智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

软件定义网络故障诊断综述

齐小刚, 单明媚, 张皓然

引用本文:
齐小刚,单明媚,张皓然. 软件定义网络故障诊断综述[J]. 智能系统学报, 2023, 18(4): 662–675.
QI Xiaogang,SHAN Mingmei,ZHANG Haoran. Summary of software-defined networking fault diagnosis[J]. *CAAI Transactions on Intelligent Systems*, 2023, 18(4): 662–675.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202205037

DOI: 10.11992/tis.202205037
网络出版地址: https://kns.cnki.net/kcms/detail/23.1538.TP.20230321.1648.018.html

软件定义网络故障诊断综述

齐小刚, 单明媚, 张皓然
(西安电子科技大学 数学与统计学院, 陕西 西安 710071)

摘　要: 复杂的传统网络结构限制了网络功能的发展, 也逐渐暴露传统故障诊断算法的缺点。为此本文从新型网络环境的角度详细分析软件定义网络 (software-defined networking, SDN) 架构中故障诊断技术的发展过程, 并从诊断方式对比其与传统网络的不同。本文介绍了 SDN 的架构、网络虚拟化的定义以及系统监控方法; 针对 SDN 中不同故障诊断算法的差异以及虚拟环境中的故障诊断问题, 总结了新型故障诊断技术, 依据不同的算法原理归纳了 5 类故障诊断方法, 并对不同类型的方法进行了优缺点比较。算法比较结果表明: 在性能方面, 结合新型网络结构的故障诊断算法在拓扑发现、数据收集、诊断速度以及算法设计的灵活性方面更加适应现代网络的发展; 在架构方面, SDN 在物理、逻辑和虚拟 3 个层面都有创新性故障诊断算法, 并在传统网络的基础上增加了网络的稳健性和可靠性; 最后从 5 个方面进行未来展望, 为后续故障诊断技术优化研究提供参考。

关键词: 软件定义网络; 可编程; 故障检测; 故障定位; 故障诊断; 系统监控; 虚拟环境; 带内网络遥测
中图分类号: TP393　　**文献标志码**: A　　**文章编号**: 1673−4785(2023)04−0662−14

中文引用格式: 齐小刚, 单明媚, 张皓然. 软件定义网络故障诊断综述 [J]. 智能系统学报, 2023, 18(4): 662–675.
英文引用格式: QI Xiaogang, SHAN Mingmei, ZHANG Haoran. Summary of software-defined networking fault diagnosis[J]. CAAI transactions on intelligent systems, 2023, 18(4): 662–675.

Summary of software-defined networking fault diagnosis

QI Xiaogang, SHAN Mingmei, ZHANG Haoran
(School of Mathematics and Statistics, Xidian University, Xi'an 710071, China)

Abstract: Traditional network structure is complicated, which limits the development of network functions. And gradually, it also exposes the shortcomings of traditional fault diagnosis algorithms. To solve this problem, this paper analyzes the development process of fault diagnosis technology in the software-defined networking (SDN) architecture in detail, and analyzes its difference from traditional network from the perspective of diagnosis mode. Firstly, the architecture of SDN, the definition of network virtualization and system monitoring methods are introduced. Then, aiming at the differences of different fault diagnosis algorithms in SDN and the problem of fault diagnosis in a virtual environment, this paper summarizes the new fault diagnosis technologies. Five types of fault diagnosis methods are summarized based on different principles of the algorithm, and the advantages and disadvantages of different types of methods are compared. Finally, this paper looks forward to the future from five aspects, which provides reference for the follow-up optimization research of fault diagnosis technology.

Keywords: SDN; programmable; fault detection; fault location; fault diagnosis; system monitoring; virtual environment; in-band network telemetry

随着网络要求层出不穷, 传统网络体系已经变得臃肿不堪, 于是人们开始重新定义新的网络形式——软件定义网络(software-defined networking, SDN)。开放网络基金会(open networking foundation, ONF)定义 SDN 是一种支持动态、弹性管理的新型网络体系架构, 是实现高带宽、动态网络的理想架构[1]。区别于传统网络紧密耦合的特点, SDN 数控分离, 实现了网络可编程和逻辑上的集中控制。随着 SDN 在虚拟化、云计算、5G 网络切片[2] 中应用越来越广泛, 网络故障问题也日益凸显。故障是指网络中的服务或者设备不能正常运行的状态[3]。故障在网络中持续时长可

收稿日期: 2022−05−23.　**网络出版日期**: 2023−03−22.
基金项目: 国家自然科学基金项目 (61877067).
通信作者: 齐小刚. E-mail: xgqi@xidian.edu.cn.

长可短, 且故障的发生对网络造成的影响或轻或重, 轻则网络服务中断, 重则网络瘫痪。比如, 从2007—2013 年, 来自 28 家云提供商的云网络由于应用程序和基础设施故障, 造成了 1600 h 的中断和约 2.73 亿美元的损失[4]。故障在网络中是不可避免的, 及时有效地进行故障诊断对网络的高效运行以及减小运营商和用户损失都有着重要意义。研究人员基于此背景提出了许多故障诊断技术, 文献 [5-8] 从信息获取、技术手段、故障场景和类型进行了调研, 这些工作详细介绍了 SDN 中的网络监控手段、SDN 架构发展和体系内部的故障类型。文献 [5, 7] 详细介绍了 SDN 架构中每层故障的分类, 围绕 SDN 应用层、控制层和数据层展开对故障诊断算法的讨论, 重点关注 SDN 软件与硬件故障却忽略了 SDN 虚拟环境下多层网络故障的情况。文献 [9] 包含了 SDN 在虚拟环境中的应用, 但侧重于介绍传统网络的故障诊断算法以及现阶段虚拟网络故障诊断算法遇到的问题, 对于 SDN 在虚拟环境中的诊断算法并没有详细说明, 文献 [10] 全面介绍了 SDN中的机器学习故障诊断算法。

本文将研究目标定位在 2015—2022 年之间的故障诊断方法, 并从基于探针的故障诊断、基于协议的故障诊断、基于模型的自诊断、基于故障注入的故障诊断、基于带内网络遥测的故障诊断 5 个方面总结故障诊断技术, 详细分析了算法的优缺点。由于海量网络信息会导致故障诊断速度减慢, 本文对故障诊断之前的准备工作进行了梳理, 从数据收集、存储和分析 3 个方面介绍了系统监控技术。本文阐述了 SDN 层内、层间故障诊断技术, 并对虚拟环境中的 SDN 故障进行了梳理, 分析了算法的特点和应用。

1　软件定义网络

1.1　SDN 体系架构

传统网络架构由专用硬件、操作系统和网络功能组成, 是一个封闭式系统。由于网络的封闭性, 网络架构之间过耦合的状态使得网络架构演变周期变长, 这种封闭、共生的网络结构虽然稳定但约束了网络发展。在这种网络发展背景下, McKeown 教授团队提出了新型网络体系 SDN。SDN 将网络分为数据平面、控制平面和应用平面等 3 个平面, 北向接口(northbound interface, NBI)和控制数据平面接口(control-data-plane interface, CDPI)等 2 个开放接口。SDN 将传统网络的控制与数据转发分离, 分成多层网络架构, 数据平面只进行简单的数据转发, 不再需要具体实现协议的控制逻辑。网络设备中的控制逻辑转交到 SDN 中的控制和应用平面, 从而实现网络的软件定义化, 演化过程如图 1 所示。

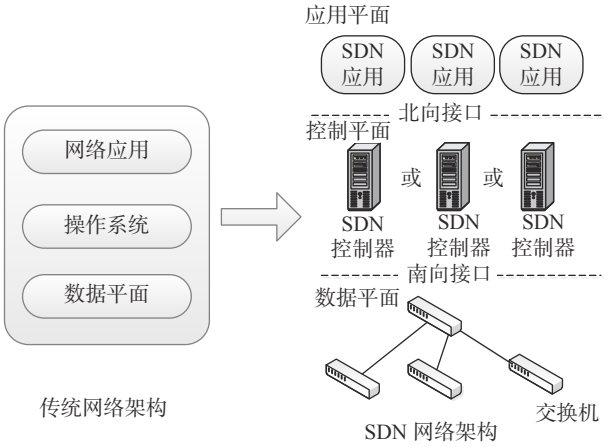

图 1　网络演化过程

Fig. 1　Network structure evolution

SDN 这种新型定义网络满足了网络快速发展的需求, 不仅重构了网络的系统功能, 还实现了数控分离。2 个平面的独立意味着只要其中一个平面更新, 就可以完成整体网络结构更新。数控分离的模式带来了另一个好处: 可编程。控制平面不再局限于简单但繁琐的功能叠加, 而是可以通过编程控制建立更高级的抽象模型, 用户可通过通用接口在控制器中编程实现对网络的自主管理。为了进一步了解 SDN, 下面介绍 SDN 架构的主要元素:

1)应用平面位于 SDN 架构中的顶层, 实现对应网络功能的应用, 允许网络运营商通过北向协议制定网络中的控制逻辑。

2)控制平面位于 SDN 架构中的中间层, 是整个网络架构的核心, 相当于传统网络中的操作系统, 可控制数据平面网络设备的行为, 完成数据包转发、下发流表等任务。

3)数据平面位于 SDN 架构中的底层, 是网络设备的集合, 一般负责网络数据的转发。

4)CDPI 作为控制平面和数据平面通信的标准, 负责传达控制器对交换机的命令或询问和交换机对控制器的请求或答复, CDPI 具有统一的通信标准, 目前南向协议代表是 OpenFlow 协议[10]。

5)NBI 是应用平面和控制平面之间的通信接口。它负责为上层 SDN 应用提供底层网络的抽象视图和访问网络资源的接口。目前, 控制器通常使用通用接口 REST API 实现用户对网络资源的访问。

6)控制器是 SDN 网络的重要组成成分, 控制

器的性能直接决定了网络的性能。自 SDN 发展以来，控制器也在不断地更新迭代。在性能、语言、功能等方面，不同厂家出厂的控制器都有一定的不同。在小、中等规模 SDN 网络中，控制器一般以单一集中的形式出现，不会对网络整体性能产生明显的影响，但在大规模网络中，为了保持网络性能一般采用分布式控制器。在学术界和工业界使用较多是 Ryu、Floodlight、OpenDaylight 和 ONOS。

1.2　SDN 与网络虚拟化

随着用户数量和需求的增多，资源划分的固定性和使用的单一性很有可能造成资源分配的不合理。为了更好地利用资源和服务客户，虚拟化技术在最大化资源的理念中产生。虚拟化技术本质上是将有限资源抽象成虚拟资源，用户可同时发出请求，映射算法会将资源自动按需分配。虚拟化技术很好地调节了用户之间使用资源的关系，既能做到资源共享、满足客户需求，也能资源隔离、保密客户信息。虚拟化技术打破了之前资源分配的局限性，推动了资源分配的弹性化发展，也为网络虚拟化技术的发展提供了思路。

当网络作为被虚拟化对象时，物理网络可以同时支撑多个虚拟网络。根据虚拟网络中租户需求的不同，比如流量、带宽，映射算法会根据物理网络的情况自动分配给租户。虚拟网络是由虚拟节点和虚拟链路组成，虚拟节点、链路是物理节点、链路的子集，物理网络的拓扑结构可虚化成任意结构。网络虚拟化的形式有“一虚多”和“多虚一”，前者指的是一个物理网络上承载多个虚拟网络，后者则是多个物理网络承载一个虚拟网络。传统网络实现网络虚拟化技术需要手动逐条部署，效率低且成本高，SDN 的出现给网络虚拟化的实现带来了便利[1]。利用 SDN 逻辑集中控制和可编程的网络操作方式，网络管理员可直接在控制器中编写程序实现网络虚拟化的自动化业务部署，显著缩短了部署周期。

SDN 逻辑集中控制的功能能够为动态网络管理提供解决方案，并为虚拟网络的实例化提供有利环境[11]。现阶段凭借 SDN 实现网络虚拟化的主要方法是搭建虚拟化平台，虚拟化平台位于底层网络与上层虚拟网络的中间层，主要完成网络资源管理、网络隔离等虚拟化任务，可以建立多个租户网络。虚拟化平台的存在既实现了底层网络与虚拟网络之间的透明性，也完成了虚拟网络之间的流量隔离。虚拟化平台有：FlowVisor、OpenVirtX、ADVisor 等，不同的虚拟化平台实现不同的资源虚拟化，比如拓扑虚拟化、节点资源虚拟化以及链路资源虚拟化。图 2 为 SDN 虚拟化的结构示意图。

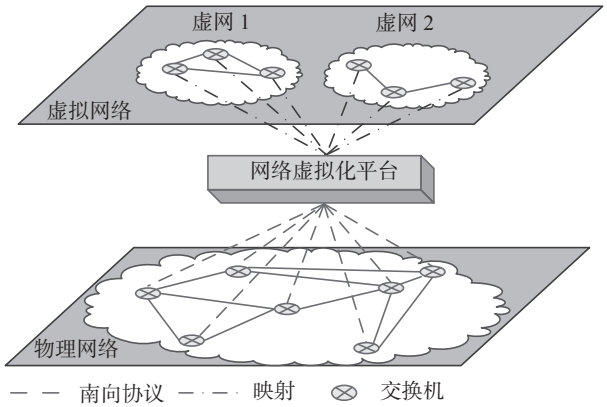

图 2　SDN 虚拟网络的构建

Fig. 2　SDN virtual network construction

1.3　SDN 故障诊断

依据文献 [4, 6, 9]，SDN 故障主要分为 2 类：第 1 类是控制平面与数据平面之间的故障，大部分是由于控制器本身或数据出现异常而引起的故障；第 2 类是 SDN 应用故障，主要侧重于 SDN 在虚拟环境中的故障，在此类故障中研究人员更倾向于研究如何根据虚拟层（物理层）故障定位物理层（虚拟层）故障。

第 1 类故障主要有以下具体表现形式：数据包故障、动态故障、控制器故障。数据包发生故障可能是在转发时行为异常、数据包头被修改、不可达故障或意外丢弃等，这类由数据包异常引发的故障一般发生在数据平面。动态故障一般是指意外发生的故障，如意外丢包、链路拥堵、延迟、出现间歇性连接[6]。控制器故障一般是控制器内部逻辑缺失或错误，对事件做出错误、延迟响应或丢弃等反应，造成网络行为异常、应用程序操作错误、控制器无法访问等后果。

第 2 类应用故障一方面可能是编码错误、用户安装、使用错误等软件故障。一般是由于用户对网络知识不了解或对网络语言的不熟悉造成网络程序错误或控制逻辑出现问题。另一方面是由虚拟层故障而引发多层故障，这类故障由于是跨层、动态且无法知道底层网络拓扑，对网络造成的影响极大。

故障诊断旨在及时发现故障、快速定位故障和准确分析故障。故障诊断中比较重要的两个方面是故障检测和定位。

故障检测一般是衡量网络状态的开端，一旦网络存在异常，即可向 SDN 控制器发出故障信息。衡量网络是否故障的标准来自服务水平协议 (service level agreement, SLA) 和服务提供商制

定的准则 2 个方面[4]。通过系统监控收集信息，并进行初步判断与分析。当监测到大量信息时，有必要先进行数据分析以及分类，以便后续故障定位工作的进行。检测性能指标也是衡量网络状态的一种方法，比如延迟、丢包率、带宽等。文献 [12] 通过结合网络链路丢包率、带宽以及拥塞性能参数得到链路拥塞的范围，再通过网络拓扑对网络中故障链路进行定位。但此方法只适用于小规模 SDN，且考虑故障的影响因素较少。

故障定位指的是从网络信息中推断出导致网络异常的组件。根据文献 [9] 的描述，故障定位可大致分为 2 种技术：白盒技术和黑盒技术。白盒技术通过对节点、事件、警报和故障之间的关系进行建模来解释故障和故障的传播，常用的技术有贝叶斯网络、Petri 网等。黑盒技术是基于数据分析的模型学习，学习的条件和结果已知，但学习的过程往往是未知且让人难以理解，常用的技术有卷积神经网络、决策树、支持向量机等。文献 [13] 提出使用机器学习方法完成自动检测网络状态和故障定位，决策树、梯度提升和极端梯度提升机器学习算法将网络状态分为了正常、拥塞和网络故障 3 个状态。SDN 在逻辑高度集中的条件下对计算、存储、网络等多种资源进行统一控制，使管理员能够使用机器学习方法有效地解决网络问题[14]。

以上故障诊断的过程将在第 2、3 节中详细介绍。

2　系统监控

系统监控是获取网络信息、排除故障的第 1 步，及时监控网络状态能够有效阻断网络中更大故障的发生。传统网络由于控制和转发相互影响以及缺少交换机等必要条件的支持，获取网络数据受到的制约因素较多[5]。SDN 的发展给系统监控带来新的实验条件，控制器可以通过南向协议收集交换机中的信息，从而达到对数据的监控，也可通过北向协议在应用层中进行数据分析。监控流程可分为 3 个步骤：数据收集、存储和处理，如表 1 所示。数据存储在 SDN 中可直接根据南向协议将数据平面的统计信息发送到控制器。SDN 中一般采用控制器轮询机制，控制器向交换机发送 Stats_Request 消息，交换机将统计信息以 Stats_Reply 消息返回，比如 OpenTM 和 penNetMon。下面将结合 SDN 实际应用情况详细介绍数据收集和数据处理 2 方面。

2.1　数据收集

1）原始数据获取：在传统网络中，数据采集方式分为主动探测和被动提取，但是网络中的计数器大多以亚秒级速度捕获数据并进行更新，这并不能满足 SDN 实时监控的需求。为此，开发者通常在传统算法基础上结合 SDN 设备功能进行速度的提升。SDN 控制器能够随时对交换机发起询问，接收并存储网络信息，提升数据采集的粒度、减少网络开销。文献 [15-16] 都提出了基于 sFlow 的流量监控系统，实时收取流量统计信息。文献 [15] 提出 OpenSample 网络测量平台，OpenSample 利用 sFlow 获取每个交换机数据包的实时样本，以提供对网络负载和流量的近实时测量，最后可达到 100 ms 的控制循环。文献 [16] 在 SDN 虚拟环境下，定期从 sFlow-RT 中获取虚拟层、物理层的流量统计信息，并将它们以时间序列格式存储在数据库中，进行数据分析和可视化。与文献 [16] 类似，NetMon[17] 系统监测也同样适用于虚拟网络服务，通过收集网络内部性能指标监测追踪整体网络功能是否退化。为了提高检测精度，文献 [18] 提出了基于机器学习的 SDN 流量监控方法 IPro，IPro 优化了探测间隔，大大减小了 SDN 控制器的负载，使得 SDN 控制器在维持基本功能的情况下及时有效地收集流量信息。

2）数据预处理：在接收到大量数据后，需要对原始数据进行简单处理，即对数据进行过滤和分类整理[5]。数据预处理将有效数据和无效数据分开，有助于提升后续数据处理的精度和速度。SDN 中 OpenSketch 测量框架先以哈希数运算减少数据量，再定制通配符规则对数据分类，最后根据不同精度的需求对数据流进行统计整合。文献 [19] 针对 SDN 多层网络流量监控问题，提出 2 阶段选择节点的算法，第 1 阶段计算单个节点对层间的影响，挑选出核心节点；第 2 阶段在最小化带宽成本的约束条件下，更新核心节点并修正实时流量参数进行流量监控，这样可以以较低的带宽成本保证监测的精度。与文献 [18-19] 相比，文献 [20] 在考虑有限的测量成本的情况下，最大程度地提高交换机中存储空间的利用率。提高对流量的获取利用率也是 SDN 的一个热门研究，SDN 中的 OpenTM[21]、iSTAMP[22]、OpenNetMon[23] 都是测量流量矩阵的方法。OpenTM 直接从流中读取信息获取流量矩阵，这种方法侧重于信息的准确性，但不适用于大规模网络；iSTAMP 将三元内容可寻址存储器（ternary content addressable memory, TCAM）条目分为 2 部分：一部分用于使用流量矩阵以最佳方式聚合部分传入流，另一部分用于对信息流进行逐流监控。

2.2　数据处理

数据处理的主要工作是将上述的原始数据进行分析与表示，由于数据分析应用的范围较广且涉及到故障诊断，故这一部分在第 3 节中有更详细地说明。

1）数据分析：此阶段将收集分析统计数据，查看网络状态并应用到特定事件中。常用的统计数据有吞吐量、延迟、丢包率、流量等。文献 [24] 在 OpenDayLight 控制器基础上实现了 SDN 监控工具 SCSCDaylight，它通过图论的拓扑分割算法将交换机划分到不同控制器区域中，提高了全局监控效率，实现了双向链路速度、丢包率的测量。为解决交换机与控制器之间的延迟以及由于链路故障导致控制器之间的通信问题，文献 [25] 提出基于延迟和负载优化的动态控制器放置方法。此算法通过构建基于任务延迟和动态约束的模型，采用启发式蚁群算法解决资源分配问题，优化资源分配的同时减少了延迟时间。除了网络内部数据统计外，此阶段还可以对网络进行异常检测，比如服务攻击[26]、异常数据包[27]、未经许可的程序入侵等。在网络服务攻击中，分布式拒绝服务 (distributed denial of service, DDoS) 是最常见和最危险的。由于层间是开放接口，SDN 架构中的层间链路更容易受到 DDoS 的攻击，造成平面之间通信拥堵。文献 [28-29] 为精准检测 DDoS 攻击，前者提出了递归神经网络分类器模型，可快速学习 DDoS 攻击的特征，准确拦截异常信息；后者提出了一个基于在线聚类的入侵检测系统，利用交换机源和目的 IP 地址和端口的熵，检测 SDN 网络中不断出现的攻击。文献 [30] 采用备份控制器来审计从主控制器及其交换机收集到的网络更新事件的处理信息，并通过识别主控制器、备份控制器和交换机之间不一致的数据来检测受损的设备。数据分析针对不同的故障问题，提取网络数据的方法也各有不同。

2）数据可视化：将上述数据分析的结果以某种形式导出，比如网络拓扑图、网络流量图等。SDN 中大多数的控制器已经拥有拓扑图形的用户界面，发现网络拓扑主要依赖于链路层发现协议（link layer discovery protocol, LLDP），控制器利用 packet_out 消息指示交换机将 LLDP 包通过指定的端口进行转发，最终以 packet_in 消息回复控制器，控制器对 packet_in 消息中的 LLDP 包进行解析，从而得知网络拓扑信息。控制器中的可视化程度也稍有不同，Ryu 中可视化虽然可以看到网络拓扑，但只显示交换机之间的连接，ONOS 则是将控制器与交换机之间的连接全部显示。不同的控制器可视化界面虽有不同，但是 SDN 这种以编程实现实时交互网络状态的方式更加适应网络的动态变化。

表 1　系统监控

Table 1　System monitor

阶段	主要方法	与传统网络的对比
数据采集	OpenSample、OpenFlow+sFlow、sFlow-RT、IPro、NetMon	SDN设备(如OpenFlow交换机)可以记录运行数据并保存，比传统网络借助外部工具收集信息更加方便；传统网络的数据采集速度比较慢，并不适用于SDN动态快速变化的网络架构
数据预处理	OpenSketch、iSTAMP、OpenTM	SDN可以直接读取由转发设备保存的信息，而传统网络需使用数学和统计假设方法读取信息
数据传输	开放接口协议	SDN可使用开放接口取代传统网络的控制方法，SDN中的信息可直接存放于控制器中
数据分析	流量矩阵、FOCES、SCSCD控制器	SDN中更偏向于自动调整SDN设备进行数据分析，网络信息可随时更新。传统网络只能在分析完网络状态后才能手动更改网络配置
数据可视化	控制器中GUI	传统网络可视化方法依赖于发现和分析结果，而SDN控制器具有交互式接口，可实时根据设备的变动提供全局视图

3　故障诊断

故障诊断是对系统监控的下一步应用，软件定义网络在某种程度上简化了繁琐的传统网络结构，但架构中存在着多层网络关联的关系和复杂的网络状态，这使得 SDN 在可扩展性、拓扑、粒度、容错等方面都面临挑战。下面将分成 5 个方面分别概述 SDN 中的故障诊断算法。

3.1　基于探针的故障诊断

一般基于探针的诊断方法分为 2 种：基于规则和路径。基于规则的方法指的是为每个测试的规则发送定制的测试数据包，而基于路径是使用一个测试数据包跨越不同交换机的多个规则，以判断或定位故障交换机[31]。

典型的基于规则的算法 Monocle, 侧重于验证交换机中规则的存在, 但并不能验证网络运行过程中转发的正确性。影响数据平面中数据包能否正确转发的关键元素是规则, 规则指定数据包的匹配和转发动作。规则如果没有安装成功, 对应地会发生缺失故障; 数据包如果没有按照优先级进行匹配, 对应地会发生优先级故障。为了解决优先级故障, 文献 [32] 提出 RuleScope, 不仅能验证规则的存在还能排除优先级故障。根据探测包的反馈(这里指的是探测包的匹配动作是否和对应的规则一致)来检测数据平面的转发故障, 当网络转发规则动态变化时, 增量算法将对新的规则生成对应的探测包。RuleScope 虽然能够检测动态网络, 但文献 [33] 中指出 RuleScope 算法中规则更新的速度可能很大程度上不能与频繁更新的网络配置同步。为此, 针对增量更新慢、探针生成慢、提高检测准确率几个问题, RuleChecker 使用多轮探测算法消除误报。与 RuleScope 通过解决布尔可满足问题生成探针的方式不同, RuleChecker 通过二元决策图执行集合运算, 快速生成探针; 在规则变动时, RuleChecker 通过判断优先级和是否有重叠的规则计算生成最少数量的探测包, 根据实验证明, 90% 的规则插入、删除等更新操作, 此算法更新时间不到 2 ms。SDNProbe[34] 将探测包数量问题看成是在有向无环图上最小合法路径问题, 以最小的探测包数量遍历所有流表, 如果控制器没有定期收到探测包, 则认为此探测包探测的路径是有可能故障, 则需对此路径上的交换机进行多次检测。SDNSpotlight[35] 根据头空间的角值创建探测包, 按照预期路径开始探测过程, 在预期路径的最后一个节点上安装一个捕获规则。如果没有遵循预期的路径, 探测数据包将通过路径附近的捕获规则发送到控制器。与 SDNProbe 相比, SDNSpotlight 不仅减少了探测包生成时间, 还填补了关于环路拓扑故障检测的内容。文献 [36] 针对链路泛洪攻击提出一种增量检测方案, 根据交换机中预定义的规则检测链路状态, 如有异常控制器发放探测包定位故障链路。增量化的检测方案将网络中的可用路径最大化, 但当链路攻击超过可控制范围时, 探测过程中丢包是不可避免的。

基于探针的故障诊断算法大多适用于判断数据层交换机的流表规则是否与控制器发放的规则一致, 进而判断交换机是否存在故障。上述算法在控制器中有比较大的开销, 且在定位故障时一般要经过多轮测试才能精准定位。大量发放探针必然影响控制器响应事件的速度, 交换机中如果存在大量规则流表也会影响有限的 TCAM 空间。基于探针的诊断方法在控制开销的同时对检测和定位故障的精度和灵活性还需提高。与传统网络相比, SDN 中的算法对网络需求的动态变化有更大的包容性。

3.2　基于协议的故障诊断

协议在 SDN 网络架构中占有很重的比例, 承担了通信、拓扑发现、故障检测、信息管理等功能, 研究者会利用部分协议进行网络故障排查。上述提到的 LLDP 协议也可以通过采集每一个交换机的信息来对数据平面进行故障发现。双向转发机制(bidirectional forwarding detection, BFD)专门用来检测 2 个转发点之间是否存在故障, 在 SDN 架构中多用在数据平面交换机之间的故障检测。

文献 [37] 提出了一种监控网络中所有路径延迟的算法, 控制器将发送 2 种不同的 LLDP 包给交换机, 一种是添加额外时间戳的 LLDP-TLV 包, 经过循环发放监控交换机之间的延迟和获取拓扑, 另一种是单纯获取拓扑的 LLDP 包。为降低控制器开销且能监控所有链路, 文献 [37] 利用贪婪算法选出需注入 LLDP-TLV 包的交换机。与传统测量方法 Ping 相比, 这种方法在往返时间上表现更好。虚拟层的延迟间接地由物理层结果相加得到, 此结果完全依赖于物理层的计算结果, 虚拟延迟计算过于简单且在实际仿真中并不准确。LLDP 中也存在漏洞, 攻击者容易伪造或拦截 LLDP 协议混淆当前的网络状态, 造成虚假链接。文献 [38] 针对拓扑攻击提出一种异常检测机制(correlation-based topology anomaly detection, CTAD), 对于异常模块, CTAD 采用计算 LLDP 包的往返时间差和 Spearman 等级相关性分析流量判别是否有攻击者对拓扑信息进行改变。文献 [39] 针对 LLDP 可能引起的泛滥攻击、重放等攻击, 在 LLDP 协议基础上提出了 SDN 链路发现协议 (SDN link discovery protocol, SLDP)。SLDP 可预防、检测和缓解由于缺乏源认证、缺乏数据包完整性检查和静态数据包重用而导致的各种安全威胁。与 LLDP 相比, SLDP 使用较少的数据包维持控制器中的全局网络拓扑。文献 [40] 利用 LLDP 协议对网络进行拓扑发现, 根据 SDN 网络中延迟、吞吐量和丢包率 3 个参数的概率密度函数提前识别具有高概率故障的链路, 确定故障链路后并计算出备份路径, 减少网络恢复的时间。为了将网络状态更加细致地分类, 文献 [41] 将链路的状态分为正常、故障和拥塞情况, 建立了基于双

向转发机制的拥塞检测系统，在网络中检测到故障后，使用基于数据平面的容错机制解决视频流的拥塞问题。BFD 用作故障检测算法比链路层发现协议、信号丢失（loss of signal, LoS）更有优势，因为 BFD 仅仅用于故障检测，且在检测时间上更优于其他方法。发送探测包或使用 LLDP 是常见的网络探测手段，但总会受到一些限制，例如流规则的预配置、校准的必要性等，文献 [42] 在 BFD 中加入 Echo 模块，实现 BFD 的链路延迟测量，为了减少控制器的开销，将延迟信息存储在 LLDP 包返回给控制器，这样既可以不提前布置流表，还可以准确计算网络延迟。

SDN 利用网络中存在的协议进行网络信息收集比对，减轻了控制器生成和回收额外资源的负担，但是随着网络规模的增大，交换机与控制器之间频繁的对话必然导致双方消息发送和接收的缓慢，从而延长最终故障检测的时间。且基于协议的故障诊断一般只能排查控制器与交换机之间通信往来的简单故障，没有类似于根本原因分析、疑似故障排除等复杂功能。协议功能的单一性使得它并不具备对复杂故障分析的能力，所以 SDN 中协议的多功能开发也一直是研究的重点。

3.3　基于模型的自诊断

在虚拟网络中，人为对网络故障进行检测和定位不再适用于虚拟网络动态变化的环境，自动检测网络状态和故障诊断的模式逐渐受到关注。基于模型的原理与白盒技术类似，研究者在建立明确模型的基础上进行故障推理。模型可以从不同的信息源构建依赖关系、网络资源和事件。该方法主要分为建模和推理 2 个模块，模型不仅要将故障和症状联系起来，还要包含网络中所有的元素。推理算法识别系统故障的根本原因，并定位故障组件。

诊断模型一般依赖于网络中依赖关系图的构建，依赖关系图最早用于传统网络故障定位，是网络监控的直观表示图。依赖关系图由警报、中间故障、故障原因构成，节点之间有边连接表示在故障传播中有因果关系。传统的依赖关系图是由专业知识手动生成，适用于静态网络拓扑，也称为网络拓扑的快照，但并不适用于类似 SDN、网络功能虚拟化 (network functions virtualization, NFV) 的动态拓扑结构。故文献 [43-47] 都针对虚拟网络中如何自建模创建了新的算法。推理算法使用较多的是贝叶斯网络（Bayesian network, BN）。BN 中节点可代表症状，节点之间的连接表示有必然的因果关系，它们之间影响的强度由条件概率决定，最终分析得到网络元素故障的概率。在 SDN 环境中基于模型的诊断算法如图 3 所示（推理算法以 BN 为例）。

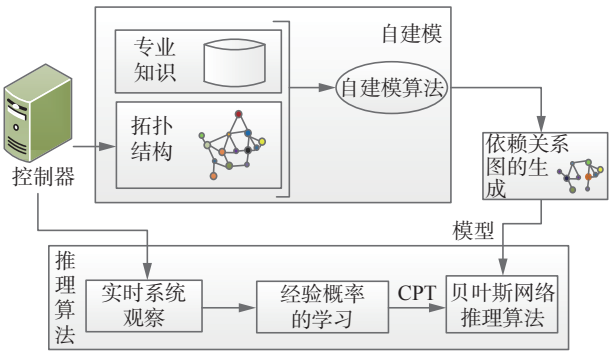

图 3　基于模型的诊断算法示意

Fig. 3　Schematic diagram of the model-based self-diagnosis algorithm

文献 [43] 提出一种自动诊断方法，将自建模与 BN 联合起来推理出最有可能故障的组件。此故障诊断方法从控制器出发，一方面从控制器中得到拓扑结构图，建立节点、链路 2 种网络模板，进行模板实例化、拓扑分类和重新组装得到依赖图；另一方面根据控制器中的实时观察，得到告警信息以及节点状态信息进行先验概率的学习，得到条件概率表作为证据推理出依赖关系图模型中故障可能性最大的组件。此方法不仅解决了网络动态变化的问题，还将故障原因扩大到网络中不同粒度的组件。文献 [44] 是对文献 [43] 工作的补充，在物理层与逻辑层的基础上加入了虚拟层、服务层，并将多个网络层之间进行关联，考虑与底层网络之间的动态依赖关系，构造依赖关系图与 BN 模型进行故障诊断。文献 [45] 中的诊断平台不仅包含了故障传播模型的自动生成和故障诊断，还增添了自我修复功能。文献 [43-45] 虽然能够针对不同的组件进行故障诊断，但依赖关系过于复杂、网络元素数量增多导致推理算法复杂度上升。文献 [46] 与文献 [43] 虽然都是采用了基于模型的方法，但是文献 [46] 中的模型并没有直接根据控制器中的信息构建依赖关系图，而是从故障注入、专家知识、服务描述和需求 3 个方面学习和验证网络中的依赖关系，相对而言文献 [46] 采用的构建方法对依赖关系图的描述更加具体。文献 [47] 采用了一种将生产环境和仿真环境结合的诊断方法，通过在线监视器跟踪和汇总的跟踪数据，建立系统行为模型。如果发生故障，则在仿真环境中对故障进行重演，经过离线诊断系统进行差异化检查，识别故障的因果关系并输出诊断书。此算法保证了在不干扰实际的生产生活的情况下还能准确地识别故障且网络性能比较稳

定, 但此模型并不包括故障检测功能。文献 [48] 实现了在大型网络中自动监控并检测网络异常, 首先通过 OpenFlow 协议收集存储流量信息, 计算信息熵对流量行为进行分析。其次对正常流量行为进行特征提取, 以此识别可疑事件, 最后生成已识别的攻击报告。与基于 k 近邻的分类检测算法和支持向量机技术相比, 文献中提出的数字签名的蚁群优化算法具有更高的检测率。

文献 [43-48] 都在自动建立模型后应用 BN 进行下一阶段的根本原因分析, 识别故障可能性最大的组件。文献 [49] 提出利用大数据处理和监督网络学习算法进行信息存储与分析, 选取易受故障影响的数据类型, 再利用 BN 进行故障分析。文献 [50] 面对 DDoS 攻击选择使用数据包进行过滤规则匹配和 BN 模型进行异常数据包的检测和过滤。文献 [51] 提出了一种将 BN 与基于案例的推理方法混合的算法, 由于 BN 的复杂度随着网络节点的数量和网络组件间的依赖关系呈指数增长, 故研究者采用基于案例的方法中和这种推理过程, 既保留了 BN 的优点也能降低算法的复杂度。贝叶斯网络故障诊断方法大部分不再单独使用, 一般作为推理算法对收集的故障信息进行故障可能性的推测。

基于模型的自诊断方法打破了需要手动构建模型的模式, 在面对复杂的网络结构时自动建立网络模型进行故障分析的形式已经在未来网络中迅速发展。自诊断算法由不同功能的模块构成, 模块之间分工不同, 相互解耦, 但网络组件间依赖关系的构建十分影响推理算法最终的分析结果, 分析的组件过多容易导致怀疑因素过多, 反之容易遗漏故障组件, 所以网络架构、依赖关系的学习和推理算法性能的提升对推理结果的合理性有很重要的修正作用。

3.4　基于故障注入的故障诊断

故障注入指的是在网络中注入已知类型的故障, 收集故障下的网络性能统计数据, 评估网络提供和维持可接受的 SLA 的能力。在传统网络中, 故障注入旨在测试网络协议的实现、系统的稳健性。在虚拟环境中, 故障注入一般为机器学习方法提供学习和验证数据集[9]。研究者根据故障注入模拟观察并记录网络的反应和症状, 当真实故障发生可查找故障类型及故障过程。

文献 [52] 针对虚拟网络故障诊断, 提出了基于故障注入的诊断算法。为了提取故障发生的路线, 将常见故障注入虚拟网络中, 对数据包进行了标记并将路线以日志的形式输出。再利用决策树对这些故障进行特征提取和分类, 以便对故障过程和结果进行分析。此算法在故障识别中只能分析前期注入的故障类型, 如果有新型故障发生, 则很有可能造成故障的误判。文献 [53] 通过本地虚拟机异常和繁重负载下全局测试 2 种不同类型的故障注入方法, 模拟了 6 种不同的异常: 中央处理器异常、内存增加、磁盘访问异常、网络丢包、网络延迟和网络负载增加。为了不对相互隔离的租户间产生影响, 文献 [54] 提出了一种新的故障注入解决方案 ThorFI。所提出的架构与云基础设施管理平台交互, 以获取有关虚拟网络的高级信息, 如虚拟路由器, 并将它们映射到低级资源, 如物理网络接口、进程和网络流。然后, ThorFI 编排跨数据中心节点部署的低级注入器, 以确保仅将故障注入目标虚拟网络, 而不会影响其他租户。不仅如此, ThorFI 包含了现有的所有故障模型, 以便在多租户网络中应用。为了能在虚拟网络更好地模拟故障发生, 文献 [55] 提出一种可编程的故障注入方法 ProFIpy, 对用户自定义和添加新的故障模型有很大的包容性。ProFIpy 支持先前导入的故障模型以及预定义故障模型, 经过对故障模型进行存储后, ProFIpy 输入目标软件的源代码, 通过对原版程序代码的改变来模仿网络中各种故障, 比如在代码转换过程中, 将参数删除或者改写、设置出厂错误、删除某方法的调用等。模拟结束后, ProFIPy 从目标系统收集日志进行数据分析。

故障注入之前是基于先前故障案例进行故障复现、识别和追踪, 随着网络结构和应用复杂化, 故障注入在应用的空间、模拟故障的数量、模拟的效果都有所提升, 故障注入一方面可以作为机器学习算法的数据库, 一方面也可以作为故障诊断算法的一部分。基于故障注入的诊断方法并不能检测故障, 如果想在网络系统中实现全面故障诊断, 则需要配置检测故障算法。另一方面故障注入诊断方法还有很多限制, 虽然能够模拟现存的故障, 但一般来说能够模拟的算法有限, 而且故障注入对新型网络的故障收集的并不全面, 由此可能导致此类诊断方法对特定的故障有效, 且十分依赖历史故障数据。

3.5　基于带内网络遥测的故障诊断

网络遥测[56] 已成为一个主流技术术语, 指代较新的网络数据收集和消费技术。遥测是用于远程收集和处理网络信息的自动化过程。随着软件定义网络的发展, 带内网络遥测应运而生。带内网络遥测利用数据平面直接驱动网络测量过程, 颠覆了传统网络测量将网络交换设备视为中间黑匣子的研究思路[57]。带内网络遥测是网络遥测的

一种，将数据包转发与网络测量相结合来收集网络状态。使用带内网络遥测，网络管理员可以直接从数据平面捕获由性能瓶颈、网络故障或错误配置引起的瞬态问题。带内网络遥测技术既不需要像主动探测式发送探针，也不需要像协议式故障诊断方法对设备进行轮询机制，它自身可通过自动化调节达到网络故障诊断低开销目的。

带内网络遥测技术相对于探针和协议故障诊断技术更适合于自动化故障诊断，但带内网络遥测技术对网络中的丢包问题存在漏洞，文献 [58] 提出的 LossSight 算法针对丢包问题提出了详细的解决方案，使用标记策略检测丢包时间、位置和原因，之后用生成对抗网络算法学习丢失的信息，进行信息恢复。文献 [59] 介绍了如何将带内网络遥测技术融入到 SDN 网络中，以及如何在 SDN 网络中进行故障诊断。文献 [60-61] 对 SDN 这类动态网络提出了低开销、细粒度的网络监控方法。文献 [60] 在 ONOS 控制器上对带内网络遥测进行了评估，经过带内遥测收集的数据在控制器中进行分析，可以以秒为单位进行更新显示，但是这对于虚拟动态网络来说更新速度还有待提高。文献 [62] 提出了一种自动对网络进行检测的系统 INT-detector，将带内遥测和生成对抗主动学习算法（generative adversarial active learning, GAAL）结合，带内遥测实现对网络每个设备的信息收集，GAAL 算法实现异常检测。自动化检测系统可以大大减小人为干预，且使用了低通滤波进行数据预处理，能够检测长时间存在的故障。文献 [63] 提出策略感知带内网络遥测技术 (policy-aware in-band network telemetry, PAINT)，指定网络运营商使用服务供应语言来定义和部署带内网络遥测服务。通过带内网络遥测收集网络内部信息进行网络故障排除，PAINT自动解析服务策略并通过推断网络组件与症状之间的因果关系，基于因果关系模型部署端到端的症状监测工具并进行分析，动态的症状−故障−遥测模型满足了对网络实时的监测与故障定位需求。

带内网络遥测技术是近几年新发展的技术，主要收集网络数据并进行故障检测，应用场景大多是动态新型网络。带内网络遥测技术符合当今网络发展的趋势，大大减少人为干预，向自动诊断技术靠拢，逐步实现网络低开销、细粒度、跨层的实时故障诊断。但同时遥测技术发展并不完善，存在对丢包问题不敏感、遥测技术额外的带宽成本、遥测数据存储问题等方面的漏洞。

本节详细介绍了在 SDN 架构中的 5 种故障诊断算法。为了更加直观地比较各种算法，表 2 介绍了每种算法的优缺点以及研究成果。

表 2　故障诊断方法对比

Table 2　Comparison of fault diagnosis methods

诊断方法	优点	缺点	研究成果
基于探针的故障诊断	简单易实现，对SDN不会造成侵入性故障	容易对控制器造成巨大的开销，且在定位故障交换机时一般都要经过多轮测试才能精准定位。大量发放探针必然会对控制器响应事件的速度产生影响，且交换机中TCAM有限，需考虑其利用率	文献[31-36]
基于协议的故障诊断	无需控制器额外发放探针，网络内部以数据包转发的形式即可完成网络故障检测与定位	检测定位故障的能力有限，对数据平面的转发异常、数据包异常简单故障有效，但对复杂故障以及在虚拟化环境中可能需要其他算法辅助	文献[37-42]
模型自诊断	可对动态网络完成实时故障自动诊断，并且考虑网络细粒度的组件	需要大量学习网络依赖关系，如果依赖关系确实或者错误则直接影响整体的故障判断	文献[43-51]
基于故障注入的故障诊断	对已知的故障能够快速追踪并复原故障的轨迹	一般来说能够模拟的算法有限，而且故障注入对新型网络的故障收集的并不全面，由此可能导致此类诊断方法对特定的故障有效，且十分依赖历史故障数据	文献[52-55]
基于带内网络遥测的故障诊断	对网络中的瞬时故障有很好的捕获能力，并达到网络细粒度和低开销的目的	遥测技术发展并不完善，存在对丢包问题不敏感、遥测技术额外的带宽成本、遥测数据测存储问题等方面的漏洞	文献[56-63]

4　未来研究方向

学术界和工业界正在对 SDN 进行更深入的研究，SDN 为网络的创新发展带来了更多的机会，但同时也面临着更多的挑战。

1）高效故障诊断：高效的故障诊断在复杂故障环境中表现为快速准确地检测、识别和分析。首先，SDN 中多种协议的存在使得数据平面更加

多样化和复杂，控制平面中也需要提供更加复杂的程序来支持多协议网络管理。但近几年的文献显示 SDN 数据平面中故障假设过于单一，没有考虑数据平面故障的复杂性，默认为数据平面中发生的故障是数据包转发故障、交换机故障、链路故障等简单的物理故障。且现有的故障管理解决方案都集中在一种协议，通常是 OpenFlow。SDN 需设计更强大的编程协议，包括网络管理、可靠性和安全性，这是未来 SDN 一个有希望发展的方向。其次，完整的故障诊断流程对网络的使用寿命来说非常重要，Cherrared 等[64] 特地为 SDN、NFV 制定故障管理框架，包括了事件和警报通知的检测和存储、过滤程序以及根本原因定位、影响分析和纠正措施的诊断检查。完整、详细的故障诊断对 SDN 网络稳定运行有着重要意义。

2）虚拟化技术性能的提升：网络虚拟化技术与 SDN 架构结合，更容易实现自动化和弹性化的未来网络。在虚拟化中，虚拟交换机，如 OVS (Open vSwitch) 和虚拟化平台的性能本身对网络整体性能有很大影响。OVS 在大规模网络中，正确转发数据的性能急转直下，给网络带来极大安全隐患。虚拟化平台集中式的管控形式虽然可以实现网络的高效管理，但架构设计存在可靠性和拓展性问题，比如主机的移动会造成网络拓扑、对应功能的改变，虚拟化平台需要对此做出调整，这极易成为 SDN 虚拟化发展的瓶颈[1]。目前，虚拟化平台的功能设计因开发者不同而有差别，平台的接口协议、对底层资源虚拟化的定义没有标准化的规定，标准化对实现多虚拟化平台的协同合作至关重要。虚拟化技术是一种扩大有限物理资源的新理念，但目前在 SDN 架构上还不够成熟，虚拟设备在性能、接口标准化、动态管理方面还需进一步优化研究。

3）SDN 管理的自动化：在现代复杂网络的需求下，网络系统需要提供许多功能，如模块化协作、状态同步、资料备份、负载均衡等[6]。除上述功能外，SDN 还需设计网络监控、故障诊断等系统。手动配置所需系统难度高且效率低，自动化在一定程度上可完成网络的便捷管理。由近几年学术工作和开源项目可得，将机器学习、人工智能、频谱故障定位[65] 引入 SDN 中实现自动化管理是一种趋势。知识定义网络[66] 是 SDN 与机器学习、人工智能相结合的新型网络范式，负责根据控制器收集的信息对网络进行分析，机器学习将这些数据转换为知识，并根据这些知识做出网络决策。OpenDayLight 和 ONOS 控制器在开源项目上也根据获取的信息利用机器学习进行网络自动管理等功能。SDN 和机器学习、人工智能的结合将简化网络以满足各种需求，并提高网络可靠性和可用性。

4）SDN 的自我修复：除了上述 SDN 管理自动化、故障诊断自动化外，网络的自动修复也是另一个研究热点。实现网络自愈机制需要一个全面的故障管理解决方案，包括在线监控系统、故障检测和定位机制、恢复网络的故障修复和恢复机制、维护正常的容错框架操作。SDN 在其架构的每一层都面临着新的故障和问题，有必要解决这些问题并保护 SDN架构的每一层，以增强 SDN 架构的容错能力[67]。SDN 的自愈系统也是提高网络整体架构可靠性的重要途径之一。

5）网络安全：SDN 以开放式接口进行信息传递，这极有可能引入新一轮的网络攻击，如通过交换机向控制器进行 DDoS 攻击、攻击者占用网络带宽[68] 等，甚至可能攻击侵占控制器，造成 SDN 整体网络瘫痪。如果 SDN 控制器被攻击入侵，那攻击者能够操控整体网络，获取所有网络信息；在虚拟环境中，虚拟化平台被入侵，租户的所有虚拟业务都会受到影响，又由于虚拟层与其他层之间的透明性，此时控制器和交换机都无法察觉，攻击者不仅可以控制网络，还可以和网络外界随意进行网络信息交易。在这种情况下，入侵检测系统代表了网络安全的重要组成部分之一。入侵检测系统的目标是通过检测、分类和警告入侵企图来减少对给定基础设施的未经授权的访问，以保持可用性、完整性和机密性[69]。入侵检测系统需要对大量实时数据进行分析，关键特征的提取对检测的复杂性和快速性有决定性影响。安全的认证机制和入侵检测系统的制定对 SDN 来说显得尤为重要。

5　结束语

随着 SDN 在未来网络中应用的增加，SDN 中的故障诊断技术在工业界和学术界也受到广泛关注。故障诊断算法能够及时有效地识别定位网络故障，为之后的网络恢复提供精确信息，在一定程度上可降低网络再次受到故障干扰的风险。本文在系统监控方面主要突出了与传统网络的区别，在故障诊断方面重点分析了各类方法的应用与差异，大部分算法不再只适用于单层、静态网络，而是逐渐转向具有动态、多层等特征的虚拟网络。在后续的研究中，研究人员还需完善 SDN 中的故障诊断模型，做到对拓扑变化的高包容

性，提高诊断的精确性、灵活度，进一步研究 SDN 的容错机制和安全漏洞等方面，做到实时风险预测、故障诊断和快速故障恢复。

参考文献：

[1] 杨泽卫, 李呈. 重构网络: SDN 架构与实现 [M]. 北京: 电子工业出版社, 2017: 1−255.
YANG Zewei, LI Cheng. Refactoring networks: SDN architecture and implementation[M]. Beijing: Publishing House of Electronics Industry, 2017: 1−255.

[2] ALEX A, BARAKABITZE. 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges[J]. Computer networks, 2020, 167: 1–40.

[3] DUSIA A, SETHI A S. Recent advances in fault localization in computer networks[J]. IEEE communications surveys & tutorials, 2016, 18(4): 3030–3051.

[4] CÉRIN C, COTI C, DELORT P, et al. Downtime statistics of current cloud solutions[EB/OL]. (2014-03-31) [2022−05−10]. http://iwgcr.org/wp-content/uploads/2014/03/downtime-statistics-current-1.3.pdf.

[5] FONSECA P C, MOTA E S. A survey on fault management in software-defined networks[J]. IEEE communications surveys & tutorials, 2017, 19(4): 2284–2321.

[6] TSAI P W, TSAI C W, HSU C W, et al. Network monitoring in software-defined networking: a review[J]. IEEE systems journal, 2018, 12(4): 3958–3969.

[7] YU Yinbo, LI Xing, LENG Xue, et al. Fault management in software-defined networking: a survey[J]. IEEE communications surveys & tutorials, 2019, 21(1): 349–392.

[8] 张朝昆, 崔勇, 唐嚣嚣, 等. 软件定义网络 (SDN) 研究进展 [J]. 软件学报, 2015, 26(1): 62–81.
ZHANG Chaokun, CUI Yong, TANG Hehe, et al. State-of-the-art survey on software-defined networking (SDN)[J]. Journal of software, 2015, 26(1): 62–81.

[9] CHERRARED S, IMADALI S, FABRE E, et al. A survey of fault management in network virtualization environments: challenges and solutions[J]. IEEE transactions on network and service management, 2019, 16(4): 1537–1551.

[10] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow[J]. ACM SIGCOMM computer communication review, 2008, 38(2): 69–74.

[11] BAYS L R, GASPARY L P. Reality shock in virtual network embedding: Flexibilizing demands for dealing with multiple operational requirements in SDNs[J]. Journal of network and computer applications, 2020, 153: 1–13.

[12] 程光, 王玉祥, 胡一非, 等. 基于 OpenFlow 的链路故障诊断方法 [J]. 北京邮电大学学报, 2015, 38(5): 42–46.
CHENG Guang, WANG Yuxiang, HU Yifei, et al. Network link fault diagnosis based on OpenFlow[J]. Journal of Beijing University of Posts and Telecommunications, 2015, 38(5): 42–46.

[13] MOHAMMED A R, MOHAMMED S A, CÔTÉ D, et al. Machine learning-based network status detection and fault localization[J]. IEEE transactions on instrumentation and measurement, 2021, 70: 1–10.

[14] ZHAO Yanling, LI Ye, ZHANG Xinchang, et al. A survey of networking applications applying the software defined networking concept based on machine learning[J]. IEEE access, 2019, 7: 95397–95417.

[15] SUH J, KWON T T, DIXON C, et al. OpenSample: a low-latency, sampling-based measurement platform for commodity SDN[C]//2014 IEEE 34th International Conference on Distributed Computing Systems. Piscataway: IEEE, 2014: 228−237.

[16] REHMAN S U, SONG W C, KANG M. Network-wide traffic visibility in OF@TEIN SDN testbed using sFlow[C]//The 16th Asia-Pacific Network Operations and Management Symposium. Piscataway: IEEE, 2014: 1−6.

[17] VULET P, BOSAK B, DIMOLIANIS M, et al. Localization of network service performance degradation in multi-tenant networks[J]. Computer networks, 2020, 168: 1–13.

[18] CASTILLO E F, RENDON O M C, ORDONEZ A, et al. IPro: an approach for intelligent SDN monitoring[J]. Computer networks, 2020, 170: 1–18.

[19] LI Feng, YAO Yiru, WANG Liangmin, et al. Multi-timescale and multi-centrality layered node selection for efficient traffic monitoring in SDNs[J]. Computer networks, 2021, 198: 1–11.

[20] YANG Tian, CHEN Weiwei, LEA C T. An SDN-based traffic matrix estimation framework[J]. IEEE transactions on network and service management, 2018, 15(4): 1435–1445.

[21] TOOTOONCHIAN A, GHOBADI M, GANJALI Y. OpenTM: traffic matrix estimator for OpenFlow networks[M]. Berlin: Springer Berlin Heidelberg, 2010: 201−210.

[22] MALBOUBI M, WANG Liyuan, CHUAH C N, et al. Intelligent SDN based traffic (de) aggregation and measurement paradigm (iSTAMP)[C]//IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. Piscataway: IEEE, 2014: 934−942.

[23] VAN ADRICHEM N L M, DOERR C, KUIPERS F A. OpenNetMon: network monitoring in OpenFlow software-defined networks[C]//2014 IEEE Network Opera-

tions and Management Symposium. Piscataway: IEEE, 2014: 1−8.

[24] WANG Lu, SUN Meng, TANG Shaoju. SCSCDaylight: network monitoring tools for software-defined networks based on opendaylight[C]//2019 International Conference on Intelligent Computing, Automation and Systems. Piscataway: IEEE, 2020: 320−323.

[25] LI Chunlei, JIANG Kun, LUO Yonlong. Dynamic placement of multiple controllers based on SDN and allocation of computational resources based on heuristic ant colony algorithm[J]. Knowledge-based systems, 2022, 241: 1–19.

[26] YAN Qiao, YU F R, GONG Qingxiang, et al. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges[J]. IEEE communications surveys & tutorials, 2016, 18(1): 602–622.

[27] ZHANG Peng, ZHANG Fangzheng, XU Shimin, et al. Network-wide forwarding anomaly detection and localization in software defined networks[J]. IEEE/ACM transactions on networking, 2021, 29(1): 332–345.

[28] POLAT H, TÜRKOLU M, POLAT O, et al. A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks[J]. Expert systems with applications, 2022, 197: 1–12.

[29] SCARANTI G F, CARVALHO L F, JUNIOR S B, et al. Unsupervised online anomaly detection in Software Defined Network environments[J]. Expert systems with applications, 2022, 191: 1–13.

[30] ZHOU Haifeng, WU Chunming, YANG Chengyu, et al. SDN-RDCD: a real-time and reliable method for detecting compromised SDN devices[J]. IEEE/ACM transactions on networking, 2018, 26(5): 2048–2061.

[31] HU Zhijun, WU Libing, LI Jianxin, et al. Everyone in SDN contributes: fault localization via well-designed rules[C]//2021 IEEE 41st International Conference on Distributed Computing Systems. Piscataway: IEEE, 2021: 370−380.

[32] WEN Xitao, BU Kai, YANG Bo, et al. RuleScope: inspecting forwarding faults for software-defined networking[J]. IEEE/ACM transactions on networking, 2017, 25(4): 2347–2360.

[33] ZHANG Peng, ZHANG Cheng, HU Chengchen. Fast data plane testing for software-defined networks with RuleChecker[J]. IEEE/ACM transactions on networking, 2019, 27(1): 173–186.

[34] KE Yuming, HSIAO H C, KIM T H J. SDNProbe: lightweight fault localization in the error-prone environment[C]//2018 IEEE 38th International Conference on Distributed Computing Systems. Piscataway: IEEE, 2018: 489−499.

[35] ARYAN R, YAZIDI A, BRATTENSBORG F, et al. SDN Spotlight: a real-time OpenFlow troubleshooting framework[J]. Future generation computer systems, 2022, 133(C): 364–377.

[36] WANG Lei, LI Qing, JIANG Yong, et al. Woodpecker: Detecting and mitigating link-flooding attacks via SDN[J]. Computer networks, 2018, 147: 1–13.

[37] LIAO Lingxia, LEUNG V C M, CHEN Min. An efficient and accurate link latency monitoring method for low-latency software-defined networks[J]. IEEE transactions on instrumentation and measurement, 2019, 68(2): 377–391.

[38] CHOU L D, LIU C C, LAI Mengsheng, et al. Behavior anomaly detection in SDN control plane: a case study of topology discovery attacks[C]//2019 International Conference on Information and Communication Technology Convergence. Piscataway: IEEE, 2019: 357−362.

[39] NEHRA A, TRIPATHI M, GAUR M S, et al. SLDP: a secure and lightweight link discovery protocol for software defined networking[J]. Computer networks, 2019, 150: 102–116.

[40] SEDDIQI H, BABAIE S. A new protection-based approach for link failure management of software-defined networks[J]. IEEE transactions on network science and engineering, 2021, 8(4): 3303–3312.

[41] YAMANSAVASCILAR B, BAKTIR A C, OZGOVDE A, et al. Enhancing QoE for video streaming considering congestion: a fault tolerance approach[C]//IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops. Piscataway: IEEE, 2019: 258−263.

[42] KIM S M, YANG G, YOO C, et al. BFD-based link latency measurement in software defined networking[C]//2017 13th International Conference on Network and Service Management. Piscataway: IEEE, 2018: 1−6.

[43] SÁNCHEZ J M, YAHIA I G B, CRESPI N. Self-modeling based diagnosis of Software-Defined Networks[C]// Proceedings of the 2015 1st IEEE Conference on Network Softwarization. Piscataway: IEEE, 2015: 1−6.

[44] SÁNCHEZ J M, YAHIA I G B, CRESPI N. Self-modeling based diagnosis of services over programmable networks[C]//2016 IEEE NetSoft Conference and Workshops. Piscataway: IEEE, 2016: 277−285.

[45] SÁNCHEZ J M, YAHIA I G B, CRESPI N. THESARD: On the road to resilience in software-defined networking

through self-diagnosis[C]//2016 IEEE NetSoft Conference and Workshops . Piscataway: IEEE, 2016: 351−352.

[46] CHERRARED S, IMADALI S, FABRE E, et al. SFC self-modeling and active diagnosis[J]. IEEE transactions on network and service management, 2021, 18(3): 2515–2530.

[47] YU Yinbo, LI Xing, BU Kai, et al. Falcon: Differential fault localization for SDN control plane[J]. Computer networks, 2019, 162: 1–13.

[48] CARVALHO L F, ABRÃO T, DE SOUZA MENDES L, et al. An ecosystem for anomaly detection and mitigation in software-defined networking[J]. Expert systems with applications, 2018, 104: 121-133.

[49] BENAYAS F, CARRERA A, IGLESIAS C A. Towards an autonomic Bayesian fault diagnosis service for SDN environments based on a big data infrastructure[C]//2018 Fifth International Conference on Software Defined Systems. Piscataway: IEEE, 2018: 7−13.

[50] SOPHAKAN N, SATHITWIRIYAWONG C. Securing OpenFlow controller of software-defined networks using Bayesian network[C]//2018 22nd International Computer Science and Engineering Conference. Piscataway: IEEE, 2019: 1−4.

[51] BENNACER L, AMIRAT Y, CHIBANI A, et al. Self-diagnosis technique for virtual private networks combining Bayesian networks and case-based reasoning[J]. IEEE transactions on automation science and engineering, 2015, 12(1): 354–366.

[52] ZHANG Huanhuan, DONG Fang, SHEN Dian, et al. Virtual network fault diagnosis mechanism based on fault injection[C]//2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design. Piscataway: IEEE, 2017: 384−389.

[53] SAUVANAUD C, LAZRI K, KAÂNICHE M, et al. Anomaly detection and root cause localization in virtual network functions[C]//2016 IEEE 27th International Symposium on Software Reliability Engineering. Piscataway: IEEE, 2016: 196−206.

[54] COTRONEO D, DE SIMONE L, NATELLA R. ThorFI: a novel approach for network fault injection as a service[J]. Journal of network and computer applications, 2022, 201: 1–14.

[55] COTRONEO D, DE SIMONE L, LIGUORI P, et al. ProFIPy: programmable software fault injection as-a-service[C]//2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Piscataway: IEEE, 2020: 364−372.

[56] YU Minlan. Network telemetry[J]. ACM SIGCOMM computer communication review, 2019, 49(1): 11–17.

[57] TAN Lizhuang. In-band network telemetry: a survey[J]. Computer networks, 2021, 186: 1–20.

[58] TAN Lizhuang, SU Wei, ZHANG Wei, et al. A packet loss monitoring system for in-band network telemetry: detection, localization, diagnosis and recovery[J]. IEEE transactions on network and service management, 2021, 18(4): 4151–4168.

[59] HAXHIBEQIRI J, ISOLANI P H, MARQUEZ-BARJA J M, et al. In-band network monitoring technique to support SDN-based wireless networks[J]. IEEE transactions on network and service management, 2021, 18(1): 627–641.

[60] VAN TU N, HYUN J, HONG J W K. Towards ONOS-based SDN monitoring using in-band network telemetry[C]//2017 19th Asia-Pacific Network Operations and Management Symposium. Piscataway: IEEE, 2017: 76−81.

[61] HAXHIBEQIRI J, MOERMAN I, HOEBEKE J. Low overhead, fine-grained end-to-end monitoring of wireless networks using In-band telemetry[C]//2019 15th International Conference on Network and Service Management. Piscataway: IEEE, 2020: 1−5.

[62] ZHANG Yan, PAN Tian, ZHENG Yan, et al. Automating rapid network anomaly detection with In-band network telemetry[J]. IEEE networking letters, 2022, 4(1): 39–42.

[63] TANG Yongning, WU Yangxuan, CHENG Guang, et al. Intelligence enabled SDN fault localization via programmable In-band network telemetry[C]//2019 IEEE 20th International Conference on High Performance Switching and Routing. Piscataway: IEEE, 2019: 1−6.

[64] CHERRARED S, IMADALI S, FABRE E, et al. LUMEN: a global fault management framework for network virtualization environments[C]//2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops. Piscataway: IEEE, 2018: 1−8.

[65] CHEN H H, LU Hanlin, HUANG S K, et al. Diagnosing SDN network problems by using spectrum-based fault localization techniques[C]//2015 IEEE International Conference on Software Quality, Reliability and Security - Companion. Piscataway: IEEE, 2015: 121−127.

[66] MESTRES A, RODRIGUEZ-NATAL A, CARNER J, et al. Knowledge-defined networking[J]. ACM SIGCOMM computer communication review, 2017, 47(3): 2–10.

[67] REHMAN A U, AGUIAR R L, BARRACA J P. Fault-tolerance in the scope of software-defined networking (SDN)[J]. IEEE access, 2019, 7: 124474–124490.

[68] CHICA J C C, IMBACHI J C, VEGA J F B. Security in SDN: a comprehensive survey[J]. Journal of network and

computer applications, 2020, 159: 1–23.

[69] MALDONADO J, RIFF M C, NEVEU B. A review of recent approaches on wrapper feature selection for intrusion detection[J]. Expert systems with applications, 2022, 198: 1–21.

作者简介：

齐小刚，教授，博士生导师，主要研究方向为系统建模与故障诊断。主持国家自然科学基金项目 1 项、陕西省自然科学基金项目 2 项，参与国家、省部级项目、中国–加拿大国际合作项目、国家重点实验室专项基金项目等 7 项。发表学术论文 100余篇。

单明媚，硕士研究生，主要研究方向为 SDN 网络的优化和故障诊断。

张皓然，硕士研究生，主要研究方向为多层复杂网络优化和故障定位。

2023 年中国粒计算与知识发现学术会议
Conference on China Granular Computing and Knowledge Discovery Society 2023(CGCKD2023)

由中国人工智能学会主办，中国人工智能学会粒计算与知识发现专委会协办，国际粗糙集学会支持，厦门理工学院承办的 2023 年中国粒计算与知识发现学术会议（第二十三届中国 Rough 集与软计算学术会议、第十七届中国粒计算学术会议、第十一届三支决策学术会议）将于 2023 年 7 月 17—20 日在“海上花园城市——厦门”召开。大会包括特邀报告、论文报告等多个会议主题。同期将举办“低质量多模态数据机器学习讲习班”、“可解释性人工智能讲习班”、“大规模学习模型与粒计算青年学者论坛”，敬请关注。

联合会议由三部分组成：CRSSC 始于 2001 年，主要研讨 Z. Pawlak 教授所提出 Rough 集理论；CGrC 于 2007 年加入，主要研讨 L.A. Zadeh 和 T.Y. Lin 提出的粒计算理论以及张钹院士和张铃教授提出的商空间理论；三支决策学术会议于 2012 年加入，主要研讨姚一豫教授提出的三支决策理论。

历届会议均邀请著名学者作主题报告，包括张钹院士（清华大学）、李德毅院士（军事科学院）、戴琼海院士（清华大学）、柴天佑院士（东北大学）、赵春江院士（中国工程院）、姚一豫教授（加拿大 Regina 大学）、Hamido Fujita 教授（日本岩手县立大学）、Thierry Denoeux 教授（法国贡比涅技术大学）、Ning Zhong 教授（日本前桥工业大学）、T.Y. Lin 教授（美国 San Jose 州立大学）、吴信东教授（美国 Louisiana 大学）、姚静涛教授（加拿大 Regina 大学）等数十位知名学者。

中国粒计算与知识发现学术会议，每年举办一次，是我国粒计算领域的学术盛会，已成为国内人工智能领域最主要的学术活动之一，为从事粒计算与知识发现的学者、研究生以及工程技术人员提供了一个交流平台，让大家了解最前沿的学术动态和分享最新研究成果，以提高国内相关领域的研究水平。

会议网站：https://cs.xmut.edu.cn/cgckd2023/index.html