



计算机科学

COMPUTER SCIENCE

基于最小生成树的vSDN故障快速恢复算法

陈港, 孟相如, 康巧燕, 翟东

引用本文

陈港, 孟相如, 康巧燕, 翟东. 基于最小生成树的vSDN故障快速恢复算法[J]. 计算机科学, 2022, 49(11A): 211200034-7.

CHEN Gang, MENG Xiang-ru, KANG Qiao-yan, ZHAI Dong. [vSDN Fault Recovery Algorithm Based on Minimum Spanning Tree](#) [J]. Computer Science, 2022, 49(11A): 211200034-7.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[节点拓扑感知的高效节能虚拟网络映射算法](#)

Energy Efficient Virtual Network Mapping Algorithms Based on Node Topology Awareness
计算机科学, 2020, 47(9): 270-274. <https://doi.org/10.11896/jsjcx.190700162>

[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
计算机科学, 2020, 47(7): 236-242. <https://doi.org/10.11896/jsjcx.190600022>

[一种基于复杂网络的图像形状及纹理描述方法](#)

Image Shape and Texture Description Method Based on Complex Network
计算机科学, 2018, 45(11A): 244-246.

[动态频谱访问无线网络的连通性分析](#)

On the Connectivity Analysis over Dynamic Spectrum Access Networks
计算机科学, 2010, 37(7): 62-65.

[Skitter与Ark探测架构下AS级Internet拓扑分析](#)

Analysis of Internet AS-level Topology under Skitter and Ark Measurement Infrastructure
计算机科学, 2010, 37(11): 38-40.

基于最小生成树的 vSDN 故障快速恢复算法

陈 港 孟相如 康巧燕 翟 东

空军工程大学信息与导航学院 西安 710077

摘 要 针对虚拟软件定义网络的故障恢复问题,提出了基于最小生成树的虚拟软件定义网络故障恢复算法,以期解决虚拟软件定义网络故障恢复时间长这一难点。该算法一方面是根据节点与链路的资源和拓扑属性,设定节点与链路重要度,并据此对节点与链路进行等级划分;在此基础上,针对不同的物理网络,调整备份与迁移的比例,从而在提升请求接受率的同时减少故障后的完全恢复时间,达成对物理网络资源的充分利用;另一方面是对虚拟网络进行连通性分析,运用最小生成树算法优先恢复虚拟网络的连通性,然后完成剩余链路的故障恢复,在保障虚拟网络连通性的基础上进一步减少故障恢复时间。仿真实验表明,该算法能在保障较高虚拟网络请求接受率与故障恢复率的基础上,缩减故障恢复时间。

关键词 网络虚拟化;虚拟软件定义网络;故障恢复;连通性;最小生成树

中图分类号 TP393

vSDN Fault Recovery Algorithm Based on Minimum Spanning Tree

CHEN Gang, MENG Xiang-ru, KANG Qiao-yan and ZHAI Dong

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

Abstract Aiming at the problem of fault recovery of virtual software-defined network, a fault recovery algorithm of virtual software-defined network based on minimum spanning tree is proposed to solve the difficulty of long recovery time of virtual software-defined network. On the one hand, the algorithm sets the importance of nodes and links according to the resources and topological attributes of nodes and links, and classifies nodes and links accordingly. On this basis, according to different physical networks, adjusting the ratio of backup and migration, improving the request acceptance rate and reducing the full recovery time after failure, so as to make full use of physical network resources. On the other hand, it analyzes the connectivity of virtual network, and uses the minimum spanning tree algorithm to restore the connectivity of virtual network first, and then completes the fault recovery of the remaining links, further reducing the fault recovery time on the basis of ensuring the connectivity of virtual network. Simulation results show that the algorithm can reduce the recovery time of virtual network on the basis of ensuring high request acceptance rate and recovery rate.

Keywords Network virtualization, Virtual software-defined network, Failure recovery, Connectivity, Minimum spanning tree

1 概述

虚拟软件定义网络(Virtual Software Defined Network, vSDN)作为一种解决原有网络僵化问题的新式网络架构,也存在着部分来自软件定义网络(Software Defined Network, SDN)和虚拟网络的故障恢复问题。一是集中控制、分散部署的架构使得少数的控制器和控制链路成为网络安全运行的关键,一旦某控制器发生故障则会导致其所控交换机全部失联;二是从物理网络基础设施中抽象出虚拟的网络服务,让单个物理交换机承担多个虚拟交换机任务,使得由物理交换机故障导致的恢复问题更加难以解决。

为此,部分学者针对 vSDN 故障恢复方案展开研究。根据网络故障与备份计算的先后顺序,故障恢复方案可分为主动备份和被动迁移两种,前者在故障发生前为相关节点和链路分配备份对象,恢复迅速、可靠但资源开销较大;后者则在故障发生后临时寻找可供资源转移的正常对象,资源利用率高但恢复时间长且可靠性低^[1-5]。由于控制器的存在,vSDN

的故障恢复方案除了对交换机以及数据链路的故障恢复,还包括对控制器的保护^[6,7]、控制信道的保护^[8]以及控制器部署^[9]和虚拟管理平台部署^[10-11]等研究内容。

为提高故障恢复率并解决负载不均问题,文献[12]将动态路径分裂方法引入故障虚拟链路的恢复,结合就近原则与负载均衡方法以提高故障虚拟链路的恢复率,但未考虑节点的故障恢复。同年,文献[13]提出了容错最短路径算法来检查节点风险故障,并找到具有更高带宽可用性且专注于节能的最短路径,适用于对故障恢复时长要求较低的虚拟网络。文献[14]进一步考虑到控制器位置部署,依据物理网络故障类型采取主动备份或被动迁移方式恢复故障节点与链路,以提高故障恢复率。而文献[15]采用一种联合主动备份和被动迁移的虚拟网络映射保障机制。该机制利用马尔可夫模型求解虚拟节点映射前后相对剩余资源的变化,若其变小则采用主动备份,否则采用被动迁移,通过权衡主动备份与被动迁移的比例,提高网络的整体故障恢复率。以上研究主要考虑提升故障恢复率,对于所承载虚拟网络

基金项目:陕西省重点研究计划(2020-GY-026)

This work was supported by the Shaanxi Province Key R & D Program, China(2020-GY-026).

通信作者:陈港(ling0chengang@163.com)

的恢复时间仍有改进空间。

综合以上研究文献,本文重点研究交换机和数据链路的故障恢复,从网络节点与链路的可靠性角度出发,综合考虑不同虚拟网络对节点与链路故障的恢复要求,提出基于生成树优先保障的故障恢复算法,从而在提高请求接受率和故障恢复率的同时,缩减故障恢复时长。

2 网络模型与故障分析

2.1 物理网络

底层物理网络用加权无向图 $G^s = (N^s, E^s, A_N^s, A_E^s)$ 来描述。其中 N^s 是物理网络节点 n^s 的集合, $n^s = (nc^s \cup nh^s \cup ns^s)$, nc^s, nh^s, ns^s 分别表示物理控制器节点、虚拟化管理平台部署节点以及物理交换机节点; E^s 是物理链路 e^s 的集合, $e^s = (ec^s \cup eg^s)$, ec^s 和 eg^s 分别表示物理控制链路和物理数据链路; A_N^s 是物理节点的属性集合, 包含节点计算资源 CPU 属性 $cu(n^s)$ 和存储资源 TCAM(Ternary Content Addressable Memory)属性 $tm(n^s)$; A_E^s 是物理链路的属性集合, 包含链路带宽 BW(Bandwidth)属性 $bw(e^s)$ 。记节点 CPU, TCAM 以及带宽 BW 的剩余资源分别为 $SA_{cu}(n^s)$, $SA_{tm}(n^s)$, $SA_{bw}(e^s)$ 。

2.2 虚拟网络

虚拟 SDN 网络请求用 vSDN 表示, 第 i 个虚拟网络表示为 $vSDN(i)$ 。

虚拟网络用加权无向图 $G^v = (N^v, E^v, A_N^v, A_E^v, Tr, Le)$ 描述。其中 N^v 是虚拟网络节点 n^v 的集合, $n^v = (nc^v \cup n^v)$, nc^v 和 n^v 分别表示虚拟控制器节点以及虚拟交换机节点; E^v 是虚拟链路 e^v 的集合, $e^v = (ec^v \cup eg^v)$, ec^v 和 eg^v 分别表示虚拟控制链路和虚拟数据链路; A_N^v 是虚拟节点的属性集合, 包含节点计算资源 CPU 属性 $cu(n^v)$ 和存储资源 TCAM 属性 $tm(n^v)$; A_E^v 是虚拟链路的属性集合, 包含链路带宽 BW 属性 $bw(e^v)$; Tr 表示虚拟网络请求运行时长; Le 表示虚拟 SDN 网络请求的优先级, 取值为 1, 2, 3, 第 3 级别最高。

2.3 网络映射

虚拟 SDN 网络映射是将虚拟 SDN 网络请求 G^v 映射到物理网络 G^s 的子集 G^s , 用 $M(G^v(i)) = G^s(i) \in G^s(i)$ 描述。在映射过程中, 需要满足节点的 CPU 约束、TCAM 约束和链路带宽的 BW 约束。本文采用基于 OpenVirteX 网络虚拟化平台的结构, 由于该平台的限制, 同一个虚拟 SDN 网络请求中, 不同的虚拟节点不能部署于同一物理交换机上, 但同一个物理交换机上可以存在多个不同虚拟 SDN 网络请求的虚拟节点。本文设定控制链路采用带内部署并假设每一个虚拟网络请求有且只有一个虚拟 SDN 控制器负责该网络的流表分发和流量控制^[16-17]。

(1) 节点映射

记某虚拟网络 $G^v(i)$ 节点总数为 n_{sum}^v , 其中虚拟交换机节点总数为 $n_{sum}^v - 1$ 。令虚拟节点 n_i^v 映射到物理节点 n_j^s 表示为 $O_{ij} = 1$, 否则记 $O_{ij} = 0$ 。将虚拟节点 n^v 映射到物理节点 n^s 的过程记为 M_N , 即:

$$M_N(n^v) = n^s \in N^s \quad (1)$$

需要满足如下约束:

$$cu(n^v) \leq SA_{cu}(M_N(n^v)) \quad (2)$$

$$tm(n^v) \leq SA_{tm}(M_N(n^v)) \quad (3)$$

$$\begin{cases} \sum_{n_j^v \in N^v} O_{ij} = 1, & \forall n_i^v \in N^v \\ \sum_{n_i^v \in N^v} O_{ij} \leq 1, & \forall n_j^s \in N^s \end{cases} \quad (4)$$

(2) 链路映射

定义底层物理网络两点间任意一条非闭环的路径为 $Pa(n_i^s, n_j^s)$ 。将虚拟链路 e^v 映射到物理链路 e^s 的过程记为 M_E , 即:

$$M_E(e^v) = Pa(M_N(n_i^v), M_N(n_j^v)) \quad (5)$$

$$\forall e^v = (n_i^v, n_j^v), 0 \leq i, j \leq n_{sum}^v, i \neq j$$

需要满足如下约束:

$$bw(e^v) \leq SA_{bw}(e^s), \forall e^s \in M_E(e^v) \quad (6)$$

2.4 故障分析

虚拟软件定义网络的整体架构以基于 OpenVirtex 的 SDN 网络虚拟化管理平台为例, 如图 1 所示。

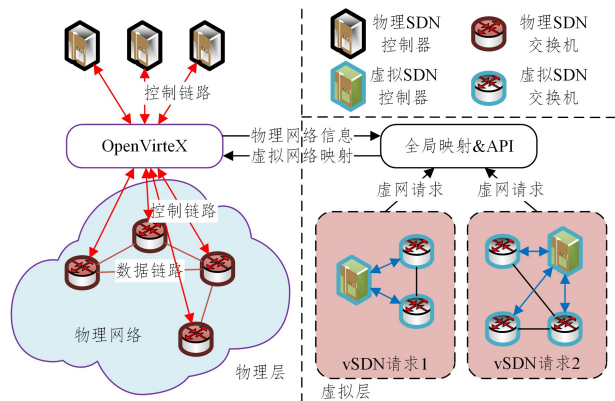


图 1 基于 OpenVirtex 的 SDN 网络虚拟化管理平台示意图

Fig. 1 SDN virtualization management platform based on OpenVirteX

根据图 1 进行故障分析可知, SDN 控制器和 OpenVirteX 平台发生故障后极易对网络造成破坏性影响, 控制链路故障也易导致部分交换机节点失去控制或控制器失联, 交换机节点和数据链路故障相较而言影响较小。本文为统筹考虑故障发生后的恢复工作, 将故障类型分为控制器与虚拟化管理平台故障、节点故障和链路故障 3 种。控制器与虚拟化管理平台故障由于影响极大, 一般而言都会采用主备切换的方式进行主动式的故障保护, 文献[10]已对其进行了部分研究; 链路故障包括由交换机故障引起的某节点和其直连链路失效的故障类型; 而节点故障特指被映射有虚拟节点的物理节点故障类型。本文重点研究由交换机和数据链路抽象出的节点与链路故障。

3 物理网络重要度分析

重要节点与链路指的是能够在更大程度上影响网络结构和功能的一类特殊节点和链路^[16]。

本节在文献[17]的基础上, 提出了更具针对性的节点与链路重要度分析算法。该算法融合局部特征和全局特征, 分别对网络中的节点和链路进行重要度评价。

以下为评价指标, 前 4 个指标描述网络的局部特征, 后两个指标描述网络的全局特征。

(1) 节点的度^[18]: 描述为节点与其邻居节点直接相邻链路的数量, 记为 $D, D(n_i) = \deg(n_i)$ 。

(2) 集聚系数^[19]: 描述为节点的邻居之间互为邻居的比例, 记为 C_c 。

(3)节点相似性度量^[20]:描述为邻居节点间的相似性,记为 lls 。

(4)节点连通程度^[20]:描述为两节点间存在的最短可达路径情况,记为 Cn ,如式(7)所示。

$$Cn(n_x, n_y) = \frac{1}{Hop} \times \log_2^{Hop+1} \quad (7)$$

$$Hop = hop(L_{\min}(n_x, n_y)), Hop \in N^+$$

其中, $L_{\min}(n_x, n_y)$ 为节点 n_x 与 n_y 之间的最短路径, $hop(L_i)$ 为路径 L_i 的链路数量, N^+ 表示正整数。当节点 n_x 与 n_y 之间存在 1 条直连路径时, $Cn(n_x, n_y) = 1$, 当节点 n_x 与 n_y 间不存在可达路径时, 记 $Cn(n_x, n_y) = 0$ 。

(5)网络效率^[21]:描述为网络任意两点间的平均信息传递效率,记为 η_{ad} ,如式(8)所示。

$$\eta_{ad} = \frac{2}{n_{\text{sum}}(n_{\text{sum}} - 1)} \sum_{i,j \in N} \frac{1}{d_{ij}}, i \neq j \quad (8)$$

其中, d_{ij} 表示节点 i, j 之间的最短路径长度。网络效率下降比例记为 $\mu, \mu = 1 - \frac{\eta_{ad}}{\eta_{ad}^0}$ 。其中 η_{ad} 表示节点或链路移除后的网络效率, η_{ad}^0 表示原始的网络效率。 μ 值越大,表示移除后的网络效率越差。

(6)极大连通系数^[22]:描述为网络中极大连通子集节点数与网络总节点数的比值,记为 η_{mc} 。

3.1 节点重要度计算

定义节点的资源度为 $p(n_i)$, 节点的局部特征指标为 $q_0(n_i)$, 节点的全局特征指标为 $q_1(n_i)$ 。节点重要度排序算法的具体流程如下:

首先计算节点的局部特征指标 $q_0(n_i)$, 如式(9)所示:

$$q_0(n_i) = D(n_i) \times Cc(n_i) \times llS(n_i) \times \frac{1}{\sum_{n_j \in nb(n_i)} bw(n_i, n_j)} \times \sum_{n_j \in nb(n_i)} \left(\frac{p(n_j)}{p(n_i)} \times bw(n_i, n_j) \right) \quad (9)$$

然后计算节点的全局特征指标 $q_1(n_i)$, 针对某静态网络(极大连通系数为 1), 计算网络效率 η_{ad}^0 , 移除某节点后, 计算网络效率 η_{ad} 和极大连通系数 $\eta_{mc}(n_i)$, 再计算 $\mu(n_i), q_1(n_i) = \mu(n_i) \times \eta_{mc}(n_i)$ 。

最后计算节点的重要度指标 $R(n_i)$ 。 $R(n_i) = p(n_i) \times [q_1(n_i) + q_0(n_i)]$, $R(n_i)$ 越大表示节点 n_i 越重要。

3.2 链路重要度计算

定义链路 e_i 的拓扑属性重要度为 $q_3(e_i)$, 带宽属性重要度为 $q_1(e_i)$ 。记链路 e_i 所连通的两个端点为 n_x 与 n_y 。链路重要度排序算法具体流程如下:

首先计算链路的拓扑属性重要度 $q_3(e_i)$ 。两个端点的重要度分别为 $R(n_x)$ 与 $R(n_y)$; 两个端点的连通程度在此简化计算为三跳内可达链路。可参考式(7)所计算的节点连通程度指标。

链路的拓扑属性重要度计算如式(10)所示:

$$q_3(e_i) = \frac{R(n_x) + R(n_y)}{2 \times \text{Max}(R(n_i))} \times Cn(n_x, n_y) \quad (10)$$

然后计算链路的带宽属性重要度 $q_1(e_i)$ 。记 $eb(n_i)$ 为与节点 n_i 直连的链路集合, 其计算式如式(11)所示:

$$\begin{cases} q_1^1(e_i) = \frac{bw(e_i) - \min[bw(e_j)]}{\max[bw(e_j)] - \min[bw(e_j)]}, & e_j \in eb(n_x) \\ q_1^2(e_i) = \frac{bw(e_i) - \min[bw(e_j)]}{\max[bw(e_j)] - \min[bw(e_j)]}, & e_j \in eb(n_y) \\ q_1^3(e_i) = \frac{bw(e_i) - \min[bw(e_j)]}{\max[bw(e_j)] - \min[bw(e_j)]}, & e_j \in e^s \end{cases}$$

$$q_1(e_i) = \frac{1}{3} \times [q_1^1(e_i) + q_1^2(e_i) + q_1^3(e_i)] \times bw(e_i) \quad (11)$$

最后计算链路的重要度 $R(e_i)$ 为:

$$R(e_i) = q_3(e_i) \times q_1(e_i) \quad (12)$$

3.3 节点与链路备份

首先,根据物理网络节点与链路的重要度以及相关属性计算其重要度等级,对物理网络节点与链路依据其重要度等级采取不同的保护措施;然后,根据故障节点和链路在物理网络中的位置以及对应虚拟网络的运行状况,对受到影响的虚拟网络采用相应的措施进行恢复。

针对节点进行重要度等级分析,简记为 RH 分析。依据重要度 $R(n_i)$ 和距离中心节点(到其余各节点最短跳数和最小的节点)的跳数,运用 k -means 聚类分析算法,将其分为 3 个等级 R1, R2, R3, 其中 R1 级最重要且占比不超过 20%, R2 级次要且占比不超过 30%, R3 级重要度最低且占比不少于 50%。针对链路进行重要度等级 RH 分析:若链路 e_i^s 两端节点重要度等级相同,则 $RH(e_i^s) = RH(n_i^s)$;若链路 e_i^s 两端节点重要度等级相差一个等级,则 $RH(e_i^s)$ 等于较低的那个等级;若链路 e_i^s 两端节点重要度等级相差两个等级,则 $RH(e_i^s) = R2$ 。

针对 R1 级节点或链路,采用主动备份措施。一方面是因为 R1 级最重要,需要始终维持其正常的工作状态;另一方面, R1 级数量较少,不超过网络中节点或链路总数的 20%。针对 R2 级与 R3 级节点或链路,采用迁移或重路由的被动恢复措施。

针对节点:在 R1 级节点三跳距离内寻找满足条件的 R2 级节点作为备份节点,同时在 R2 级节点的三跳距离内寻找满足条件的 R3 级节点作为前两个节点的公共备份节点;针对链路:寻找同端点的另一条最短路径作为备份路径。

4 虚拟网络连通性分析

对于受损的虚拟网络,节点故障后只能寻求空余的其他物理节点进行重映射,链路故障后若有备份路径则可直接迁移恢复,或者进行重路由。但当虚拟网络出现多条链路故障时,可以优先恢复能保障该虚拟网络所有节点连通的链路,然后根据虚拟网络等级和物理网络资源利用情况恢复其余链路,以实现虚拟网络所承载业务的快速连通要求,降低恢复等待时延。

4.1 连通性理论分析

该小节关键在于明确属于最大连通子图的链路,其中所述链路故障为断路故障。给定一个确定网络 $G_i(N, E)$, 节点总数为 n_{sum} , 边总数为 e_{sum} , 其中 $n_{\text{sum}} \geq 2, n_{\text{sum}} - 1 \leq e_{\text{sum}} \leq (n_{\text{sum}} \times n_{\text{sum}} - 1) / 2$, 则它的最大连通子图 $G_{z_i}(N', E')$ 的节点和边的数量固定,且分别为 n_{sum} 和 $n_{\text{sum}} - 1$ 。

记网络 $G_i(N, E)$ 的连通概率为 K , 表示为边的数量占相同节点的完全图中边的数量的比例,计算式为:

$$K = \frac{2 \times e_{\text{sum}}}{n_{\text{sum}} \times (n_{\text{sum}} - 1)}, 0 \leq K \leq 1 \quad (13)$$

则恢复连通性所需考虑的链路数量相比完全恢复所需考虑的链路数量缩减的比例 η 为:

$$\eta = 1 - \frac{2}{n_{\text{sum}} \times K}, 0 \leq \eta \leq 1 - \frac{2}{n_{\text{sum}}} \quad (14)$$

4.2 最小生成树计算步骤

记正常链路连通性为 $Co = 1$, 故障链路连通性为 $Co = 0$ 。

由于 $G_{z_i}(N', E')$ 的边无法具体确定, 因此最大连通子图在一般情况下存在多种解。为此, 受到 k -壳分解算法启发, 本节创新性地设计了一种基于最小生成树的虚拟网络连通性分析算法, 以在虚拟网络多条边受损时确定边恢复的优先序列。

首先搜寻必要链路, 所有链路权重为 MAX 的都为必要链路。

步骤 1 给予所有链路权重为链路的重要度。

步骤 2 遍历网络中度数为 1 的节点, 将其直连链路权重改为 MAX。

步骤 3 删掉度数为 1 的节点和其相连链路。

步骤 4 重复步骤 2 和步骤 3, 直至网络中再无度数为 1 的节点。然后进行故障链路约束分析。

步骤 5 若剩余网络 G_a 中仍存在节点, 确定剩余节点数量 N_m 和链路的数量 E_m , 且满足如下约束: $3 \leq N_m \leq E_m \leq \frac{1}{2} N_m(N_m - 1)$ 。

步骤 6 对于剩余网络 G_a 中的链路, 计算当该链路断路时两端节点间的节点连通程度 C_n , 若 C_n 为 0, 则将该链路权值改为 MAX。

步骤 7 为保持其连通性, 记同时故障链路数量为 E_f , 则可设立确保网络连通性的约束条件如下:

$$(1) E_f \leq E_m - N_m + 1;$$

$$(2) \sum_{e_i \in nb(n_i)}^{D(n_i)} Co(e_i) \geq 1, \text{ 即每个节点至少有一条直连链路连通。}$$

最后确保网络连通的链路, 明确链路恢复时顺序。

步骤 8 当多条链路发生故障时, 先恢复权重为 MAX 的链路即必要链路; 针对剩余的故障链路, 判断是否满足步骤 6 和步骤 7 中的约束条件, 若满足则延迟恢复, 若不满足则依次恢复冲突链路集合中权重高的链路直至符合约束条件。

以上步骤的计算演示如图 2 所示。

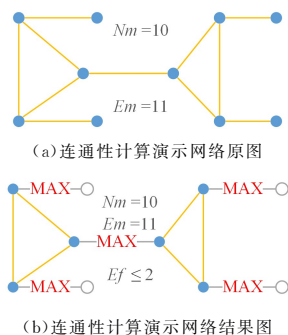


图 2 连通性计算演示图

Fig. 2 Diagram of connectivity computing demo

5 虚拟网络恢复算法构建

基于 OpenVirtex 的 SDN 网络虚拟化平台发生故障后的恢复方案分为两步。当平台正常运行遭遇突发性故障时, 首先恢复虚拟网络的连通性, 再恢复虚拟网络的其余链路。

基于最小生成树优先保障虚拟软件定义网络连通的故障恢复算法 (Fault Recovery algorithm based on Minimum Spanning Tree to guarantee the connectivity of Virtual Software-Defined Network, MST-vSDNFR) 流程如下, 其伪代码如算法 1 所示。

当发生物理网络节点或链路故障导致相应虚拟网络受损时, 首先确定受损物理网络对象的重要度等级, 若都为 R1

级, 则采用备份迁移措施, 将相关受损虚拟网络的资源转移到提前进行备份的节点或链路中, 完成故障恢复 (见算法 1 第 1—2 行)。

若存在不为 R1 级的对象, 则将受损虚拟网络进行分离, 确认故障物理对象具体情况以及受损虚拟网络优先级。虚拟网络受损恢复优先级判断细则: 优先恢复优先级高的虚拟网络, 同优先级条件下优先恢复受损对象少的虚拟网络, 同优先级且同数量受损对象情况下优先恢复整体资源使用率低的虚拟网络, 同资源使用率条件下随机顺序恢复 (见算法 1 第 3—13 行)。

分离后的受损虚拟网络恢复可分为 3 个阶段: 第一阶段恢复所有节点; 第二阶段恢复网络的一个生成树; 第三阶段恢复网络其余链路。

第一阶段: 若出现承担虚拟节点映射的物理节点故障, 则执行节点迁移算法。首先进行空闲节点搜寻, 将虚拟节点资源迁移到具备足够空闲资源的完好物理节点中, 然后恢复故障物理节点相连链路所承载的虚拟链路 (见算法 1 第 14—22 行)。

第二阶段: 若仅出现链路故障或故障物理节点所承载虚拟节点完成迁移后, 对于已经分离的虚拟网络, 对每一个虚拟网络进行连通性分析, 根据已确定的链路恢复顺序优先恢复使该虚拟网络连通的故障链路 (见算法 1 第 23—26 行)。

第三阶段: 对受损虚拟网络的其余链路进行重路由恢复 (见算法 1 第 27—33 行)。

算法 1 MST-vSDNFR

输入: 物理网络, 虚拟网络, 故障节点, 故障链路

输出: 故障节点与链路恢复方案

```

1. if all  $n_i^s|_f$  is  $n_i^1$  and all  $e_i^s|_f$  is  $e_i^1$ 
2.   make 节点排序 and 链路排序
3. else
4.   for  $n_i^s|_f$  in  $G^s$ 
5.     Split  $n^v|_f$  in  $n_x^s|_f$ 
6.   end for
7.   for  $e_x^s|_f$  in  $G^s$ 
8.     Split  $e_i^v|_f$  in  $e_x^s|_f$ 
9.   end for
10.  Cal and find  $G^v|_f$ 
11.  Sort  $G^v|_f$  by LA, the num of  $n_i^s|_f$  and  $e_i^s|_f$ , the ratio of resource utilization
12.  Or by random
13. end if
14. Stage one:
15.  for  $n_i^s|_f$  in  $G^v|_f$  which has  $n^v|_f$ 
16.    if the num of  $n_i^s|_f$  is 1
17.      make 单节点迁移
18.    else
19.      make 多节点迁移
20.    end if
21.  end for
22.  OUTPUT:  $M_N(n^v|_f) \leftrightarrow n^s$ 
23. Stage two:
24.  if have  $e_x^s|_f$  or stage one complete
25.    for  $G^v|_f$  make 连通性分析
26.  end if
27. Stage three:
28.  if the num of  $e_x^s|_f$  is 1
29.    make 单链路迁移

```

```

30. else
31.     make 多链路迁移
32. end if
33. OUTPUT:  $M_N(e^v|_f) \leftrightarrow e^s$ 

```

恢复流程图如图 3 所示。

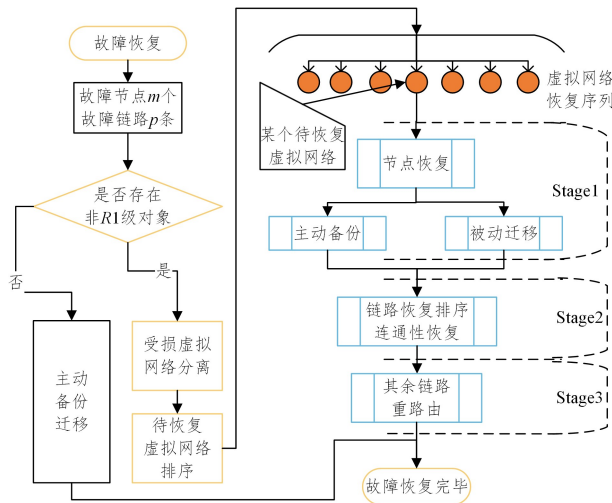


图 3 物理网络故障恢复流程图

Fig. 3 Flowchart of physical network fault recovery

6 实验仿真

实验在同样的底层物理网络环境下,面对随机的虚拟网络请求,研究分析文献[14]的 SVNE-DNRE 方法、文献[15]的 D-SViNE 方法和本文提出的基于最小生成树优先保障的故障恢复算法 (MST-vSDNFR) 在故障恢复时长以及故障恢复率上的差异。对比算法介绍如表 1 所列。

表 1 算法对照

Table 1 Algorithm control

Algorithm name	A brief description of the algorithm
MST-vSDNFR	考虑节点与链路的网络拓扑与资源属性,对不同等级的节点与链路制定不同的主动备份与被动迁移故障保护方案;通过优先恢复虚拟网络的生成树,提高故障恢复率并缩短恢复时长
SVNE-DNRE	考虑时延权重部署控制器,基于网络资源关键度对节点和链路进行映射,故障恢复时针对不同网络故障类型采取备份切换或可行迁移的策略 ^[15]
D-SViNE	联合先验式保护和后验式故障恢复的方式,通过对底层网络剩余资源感知和预测,尽可能地提高虚拟网络请求的接受率,同时兼顾虚拟网络和物理关键性节点和链路的可靠性 ^[16]

表 2 网络分类及参数设置

Table 2 Network classification and parameter setting

NO.	Network name	Topology type	Total number of nodes	K	Bandwidth resources	Computing resource	
			(Switch+Controller)		(Data/control) Unit;Gbit/s	Cpu Unit;GB (Switch/Controller)	Tcam Unit;GB (Switch/Controller)
1	Underlying physical network	topo-fat-tree	96+4	About 0.026	40/40	12/48	250/1000
2		topo-random	96+4	0.026	$U[35,45]/40$	$U[8,16]/4 * cu(n_g^s)$	$U[200,300]/4 * tm(n_g^s)$
3	Virtual network		$U[2,9]+1$	0.5	11/11	10/10	10/10

6.1 仿真环境设置

为验证本文提出的算法的适用性,分别采用 Fat-tree 网络和随机拓扑网络两种类型作为底层物理网络来进行仿真验证(以上两类网络分别记为 topo-fat-tree 和 topo-random)。由于 Fat-tree 具有较为固定的拓扑结构,当节点固定时链路数也相对固定,故本文实验中节点数都取 100。虚拟网络采用随机拓扑生成。详细参数如表 2 所列,其中 $U[a,b]$ 表示数据服从从 a 到 b 的均匀分布。

受限于实验条件,本文实验经 Visual Studio 软件平台使用 C# 语言编写,在 1000 个时间单元内,每 5 个时间单元产生一个虚拟网络请求,持续时长记为 $T, T \sim P(\lambda), \lambda=50$ 的泊松分布,单位为秒。故障的发生时刻记为 t_{fail} ,在每 100 个时间单元内服从 $\lambda=50$ 的泊松分布,每次故障发生时节点与链路故障的数量 m 和 q 都服从 $[0,4]$ 的均匀分布,具体故障节点与链路随机产生。

6.2 实验对比分析

6.2.1 实验一

对比分析算法 MST-vSDNFR 在 topo-random 上各等级节点与链路比例 R1-R2-R3 分别为 10%-30%-60%, 15%-30%-55%, 20%-30%-50%, 10%-25%-65% 以及 10%-20%-70% 时,在 1000 个单位时间内接受到 200 个相同虚拟网络后的虚拟网络请求接受率,以及在遭遇 9 次随机故障后的恢复时长和故障恢复率情况。

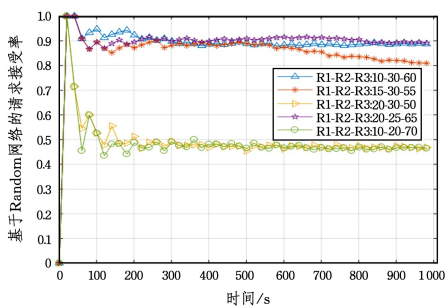


图 4 基于 Random 网络的不同备份比例下的请求接受率

Fig. 4 Request acceptance rate based on Random network with different backup ratios

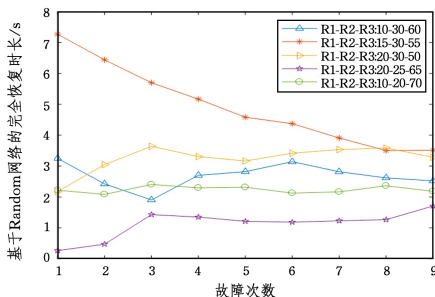


图 5 基于 Random 网络的不同备份比例下的恢复时长

Fig. 5 Recovery duration at different backup ratios based on Random network

从图 4、图 5 可看出,当 R1-R2-R3 为 10%-25%-65% 时, Random 网络能在取得最高请求接受率的同时,获得最低的故障恢复时长。分析图 4、图 5 中的 5 种比例数值可看出,恒定 R2 为 30% 时, R1 取 10% 性能最佳;恒定 R1 为 10% 时, R2 取 25% 性能最佳。所以算法 STP-vSDNFR 在面对 Random 网络时应将备份与迁移的比例 R1-R2-R3 调整为 10%-25%-65%,从而获取最大的收益。

从图 6、图 7 可看出,当 R1-R2-R3 为 10%-20%-70% 时, Fat-tree 网络能在取得最高请求接受率的同时,获得最低的故障恢复时长。分析图 6、图 7 中的 5 种比例数值可看出,恒定 R2 为 30% 时, R1 取 10% 性能最佳;恒定 R1 为 10% 时, R2 取 20% 性能最佳。所以算法 MST-vSDNFR 在面对 Fat-tree 网络时应将备份与迁移的比例 R1-R2-R3 调整为 10%-20%-70%,从而获取最大的收益。

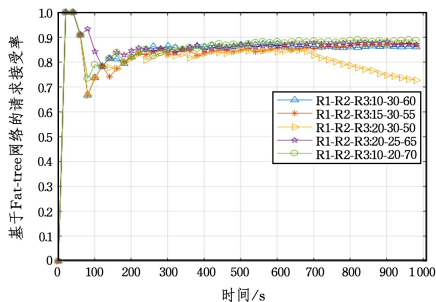


图 6 基于 Fat-tree 网络的不同备份比例下的请求接受率
Fig. 6 Request acceptance rates in Fat-tree networks at different backup ratios

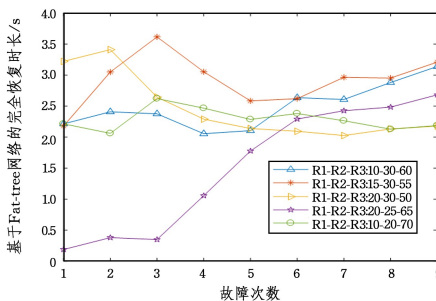


图 7 基于 Fat-tree 网络的不同备份比例下的恢复时长
Fig. 7 Recovery duration based on different backup ratios of Fat-tree networks

从图 4—图 7 可看出,算法 MST-vSDNFR 能较好地适应 Random 网络和 Fat-tree 网络,通过合理调整备份与迁移的节点和链路比例,可以同时取得较高的虚拟网络请求接受率和较低的故障恢复时长。

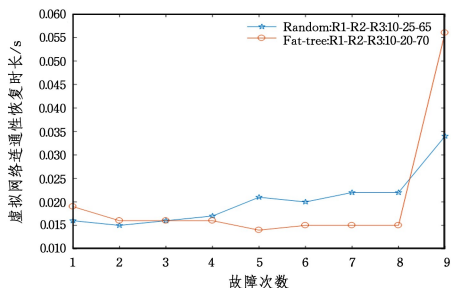


图 8 基于不同物理网络的虚拟网络连通性恢复时长
Fig. 8 Recovery duration of virtual network connectivity based on different physical networks

从图 8 可看出,算法 MST-vSDNFR 能在极短的时间内恢复故障受损虚拟网络的连通性,保障其故障恢复的快速响应。这主要是因为恢复虚拟网络的连通性往往只需要保证 n 个节点和 $n-1$ 条链路的正常运转,远少于总体链路数的要求。

6.2.2 实验二

对比分析算法 MST-vSDNFR、算法 SVNE-DNRE 以及算法 D-SViNE 在 Random 物理网络上,在 1000 个单位时间内接受到 200 个相同虚拟网络后的虚拟网络请求接受率,以及在遭遇 9 次随机故障后的故障恢复情况。

从图 9 可看出,算法 STP-vSDNFR 请求接受率居于算法 SVNE-DNRE 和算法 D-SViNE 中间,但差值都在 5% 以内,且都高于 70%。

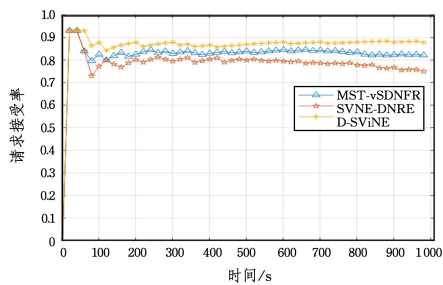


图 9 请求接受率对比分析图

Fig. 9 Comparison analysis of request acceptance rate

从图 10 可看出,算法 STP-vSDNFR 的虚拟网络在故障后的完全恢复时长更短,缩减了约 50%,远优于其余两种算法。

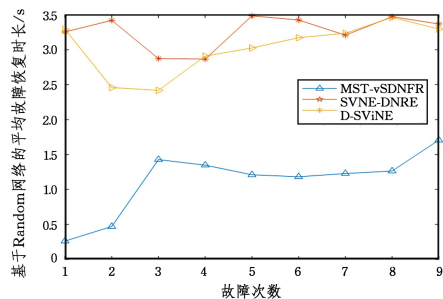


图 10 基于 Random 网络的平均故障恢复时长

Fig. 10 Average fault recovery time based on Random network

其原因在于本文算法为缩短故障恢复时长,使用了较多的资源用于主动备份,导致可用映射资源较少,一定程度上降低了请求接受率。

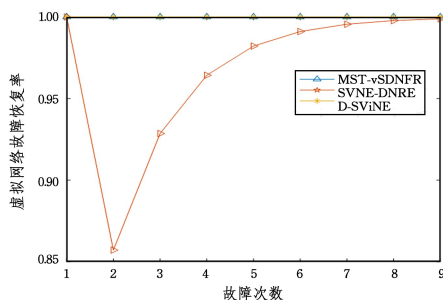


图 11 虚拟网络故障恢复率对比分析图

Fig. 11 Comparative analysis of virtual network failure recovery rate

从图 11 可看出,算法 MST-vSDNFR 与算法 D-SViNE 的故障恢复率相同,且优于算法 SVNE-DNRE。在遭遇故障后算法 MST-vSDNFR 与算法 D-SViNE 都能完全恢复故障

受损的虚拟网络。分析原因在于算法 MST-vSDNFR 与算法 D-SViNE 都采用了主动备份与被动迁移混合的故障恢复方案,提升了在遭遇故障后的故障恢复能力。

总结以上分析,说明相比算法 SVNE-DNRE 和 D-SViNE,算法 MST-vSDNFR 能更快、更好地恢复故障后受损的虚拟网络,在保障虚拟网络请求接受率和故障恢复率的同时,缩短虚拟网络故障恢复时间。

结束语 本文重点研究如何提升虚拟软件定义网络的故障恢复性能,提出了基于生成树优先保障的故障恢复算法。通过遍历搜寻的方式,调整主动备份和被动迁移的节点和链路比例,提升在不同物理网络下的请求接受率,并缩短故障恢复时长;通过采用改进的生成树方法优先恢复故障虚拟网络的连通性,保障虚拟网络故障恢复率的同时,减少恢复时间。

下一步将研究虚拟软件定义网络的故障机理以及故障预测,以提升网络整体的可生存性。

参 考 文 献

- [1] LI Q, LIN Y, ZHU Z J, et al. BOND: Flexible Failure Recovery in Software Defined Networks[J]. *Computer Networks*, 2018, 149:1-12.
- [2] LIU X R. Research on Virtual Network Mapping Strategy based on survivability [D]. Beijing: Beijing University of Posts and Telecommunications, 2018.
- [3] SUN G, YU H, LI L, et al. Efficient algorithms for survivable virtual network embedding[C]// *Asia Communications and Photonics Conference and Exhibition*. 2010:531-532.
- [4] GHALEB A M, KHALIFA T, SHABAN K B. Enhancing the performance of post-failure restoration schemes in multi-tenant networks[C]// *2016 12th International Conference on Network and Service Management (CNSM)*. 2016:359-362.
- [5] SWARNA D B, MUTHUMANIKANDAN V. Nested Failure Detection and Recovery in Software Defined Networks[C]// *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. 2019:1-6.
- [6] HE F, SATO T, OKI E. Master and Slave Controller Assignment Model Against Multiple Failures in Software Defined Network[C]// *2019 IEEE International Conference on Communications (ICC 2019)*. 2019:1-6.
- [7] KIM W, HONG J W, SUH Y. T-DCORAL: A Threshold-Based Dynamic Controller Resource Allocation for Elastic Control Plane in Software-Defined Data Center Networks[C]// *IEEE Communications Letters*. 2019:198-201.
- [8] ALOWA A, FEVENS T. A Dynamic Recovery Module for In-band Control Channel Failure In Software Defined Networking [C]// *2020 6th IEEE Conference on Network Software Virtualization (NetSoft)*. 2020:209-217.
- [9] YANG K, ZHANG B, GUO D, et al. Partitioned Controller Placement in SDWANs for Reliability Maximization with Latency Constraints[C]// *2019 IEEE Globecom Workshops (GC Wkshps)*. 2019:1-6.
- [10] CHEN S, SUN W, HU W. On Dynamic Hypervisor Placement in Virtualized Software Defined Networks (vSDNs)[C]// *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. 2020:1-5.
- [11] BLENK A, BASTA A, ZERWAS J, et al. Pairing SDN with network virtualization: The network hypervisor placement problem [C]// *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*. 2015:198-204.
- [12] XIE F, MENG X R, MENG Q W, et al. Virtual network reconfiguration algorithm based on dynamic path splitting [J]. *Fire & Command Control*, 2019, 44(11):29-34, 40.
- [13] SANDHU A K, SINGH BATH R, NAGPAL A. Improved QoS Using Novel Fault Tolerant Shortest Path Algorithm in Virtual Software Defined Network (VSDN)[C]// *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*. 2019:383-388.
- [14] RAN J P, ZHAO S H, WANG X, et al. Survivability virtual network mapping algorithm for SDN [J]. *Systems Engineering and Electronics*, 2020, 42(5):1182-1189.
- [15] ZHAO J H, WU D D, QU Y, et al. Reliability virtual network mapping guarantee mechanism based on software-defined network [J]. *Computer Applications*, 2020, 40(3):770-776.
- [16] LI R, ZHANG J, TAN Y, et al. An enhanced virtual cluster embedding strategy with virtualized SDN[C]// *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. 2017:974-981.
- [17] LI R, ZHANG J, TAN Y, et al. Topology-aware VSDN embedding approach on shared software defined networks[C]// *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*. 2016:455-460.
- [18] REN Z M. Research progress of node influence in dynamic complex networks [J]. *Acta Physica Sinica*, 2020, 69(4):24-32.
- [19] LIU Y Y, LI L J, SHAN N, et al. Link prediction method based on fusion aggregation coefficient [J]. *Computer Applications*, 2020, 40(1):28-35.
- [20] LIU S Q. Research on Similarity Measurement of Network Nodes Based on Relative Entropy [D]. Taiyuan: Shanxi University, 2020.
- [21] LI F, MA J Z, ZHU P C, et al. Review of reliability evaluation methods for complex network [J]. *Measurement & Control Technology*, 2017, 36(4):1-5, 10.
- [22] WANG Z X, JIANG D L, QI L, et al. Complex Network Invulnerability and Node Importance Evaluation Model Based on Redundancy [J]. *Complex Systems and Complexity Science*, 2020, 17(3):78-85.



CHEN Gang, born in 1997, postgraduate. His main research interests include network virtualization and network survivability.