

面向软件定义广域网的路径可编程性保障研究综述

郭泽华^{*①②} 窦松石^{①②} 齐力^{①②} 兰巨龙^③

^①(北京理工大学 北京 100081)

^②(复杂系统智能控制与决策教育部重点实验室 北京 100081)

^③(国家数字交换系统工程技术研究中心 郑州 450002)

摘要: 软件定义网络(SDN)被誉为下一代网络的关键技术。近年来, SDN已经成为学术界与工业界的热点。广域网是SDN应用到工业界的一个重要的场景。基于SDN的广域网被称为软件定义广域网(SD-WAN)。在SD-WAN中, SDN控制器通过控制流转发路径上的SDN交换机来实现流的路径可编程性。然而, 控制器失效是SD-WAN中一种常见的现象。当控制器失效时, 流转发路径上的交换机会失去控制, 流的路径可编程性将无法得到保障, 从而无法实现对网络流量的灵活调度, 导致网络性能下降。该文对SD-WAN控制器失效场景下保证路径可编程性的研究工作进行了综述。该文首先阐述了当控制器失效时, SD-WAN中路径可编程性保障研究的背景及意义。随后, 在查阅分析了国内外相关文献的基础上, 介绍了当前在控制器失效时SD-WAN对交换机的主流控制方案。最后, 对现有研究成果可能的进一步提高之处进行了总结, 并对此研究的未来发展与研究前景进行了展望。

关键词: 软件定义网络; 广域网; 路径可编程性

中图分类号: TN929.5; TP393

文献标识码: A

文章编号: 1009-5896(2023)05-1899-12

DOI: [10.11999/JEIT220418](https://doi.org/10.11999/JEIT220418)

A Survey of Maintaining the Path Programmability in Software-Defined Wide Area Networks

GUO Zehua^{①②} DOU Songshi^{①②} QI Li^{①②} LAN Julong^③

^①(Beijing Institute of Technology, Beijing 100081, China)

^②(Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China)

^③(National Digital Switching System Engineering & Technological Research Center, Zhengzhou 450002, China)

Abstract: Software-Defined Networking (SDN) is the key technique of the next-generation network. Recently, SDN has become a hot spot in both academia and industry. Wide Area Network (WAN) is one of the primary application scenarios in the industry for SDN, which is known as Software-Defined WAN (SD-WAN). In SD-WAN, flexible traffic scheduling and network performance improvement are realized by the flow path programmability, which is enabled by the SDN controller to change dynamically the paths of flows traversing SDN switches. However, controller failure is a common phenomenon. When the controller fails, the switches controlled by the failed controller become offline, and the flows traversing the offline switches become offline too. In this way, the path programmability can not be guaranteed, and thus flexible flow control becomes invalid, leading to severe network performance degradation. This survey is presented to introduce the research works on maintaining path programmability in SD-WAN. First, the path programmability and the important feature for maintaining the path programmability in SD-WAN are introduced. Second, different types of existing solutions for coping with the controller failure in SD-WAN are proposed. Finally, potential improvements and future directions on this research topic are proposed.

Key words: Software-Defined Networking (SDN); Wide Area Network (WAN); Path programmability

收稿日期: 2022-04-08; 改回日期: 2022-06-17; 网络出版: 2022-06-23

*通信作者: 郭泽华 guolizihao@hotmail.com

基金项目: 国家自然科学基金(62002019), 北京理工大学青年教师学术启动计划, 国家重点研发计划(2021YFB1714800)

Foundation Items: The National Natural Science Foundation of China (62002019), Beijing Institute of Technology Research Fund Program for Young Scholars, The National Key Research and Development Program of China (2021YFB1714800)

1 引言

网络技术是信息时代发展的基础,而软件定义网络(Software-Defined Networking, SDN)被誉为下一代网络的关键技术。在传统网络中,如果要部署一个新的网络功能,必须考虑对现有网络设备的前后兼容,这使得新功能难以实现。SDN采用控制平面和转发平面相互分离的方式,控制平面负责网络策略控制以及对转发平面的资源调度,转发平面则根据控制平面的动态策略进行数据转发,从而实现了网络流量的灵活控制,便于新功能的拓展^[1]。

近年来,SDN已经成为学术界的研究热点,并开始被工业界所应用。广域网(Wide Area Network, WAN)是SDN应用在工业界的一个重要场景。基于SDN的广域网被称为软件定义广域网(Software-Defined WAN, SD-WAN)。全球领先的互联网公司谷歌(Google)和微软(Microsoft)分别对其连接全球数据中心的广域网进行了SDN化改造来提高其网络性能^[2,3]。世界上最大的网络运营商之一的美国AT&T公司对其广域网进行基于SDN交换机的虚拟化,提高其服务质量并降低运营成本。据报道,到2018年底,AT&T已经实现其广域网65%的虚拟化,并预期在近年内将虚拟化比例提高到75%^[4]。随着SDN的不断发展,在不久的将来,广域网中的网络设备都将被逐步替换为SDN交换机,从而实现广域网的全面SDN化。

路径可编程性是SDN技术赋予网络的一种重要能力,是实现流量灵活转发的核心。SDN控制器在感知网络状态变化的基础上,将应用平面的流量调度决策转换为具体流量调度执行策略的流表项,并通过下发流表项来改变流的转发路径,进而优化网络性能。然而,控制器失效是一种常见的现象。当控制器失效时,流转发路径上的交换机会失去控制,流的路径可编程性将无法得到保障,从而无法实现对网络流量的灵活调度,这将会导致网络性能下降。控制器失效对SD-WAN的性能有着很严重的影响,因此面对SD-WAN的路径可编程性保证研究已经成为国内外学者的研究热点。

本文对SD-WAN控制器失效场景下保证路径可编程性的研究工作进行了综述。首先,本文阐述了当控制器失效时,SD-WAN中路径可编程性保障研究的背景及意义。随后,在查阅分析了国内外相关文献的基础上,本文详细阐述了主流的两种在控制器失效时SD-WAN对交换机的控制方案,并将当前的研究分类为静态的控制方案与动态的控制方案,更进一步地将其中的动态的控制方案细分为

交换机-控制器映射方案与流-控制器映射方案。最后,本文对现有研究成果可能的进一步提高之处进行了总结,并对本研究的未来发展与研究前景进行展望。

2 路径可编程性保证概述

2.1 路径可编程性介绍

路径可编程性(path programmability)是SDN技术提供的一个重要能力。其主要思想是,SDN控制器在感知网络状态变化的基础上,动态调整网络中流的转发策略,并通过下发流表项来改变流的转发路径,进而优化网络性能。OpenFlow协议是SDN中最具代表性、应用最广的协议^[5]。在OpenFlow中,网络状态监测模块通过从SDN交换机中提取流状态,来动态地获得当前整个网络的状态。然后,控制器对当前状态和前一状态进行对比,判断是否需要网络进行控制。如果对比的结果是网络性能稳定,那么控制器等待下一次状态对比;如果对比的结果是网络性能下降,那么控制器就会进行策略调整(例如,更新网络中一些流的转发路径)来提升网络的性能。流的转发路径更新操作主要包括:流的选择、流的路径计算、通过流表的安装更新和删除来更新流的路径。为了保证网络流转发的正确性,在路径更新的过程中,流通常会在原来的路径上转发,直到控制器确认新的路径建立完毕后,流开始在新的路径上转发。

在SD-WAN中,SDN控制器通过控制流转发路径上的SDN交换机来实现流的路径可编程性。由于广域网范围大、设备多,一个SD-WAN通常被分为多个域,每个域中由一个控制器对该域中的SDN交换机进行控制。虽然这些控制器部署在不同的物理位置,但是它们通过状态同步来维持全网一致的状态信息,构建了逻辑上统一的控制平面。[图1](#)描述了由网络域D1、D2和D3组成的一个SD-WAN。其中,控制器C1、C2和C3部署在不同的物理位置,每个控制器控制一个网络域,并对域中的交换机进行管理。控制器之间通过状态同步来保证全局统一的控制信息^[6]。在[图2](#)中,流f从交换机s21转发至s24。在交换机s21上,控制器可以决定该流转发路径的下一跳为s22或者s23,从而调整该流的转发路径,实现路径可编程性。

2.2 SD-WAN中保证路径可编程性的重要性

流的可编程性依赖于控制器。在SD-WAN中,当SDN控制器失效时,与之相连接的交换机都会失去该控制器的控制,从而变为离线交换机。在[图1](#)中,控制器C2失效后,产生了离线交换机s20、s21、s22、s23和s24。与此同时,流经这些离线交换机中

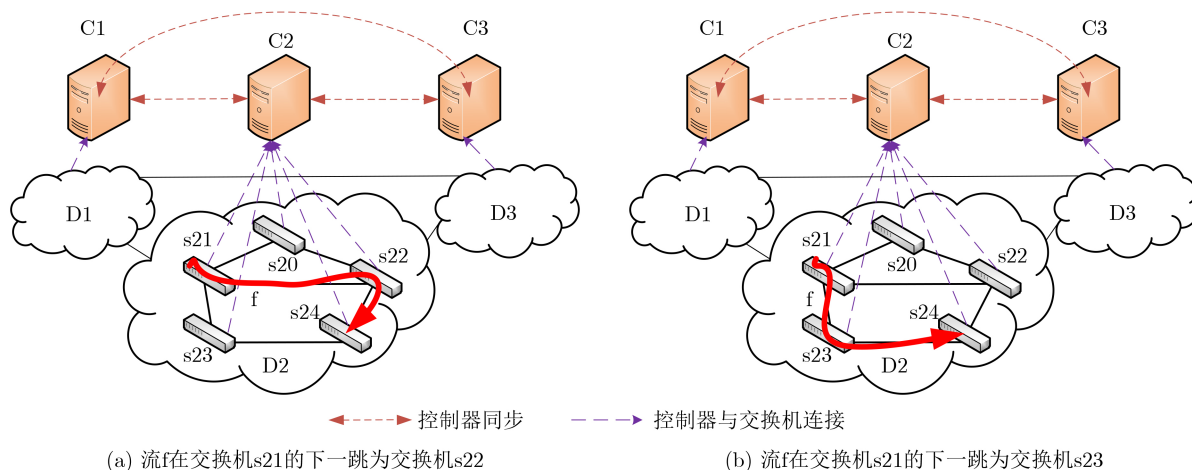


图1 SD-WAN中流f的路径可编程性例子

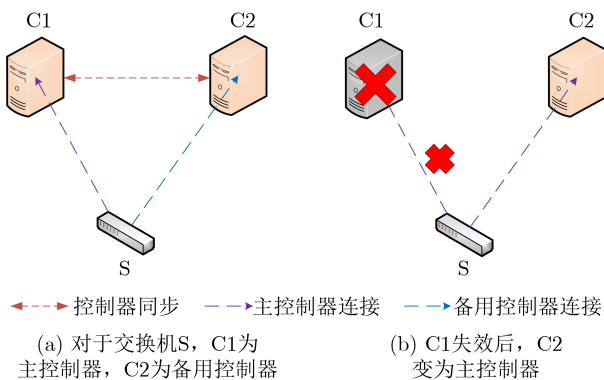


图2 OpenFlow协议中多控制器连接方案的例子

的流也无法受到控制器的控制,从而变成不具有路径可编程性的离线流。因此,当控制器失效时,该失效控制器控制下的流的路径可编程性将无法得到保障。如果离线流的流量产生巨大的变化,可能造成网络拥塞,从而导致网络性能下降。

控制器失效是一种常见的现象^[7]。从本质上来说,SDN控制器是安装了网络操作系统的物理服务器或虚拟机。由于意外因素(例如,自然灾害、硬件/软件错误、电源故障、黑客攻击等),正常运行的控制器可能失效。工业界通常采用集群部署来应对单个控制器失效的问题。例如,部署一个由多个控制器组成的控制器集群来应对单个控制器失效问题。典型的应用案例是工业界广泛应用的SDN控制器开放式网络操作系统(Open Network Operating System, ONOS)^[8]和OpenDayLight (ODL)^[9]。通常情况下,如果控制器集群中一个控制器崩溃,其他工作的控制器仍然可以运行,提供不中断的控制功能。ONOS和ODL通常使用3个控制器来组成控制器集群,3个控制器彼此通信(例如,使用Raft协议^[10])保证网络状态的一致性。控制器集群方案虽然解决

了单个控制器失效的问题,但是无法处理控制器集群物理位置失效的问题,不适用于SD-WAN。如果控制器对应集群所在的机房出现电源故障而造成整个机房断电,那么这个控制集群将无法提供控制服务,进而对整个网络造成不良影响。

为了提供弹性的网络控制,OpenFlow协议^[5]提出了多控制器连接方案。具体来说,一个SDN交换机可以连接到主控制器和几个备份控制器。SDN交换机和控制器之间通过TCP协议的keepalive来保持连接。当主控制器失效后,它与交换机的连接变为不活跃状态。感知到此变化后,交换机将请求它的其他备份控制器成为其新的主控制器。当一个备份控制器切换到主控制器角色以后,就可以对该交换机进行正常的控制。图2给出了一个OpenFlow采用多控制器连接方案来避免控制器失效的例子。虽然OpenFlow协议给出了交换机控制恢复的基本解决方法,但是没有考虑控制器失效时如何为多个离线交换机合理选择控制器的问題,无法直接应用于SD-WAN。因此,在多控制器失效情况下路径可编程性研究是SD-WAN部署过程中面临的一个重要研究问题。

3 研究现状分析

当前国内外的研究大多基于OpenFlow多控制器连接方案,采用1个交换机由1个主控制器控制的思想,实现了静态恢复与动态恢复的两种方案。静态的方案主要考虑如何降低失效概率以及降低失效后的影响两个方面。而动态方案主要考虑交换机-控制器和流-控制器之间的映射关系。其中,动态可编程性恢复大多都是以交换机为单位的恢复方法。但随着当前SDN技术和设备的开发与改进,多个控制器可以同时控制一个交换机内的不同

流,从而实现了细粒度的流-控制器控制方案。综上,保证路径可编程性的研究方案可以归纳为两大类,如图3所示。当前研究现状的详细信息如表1所示。

3.1 静态可编程性恢复方案

3.1.1 降低失效概率的方案

最优控制器放置。最优控制器放置方案基于某些特定的目标(例如,控制器负载、链路负载)来精心地挑选控制器放置的位置,从而降低潜在的控制器失效概率。该方案在设计中的挑战在于如何设计合理的目标函数以实现最优控制器放置。基于帕累托的最优控制器放置方案^[11]提出最小化不同的目标(例如,交换机与控制器之间的延迟、控制器之间的延迟)。当考虑多种性能以及控制弹性指标时,通常不会存在一种最优的控制方案。所以作者在多种指标之间做出了权衡,提出了基于帕累托最优的控制器放置方案。Tanha等人^[12]定义了一个弹性控制器放置问题,目的是尽可能提高网络可靠性,并试图在多约束条件下(例如,控制器部署成本、网络延迟和所需控制器数量)寻找最佳的网络性能和恢复成本的折中,从而实现有控制弹性的控制器放置,并将问题建模为一个整数线性规划(Integer Linear Programming, ILP)问题。为了提高SD-WAN控制平面的可靠性和鲁棒性,作者在另一篇工作中加入了新的考量,同时考虑交换机控制器、控制器之间的延迟要求和控制器的能力等因素来选择控制器的位置^[13],并将问题建模为ILP问题。作者分析该问题复杂度为NP-Hard,并提出了一种多项式时间复杂度的启发式算法高效求解该问题。为了解决重映射过程后交换机控制器之间的最大延迟可能会剧烈增加的问题,Killi等人^[14]综合考虑了控制器控制能力和SDN控制平面可靠性,并将控制器失效时的网络性能定义为两部分:主控制器和交换机之间的延迟、主控制器和备份控制器之间的延迟,将问题建模为一个混合整数线性规划(Mixed Integer Linear Programming, MILP)问题,并提出

了优化此目标的控制器放置方案,利用模拟退火算法求解该问题。Alshamrani等人^[15]提出了一种弹性的控制器部署模型,目的是最大限度提高控制平面在发生故障时的弹性和鲁棒性。该模型以最小化每个交换机与其最接近控制器之间的距离为目标。为了提高网络弹性,Alenazi等人^[16]引入了一个新的度量标准,该度量标准根据一个节点与其他节点的不同连接性来衡量节点的重要性,并基于该度量标准来确定备份控制器的位置。为了在确保网络服务质量的同时尽可能提高网络的鲁棒性,Santos等人^[17]联合考虑了控制器放置问题和链路升级问题,并引入一个新的度量指标来衡量不同路径之间地理上的隔离性,同时基于该度量指标来确定控制器放置位置和数量以及链路升级的位置和数量,在保证网络延迟、路径的地理多样性和网络鲁棒性的同时,尽可能降低升级成本。

弹性控制结构设计。弹性控制结构设计方案通过在网络中设计与部署新型的协议、框架以及度量指标等方法来提升控制平面的控制弹性。该方案在设计中所面临的挑战在于如何设计可面对不同失效情形的高扩展性弹性控制结构。为了保护给定网络中的所有单链路故障,Yang等人^[18]提出通过将最小的IP路由器子集更新为SDN交换机。高洁等人^[19]针对控制器的单点故障、未知的漏洞等安全性问题,提出一种基于拜占庭协议的安全结构,通过控制器之间执行拜占庭协议,将动态性、异构性引入到控制平面的设计中,增强了网络的主动防御能力。同时,设计了两阶段控制器视图选举算法,提高了网络的可靠性和安全性。Xie等人^[20]考虑了分布式控制平面在不同失效情形以及失效恢复策略下的鲁棒优化问题,提出了一种控制平面鲁棒性的验证框架,该框架可以验证控制平面在各种失效场景和恢复策略下能否维持可靠的表现,并进一步提出两种控制平面故障恢复策略,达到近似最优的恢复效果。为了最大化控制平面可靠性,Hu等人^[21]提出一种新型度量指标:控制路径损耗的预期百分

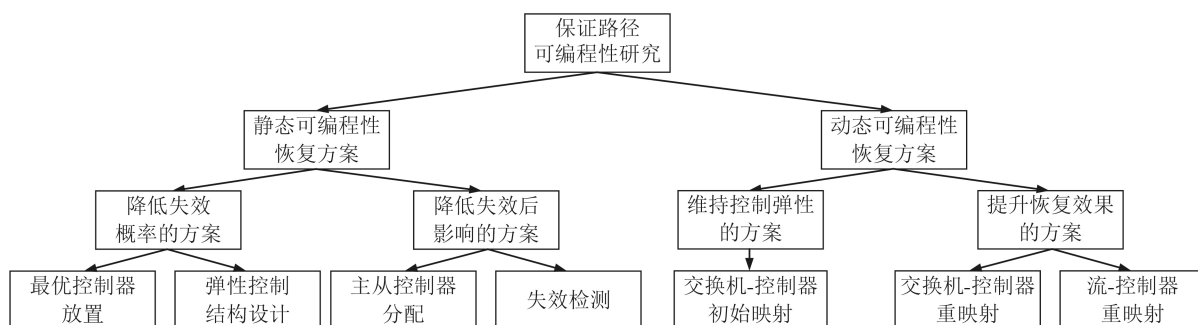


图3 保证路径可编程性的相关工作分类

表 1 保证路径可编程性的研究现状

恢复类型	恢复目标	恢复方法	优化目标	求解方法	参考文献
静态	降低失效概率	最优控制器放置	控制延迟	帕累托最优	[11]
			控制器部署代价和路由代价	ILP	[12]
			所需控制器数量	ILP和启发式算法	[13]
			控制延迟	MILP和模拟退火算法	[14]
			控制延迟	启发式算法	[15]
			节点重要程度	启发式算法	[16]
			链路升级成本	ILP	[17]
			IP路由器更新数量	启发式算法	[18]
			控制器视图异构度	启发式算法	[19]
		弹性控制结构设计	控制器利用率	ILP和启发式算法	[20]
			控制路径失效数量	ILP	[21]
	降低失效后影响	主从控制器分配	映射鲁棒性	ILP和启发式算法	[22]
			负载变化	ILP和启发式算法	[23]
			控制延迟、控制器负载均衡和映射鲁棒性	ILP和启发式算法	[24]
			控制延迟	ILP和贪婪算法	[25]
			控制器负载均衡	ILP和模拟退火算法	[26]
			控制延迟	MILP和贪婪算法	[27]
		失效检测	控制器恢复效果	基于区块链的启发式算法	[28]
			应用服务质量	控制器负载迁移框架	[29]
			故障恢复速度	高级消息队列协议	[30]
			网络可靠性、电力成本和映射鲁棒性	ILP、基于SVM的分类法和贪婪算法	[31]
			重映射成本	ILP	[32]
	维持控制弹性	交换机-控制器初始映射	控制器负载均衡	基于控制器负载的贪婪算法	[33]
			控制器负载均衡	ILP和模拟退火算法	[34]
			控制延迟、控制器负载均衡和映射鲁棒性	深度Q学习	[35]
			所需控制器数量	LP	[36]
			控制器负载均衡和控制器失效概率	MILP和遗传算法	[37]
			所需控制器数量	LP和启发式算法	[38]
			流建立时间	MILP	[39]
动态	提升恢复效果	交换机-控制器重映射	控制器交换机信息交换时长	ILP	[40]
			负载变化和交换机迁移代价	MILP和启发式算法	[41]
			控制器负载均衡和控制延迟	MILP和启发式算法	[42]
			恢复流的数量	MILP和启发式算法	[43]
		流-控制器重映射	可编程性均衡性、总体可编程和控制延迟	MILP和启发式算法	[44]
			可编程性均衡性、总体可编程性	MILP和启发式算法	[45]

比，该指标可以表征控制平面的可靠性，并基于该指标来确定控制器放置数量和位置来最大化控制平面的可靠性。Zhang等人^[22]考虑控制平面在故障场景下的生存能力，并提出了基于生存能力模型的低延迟控制平面设计方案，该方案可以确保交换机恢复效果并保证低控制延迟。

3.1.2 降低失效后影响的方案

主从控制器分配。主从控制器分配方案通过为每一个交换机分配若干备用控制器来降低控制器失效后对网络影响。该方案所面临的主要挑战为如何在同时考虑多个不同的目标对网络性能的潜在影响的基础上为每个交换机选定合适的备用控制器集合。

针对为交换机分配备用控制器的问题, Hu等人^[23]发现不合理的备用控制器分配可能会导致控制链路失效, 甚至可能会造成整体网络的瘫痪。作者在设计主从控制器分配时将控制器失效前后的链路负载变化加入考量, 建立ILP模型并提出了一种启发式算法高效解决该问题。作者在另一篇工作中对之前的研究进行了扩展, 提出了综合考虑延迟和控制器负载均衡的控制器分配方案^[24], 建立了ILP问题并提出了一种启发式算法来高效解决该问题。He等人^[25]同样设计了一个主从控制器分配模型, 在保证交换机和控制器之间传播延迟的约束下, 降低多控制器失效对网络性能的影响。作者又将控制器负载均衡作为目标, 提出了一种应对多控制器失效的新型主从控制器分配ILP模型^[26], 并提出了一种启发式算法高效解决该问题, 能够实现控制器失效场景下快速恢复并均衡各个控制器负载。文献^[27]还提出了基于多目标优先级的多控制器分配问题模型, 该模型考虑了3个目标: 最小化平均延迟、最小化最坏情况下的延迟以及最大化特定延迟范围内的交换机数量, 并基于上述目标为每个交换机分配一个控制器集合。同时设计了基于策略的主控制器选择机制, 为每个交换机确定主控制器, 并提出一种贪婪算法确定控制器集合, 该算法能在多项式时间内解决控制器分配问题。

失效检测。失效检测方案可实现对失效控制器的检测与识别, 并针对失效控制器进行相应的恢复来降低控制器失效的影响。该方案的主要挑战为如何精准且实时地对失效控制器进行检测。为有效地对失效控制器进行检测并恢复, Misra等人^[28]提出了一种基于区块链的控制器恢复方法, 该方法在每个交换机处用一个轻量级的极简区块链记录控制器信息, 当有新事件发生时, 相应的网络流表信息就会被记录到区块链中, 当控制器恢复之后, 控制器就可以从区块链中下载所有交换机的流表信息, 实现将失效控制器恢复至失效之前的状态。Basu等人^[29]提出了一种应对控制器失效的框架, 该框架包含基于云的实时流集中存储模块和虚拟连接管理模块, 并通过对控制器负载的迁移来降低控制器失效对应用服务质量的影响。乐宗港等人^[30]针对控制器存在单点失效故障问题, 提出一种基于高级消息队列的协议, 将多个控制器和控制代理相连接, 控制代理通过心跳机制检测出失效控制器并将其移除。Ren等人^[31]提出了一种基于支持向量机分类方法的适应性恢复策略, 该策略综合考虑了网络可靠性、电力成本和控制延迟, 从而得出控制器的效益矩阵, 并依据该效益矩阵进行控制器失效的应对。针

对混合SDN网络中控制器失效的问题, Guillen等人^[32]综合考虑了混合SDN网络的混合特性和控制平面故障的规模, 提出了一种弹性交换机恢复方案, 该方案采用故障预警机制, 提升了转发平面中离线交换机的恢复数量。Dharam等人^[33]基于对离线交换机的识别与检测, 提出两种离线交换机快速恢复方案来解决控制器失效问题, 通过调整交换机到控制器的控制代价, 实现了正常运行控制器中的负载均衡并保证了离线交换机的恢复效果。

3.1.3 静态可编程性恢复方案小结

以上静态可编程性恢复方案通过考虑一些特定的目标来尝试优化控制器失效后网络的服务质量, 但是它们忽略了网络状态的动态性。文献^[11–33]在部署控制器的时候, 假定每个离线交换机及其控制代价是相同的, 而且控制器的处理负载是给定不变的。然而, 在真实的网络环境中, 控制器和交换机的状态都是动态变化的。例如, 采用以流为单位提取网络状态的机制, 控制器进行网络状态提取的开销会根据每个交换机中流的数量而变化^[46,47]。当流的转发路径发生变化时, 相应交换机中的提取网络状态的开销就会发生相应的改变。如果一个控制器控制过多的离线交换机, 这些离线交换机可能产生过多的提取处理请求, 超过该控制器的处理能力, 进而引发控制器超载现象。在超载情况下, 控制器处理的请求将会经历长尾延迟^[48]。极端情况下, 超载控制器还会出现失效, 从而产生更加严重的控制器集联失效现象, 即在一个控制器失效的情况下, 由于将恢复离线交换机的控制不合理的分配, 导致正常运行的控制器超载, 出现控制器连续的链式失效, 最终造成整个网络的控制平面瘫痪^[23,24,49]。由此可见, 考虑动态性对控制映射方案设计至关重要。

3.2 动态可编程性恢复方案

3.2.1 维持控制弹性的方案

交换机-控制器初始映射。交换机-控制器的初始映射方案旨在通过分析网络的当前状态 (例如, 控制器控制负载、控制器控制能力上限和链路的鲁棒性) 来设定交换机与控制器之间的初始映射关系。该方案的挑战在于如何建立起网络当前状态与控制弹性之间的内在联系。为维持弹性的网络控制, Açan等人^[34]针对网络中的交换机、链路以及控制器失效问题, 在考虑控制器负载的基础上, 提出了一个整数线性规划问题来解决交换机-控制器的初始映射问题。作者同时设计了基于模拟退火算法的启发式算法来解决上述网络失效问题。Chen等人^[35]考虑网络延迟、负载均衡以及网络的可靠性, 提出用深度强化学习来解决交换机控制器

映射问题。Mohan等人^[36]为应对多控制器同时进行的拜占庭式攻击,提出了一种交换机控制器映射的线性规划(Linear Programming, LP)问题。该问题以最小化所需控制器数量为目标,并考虑了控制器的控制能力限制以及交换机控制器之间的通信延迟限制。

3.2.2 提升恢复效果的方案

交换机-控制器重映射。交换机-控制器重映射方案在控制器失效后,通过为离线交换机选择合理的在线控制器进行重映射而提升网络的整体可编程性。该方案需要考虑如何设计有效的重映射策略以最大化离线交换机以及离线流的恢复效果。为恢复因控制器失效而丧失的路径可编程性, Güner等人^[37]在其工作PREFCP中,将控制器负载分配、离线交换机重新映射代价以及控制器失效概率进行考虑,提出了一种基于遗传算法的启发式算法来解决交换机映射问题。Mohan等人^[38]在基于之前对控制器数量、通信延迟等指标的基础上,考虑了控制器失效时的交换机-控制器重映射问题。为进一步优化SD-WAN的网络性能表现,在文献^[39]中, Sridharan等人以最小化流的建立时间为目标,以控制器失效时的控制弹性为限制条件,设计了一种有效的交换机控制器映射算法。文献^[40]还将隐私保护加入到交换机与控制器的映射的设计之中,提出了名为DPAM的算法。该算法通过动态地调整交换机与控制器的映射关系,从而避免控制器长时间监视相同的交换机。Al-Tam等人^[41]发现静态的交换机控制器映射方案在实际应用中是不切实际的,因为网络状况的波动与复杂性使其无法实现被预测。因此,作者设计了一种在交换机控制器动态映射中通过交换机迁移而实现对网络动态适应的启发式算法,保证网络中的控制器负载均衡。为进一步提升SDWAN的控制弹性,文献^[42]还考虑了映射的稳定性与鲁棒性,并建立了相应的凸二次规划问题。针对离线交换机相对固定的控制代价不能适应控制器的剩余控制能力的问题, Dou等人^[43]提出了一种多控制器失效场景下具有适应性的离线交换机恢复方案Matchmaker,该方案通过动态调整一些离线流的路径,从而调整了离线交换机的控制成本,进而增加了恢复的离线流数量并提高了在线控制器的控制资源利用率。

流-控制器重映射。以上粗粒度的交换机-控制器映射方案都是基于一个交换机只能被一个控制器控制的模式,本质上只能实现粗粒度的控制,恢复效果不理想。而细粒度的流-控制器映射方案ProgrammabilityGuardian (PG)^[44]打破了当前1个交换

机只能被1个控制器控制的固化管理模式,实现多个控制器同时对1个交换机内不同流的控制。图4显示了拟采用的新型细粒度控制方法的设计框架。该控制方法由1个主单元和连接每个交换机的代理组成。主单元连接到控制器以检测其运行状态。如果1个或多个控制器被发现失效,主单元将计算控制器对离线流的控制映射关系,并将映射关系部署到离线交换机的代理中。代理位于具有交换机的转发平面和具有多个控制器的控制平面之间,每个代理通过对交换机进行虚拟化操作,可以将1个物理SDN交换机分割为不同的虚拟逻辑交换机,从而允许多个控制器管理同一个物理交换机,并根据映射将离线流的控制权调度到选定运行控制器。该方法的实现可以借助已有的SDN虚拟化技术FlowVisor^[50],并在已有的技术上进行2次开发。

PG是目前唯一在应对多控制器失效时借助FlowVisor实现流粒度恢复的方案。除控制器失效场景以外, FlowVisor技术也已被应用在实现细粒度的控制器-交换机映射的其他方面。Bera等人^[51]借助FlowVisor技术,提出了一种流量感知的动态控制器分配模型,该模型分为两阶段,即窗口调整阶段和流分配阶段。第1阶段FlowVisor根据网络中的流量适应性地选择窗口大小,第2阶段则将该窗口之后时刻的流根据动态表匹配策略分配给控制器,从而减少了控制器响应时间并保证了大多数流的服务质量。Yang等人^[52]考虑了交换机的最小可用剩余容量并提出一个非线性整数规划问题,同时将其分解为两个阶段的问题,即首先将离线交换机映射给多个控制器,再赋予不同控制器不同权重,从而增加了离线交换机的最小可用剩余容量并降低控制器之间的同步成本。

虽然上述借助FlowVisor的新型流粒度的重映射设计能实现最优的恢复效果,但是FlowVisor的引入却带来了新的问题。FlowVisor会因为引入的

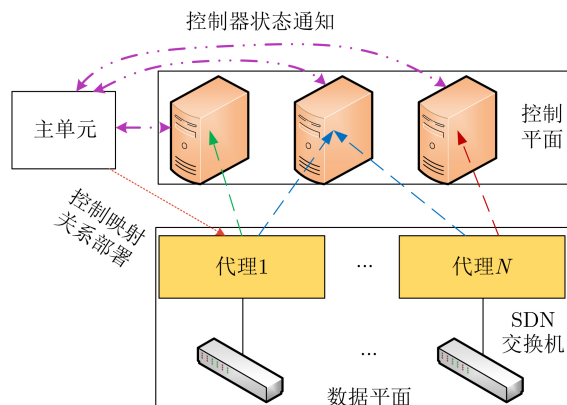


图4 新型流粒度控制方法架构

中间层而需要额外的时间去处理请求,从而显著地增加了通信开销。而SDN支持的另一项技术,即混合路由模式为解决该问题带来了新的思路。当前部分商用SDN交换机支持的混合路由模式,为提升控制器失效下的路径可编程恢复效果带来新的方法。新型SDN交换机中提供的混合路由模式(即SDN模式和传统模式)切换给控制映射方案的设计提供了新的思路。SDN模式下,控制器下发路由决策到交换机的OpenFlow表,交换机使用OpenFlow表进行路由。传统模式下,交换机采用传统路由表中基于目的的路由表项进行路由,而路由表中的路由表项由传统路由协议产生。控制器可以通过控制信息来配置交换机的路由模式。图5给出了一个SDN交换机中的路由模式切换流程的例子。当前的商用SDN交换机(例如,MLX-8 PE^[53])已经支持了SDN混合路由模式,并已被工业界与学术界认可。SDN混合路由模式已经被测试于真实的网络中(例如,CHN-IX^[54]),此外,Xu等人^[55]也已经证明了其提高网络表现的有效性。

得益于混合路由模式,ProgrammabilityMedic (PM)^[45]通过合理的路由模式配置的方式,将一个由纯SDN路由方式的SDN网络变为一个由SDN路由方式和传统路由方式相结合的混合SDN网络。该方法的核心思想是在不引入SDN虚拟化技术的前提下,借助SDN交换机中的混合路由模式,通过动态调整待恢复的离线流在交换机上的路由模式,从而调整离线交换机所需要的控制代价,实现一种近似流级别的细粒度恢复方案。相较于PG,该方案不需要额外的专用SDN虚拟化设备,并且减少了SDN虚拟化设备带来的额外处理延迟。

3.2.3 动态可编程性恢复方案小结

不同于静态可编程性恢复方案所假定的“每个离线交换机及其控制代价是相同的,且控制器的处

理负载是给定不变的”这一假设,动态可编程性恢复方案考虑了在真实的网络环境中,控制器和交换机的状态都是动态变化的,并根据以流为单位提取网络状态的机制,动态地考虑控制器进行网络状态提取的开销变化来进行重映射设计。因此动态可编程性恢复方案更符合实际的网络状态,可以提供更好的恢复效果。

而相较于交换机-控制器粒度的映射方案,流-控制器粒度的方案借助现有SDN的相关技术(例如,FlowVisor和SDN路由混合模式等),实现了细粒度的可编程性恢复。得益于更细粒度的重映射操作,流-控制器映射方案还将恢复的离线流可编程性均衡性加入考虑。从统计上说,如果一个离线流的路径较长,它被恢复的概率高于一个较短路径长度的离线流,因为较长的路径增加了这个流可以被恢复的可能性。同样地,对比两个不同长度的流。通常情况下,路径较长流的可编程性高于路径较短的流。然而,这种简单通过路径长度来区分恢复离线流可编程性的方法,会导致恢复离线流的可编程性不均衡。如果短路径流的流量突增导致网络拥塞,则无法通过调整这个流的路径来降低网络拥塞,增加了网络性能的不确定性。流-控制器映射方案以恢复可编程性的均衡性作为目标,实现了公平的可编程性恢复。

4 未来研究方向

4.1 当前工作存在的问题

尽管研究人员已经对SD-WAN部署过程中在控制器失效情况下对交换机的控制方案研究进行了大量工作,可以总结为静态与动态两种控制器映射方案。但是,目前已有工作存在以下问题:

(1) 恢复离线流的数量有限。实验结果表明,静态恢复方案和动态恢复方案均只能恢复一定数量的离线流。OpenFlow协议中,控制器通过建立与交换机的映射关系来实现对交换机中流的控制,进而实现对流的路径可编程。如果一个控制器中的控制能力不足以控制一个交换机时,这种控制映射关系就无法建立。在控制器失效情况下,这种粗粒度的控制映射关系,会导致运行控制器有限的剩余控制能力只能恢复部分离线交换机,从而只能恢复部分离线流。基于粗粒度的控制映射关系,使用混合路由模式可以在一定程度上缓解正常运行控制器的控制能力和离线交换机恢复控制需要之间的矛盾,有限地提高控制器的控制资源的利用率,但无法从根本上解决该问题。

(2) 增加了控制器的通信开销。控制器的通信开销主要由两部分引起:(a)控制器与交换机之间

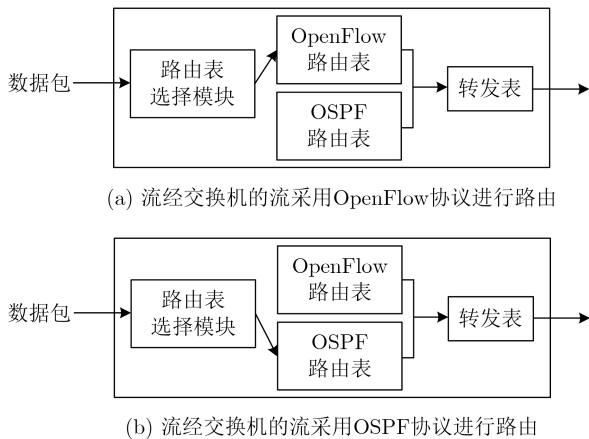


图5 混合路由模式

的传播延迟; (b)流经交换机的流的数量。在广域网中, 控制器与交换机之间的传播延迟是控制器的通信开销最主要的组成部分, 因为它很大程度上影响了控制器的响应速度。较大的通信延迟可能会限制路由收敛以及网络状态信息更新。所以, 在进行交换机控制器的映射时, 需要将交换机与控制器之间的通信延迟加入考虑。否则, 控制器的响应速度会随着网络状态的动态波动而下降。

4.2 未来研究方向

基于现有的研究工作, 分别从路径可编程性的计算、网络性能感知、失效类型以及失效概率4个方面预测了在未来可能有意义的SD-WAN部署过程中保证路径可编程性的研究方向。

(1) 路径可编程性的精确计算。如前所述, 路径可编程性是SDN技术提供的一个重要能力。如果在恢复离线流控制的过程中不考虑该因素, 则网络性能会产生严重的不确定性。然而, 当前研究并没有明确的方法对流的路径可编程性进行明确的度量。因此, 如果研究人员能够在流的路径上多个节点被控制器控制时, 精确地计算出其控制节点位置和其路径可编程性的确定性关系, 并能够在不同规模的网络拓扑中, 快速求解出多个流的路径可编程性, 定能极大地帮助保证路径可编程性的研究。

(2) 网络性能的感知。当前的研究都是以恢复交换机或是流的数量等间接衡量标准作为目标的恢复方法。然而, 网络中的流存在不同的重要性。以流量为例, 某些流的流量可能一直较小, 对网络负载均衡性能的影响微乎其微; 而某些流的流量很大, 转发路径的合理选择决定了整个网络的负载均衡性能。因此, 在不区分流的重要性的前提下进行离线流可编程性的恢复, 可能得不到较好的恢复效果, 并造成网络性能的抖动。所以, 如果研究人员能够以网络性能作为直接的目标来进行可编程性保证的研究, 定能适用于真实网络环境下的软件定义广域网路径可编程性保证的问题。

(3) 不同的失效形式。在软件定义广域网中, 控制器失效只是其中一种失效的形式。除了控制器失效, 网络中还可能会出现单点失效以及链路失效等不同形式的失效形式。当前已经有学者对于这些不同失效形式下的保证控制弹性进行了研究。但是, 当前的研究基本是分别考虑单一失效形式, 而在真实网络环境下, 不同形式的失效有可能会同时发生, 对网络的表现造成严重的影响。因此, 如果研究人员能够将不同失效形式统一进行考虑并设计可行的算法, 定能更好地保证软件定义广域网中的路径可编程性。

(4) 不同的失效概率。在真实网络环境中, 网络流量会实时地发生波动, 从而导致每条链路上的负载会随之产生变化。同样地, 每台控制器的负载也会出现差异。不同负载下的控制器或者链路会有不同失效的概率。负载量大的控制器或链路等会更容易发生失效, 而负载量小的控制器或链路等则不太容易发生失效。因此, 如果研究人员能够将失效的概率加入算法设计的考量之中, 定能更适用于真实网络环境下的软件定义广域网路径可编程性保证的问题。

5 结束语

路径可编程性作为SDN提供的重要技术之一, 为灵活流量调度的实现带来了可能。但当控制器失效时, 当前国内外的研究大多以OpenFlow多控制器连接方案为前提, 采用1个交换机由1个主控制器控制的交换机粒度恢复策略, 提出静态可编程性恢复和动态可编程性恢复两种解决方案。静态可编程性恢复方案的主要思想是在控制器部署时考虑控制器失效对网络的影响, 来决定交换机主备控制器的部署位置。虽然该方案通过考虑一些特定的目标来尝试保障控制器失效后网络的服务质量, 但是它们忽略了网络状态的动态变化, 在真实网络中容易造成控制资源浪费, 并且无法保证所有流的服务质量。针对以上问题, 动态可编程性恢复方案根据实时的动态网络状态来进行控制器和离线交换机的控制关系映射, 进一步降低了控制器失效对网络服务质量的影响。但是, 动态可编程性恢复方案依然是基于“1个交换机由1个主控制器控制”这一粗粒度的现有控制方法。在多控制器失效的情况下, 动态可编程性恢复方案无法充分利用控制资源来尽可能多地恢复离线流。因此, 有学者借助当前的SDN虚拟化技术以及SDN混合路由模式, 打破了当前1个交换机只能被1个控制器控制的固化管理模式, 实现了细粒度的恢复策略, 提升了控制器失效下离线流可编程性的恢复效果。

综上, 在多控制器失效情况下路径可编程性研究是SD-WAN部署过程中面临的一个重要研究问题, 相信将来会有越来越多的基于SD-WAN的网络服务的出现。如何充分保障SD-WAN内的路径可编程性进而提升网络的服务质量, 有待工业界和学术界的共同探索。

参考文献

- [1] KREUTZ Diego, RAMOS M V F, VERÍSSIMO P E, *et al.* Software-defined networking: A comprehensive survey[J].

- Proceedings of the IEEE*, 2015, 103(1): 14–76. doi: [10.1109/JPROC.2014.2371999](https://doi.org/10.1109/JPROC.2014.2371999).
- [2] JAIN S, KUMAR A, MANDAL S, *et al.* B4: Experience with a globally-deployed software defined WAN[J]. *ACM SIGCOMM Computer Communication Review*, 2013, 43(4): 3–14. doi: [10.1145/2534169.2486019](https://doi.org/10.1145/2534169.2486019).
 - [3] HONG Chiyao, KANDULA S, MAHAJAN R, *et al.* Achieving high utilization with software-driven WAN[C]. The ACM SIGCOMM 2013 Conference on SIGCOMM, Hong Kong, China, 2013: 15–26. doi: [10.1145/2486001.2486012](https://doi.org/10.1145/2486001.2486012).
 - [4] First in the U. S. to Mobile 5G – What’s next? Defining AT&T’s network path in 2019 and beyond[EB/OL]. https://about.att.com/story/2019/2019_and_beyond.html, 2019.
 - [5] OpenFlow Switch Specification. Version 1.5. 1 (Protocol version 0x06)[EB/OL]. <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>, 2015.
 - [6] LEVIN D, WUNDSAM A, HELLER B, *et al.* Logically centralized?: State distribution trade-offs in software defined networks[C]. The First Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, 2012: 1–6. doi: [10.1145/2342441.2342443](https://doi.org/10.1145/2342441.2342443).
 - [7] HELLER B, SHERWOOD R, and MCKEOWN N. The controller placement problem[J]. *ACM SIGCOMM Computer Communication Review*, 2012, 42(4): 473–478. doi: [10.1145/2377677.2377767](https://doi.org/10.1145/2377677.2377767).
 - [8] ONOS controller[EB/OL]. <https://onosproject.org/>.
 - [9] OpenDayLight controller [EB/OL]. <https://www.opendaylight.org/>.
 - [10] ONGARO D and OUSTERHOUT J. In search of an understandable consensus algorithm[C]. The 2014 USENIX conference on USENIX Annual Technical Conference, Philadelphia, USA, 2014: 305–320.
 - [11] HOCK D, HARTMANN M, GEBERT S, *et al.* Pareto-optimal resilient controller placement in SDN-based core networks[C]. The 2013 25th International Teletraffic Congress (ITC), Shanghai, China, 2013: 1–9. doi: [10.1109/ITC.2013.6662939](https://doi.org/10.1109/ITC.2013.6662939).
 - [12] TANHA M, SAJJADI Dawood, and PAN Jianping. Enduring node failures through resilient controller placement for software defined networks[C]. 2016 IEEE Global Communications Conference (GLOBECOM), Washington, USA, 2016: 1–7. doi: [10.1109/GLOCOM.2016.7841786](https://doi.org/10.1109/GLOCOM.2016.7841786).
 - [13] TANHA M, SAJJADI D, RUBY R, *et al.* Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs[J]. *IEEE Transactions on Network and Service Management*, 2018, 15(3): 991–1005. doi: [10.1109/TNSM.2018.2829661](https://doi.org/10.1109/TNSM.2018.2829661).
 - [14] KILLI B P R and RAO S V. Capacitated next controller placement in software defined networks[J]. *IEEE Transactions on Network and Service Management*, 2017, 14(3): 514–527. doi: [10.1109/TNSM.2017.2720699](https://doi.org/10.1109/TNSM.2017.2720699).
 - [15] ALSHAMRANI A, GUHA S, PISHARODY S, *et al.* Fault tolerant controller placement in distributed SDN environments[C]. 2018 IEEE International Conference on Communications (ICC), Kansas City, USA, 2018: 1–7. doi: [10.1109/ICC.2018.8422593](https://doi.org/10.1109/ICC.2018.8422593).
 - [16] ALENAZI M J F and ÇETINKAYA E K. Resilient placement of SDN controllers exploiting disjoint paths[J]. *Transactions on Emerging Telecommunications Technologies*, 2020, 31(2): e3725. doi: [10.1002/ett.3725](https://doi.org/10.1002/ett.3725).
 - [17] SANTOS D, GOMES T, and TIPPER D. SDN controller placement with availability upgrade under delay and geodiversity constraints[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(1): 301–314. doi: [10.1109/TNSM.2020.3049013](https://doi.org/10.1109/TNSM.2020.3049013).
 - [18] YANG Ze and YEUNG K L. SDN candidate selection in hybrid IP/SDN networks for single link failure protection[J]. *IEEE/ACM Transactions on Networking*, 2020, 28(1): 312–321. doi: [10.1109/TNET.2019.2959588](https://doi.org/10.1109/TNET.2019.2959588).
 - [19] 高洁, 邬江兴, 胡宇翔, 等. 基于拜占庭容错的软件定义网络控制面的抗攻击性研究[J]. 计算机应用, 2017, 37(8): 2281–2286. doi: [10.11772/j.issn.1001-9081.2017.08.2281](https://doi.org/10.11772/j.issn.1001-9081.2017.08.2281).
GAO Jie, WU Jiangxing, HU Yuxiang, *et al.* Research of control plane’ anti-attacking in software-defined network based on Byzantine fault-tolerance[J]. *Journal of Computer Applications*, 2017, 37(8): 2281–2286. doi: [10.11772/j.issn.1001-9081.2017.08.2281](https://doi.org/10.11772/j.issn.1001-9081.2017.08.2281).
 - [20] XIE Junjie, GUO Deke, QIAN Chen, *et al.* Validation of distributed SDN control plane under uncertain failures[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(3): 1234–1247. doi: [10.1109/TNET.2019.2914122](https://doi.org/10.1109/TNET.2019.2914122).
 - [21] HU Yannan, WANG Wendong, GONG Xiangyang, *et al.* On reliability-optimized controller placement for software-defined networks[J]. *China Communications*, 2014, 11(2): 38–54. doi: [10.1109/CC.2014.6821736](https://doi.org/10.1109/CC.2014.6821736).
 - [22] ZHANG Lingyu, WANG Ying, LI Wenjing, *et al.* A survivability-based backup approach for controllers in multi-controller SDN against failures[C]. 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, Korea (South), 2017: 100–105. doi: [10.1109/APNOMS.2017.8094186](https://doi.org/10.1109/APNOMS.2017.8094186).
 - [23] HU Tao, GUO Zehua, ZHANG Jianhui, *et al.* Adaptive slave controller assignment for fault-tolerant control plane in software-defined networking[C]. 2018 IEEE International Conference on Communications (ICC), Kansas City, USA, 2018: 1–6. doi: [10.1109/ICC.2018.8422598](https://doi.org/10.1109/ICC.2018.8422598).

- [24] HU Tao, YI Peng, GUO Zehua, *et al.* Dynamic slave controller assignment for enhancing control plane robustness in software-defined networks[J]. *Future Generation Computer Systems*, 2019, 95: 681–693. doi: [10.1016/j.future.2019.01.010](https://doi.org/10.1016/j.future.2019.01.010).
- [25] HE Fujun, SATO T, and OKI E. Master and slave controller assignment model against multiple failures in software defined network[C]. ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019: 1–6. doi: [10.1109/ICC.2019.8761291](https://doi.org/10.1109/ICC.2019.8761291).
- [26] HE Fujun and OKI E. Load balancing model against multiple controller failures in software defined networks[C]. ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020: 1–6. doi: [10.1109/ICC40277.2020.9148661](https://doi.org/10.1109/ICC40277.2020.9148661).
- [27] HE Fujun and OKI E. Main and secondary controller assignment with optimal priority policy against multiple failures[J]. *IEEE Transactions on Network and Service Management*, 2021, 18(4): 4391–4405. doi: [10.1109/TNSM.2021.3064646](https://doi.org/10.1109/TNSM.2021.3064646).
- [28] MISRA S, SARKAR K, and AHMED N. Blockchain-based controller recovery in SDN[C]. IEEE INFOCOM 2020 – IEEE IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, Canada, 2020: 1063–1068. doi: [10.1109/INFOCOMWKSHPS50562.2020.9162659](https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162659).
- [29] BASU K, HAMDULLAH A, and BALL F. Architecture of a cloud-based fault-tolerant control platform for improving the QoS of social multimedia applications on SD-WAN[C]. 2020 13th International Conference on Communications (COMM), Bucharest, Romania, 2020: 495–500. doi: [10.1109/COMM48946.2020.9142038](https://doi.org/10.1109/COMM48946.2020.9142038).
- [30] 乐宗港, 黄刘生, 徐宏力. 基于AMQP的SDN控制器故障恢复机制[J]. 通信技术, 2017, 50(3): 487–491. doi: [10.3969/j.issn.1002-0802.2017.03.018](https://doi.org/10.3969/j.issn.1002-0802.2017.03.018).
LE Zonggang, HUANG Liusheng, and XU Hongli. Failure recovery mechanism of SDN controller based on AMQP[J]. *Communications Technology*, 2017, 50(3): 487–491. doi: [10.3969/j.issn.1002-0802.2017.03.018](https://doi.org/10.3969/j.issn.1002-0802.2017.03.018).
- [31] REN Xiaodon, AUJLA S G, JINDAL A, *et al.* Adaptive recovery mechanism for SDN controllers in Edge-Cloud supported FinTech applications[J]. *IEEE Internet of Things Journal*, 2023, 10(3): 2112–2120. doi: [10.1109/JIOT.2021.3064468](https://doi.org/10.1109/JIOT.2021.3064468).
- [32] GUILLEN L, IZUMI S, ABE T, *et al.* A resilient mechanism for multi-controller failure in hybrid SDN-based networks[C]. 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), Tainan, China, 2021: 285–290. doi: [10.23919/APNOMS52696.2021.9562671](https://doi.org/10.23919/APNOMS52696.2021.9562671).
- [33] DHARAM P and DEY M. A mechanism for controller failover in distributed software-defined networks[C]. 2021 8th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 2021: 196–201. doi: [10.1109/ICCCE50029.2021.9467174](https://doi.org/10.1109/ICCCE50029.2021.9467174).
- [34] AÇAN F, GÜR G, and ALAGÖZ F. Reactive controller assignment for failure resilience in software defined networks[C]. 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 2019: 1–6. doi: [10.23919/APNOMS.2019.8892940](https://doi.org/10.23919/APNOMS.2019.8892940).
- [35] CHEN Jia, CHEN Shihua, CHENG Xin, *et al.* A deep reinforcement learning based switch controller mapping strategy in software defined network[J]. *IEEE Access*, 2020, 8: 221553–221567. doi: [10.1109/ACCESS.2020.3043511](https://doi.org/10.1109/ACCESS.2020.3043511).
- [36] MOHAN P M, TRUONG-HUU T, and GURUSAMY M. Primary-backup controller mapping for Byzantine fault tolerance in software defined networks[C]. GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017: 1–7. doi: [10.1109/GLOCOM.2017.8254755](https://doi.org/10.1109/GLOCOM.2017.8254755).
- [37] GÜNER S, GÜR G, and ALAGÖZ F. Proactive controller assignment schemes in SDN for fast recovery[C]. 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 2020: 136–141. doi: [10.1109/ICOIN48656.2020.9016544](https://doi.org/10.1109/ICOIN48656.2020.9016544).
- [38] MOHAN P M, TRUONG-HUU T, and GURUSAMY M. Byzantine-resilient controller mapping and remapping in software defined networks[J]. *IEEE Transactions on Network Science and Engineering*, 2020, 7(4): 2714–2729. doi: [10.1109/TNSE.2020.2981521](https://doi.org/10.1109/TNSE.2020.2981521).
- [39] SRIDHARAN V, GURUSAMY M, and TRUONG-HUU T. On multiple controller mapping in software defined networks with resilience constraints[J]. *IEEE Communications Letters*, 2017, 21(8): 1763–1766. doi: [10.1109/LCOMM.2017.2696006](https://doi.org/10.1109/LCOMM.2017.2696006).
- [40] SRIDHARAN V, LIYANAGE K S K, and GURUSAMY M. Privacy-aware switch-controller mapping in SDN-based IoT networks[C]. 2020 International Conference on Communication Systems & NETworks (COMSNETS), Bengaluru, India, 2020: 1–6. doi: [10.1109/COMSNETS48256.2020.9027467](https://doi.org/10.1109/COMSNETS48256.2020.9027467).
- [41] AL-TAM F and CORREIA N. On load balancing via switch migration in software-defined networking[J]. *IEEE Access*, 2019, 7: 95998–96010. doi: [10.1109/ACCESS.2019.2929651](https://doi.org/10.1109/ACCESS.2019.2929651).
- [42] AL-TAM F and CORREIA N. Fractional switch migration in multi-controller software-defined networking[J]. *Computer Networks*, 2019, 157: 1–10. doi: [10.1016/j.comnet.2019.04.011](https://doi.org/10.1016/j.comnet.2019.04.011).
- [43] DOU Songshi, MIAO Guochun, GUO Zehua, *et al.*

- Matchmaker: Maintaining network programmability for Software-Defined WANs under multiple controller failures[J]. *Computer Networks*, 2021, 192: 108045. doi: [10.1016/j.comnet.2021.108045](https://doi.org/10.1016/j.comnet.2021.108045).
- [44] GUO Zehua, DOU Songshi, and JIANG Wenchao. Improving the path programmability for software-defined wans under multiple controller failures[C]. 2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS), Hangzhou, China, 2020: 1–10. doi: [10.1109/IWQoS49365.2020.9212884](https://doi.org/10.1109/IWQoS49365.2020.9212884).
- [45] DOU Songshi, GUO Zehua, and XIA Yuanqing. ProgrammabilityMedic: Predictable path programmability recovery under multiple controller failures in SD-WANs[C]. 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), Washington DC, USA, 2021: 461–471. doi: [10.1109/ICDCS51616.2021.00051](https://doi.org/10.1109/ICDCS51616.2021.00051).
- [46] VAN ADRICHEM N L M, DOERR C, and KUIPERS F A. Opennetmon: Network monitoring in openflow software-defined networks[C]. 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 2014: 1–8. doi: [10.1109/NOMS.2014.683822](https://doi.org/10.1109/NOMS.2014.683822).
- [47] TOOTOONCHIAN A, GHOBADI M, and GANJALI Y. OpenTM: Traffic matrix estimator for OpenFlow networks[C]. 11th International Conference on Passive and Active Network Measurement, Zurich, Switzerland, 2010: 201–210. doi: [10.1007/978-3-642-12334-4_21](https://doi.org/10.1007/978-3-642-12334-4_21).
- [48] XIE Junjie, GUO Deke, LI Xiaozhou, *et al.* Cutting long-tail latency of routing response in software defined networks[J]. *IEEE Journal on Selected Areas in Communications*, 2018, 36(3): 384–396. doi: [10.1109/JSAC.2018.2815358](https://doi.org/10.1109/JSAC.2018.2815358).
- [49] YAO Guang, BI Jun, and GUO Luyi. On the cascading failures of multi-controllers in software defined networks[C]. 2013 21st IEEE International Conference on Network Protocols (ICNP), Goettingen, Germany, 2013: 1–2. doi: [10.1109/ICNP.2013.6733624](https://doi.org/10.1109/ICNP.2013.6733624).
- [50] SHERWOOD R, GIBB G, YAP K K, *et al.* Flowvisor: A network virtualization layer[R]. OpenFlow Switch Consortium, Tech. Rep, 2009, 1: 132.
- [51] BERA S, MISRA S, and SAHA N. Traffic-aware dynamic controller assignment in SDN[J]. *IEEE Transactions on Communications*, 2020, 68(7): 4375–4382. doi: [10.1109/TCOMM.2020.2983168](https://doi.org/10.1109/TCOMM.2020.2983168).
- [52] YANG Xuwei, XU Hongli, CHEN Shigang, *et al.* Indirect multi-mapping for burstiness management in software defined networks[J]. *IEEE/ACM Transactions on Networking*, 2021, 29(5): 2059–2072. doi: [10.1109/TNET.2021.3078132](https://doi.org/10.1109/TNET.2021.3078132).
- [53] Brocade MLX-8 Pe[EB/OL]. [2022–03-29]. https://www.dataswitchworks.com/datasheets/MLX_Series_DS.pdf.
- [54] CHN-IX[EB/OL]. [2022–03-29]. <http://www.chn-ix.net/>.
- [55] XU Hongli, HUANG He, CHEN Shigang, *et al.* Achieving high scalability through hybrid switching in software-defined networking[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(1): 618–632. doi: [10.1109/TNET.2018.2789339](https://doi.org/10.1109/TNET.2018.2789339).
- 郭泽华: 男, 研究员, 博士生导师, 研究方向为可编程网络、机器学习以及网络安全。
- 窦松石: 男, 硕士生, 研究方向为可编程网络。
- 齐 力: 男, 硕士生, 研究方向为可编程网络。
- 兰巨龙: 男, 教授, 博士生导师, 研究方向为新型网络体系。

责任编辑: 余 蓉