

网络切片下基于粒子群的虚拟业务故障恢复算法

张筱筠

国网河南能源互联网电力设计院有限公司, 河南 郑州
Email: 13525510160@163.com

收稿日期: 2021年5月28日; 录用日期: 2021年6月14日; 发布日期: 2021年6月28日

摘要

网络切片环境下, 如何在有限资源约束下恢复尽可能多的虚拟网服务是一个急需解决的问题。为解决此问题, 本文将虚拟网服务恢复问题建模为失效业务恢复数量最大化问题。提出了网络切片下基于粒子群的虚拟业务故障恢复算法, 该算法首先将故障资源和虚拟业务构建为二层关联模型。其次将网络资源的恢复问题建模为粒子群问题, 并采用粒子群优化算法进行求解。在实验部分, 通过与已有算法进行比较, 验证了本文算法能够恢复更多的发生故障的虚拟网业务。

关键词

网络切片, 底层网络, 虚拟网络, 粒子群, 故障恢复

Virtual Service Failure Recovery Algorithm Based on Particle Swarm in Network Slicing

Xiaoyun Zhang

State Grid Henan Energy Internet Electric Power Design Institute Co., Ltd., Zhengzhou Henan
Email: 13525510160@163.com

Received: May 28th, 2021; accepted: Jun. 14th, 2021; published: Jun. 28th, 2021

Abstract

In a network slicing environment, how to restore as many virtual network services as possible under the constraints of limited resources is an urgent problem to be solved. To solve this problem, this paper models the virtual network service recovery problem as a problem of maximizing

the number of failed services recovery. A particle swarm-based virtual service failure recovery algorithm under network slicing is proposed. The algorithm first constructs the faulty resource and virtual service as a two-layer correlation model. Secondly, the network resource recovery problem is modeled as a particle swarm problem, and the particle swarm optimization algorithm is used to solve it. In the experimental part, by comparing with existing algorithms, it is verified that the algorithm in this paper can recover more virtual network services that have failed.

Keywords

Network Slicing, Underlying Network, Virtual Network, Particle Swarm, Failure Recovery

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

网络切片环境下, 基础网络被划分为底层网络和虚拟网络。底层网络包括底层节点和底层链路。虚拟网络包括虚拟节点和虚拟链路。服务提供商通过租用底层网络的资源组建虚拟网络, 从而快速为目标用户提供特定的虚拟网服务[1] [2]。由于网络切片后, 底层网络上承载的虚拟网服务数量越来越多, 如果底层网络出现故障, 将导致越来越多的虚拟网服务不可用。为解决此问题, 网络故障恢复已成为一个重要的研究内容。

为提升故障的虚拟网业务恢复的速度, 文献[3]基于虚拟网的动态迁移特性, 设计了基于迁移策略的故障恢复算法。为提升关键业务的可靠性, 文献[4]分析了网络业务的优先级, 对于关键业务进行优先恢复。针对大规模环境下的故障恢复问题, 文献[5]设计了一种分布式环境下的虚拟网资源恢复算法, 提升了虚拟网资源的恢复速度。为进一步提升恢复资源率, 文献[6]针对需要恢复的故障资源的特点, 提出分阶段恢复策略, 从而提升了算法的性能。对于具体网络环境下的故障资源恢复问题, 文献[7]针对光网络中的故障链路恢复能力低的问题, 将流量多播技术应用到流量疏导策略中, 提升了故障资源的恢复能力。文献[8]针对已有研究故障恢复能力低的问题, 以 SDN 网络为实验场景, 基于客户信息链路的故障阈值对需要恢复的故障链路进行判断, 从而有效提升故障恢复的准确度。

已有研究已经取得了较多的研究成果, 但是, 如何在有限资源约束下恢复尽可能多的虚拟网服务, 仍然需要进一步研究。为解决此问题, 本文将虚拟网服务恢复问题建模为失效业务恢复数量最大化问题, 并采用智能化算法, 求解最优的失效业务恢复策略。通过实验, 验证了本文算法能够恢复更多的发生故障的虚拟网业务。

2. 问题描述

网络切片环境下, 基础网络被划分为底层网络和虚拟网络。底层网络使用 $G_S = (N_S, E_S)$ 表示, 其中, N_S 表示底层节点集合, 每个底层节点 $n_i^s \in N_S$ 具有计算资源属性, 使用 $cpu(n_i^s)$ 表示。 E_S 表示底层链路集合, 每条底层链路 $e_j^s \in E_S$ 具有带宽资源属性, 使用 $bw(e_j^s)$ 表示。虚拟网络使用 $G_V = (N_V, E_V)$ 表示。其中, N_V 表示虚拟节点集合, 每个虚拟节点 $n_i^v \in N_V$ 需要向底层节点申请的计算资源量使用 $cpu(n_i^v)$ 表示。 E_V 表示虚拟链路集合, 每条虚拟链路 $e_j^v \in E_V$ 需要向底层链路申请的带宽资源量使用 $bw(e_j^v)$ 表示。

服务提供商向底层网络提供商申请资源时, 使用 $G_V = (N_V, E_V)$ 描述需要的资源量。底层网络服务商

为虚拟网络服务提供商提供资源时,使用 $G_v \downarrow G_s$ 描述资源分配情况。其中,底层节点 n_i^s 为虚拟节点 n_i^v 分配资源,使用 $n_i^v \downarrow n_i^s$ 表示。底层路径 p_j^s 为虚拟链路 e_j^v 分配资源,使用 $e_j^v \downarrow p_j^s$ 表示。底层路径 p_j^s 是指由多条底层链路相连接而组成的一条底层路径,该路径的两个端节点为虚拟链路的两个虚拟端点映射的两个底层节点。虚拟网服务提供商租用虚拟网后,可以部署个性化的业务,为最终用户提供服务。本文主要研究端到端的虚拟网业务,使用 S_{ij}^v 表示从虚拟节点 n_i^v 到虚拟节点 n_j^v 的一个端到端业务。

3. 故障恢复模型

故障恢复问题的难点是如何使用有限数量的恢复资源,恢复尽可能多的故障资源,从而实现恢复业务的最大化。对于有限数量的恢复资源,包括总的节点恢复资源、总的链路恢复资源,分别使用 Re^n 和 Re^e 表示。

对于底层节点恢复的资源限制问题,可以使用公式(1)描述。公式的含义是成功恢复的底层节点占用的恢复资源的总数量,不能超过总的节点恢复资源量。其中, re_m^n 表示故障节点 m 需要消耗的节点资源数量, F^n 表示需要恢复的发生故障的节点的集合。 R_m^n 表示故障节点 m 是否成功恢复的标识,取值为 $R_m^n = \{0,1\}$ 。当取值为 1 时,表示故障节点 m 已经被成功恢复。当取值为 0 时,表示故障节点 m 没有被成功恢复。

$$\sum_{m \in F^n} R_m^n re_m^n \leq Re^n \quad (1)$$

对于底层链路恢复的资源限制问题,可以使用公式(2)描述。公式的含义是成功恢复的底层链路占用的恢复资源的总数量,不能超过总的链路恢复资源量。其中, r_{mn}^e 表示恢复故障链路(mn)需要消耗的链路资源数量, F^e 表示需要恢复的发生故障的链路组成的集合。 R_{mn}^e 表示故障链路(mn)是否成功恢复的标识,取值为 $R_{mn}^e = \{0,1\}$ 。当取值为 1 时,表示故障链路(mn)被成功恢复。当取值为 0 时,表示故障链路(mn)没有被恢复成功。

$$\sum_{(mn) \in F^e} R_{mn}^e r_{mn}^e \leq Re^e \quad (2)$$

当发生故障的底层节点和底层链路被恢复成功后,可以用于恢复虚拟网业务。在恢复虚拟网业务时,恢复业务占用的底层节点资源和底层链路资源不能超过其自身的空闲资源量。在底层节点分配资源约束方面,底层节点 $n_m^s \in N_s$ 的资源分配约束使用公式(3)进行计算。该公式表示恢复的业务占用的底层节点的资源量不能超过底层节点修复后的资源限制。其中, $c_{S_{ij}^v}^m$ 表示被恢复的发生故障业务 S_{ij}^v 需要底层节点 n_m^s 为其分配的计算资源数量; c_m 表示发生故障的底层节点 n_m^s 剩余的计算资源容量, C_m 表示底层节点 n_m^s 总共具有的计算资源数量, C_g 表示底层节点 n_m^s 发生故障后仍然可承载的虚拟业务所占用的计算资源数量。

$$\sum_{S_{ij}^v \in S_f^v} c_{S_{ij}^v}^m R_{S_{ij}^v} \leq c_m + (C_m - c_m) R_m^n - C_g \quad (3)$$

在底层链路分配资源约束方面,底层链路 $e_j^s \in E_s$ 的资源分配约束使用公式(4)进行计算。该公式表示恢复的业务占用的底层链路的资源量不能超过底层链路修复后的资源限制。其中, $e_{S_{ij}^v}^{mn}$ 表示被恢复的发生故障业务 S_{ij}^v 需要底层链路(mn)为其分配的带宽资源数量; e_{mn} 表示发生故障的底层链路(mn)剩余的带宽资源容量, B_{mn} 表示底层链路(mn)总共具有的带宽资源数量, B_g 表示底层链路(mn)发生故障后仍然可承载的虚拟业务所占用的带宽资源数量。

$$\sum_{S_{ij}^v \in S_f^v} e_{S_{ij}^v}^{mn} R_{S_{ij}^v} \leq e_{mn} + (B_{mn} - e_{mn}) R_{mn}^e - B_g \quad (4)$$

基于此,本文定义虚拟网业务故障恢复问题的目标函数为公式(5),其约束条件为公式(1~4),公式的

含义是实现失效业务的恢复数量最大化,即在资源受限的条件下,恢复尽可能多的虚拟网业务。公式中, $f(S_f^v)$ 表示成功恢复的虚拟网业务的数量, S_f^v 表示发生故障的虚拟网业务集合。 $R_{S_{ij}^v}$ 表示虚拟业务 $S_{ij}^v \in S_f^v$ 是否被成功恢复的标识,取值为 $R_{S_{ij}^v} = \{0,1\}$ 。当取值为1时,表示虚拟业务 $S_{ij}^v \in S_f^v$ 被成功恢复。当取值为0时,表示虚拟业务 $S_{ij}^v \in S_f^v$ 未被成功恢复。

$$\max f(S_f^v) = \sum_{S_{ij}^v \in S_f^v} R_{S_{ij}^v} \quad (5)$$

4. 算法

从目标函数及其约束条件可知,要求解最优化的解是 NP 难题。考虑到粒子群优化算法具有求解速度快、求解效果好等优点,为了获得最优解,本文采用粒子群优化算法进行求解。粒子群优化算法是一种通过粒子间互相学习而指导搜索最优解的智能化算法。粒子群优化算法的两个关键参数是粒子的位置和粒子的速度。

4.1. 算法分析

本文中,粒子的位置是指一个资源恢复方案。粒子的速度是指粒子从一个粒子位置向更优化的粒子位置运动的过程。使用 $X_i = [x_i^1, x_i^2, \dots, x_i^D]$ 表示由 D 个粒子位置构成的向量,被定义为第 i 个恢复方案。其中, D 表示需要恢复的故障底层节点 F^n 和故障底层链路 F^e 构成的集合。集合中的每个元素 x_i^j 表示当前的故障底层资源是否恢复,取值为二进制数值。当其取值为1时,表示对当前的故障底层资源进行恢复。当其取值为0时,表示对当前的故障底层资源不进行恢复。

粒子的速度用来表示资源恢复方案的优化策略,使用向量 $V_i = [v_i^1, v_i^2, \dots, v_i^D]$ 表示。其中,向量元素 v_i^j 表示第 j 个需要恢复的网络资源的恢复状态是否需要改变。当 $v_i^j = 0$ 表示需要进行改变。当 $v_i^j = 1$ 表示不需要进行改变。每个粒子运动时,使用 X_{pb} 表示其历史最优的位置,使用 X_{gb} 表示全局历史最优的位置。

在粒子群优化算法中,为了对恢复方案进行优化,一般采用粒子的加法、减法、乘法三种运算。粒子的加法运算使用 \oplus 表示,用于计算恢复方案的优化策略。计算方法如公式(6)所示。其中, P_i 和 P_j 表示采取现有恢复方案的概率,且 $P_i + P_j = 1 (0 \leq P \leq 1)$ 。当某个恢复方案的概率不确定时,使用*表示。例如, $0.1 \times (1,0,0,1,1) \oplus 0.9 \times (1,0,1,0,1) = (1,0,*,*,1)$ 。其中,第一个*表示当前故障资源的恢复方案是以0.1的概率取0,以0.9的概率取1。

$$V_i \text{ 和 } V_j \text{ 的优化策略} = P_i V_i \oplus P_j V_j \quad (6)$$

减法运算使用 \ominus 表示,用于比较两种恢复方案的区别。计算方法如公式(7)所示。计算方法为比较每维度上的取值。如果相等,减法的计算结果为1。如果不相等,减分的计算结果为0。例如, $(1,1,0,0,0) \ominus (1,0,1,0,1) = (1,0,0,1,0)$ 。

$$X_i \text{ 和 } X_j \text{ 区别} = X_i \ominus X_j \quad (7)$$

乘法运算使用 \otimes 表示,用于计算出新的更加优化的方案。计算方法如公式(8)所示。

例如, $(1,1,1,0,0) \otimes (1,0,1,1,1)$ 表明恢复方案中第二个需要恢复的网络资源的恢复方案需要调整。

$$\text{粒子位置 } X_i \text{ 的新位置} = X_i \otimes V_i \quad (8)$$

基于上述计算方法,可以得到粒子位置的优化方法如公式(9),粒子速度的优化公式如公式(10)。

$$X_{i+1} = X_i \otimes V_{i+1} \quad (9)$$

$$V_{i+1} = P_1 V_i \oplus P_2 (X_{pb} \ominus X_i) \oplus P_3 (X_{gb} \ominus X_i) \quad (10)$$

4.2. 算法描述

通过粒子群算法的过程分析, 本文提出的基于粒子群的虚拟业务故障恢复算法(Virtual Service Failure Recovery Algorithm based on Particle Swarm, VSFRAoPW)如表 1 所示。算法包括下面七个步骤。1) 构建二层关联模型: 顶层节点为故障资源, 底层节点为虚拟业务, 从顶层节点到底层节点的连线, 表示当前的故障资源为其连接的虚拟业务提供网络资源。2) 构建故障资源的二进制字符串。使用故障的网络节点集合 F^n 、故障的网络链路集合 F^e 构建一个二进制字符串 X_i , 每位表示当前网络资源是否被恢复。3) 参数初始化。根据粒子群建模和分析可知, 需要初始化的参数包括算法的迭代次数 MG 、粒子群规模 N 、随机生成粒子的初始位置 X_i 、随机生成粒子的初始速度 V_i 。4) 计算初始位置。虚拟网业务故障恢复问题的目标函数为公式(5), 所以, 将公式(5)设置为每个粒子位置的适应度函数值 $f(X_i)$ 。5) 粒子速度和粒子位置更新。在恢复虚拟网业务时, 需要满足的业务资源分配约束包括底层节点分配资源约束、底层链路分配资源约束。所以, 判断各个粒子位置是否符合约束条件时, 将约束条件设置为公式(1)到公式(4), 并进行相应操作。6) 更新全局最优初始位置和个体最优初始位置。通过判断全局最优初始位置 X_{gb} 和个体最优初始位置 X_{pb} 的取值, 对全局最优初始位置和个体最优初始位置进行更新。7) 判断是否达到结束条件。判断的条件为最大的迭代次数 MG ; 如已达到最大迭代次数, 输出最优的 X_i 。

Table 1. Virtual service failure recovery algorithm based on particle swarm

表 1. 基于粒子群的虚拟业务故障恢复算法

输入: 虚拟网 $G_v = (N_v, E_v)$ 、底层网络 $G_s = (N_s, E_s)$ 、发生故障的网络节点集合 F^n 、发生故障的网络链路集合 F^e 、节点总恢复资源 Re^n 、链路总恢复资源 Re^e 、迭代次数 MG 、粒子群规模 N ;
输出: 发生故障的网络资源的恢复方案 X_i

- 1) 构建故障资源和虚拟业务的二层关联模型。基于故障资源和虚拟业务的映射关系, 构建二层关联模型;
- 2) 构建故障资源的二进制字符串 X_i , 每位表示当前网络资源是否被恢复;
- 3) 参数初始化。初始化的参数包括迭代次数 MG 、粒子群规模 N 、随机生成粒子的初始位置 X_i 、随机生成粒子的初始速度 V_i ;
- 4) 计算初始位置。使用公式(5)计算每个粒子位置的适应度函数值 $f(X_i)$, 并将最优的初始位置 X_i 设置为全局最优初始位置 X_{gb} , 将每个粒子的初始位置 X_i 设置为个体最优初始位置 X_{pb} ;
- 5) 粒子速度和粒子位置更新。判断各个粒子位置是否符合约束条件, 并进行相应操作。如符合公式(1)到公式(4)的约束条件, 使用公式(6)、公式(7)分别对粒子速度、粒子位置进行更新; 如不符合约束条件, 随机生成新的粒子位置和粒子速度。
- 6) 更新全局最优初始位置和个体最优初始位置。对全局最优初始位置 X_{gb} 、个体最优初始位置 X_{pb} 进行更新。当 $f(X_i) < f(X_{pb})$ 时, 设置 $X_{pb} = X_i$; 当 $f(X_{pb}) < f(X_{gb})$ 时, 设置 $X_{gb} = X_{pb}$ 。
- 7) 判断是否达到结束条件。判断是否达到最大的迭代次数 MG ; 如已达到最大迭代次数, 输出最优的 X_i 。否则, 返回步骤 5。

5. 性能分析

实验环节使用 GT-ITM [9]工具生成网络拓扑环境。在底层网络方面, 底层网络节点数量从 100 个增加到 700 个, 用于描述不同网络规模环境下的网络环境。每个底层节点的资源、每条底层链路的资源, 都服从(10, 30)的均匀分布。在虚拟网络方面, 虚拟网络节点数量服从(10, 30)的均匀分布, 每个虚拟节点的资源、每条虚拟链路的资源, 都服从(1, 5)的均匀分布。因底层链路故障和底层节点故障的恢复类似, 本文实验以底层链路故障为研究对象。在仿真底层链路故障方面, 每条链路发生故障的概率为 0.05。当某条底层链路发生故障后, 其剩余的链路容量服从(0, 10)的均匀分布。

在算法比较方面, 将本文算法 VSFRAoPW 与启发式恢复算法(Heuristic recovery algorithm, VSFR-AoHR)进行比较。算法 VSFRAoHR 以恢复故障业务数量最大化为目标, 采取优先恢复资源需求少的虚拟网业务对应的底层网络链路资源。在比较指标方面, 使用故障恢复率、用户满意度两个指标进行评价。故障恢复率是指被成功恢复的发生故障的虚拟网业务数量在总的虚拟网业务数量种的占比。用户满意度 u 使用公式(11)计算。其中, $flow_{\theta}$ 表示恢复的虚拟网业务的流量。 f_{θ} 表示恢复的虚拟网业务的数量。 Θ 表示已成功恢复的虚拟网业务集合。 Ω 表示需要恢复的虚拟网业务集合。 α 和 β 是调节恢复业务流量和恢复业务数量的权重因子。从公式(11)可知, 恢复的虚拟网业务的数量和流量越多, 公式(11)的分子越大, 用户满意度 u 的取值越大, 算法恢复的效果越好。

$$u = \frac{\alpha \sum_{\theta \in \Theta} flow_{\theta} + \beta \sum_{\theta \in \Theta} f_{\theta}}{\alpha \sum_{\phi \in \Omega} flow_{\phi} + \beta \sum_{\phi \in \Omega} f_{\phi}} \quad (11)$$

故障恢复率的实验结果如图 1 所示。X 轴表示底层网络节点的数量从 100 个增加到 700 个, 用于验证不同的底层网络规模下算法的性能。Y 轴表示虚拟网业务故障的恢复率。实验中使用的底层链路恢复资源总量为 300 个。从实验结果可知, 随着底层网络规模的增加, 两个算法的虚拟网业务故障恢复率都在降低。这是因为底层网络规模增加, 发生故障的虚拟网业务数量增加, 需要的恢复资源也需要增加。从两个算法的性能分析方面, 各种网络环境下, 本文算法的虚拟网业务故障恢复的恢复率都高于对比算法。这是因为本文算法采取的粒子群优化算法, 可以较好的获得全局最优解, 从而提升虚拟网业务的恢复成功率。

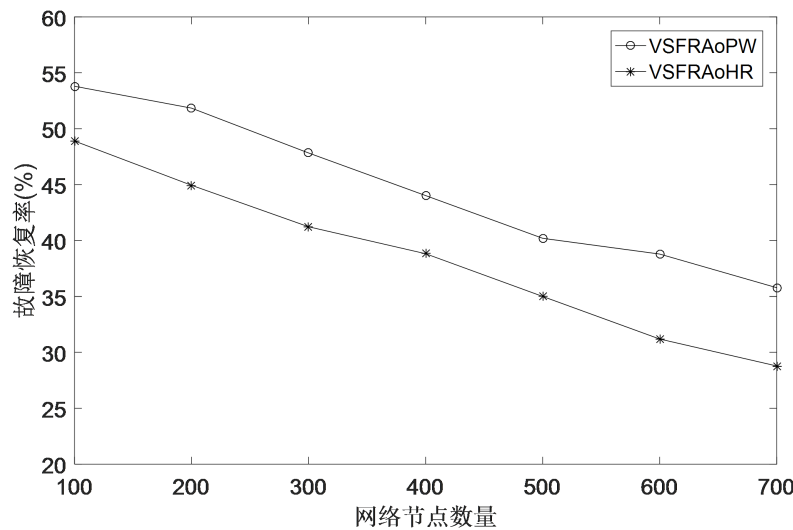


Figure 1. Comparison of failure recovery rates of virtual business
图 1. 虚拟业务故障的恢复率比较

恢复资源总量对用户满意度影响的实验结果如图 2 所示。图中 X 轴表示恢复资源的总量从 100 个增加到 500 个。Y 轴表示用户满意度取值。实验结果展示了底层网络节点数量在 400 个的网络环境下的用户满意度情况。从图可知, 随着恢复资源总量的增加, 两个算法下的用户满意度都在增加, 这是因为恢复资源总量增加后, 可以恢复更多的发生故障的底层网络资源, 从而恢复了较多的发生故障的虚拟网业务。从两个算法的用户满意度分析可知, 各种恢复资源总量环境下, 本文算法的用户满意度都高于比较算法。这是因为本文算法能够找到最优的恢复策略, 从而恢复了较多发生故障的虚拟业务。

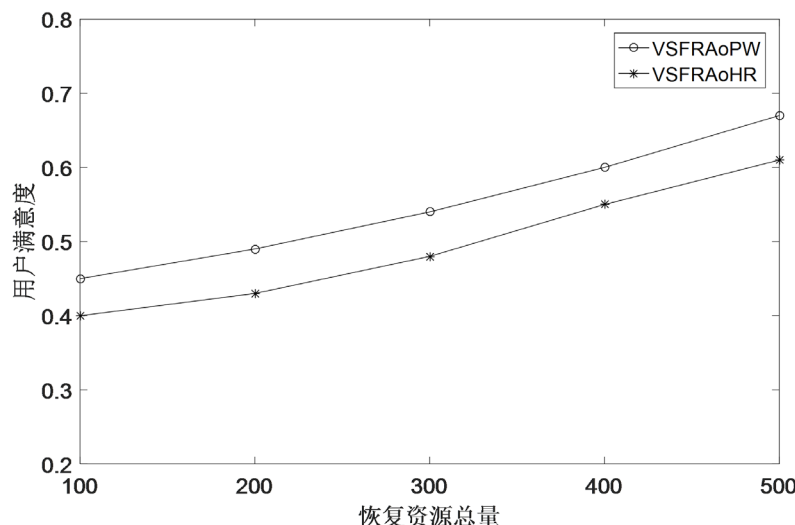


Figure 2. The impact of total restoration resources on user satisfaction
图 2. 恢复资源总量对用户满意度影响

6. 总结

网络切片环境下,网络故障恢复已成为一个重要的研究内容。但是,已有研究对于有限资源约束下的网络故障恢复能力还需进一步提升。为解决此问题,本文将虚拟网服务恢复问题建模为失效业务恢复数量最大化问题,提出了网络切片下基于粒子群的虚拟业务故障恢复算法,并通过实验验证了本文算法能够恢复更多的发生故障的虚拟网业务。由于虚拟网承载在底层网络上,存在多个虚拟网资源共享同一底层网络资源的情况。所以,如果能够恢复被较多虚拟资源共享的底层链路资源,将有效提升虚拟网服务故障的恢复能力。下一步工作中,将基于本文的研究成果,基于虚拟网和底层网络的映射关系,进一步提升虚拟网业务的故障恢复能力。

参考文献

- [1] Peng, M., Li, Y., Jiang, J., *et al.* (2014) Heterogeneous Cloud Radio Access Networks: A New Perspective for Enhancing Spectral and Energy Efficiencies. *IEEE Wireless Communications*, **21**, 126-135. <https://doi.org/10.1109/MWC.2014.7000980>
- [2] Tang, J., Tay, W.P. and Quek, T.Q.S. (2014) Cross-Layer Resource Allocation in Cloud Radio Access Network. 2014 *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, 3-5 December 2014, 158-162. <https://doi.org/10.1109/GlobalSIP.2014.7032098>
- [3] Hawilo, H., Shami, A., Mirahmadi, M., *et al.* (2014) NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC). *IEEE Network*, **28**, 18-26. <https://doi.org/10.1109/MNET.2014.6963800>
- [4] Wang, X. (2011) Network Recovery and Augmentation under Geographically Correlated Region Failures. *Proceedings of the Global Communications Conference, GLOBECOM 2011*, Houston, 5-9 December 2011, 1-5.
- [5] Mijumbi, R., Serrat, J., Gorricho, J.L., *et al.* (2015) Design and Evaluation of Algorithms for Mapping and Scheduling of Virtual Network Functions. *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, 13-17 April 2015, 1-9. <https://doi.org/10.1109/NETSOFT.2015.7116120>
- [6] Yu, H. and Yang, C. (2011) Partial Network Recovery to Maximize Traffic Demand. *IEEE Communications Letters*, **15**, 1388-1390. <https://doi.org/10.1109/LCOMM.2011.103111.111668>
- [7] 鲍宁海, 袁园, 刘自谦, 等. 基于链路生命期的光数据中心网络业务恢复方案[J]. *通信学报*, 2018, 39(8): 125-132.
- [8] 潘志安, 刘庆杰, 王小英. 软件定义网络客户信息链路故障恢复仿真[J]. *计算机仿真*, 2018(5): 241-244.
- [9] Zegura, E.W., Calvert, K.L. and Bhattacharjee, S. (1996) How to Model an Internetwork. *Proceedings of IEEE INFO-COM'96. Conference on Computer Communications*, San Francisco, 24-28 March 1996, 594-602.