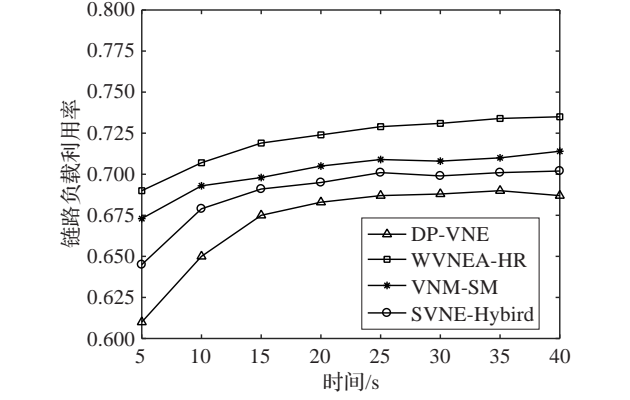


图7 链路负载利用率

曲线中,本文提出的算法具有较优的性能,由于WVNEA-HR在节点映射方面综合考虑节点资源和扩展资源,使得节点扩展资源较小的节点被排除,故节点资源负载利用率略小于VNM-SM算法,而在链路映射阶段由于采用最小链路代价策略,有效地避免了底层某条链路被多次映射,故其链路负载利用率优于其余算法,SVNE-Hybrid算法的负载利用率高于DP-VNE,DP-VNE算法简单将每条虚拟链路映射到底层不交叉的链路上,没有更进一步的约束条件,其效率最低。总体而言,本文算法具有较高的网络资源利用率。

## 5 结束语

本文主要针对虚拟网络映射过程中链路映射阶段资源不足以及网络故障问题进行研究,为确保链路映射的成功率,在节点映射阶段采用基于节点扩展资源的映射策略,以此提高链路映射成功率,在链路映射阶段则采用最小链路代价的映射方法。此外在底层网络节点发生故障时,文中所提映射算法能够最大程度保证虚拟请求的连通性,并能够完成对虚拟请求失效部分的重映射,结果表明本文所提算法在满足虚拟请求可靠映射的同时,能够提高虚拟映射接受率和虚拟请求恢复成功率。现阶段关于虚拟网络可靠性映射的研究主要围绕单节点失效,关于多节点失效和多域映射的研究较少,今后将在现阶段的基础上进一步关于多节点失效和多域映射展开研究。

## 参考文献:

- [1] Zhang Ning, Yang Peng, Zhang Shan, et al. Software defined networking enabled wireless network virtualization: challenges and solutions[J]. IEEE Network, 2017, 31(5): 42-49.
- [2] Wang A, Iyer M, Dutta R, et al. Network virtualization: technologies, perspectives, and frontiers[J]. Journal of Light-wave Technology, 2013, 31(4): 523-537.
- [3] Fischer A, Botero J F, Beck M T, et al. Virtual network embedding: a survey[J]. IEEE Communications Surveys & Tutorials, 2013, 15(4): 1888-1906.
- [4] 朱强, 王慧强, 马春光, 等. 虚拟网络可生存的启发式可靠映射算法[J]. 通信学报, 2015, 36(7): 109-119.
- [5] 许倩, 易辉跃, 朱军, 等. 基于熵权法的虚拟网络映射算法[J]. 计算机工程与应用, 2015, 51(23): 94-99.
- [6] Mosharaf N M, Chowdhury K, Rahman M R, et al. Virtual network embedding with coordinated node and link mapping[C]//2009 IEEE INFOCOM 28th International Conference on Computer Communications, Rio de Janeiro, Brazil, 2009: 783-791.
- [7] 余建军, 吴春明. 虚拟网映射问题的计算复杂性分析[J]. 计算机科学, 2018(11): 87-91.
- [8] 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.
- [9] Zheng Q, Li J, Wang Z, et al. A 2-layers virtual network mapping algorithm based on node attribute and network simplex[J]. IEEE Access, 2018, 6: 77474-77484.
- [10] Chowdhury M, Rahman M R, Boutaba R. ViNEYard: virtual network embedding algorithms with coordinated nodes and link mapping[J]. IEEE/ACM Transactions on Networking, 2012, 20(1): 206-219.
- [11] 李志敏, 汤创为, 谭敏生, 等. 基于最小节点负载优先的逻辑承载网构建方法[J]. 计算机工程与应用, 2014, 50(10): 87-91.
- [12] Ghazisaeedi E, Huang C. Off-peak energy optimization for links in virtualized network environment[J]. IEEE Transactions on Cloud Computing, 2017, 5(2): 155-167.
- [13] Mechtri M, Hadji M, Zeghlache D. Exact and heuristic resource mapping algorithms for distributed and hybrid clouds[J]. IEEE Transactions on Cloud Computing, 2017, 5(4): 681-696.
- [14] 刘光远, 苏森. 面向底层单节点失效的轻量级可靠虚拟网络映射算法[J]. 电子与信息学报, 2013, 35(11): 2644-2649.
- [15] Xiao A, Wang Y, Meng L, et al. Topology-aware virtual network embedding to survive multiple node failures[C]//Global Communications Conference, 2014: 1823-1828.
- [16] 罗娟, 刘川川, 李仁发. 基于链路可靠性的无线虚拟网络分配方法[J]. 通信学报, 2012(s1): 88-95.
- [17] Alam Khan M M, Shahriar N, Ahmed R, et al. Multi-path link embedding for survivability in virtual networks[J]. IEEE Transactions on Network and Service Management, 2016, 13(2): 253-266.
- [18] Zheng X, Tian J, Xiao X, et al. A heuristic survivable virtual network mapping algorithm[J]. Soft Computing, 2018, 23(5): 1453-1463.
- [19] Yi Z, Xu J, Zhang Q, et al. Survivable virtual network mapping based on two-stage potential games for cloud infrastructure[C]//2018 International Conference on Computing, Networking and Communications (ICNC), 2018: 866-872.
- [20] Abdelwahab S, Hamdaoui B, Guizani M, et al. Efficient virtual network embedding with backtrack avoidance for dynamic wireless networks[J]. IEEE Transactions on Wireless Communications, 2016, 15(4): 2669-2683.
- [21] Lira V, Tavares E, Oliveira M, et al. Virtual network mapping considering energy consumption and availability[J]. Computing, 2018(2): 1-31.
- [22] 刘光远, 苏森. 面向底层单节点失效的轻量级可靠虚拟网络映射算法[J]. 电子与信息学报, 2013(11): 2644-2649.
- [23] Rahman M R, Boutaba R. SVNE: survivable virtual network embedding algorithms for network virtualization[J]. IEEE Transactions on Network and Service Management, 2013, 10(2): 105-118.
- [24] 贾伟, 夏靖波. 一种生存性军事虚拟网络映射新方法[J]. 电讯技术, 2015(5): 545-551.