软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(9):2709−2722 [doi: 10.13328/j.cnki.jos.005940]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

SDN 数据平面软件一致性测试用例生成方法*

张雯雯[1,2], 许天予[3], 章 玥[1,2], 郑孝遥[4]

[1](上海市高可信计算重点实验室(华东师范大学),上海 200062)

[2](华东师范大学 教育部软硬件协同设计技术与应用工程研究中心,上海 200062)

[3](上海移动通信有限公司,上海 200070)

[4](安徽师范大学 计算机与信息学院,安徽 芜湖 241002)

通讯作者: 章玥, E-mail: yzhang@sei.ecnu.edu.cn

摘 要: SDN(software-defined network)旨在解决架构复杂且分散的传统网络出现的问题,使网络具有更强的灵活性.P4 编程语言的特征在于用户可以直接根据自己对处理数据包的需求定义 P4 程序,然后经过编译过程,生成适配文件将用户需求配置到网络设备.面向 P4 编程语言的 SDN 数据平面一致性测试,是将一致性测试用例发送给 P4 网络设备,评估实际输出和预期输出的一致程度.一致性测试用例是执行一致性测试的载体,而传统的人工构造测试用例是一项繁琐耗时费力的工作.重点分析了面向 P4 编程语言的 SDN 数据平面软件一致性测试用例设计原则和生成方法,给出了一致性测试用例覆盖标准,设计了命令信息实体结构和测试用例实体结构,以装载 P4 程序的 simple_switch 虚拟交换机为测试对象,说明一致性测试用例生成过程,实现了一个用于 P4 网络设备一致性测试的测试用例自动生成工具,并验证了该工具自动生成测试用例的有效性,实现了一致性测试用例构造过程简易性.

关键词: 一致性测试;测试用例生成;测试覆盖;SDN(software-defined networks);P4(programming protocol-independent packet processors)中间节点编程语言

中图法分类号: TP311

中文引用格式: 张雯雯,许天予,章玥,郑孝遥.SDN 数据平面软件一致性测试用例生成方法.软件学报,2020,31(9):2709−2722. http://www.jos.org.cn/1000-9825/5940.htm

英文引用格式: Zhang WW, Xu TY, Zhang Y, Zheng XY. SDN data plane software conformance test case generation method. Ruan Jian Xue Bao/Journal of Software, 2020,31(9):2709−2722 (in Chinese). http://www.jos.org.cn/1000-9825/5940.htm

SDN Data Plane Software Conformance Test Case Generation Method

ZHANG Wen-Wen[1,2], XU Tian-Yu[3], ZHANG Yue[1,2], ZHENG Xiao-Yao[4]

[1](Shanghai Key Laboratory for Trustworthy Computing (East China Normal University), Shanghai 200062, China)

[2](MOE Engineering Research Center of Hardware & Software Co-design Technology and Application, East China Normal University, Shanghai 200062, China)

[3](Shanghai Mobile Communications Co. Ltd., Shanghai 200070, China)

[4](School of Computer and Information, Anhui Normal University, Wuhu 241002, China)

* 基金项目: 国家重点研发计划(2018YFB2101301); 上海市高可信计算重点实验室开放课题(07dz22304201607); 国家自然科学基金(61772034); 安徽省自然科学基金(1808085MF172); 高校优秀青年人才支持计划(gxyqZD2019010)

Foundation item: National Key Research and Development Program of China (2018YFB2101301); Open Project Fund of Shanghai Key Lab for Trustworthy Computing (07dz22304201607); National Nature Science Foundation of China (61772034); Natural Science Foundation of Anhui Province (1808085MF172); Youth Elite Support Plan in Universities of Anhui Province (gxyqZD2019010)

本文由“智能嵌入式系统”专题特约编辑王泉教授、吴中海教授、陈仪香教授、苗启广教授推荐.

收稿时间: 2019-07-01; 修改时间: 2019-08-18; 采用时间: 2019-11-02; jos 在线出版时间: 2020-01-13

CNKI 网络优先出版: 2020-01-14 11:26:58, http://kns.cnki.net/kcms/detail/11.2560.TP.20200114.1126.022.html

Abstract: SDN (software-defined network) is designed to solve the problems of traditional networks with complex and scattered architecture, making the network more flexible.The characteristic of the P4 programming language is that users can directly define P4 programs according to their needs for processing data packets,and then compile and configure user requirements to network equipment through the adaptation file.The SDN data plane conformance test for the P4 programming language is to send the consistency test cases to the P4 network equipment to evaluate the consistency of the actual output and the expected output.Conformance test cases are the carriers for performing conformance tests,and the traditional manual construction of test cases is a tedious and time-consuming work.This article focuses on the design principles and generation method of SDN data plane software conformance test cases for P4 programming language, gives the conformance test case coverage standards,designs the command information entity structure and the test case entity structure, and uses the simple_switch virtual switch loaded with the P4 program as the test object to illustrate the conformance test case generation process.This paper also implements an automatic test case generation tool for P4 network device conformance testing, and verifies the effectiveness of the tool to automatically generate test cases, and realizes the simplicity of the conformance test case construction process

Key words: conformance test; test case generation; test coverage; SDN (software-defined networks); P4 (programming protocol-independent packet processors) intermediate node programming language

在网络需求不断增加、网络灵活性要求不断提高的背景下,SDN(software-defined network)解决了架构复杂且分散的传统网络出现的问题,使网络具有更强的灵活性.SDN 使网络控制平面和转发平面分离,使网络具有可编程性.斯坦福大学的 Nick 教授及其团队研究出 P4 高级编程语言,P4 是一种协议独立的针对数据转发平面编程的语言,旨在允许我们定义交换机将识别(或“解析”)的表头、如何匹配每个表头以及我们希望交换机在每个表头上执行的操作.P4 语言的特征在于,用户可以直接根据自己对处理数据包的需求定义 P4 程序.

OpenFlow 协议是比较流行的 SDN 南向协议,目前,针对 OpenFlow 协议的一致性测试工作展开了很多,但是面向 P4 编程语言的一致性测试研究甚少,缺乏成熟的一致性测试方案.P4 编程语言具有巨大的灵活性且传统的人工构造测试用例是一项繁琐耗时费力的工作,面向 P4 编程语言研究 SDN 数据平面软件一致性测试用例生成方法具有重要意义.

面向 P4 编程语言的一致性测试,本质上是将测试用例发送给 P4 网络设备,通过比较实际输出和预期输出的一致程度,保证了 P4 网络设备在错综复杂的网络环境中的表现与预期表现一致性.面向 P4 编程语言的 SDN 数据平面软件一致性测试,目的在于保障控制器与网络设备之间的互操作性.

装载了 P4 程序的网络设备基本的工作机制为:接收数据包;解析器以有限状态机的形式解析数据包包头,从初始状态开始,每一种状态解析一种协议,从低层协议解析匹配首部定义的协议进而转移到高层协议字段,最终转移至目标状态;执行 Match-Action 动作,数据包头的目的地址与表的匹配字段进行匹配,当数据包中的字段和匹配字段根据给定的匹配类型匹配成功时,就会执行对应的动作,如果匹配不成功,则会执行默认动作(default_action).

本文重点描述面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成方法进行综述.具体来说,通过研究 SDN 架构、P4 程序的工作原理,我们发现表和动作是实现转发功能的核心部分:动作可以处理数据,定义转发等动作;而表是匹配动作表(match-action tables)的简称,表会定义匹配字段(key)、匹配方式以及匹配成功后执行的对应动作.本文生成测试用例的核心思想是:通过划分测试用例类型,获取 P4 目标网络设备控制命令集,依据 P4 程序表和动作具体信息构造测试用例.本文设计并实现了自动生成批量测试用例的工具,并以装载 P4 程序的 simple_switch 虚拟交换机为测试对象说明测试用例的生成过程.

1 相关工作

本节介绍 SDN 体系架构、SDN 领域的 P4 编程语言介绍及相关研究和面向 OpenFlow 协议以及面向 P4 编程语言的一致性测试的相关研究.

1.1 SDN体系架构

SDN 是由美国斯坦福大学 Clean State 课题研究组提出的,是网络虚拟化的一种实现方式[1]. 2007 年,Casado

博士基于 SANE 实现了采用集中式控制器来管理网络的 Ethane 项目[2].SANE/Ethane 项目正是 SDN 和 OpenFlow 的前身.SDN 的核心在于:利用分层的思想,实现数据平面与控制平面相分离. SDN 的优势在于增强了传统网络灵活性,解决了传统网络可控性差的问题.

如图 1 所示,SDN 体系架构[3]分为 3 层:最底层为架构层(也被称为数据层、转发层),主要功能是实现数据转发;中间层为控制层,实现 SDN 的主要功能,集中存储网络设备的控制逻辑,其控制逻辑控制数据平面数据包转发;最上层为应用层,包含错综复杂的业务应用.控制层的控制逻辑通过 SDN 北向接口(northbound interface,简称 NBI)与应用层进行通信,使用户按实际需求定制开发.控制层和数据层通过 SDN 南向接口(southbound interface,简称 SBI)进行通信,控制数据层数据转发.

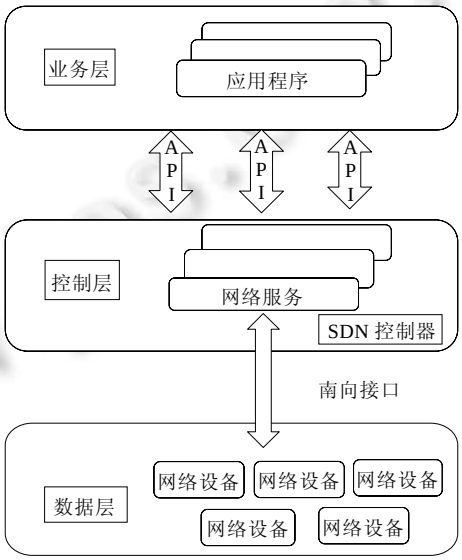


Fig.1 SDN network architecture

图 1 SDN 网络架构

交换机等路由设备是数据层的基本网络元素,不同的转发规则形成元素之间不同的数据通路连接.P4 编程语言具有设备无关性,用户无需关心底层元素的实现细节,仅仅通过简明易懂的编程就可以实现快捷开发迅速部署.因而,程序员可以通过可编程的控制逻辑控制整个 SDN 网络.具体来说,控制逻辑通过转换应用程序对数据层元素进行低级控制,同时提供相关信息,还可以根据策略协调对于有限网络资源的需求,从而真正实现了软件定义网络.

1.2 P4编程语言介绍

随着 SDN 的发展,OpenFlow 协议作为 SDN 的南向协议,一时间成为流行的标准用于控制平面和数据平面的交互.OpenFlow 假设交换机具有固定的、众所周知的行为,通常在交换机 ASIC 的数据表中描述.传统的高性能交换芯片支持一组固定的协议,因为它们直接在芯片中实现 IEEE 和 IETF 标准协议.用户无法更改其行为并添加新协议或测量和控制数据路径的新方法.

然而 OpenFlow 并不真正控制交换机行为,它为我们提供了一种填充一组众所周知的表的方法.SDN 通常与 OpenFlow 协议相关联,OpenFlow 协议是 SDN 的南向接口,目前已经发布了许多版本,其匹配域个数也不断增加,但是每次更新匹配域,对应交换机的协议栈和数据包处理逻辑就要更新,降低了 OpenFlow 协议的可扩展性、可编程性,因此对其协议的推广造成负面影响.

针对此问题,斯坦福大学的 Nick 教授及其团队研究出 P4 高级编程语言.2015 年 6 月在斯坦福大学[4]展开对 P4 的相关工作.P4 是一种协议独立的语言,提供了编码结构,其范围从通用 CPU 到网络处理器、FPGA 和定

制 ASIC.P4 编程语言的特征在于用户可以直接根据自己对处理数据包的需求定义 P4 程序,然后经过编译过程生成适配文件将用户需求配置到网络设备.

P4 编程语言如今分为 $P4_{14}$ 和 $P4_{16}$ 两个版本,最新的版本 $P4_{16}$ 相比 $P4_{14}$ 在语法和语义方面的进行了许多修改.P4 程序有 5 个基本组成部分,分别是首部(header)、解析器(parser)、动作(actions)、表(tables)、控制模块(control blocks)[5].首部类型是由成员字段组成的有序列表,每一个成员字段都有其名称和字段长度,用户可以根据需要定义协议字段;解析器的工作是把数据包中的头和元数据解析出来,解析器是以有限状态机的形式实现;表和动作是实现数据包处理的核心部分,动作可以处理数据,定义转发等操作;表是匹配动作表(match-action tables)的简称,表会定义匹配字段(key)、匹配方式以及匹配成功后执行的动作;控制平面制定控制流,将管理数据包处理的规则插入到流水线中,当规则与数据包匹配时,将使用控制平面提供的参数作为规则的一部分来调用其操作.

P4 的 3 个主要目标是协议无关性、可重新配置性以及可移植性:协议无关性是指协议和设备无关性,设备不会指定协议;可重新配置性指协议重构性,程序员能够在部署后继续根据需求修改设备的行为;可移植性指 P4 程序不与特定的网络设备相关联.通过 P4 程序,我们可以定义交换机将识别(或“解析”)的表头、如何匹配每个表头以及我们希望交换机在每个表头上执行的操作.因此,通过 P4 语言,用户可以对网络设备的硅处理芯片(如交换机、路由器、网络接口等)进行管理.由于传统交换机只有一种转发数据包的方式,功能固定,所以传统网络功能呈“自下而上”构建;P4 可编程网络则呈现“自上而下”的网络功能.

现在,P4 语言已经成为 Linux 基金会投资组织的一部分,并作为开放网络基金会(open networking foundation,简称 ONF)[6]旗下的项目之一.P4 开发环境也在逐渐发展壮大起来,目前,P4 社区不仅拥有强大的编译器,还拥有了架构独立的 P4 交换机,其主要可编程目标包括可编程 NIC、高端交换芯片、软件交换机(如 OVS 和 eBPF)等.P4 成为开发新型数据平面设计的一种越来越受欢迎的选择,目前,阿里巴巴、AT&T、思科、Juniper、Netronome、Vmare、Xilinx 和中兴通讯等大厂也加入了 P4 联盟,其应用包括 NetFlow,Paxos 等方面的研究,且范围范围还在扩展,但有关 P4 程序的测试工作有待开展.

在 Sigcomm 2017 年会上,来自清华大学的余舟等人进行了 P4 模块化编程[7]以及 P4 驱动的网络模拟器[8]的演示,其 P4 模块化编程体系结构 ClickP4 简化了 P4 程序的开发过程,ClickP4 的网络策略具有可靠性.P4 首次应用于模拟网络的研究成果是网络模拟器 NS4,NS4 简化了模拟器内部模型开发过程,构建了虚拟的 P4 网络设备及环境系统,是一款拓展性强的 P4 行为模型验证工具.2019 年 11 月,ONF 开源了包括 NG-SDN 接口和模型的交换机操作系统 Stratum[9],以提供全生命周期的控制、配置和接口操作管理.Stratum 对外呈现 P4、P4Runtime、gNMI/OpenConfig 和 gNOI 等接口,基于上述接口的 Stratum 一致性测试框架正在研究制定中.

P4 和 OpenFlow 协议的主要区别在于:OpenFlow 协议功能固定、协议更新时间长,而 P4 编程语言可以告诉交换机如何识别、处理数据包,让网络架构师或程序员可以控制交换机、指定转发行为,让填充定义表成为可能.某种意义上来说,P4 编程语言可以真正意义上的完全实现软件定义网络.

1.3 一致性测试相关研究

全球已经存在许多独立的针对 OpenFlow 协议一致性测试的实验室,如北京互联网研究院(BII)、韩国电子通信研究院(ETRI),这些实验室已经提出 OpenFlow 协议一致性测试测试方案并获得了 SDN 标准化组织 ONF 的认可.2013 年,ONF 宣布启动 OpenFlow 一致性测试项目,目的在于认证网络供应商在交换机、路由器和网络软件中 SDN 协议的实施.ONF 提供了完全一致性、三层一致性以及二层一致性这 3 种针对 OpenFlow 的一致性测试规范,网络供应商可以根据实际要求选择测试规范对数据转发设备进行测试.目前,ONF 已经认证并投入使用的工具包括 OFCheck,OFsuite,OFTest 等.

P4 编程语言的可编程性和灵活性是提高网络性能的关键,符合错综复杂的网络需求对未来网络发展方向的要求.通信协议按照标准落地,是协议设计的初衷和目的,是保证设备之间按照协议设定进行通信的关键.检测评估协议是否落地,一致性测试必不可少.针对 P4 编程语言的测试工具有 p4pktgen,p4app 以及 ptf 数据平面测试框架.

- p4pktgen[10]是一个可以为 P4 程序自动化生成测试用例的开源工具. p4pktgen 不直接依赖 P4 程序,而是解析 P4 程序经编译后的 JSON 格式文件,并为给定程序中的路径生成数据包和表配置,用户可以选择生成所有路径的测试用例,也可以生成优先考虑分支覆盖范围的测试用例.随后, p4pktgen 配置 BMv2 交换机、发送测试包并验证数据包的转发是否和预期的路径一致.p4pktgen 支持 $P4_{14}$ 和 $P4_{16}$ 两个版本的 P4 程序,其功能上尚且有一些没有修复的 bug,生成的数据包不能随 P4 程序更新而灵活改变,其针对性有待改进;
- p4app 是一个可以构建、运行、调试和测试 P4 程序的工具,其工作过程具体包括:p4app 运行一个以.p4app 为后缀的程序包,该程序包可以封装 P4 程序、JSON 格式文件等,还可以封装 P4 编译器、抓包、发包等工具.在测试过程中,stf 测试框架直接编译 p4app 程序包将生成一个.stf 文件,发送.stf 文件中的报文并验证 BMv2 交换机发出的报文与预期是否一致;
- ptf 是一个用 python 实现的数据平面测试框架,主要功能来自 OFTest 测试框架.ptf 专注于数据平面,并且不依赖于 OpenFlow,且添加了 Filters、Ternary matching 和 Test timeout 等功能.

上述常见的测试工具已经部分应用于 P4 程序的功能测试.2019 年 11 月,ONF 先后发布了交换机操作系统 Stratum 和目前正在开发中的针对 Stratum 交换机测试的 Test Vector Framework,该框架正在致力于开发一系列供应商无关的测试来检验目标交换机与 Stratum 的一致性.本文着重针对面向 P4 编程语言的一致性测试用例生成方法进行相关研究.

2 一致性测试用例覆盖标准

软件测试的覆盖指标包括语句覆盖、判定覆盖、路径覆盖等,而面向 P4 程序的一致性测试指标定义尚未明确.通过分析 P4 程序,我们发现首部(header)、解析器(parser)、动作(actions)、表(tables)、控制模块(control blocks)这 5 个 P4 编程语言的基本组成部分的工作都是与匹配动作表(match-action tables)相关联,而 P4 网络设备也是通过匹配流水线(pipeline)中的表来执行相应处理数据包的操作.因此,我们可以通过 P4 网络设备的控制命令来进行面向 P4 编程语言的 SDN 数据平面软件一致性测试,则相应的测试用例为控制命令.

由于每条控制命令所含字段不同,每条字段的合法类型也不同,经过分析,本文将面向 P4 编程语言的 SDN 数据平面软件一致性测试用例覆盖标准确立为:生成足够的测试用例,使得所有针对 match-action tables 操作的控制命令都被执行,其中每条命令的每个字段都要覆盖合法和不合法两种测试结果.

3 面向 P4 编程语言的一致性测试用例自动生成方法设计

本节设计了一致性测试用例自动化生成方法,详细介绍了 P4 程序经过编译、解析、测试用例生成的整个自动化过程.

如图 2 所示,本节设计的测试用例自动生成方法基本流程为:P4 程序经过编译过程,生成解析过程所需要的 P4 信息文件(.p4info 文件)和 P4 数据平面描述文件;所述 P4 数据平面描述文件包括 JSON 格式的网络设备配置文件;解析过程包括获取目标网络设备控制命令集,依据每条控制命令格式,从所述 P4 程序编译单元生成的 P4 信息文件中解析出 TABLE_NAME,ACTION_NAME,MATCH_KEY,PARAMETERS 字段信息,最终生成控制信息实体集合;测试用例生成过程划分了测试用例类型,将所述解析过程生成的控制信息实体集按照测试用例类型组织,最终生成面向 P4 编程语言 SDN 数据平面软件一致性测试用例集.

3.1 编译过程

p4c 编译器是为 P4 编译器设计的编译器基础设施,同时支持 $P4_{14}$ 以及 $P4_{16}$ 两个版本的 P4 语言.如图 3 所示: p4c 提供了标准的前端和中间件,通过在编译时添加选项,可以和特定目标的后端相结合使用.p4c 编译器支持 EBPF Backend,BMv2 Backend 和 P4test Backend.其中 EBPF 能够生成可以扩展 Berkeley 数据包过滤程序的 C 代码, BMv2 能够生成用于简单的网络交换仿真的 JSON 格式文件,P4test 是一个测试用的后端编译器.最新的

后端编译器详情请见 https://p4.org/.

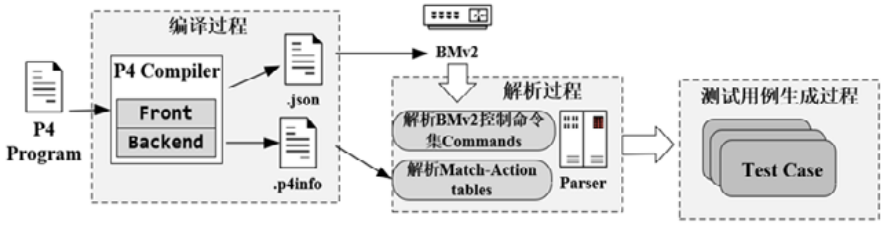


Fig.2 SDN data plane software conformance testcase generation tool data flow diagram for P4 programming language

图 2 面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成工具数据流图

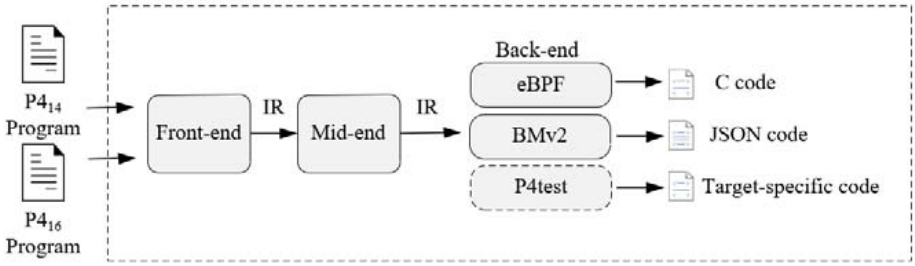


Fig.3 P4 program compilation process diagram

图 3 P4 程序编译过程图

图 3 所示为 P4 程序的编译过程, 本文使用的是 p4c-bm2-ss 来编译 P4 程序,p4c-bm2-ss 是 p4c 项目编译完后的产物之一,具体编译过程为: p4c 编译器首先将 $P4_{14}$ 和 $P4_{16}$ 编写的 P4 源程序转换成通用的中间表示 IR,然后经过前端编译器,生成为通用的后端中间表示 IR 和 P4 信息文件(.p4info 文件),中间表示 IR 经过 p4c-bm2-ss 编译生成一个 JSON 格式的数据文件,此文件专门用于将 P4 程序编译生成 BMv2 使用的数据平面描述文件.我们研究发现:由于 P4 官方提供了一个 Helper 类以帮助解析.p4info 文件,该文件格式更加便于处理,因此,.p4info 文件可用于 P4 表信息的解析过程.

3.2 解析过程

由于面向 P4 编程语言的 SDN 的实现,是通过定义匹配动作表(match-action tables)来部署 P4 网络设备.匹配动作表(match-action tables)是面向 P4 编程语言的 SDN 实现的关键载体,也是体现 P4 编程语言特性的重要部分.用户可以直接把自己对处理数据包的需求编写成 P4 程序来部署网络设备,因而网络架构师等人员可以通过控制网络设备、指定转发行为、填充定义表来实现 SDN.而确定装载 P4 程序的网络设备是否真正实现用户需求,则需要对目标网络设备进行一致性测试.一致性测试用例的生成,需要通过解析 P4 目标设备的控制命令集来获取每条命令所需要的表名、匹配字段、动作、动作参数信息.

- 控制命令信息类结构设计

我们以面向对象的思想来建模控制命令信息 Command 类,该类成员包括 CommandName,CommandFormat, TableName,ActionName,MatchField,ActionParameter1,ActionParameter2,如图 4 所示.

- 解析控制命令集

我们研究发现,不同的 P4 网络设备具有不同的控制命令集.生成用于 P4 网络设备的测试用例首先需要解析 P4 网络设备的控制命令集,进而生成包含控制命令头、控制命令格式以及控制命令字段的控制命令信息实体.本节设计了用以解析控制命令集的方法,生成控制命令信息实体列表,其工作流程如图 5 所示.

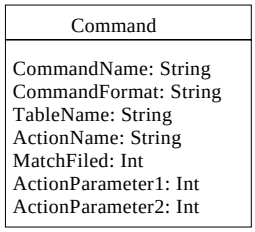


Fig.4 Control command information storage structure

图 4 控制命令信息存储结构

- 解析 P4 信息文件

.p4info 文件记录了 P4 程序匹配动作表(match-action tables).P4 官方提供了一个 Helper 类,以帮助解析.p4info 文件,实现了根据 id 查找表名、动作名,根据表名、动作名查询对应的 id、匹配字段、动作参数的功能.因此考虑从.p4info 文件中解析命令所需要的字段,其工作流程如图 6 所示.

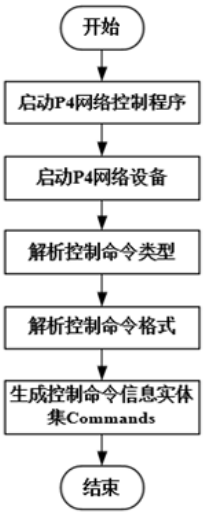


Fig.5 Control command parsing flowchart

图 5 控制命令解析流程图

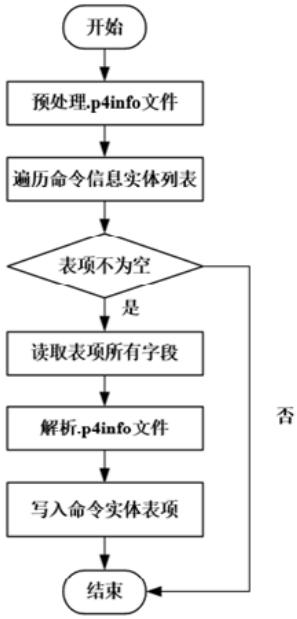


Fig.6 P4 program compilation process diagram

图 6 P4 信息文件解析流程图

其中,解析.p4info 文件具体过程为:解析命令信息实体命令名称,解析该命令信息实体所需字段信息,利用 Helper 类所提供的方法对.p4info 文件中所有匹配动作表(match-action tables)进行线性查找,将控制命令格式和每个匹配动作表(match-action tables)对应的 TABLE_NAME,ACTION_NAME,MATCH_KEY,PARAMETERS 字段信息写入命令信息实体.

(1) 解析表名

表名层级关系为 table→preamble→name.解析表名的流程如图 7 所示,每获取一个表名,会判断当前表是否为最后一张表:若有,则继续获取;若没有,则结束.随后,所有的表名都会从文件中被提取出来.

(2) 解析匹配字段

匹配字段的层级关系为 table→match_fields→(id,name,bitwidth).解析匹配字段的流程如图 8 所示,匹配字段

和表名一一对应,输入表名后判断该表是否存在:若存在,则能成功获取匹配字段;否则报错.在操作正确的情况下,表的对应匹配字段会从文件中被提取出来.

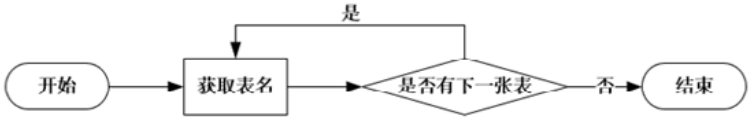


Fig.7 Parsing the table name process

图 7 解析表名流程

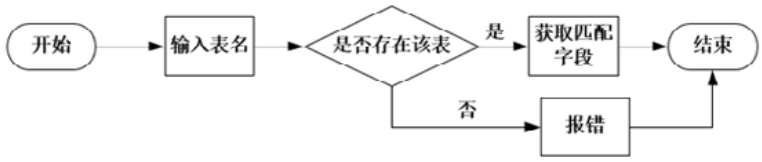


Fig.8 Procedure of extracting match field

图 8 解析匹配字段流程

(3) 解析动作

动作名的层级关系为 actions→preamble→name.解析动作的流程如图 9 所示,根据表名获取该表的所有动作 ID 后,根据 ID 获取动作名,判断该 ID 是否为最后一个 ID:若不是,则继续获取 ID.表对应的所有动作名都会被从文件解析出来.

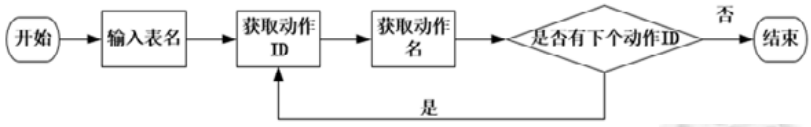


Fig.9 Procedure of extracting actions

图 9 解析动作流程

(4) 解析动作参数

动作参数的层级关系为 actions→params→(id,name,bitwidth),解析动作参数的流程如图 10 所示,根据动作名判断该动作是否有参数:若有,则获取所有参数的信息;若没有,则结束.

每解析得到一个命令信息实体,命令信息实体即被插入命令信息实体列表.经过以上解析过程,最终生成测试用例所需的命令信息实体集合.

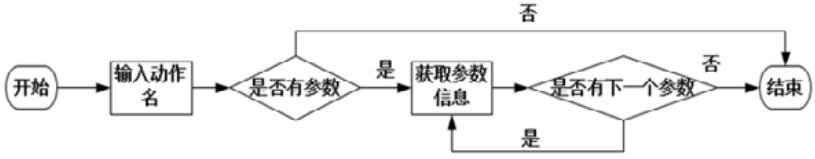


Fig.10 Procedure of extracting action parameters

图 10 解析动作参数流程

3.3 测试用例生成过程

前两节生成的命令信息实体集合为面向 P4 编程语言的 SDN 数据平面软件一致性测试用例的生成提供了数据基础,然而测试用例实体的生成需要一定的组织规范.本节不仅划分了测试用例类型,还分别设计了针对每种测试用例类型的测试用例构造方法.测试用例生成方法可分为生成单条测试用例和批量生成全部测试用例.

- 测试用例类型

本节依据测试用例结果类型,将最终生成的测试用例实体分为 5 种类型:命令合法、表名不合法、动作不合法、匹配域不合法、参数不合法.

- 构造测试用例

单条测试用例生成过程为:用户依次选择测试用例类型、控制命令类型,再根据命令类型所需字段选择或编辑表名、动作名、匹配域、参数信息,最后生成单条测试用例.

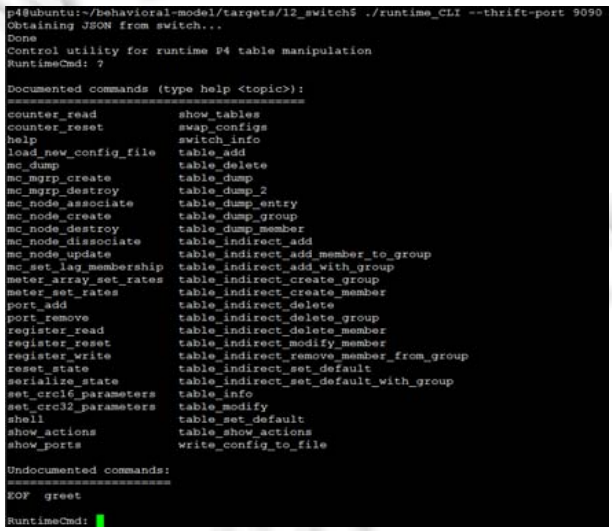
批量生成全部测试用例集的过程为:以测试用例类型为组织单位构造测试用例,遍历解析过程生成的命令信息实体列表.若测试用例类型为命令合法,则直接依据每个命令信息实体的命令格式和字段信息构造测试用例;若测试用例类型为表名不合法,则将命令信息实体中的 TABLE_NAME 字段修改为错误的 TABLE_NAME 字段,再依据命令格式与其他字段信息构造测试用例;同理,动作不合法、匹配域不合法、参数不合法类型的测试用例构造方法与命令不合法类似.

4 测试用例生成示例

本节以装载 P4 程序的 simple_switch 虚拟交换机为测试对象,说明一致性测试用例生成过程.

4.1 解析控制命令集

simple_switch 虚拟交换机的控制命令集如图 11 所示.



```
p4@ubuntu:~/behavioral-model/targets/l2_switch$ ./runtime_CLI --thrift-port 9090
Obtaining JSON from switch...
Done
Control utility for runtime P4 table manipulation
RuntimeCmd: ?

Documented commands (type help <topic>):
========================================
counter_read            show_tables
counter_reset           swap_configs
help                    switch_info
load_new_config_file    table_add
mc_dump                 table_delete
mc_mgrp_create          table_dump
mc_mgrp_destroy         table_dump_2
mc_node_associate       table_dump_entry
mc_node_create          table_dump_group
mc_node_destroy         table_dump_member
mc_node_dissociate      table_indirect_add
mc_node_update          table_indirect_add_member_to_group
mc_set_lag_membership   table_indirect_add_with_group
meter_array_set_rates   table_indirect_create_group
meter_set_rates         table_indirect_create_member
port_add                table_indirect_delete
port_remove             table_indirect_delete_group
register_read           table_indirect_delete_member
register_reset          table_indirect_modify_member
register_write          table_indirect_remove_member_from_group
reset_state             table_indirect_set_default
serialize_state         table_indirect_set_default_with_group
set_crc16_parameters    table_info
set_crc32_parameters    table_modify
shell                   table_set_default
show_actions            table_show_actions
show_ports              write_config_to_file

Undocumented commands:
======================
EOF  greet

RuntimeCmd:
```

Fig.11 simple_switch virtual switch control command set

图 11 simple_switch 虚拟交换机控制命令集

由于控制命令集包含许多功能类似的表项操作命令,并且不同的 P4 网络设备控制命令集并不完全相同,因此,本实验并没有对全部控制命令生成一致性测试用例,而是选取 simple_switch 虚拟机中具有代表性的表操作控制命令 table_add,table_delete,table_dump,table_modify 来生成一致性测试用例.

4.2 控制命令类型及格式

- table_add 表项增加命令

合法的增加表项的命令格式如下:

table_add⟨table name⟩⟨action name⟩⟨match fields⟩⇒⟨action parameters⟩[priority].

插入一个表项需要确定表名、动作名、匹配域、动作参数.以命令“table_add ipv4_lpm drop 0/32 0/32⇒0”

为测试用例进行说明.其中,ipv4_lpm 指定了该命令是向 ipv4_lpm 这张表插入表项,ipv4_forward 指定了表项匹配成功时执行的动作为 ipv4_forward,0*x*ffffffff/32 指定了匹配字段是 16 进制的 ffffffff,⇒是命令的格式需要所定义的箭头形状,⇒后的 0*x*0 0*x*0 指定了 ipv4_forward 动作的两个参数均为 0,命令中的动作需要几个参数,⇒符号后就应当给出几个参数.命令中/后的数字如 0/32 表示的是该数长度为 32 位.

- table_delete 表项删除命令

合法的删除表项的命令格式如下:

table_delete⟨table name⟩⟨entry handle⟩.

删除一个表项需要确定表项所在表的表名以及该表项在表中的编号,在这里被称为 entry handle.以命令“table_delete ipv4_lpm 0”为测试用例进行说明.其中,ipv4_lpm 指定了该命令删除的表项在表 ipv4_lpm 中,0 说明了该命令会删除表中编号为 0 的表项.

- table_dump 表项查询命令

合法的查询表项的命令格式如下:

table_dump⟨table name⟩.

查询表项需要确定需要查询的表项所在表的表名.以命令“table_dump ipv4_lpm”为测试用例进行说明.其中,ipv4_lpm 指定了该命令查询的是名为 ipv4_lpm 这张表中的所有表项.

- table_modify 表项修改命令

合法的修改表项的命令格式如下:

table_modify⟨table name⟩⟨action name⟩⟨entry handle⟩[action parameters].

修改表项需要确定表项所在表的表名、表项中的动作名、表项的编号、动作的参数.以命令“table_modify ipv4_lpm ipv4_forward 0 0*x*123456654321 1”为测试用例进行说明.其中,ipv4_lpm 指定了该命令修改的是表 ipv4_lpm 中的表项,ipv4_forward 说明了该命令修改的表项中的动作为 ipv4_forward,0 指定了该命令修改的是表 ipv4_lpm 中编号为 0 的表项,0*x*123456654321 1 指定了该命令是将 ipv4_forward 的两个参数改成 0*x*123456654321 1.命令要修改的表项中的动作有几个参数,就要给出几个参数.

4.3 生成测试用例集

如图 12 所示为测试用例生成的算法映射图,以测试用例类型命令合法、表名不合法、动作不合法、匹配域不合法、参数不合法为组织单位,以一个个命令信息实体为信息体,构造对应类型的 TABLE_NAME, ACTION_NAME,MATCH_KEY,PARAMETERS 字段,生成 table_add,table_delete,table_dump,table_modify 命令.

如图 13 所示,本节以增删改查命令类型的测试用例生成过程为例进行说明.在完成了对 P4 信息文件(.p4info 文件)的解析后,以测试用例类型为组织单位,构造符合测试用例类型的字段,依据命令信息实体的命令格式,生成测试用例.表项的增加和修改动作需要用到表名、匹配字段、动作名和动作参数,而表项的查询和删除命令需要用到表名.本节实现的 SDN 数据平面软件一致性测试用例生成工具不直接依赖于 P4 程序,而是直接依赖于编译后的 P4 信息文件(.p4info 文件).

- 参数默认值的设定原则

命令中,用户需要自行填写的参数有匹配字段(match_field)和动作参数(action_params),这两个参数不能由.p4info 确定,需要用户根据从.p4info 文件中提取出参数的位数自行填入.程序可以识别十进制数并转换成十六进制,但是交换机中记录的是十六进制的格式,目前的 IP 地址格式是 32 位二进制数.为了方便也分成了 4 段,每段 8 位.考虑到以十进制数表示 IP 地址不直观也非当今主流,所以需要根据参数的位数给出一个默认的参数,以确定一个参数的格式.匹配字段中的参数往往是目的地址或者是源地址之类的 IP 地址.匹配字段的长度必须是 8bits 的整数倍.参考参数见表 1.

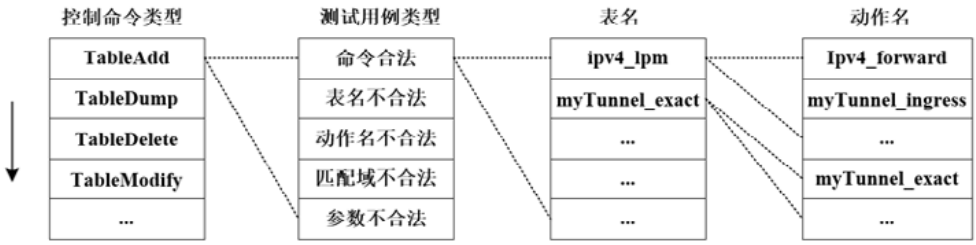


Fig.12 SDN data plane software conformance test case generation algorithm map for P4 programming language

图 12 面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成算法映射图

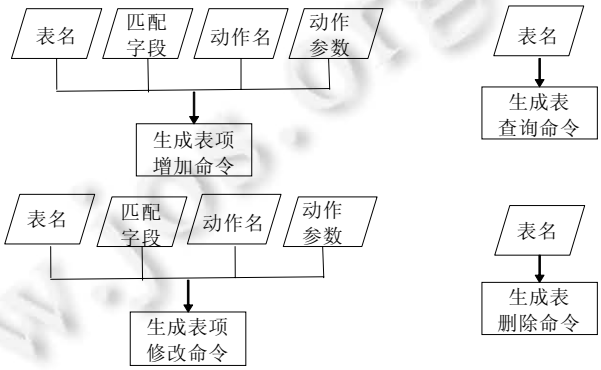


Fig.13 Procedure of generating commands

图 13 命令生成流程

Table 1 Reference parameters

表 1 参考参数

参数	位数	默认参数
match_field	8	00
match_field	16	00:00
match_field	24	00:00:00
match_field	32	00:00:00:00
match_field	40	00:00:00:00:00
match_field	48	00:00:00:00:00:00
action_params	9	0

- 单个测试用例示例

本节设计了测试用例实体结构,每个测试用例实体分为 4 部分:命令格式 CF(CommandFormat)、前提条件 PC(Precondition)、测试用例 TC(TestCase)、期望输出结果 EO(ExpectedOutput).设计该结构的目的在于规范测试用例实体,使测试用例实体信息更加清晰明了,方便面向 P4 编程语言的 SDN 数据平面软件一致性测试实际结果与预期结果比对过程.前提条件 PC 和期望输出结果 EO 为测试系统初期部署过程所作工作,本文不作赘述.

图 14 所示为一个测试用例实例,其中,

- 命令格式 CF 为“table_add ipv4_lpm myTunnel_ingress 0*xffffffff*/32⇒0*x*0 0*x*0”;
- 前提条件 PC 为“The entity to be added should not be duplicate”;
- 测试用例 TC 为“table_add ipv4_lpm myTunnel_ingress 0*xffffffff*/32⇒0*x*0 0*x*0”;
- 期望输出结果 EO 为“Entry has been added with handle *N*”.

TC 中,table_add 和 ipv4_lpm 代表向 ipv4_lmp 表插入表项,0*xffffffff*/32 代表匹配字段是 32 位的 0*xffffffff*,匹配成功则向表中插入参数为 0*x*0 和 0*x*0 的动作 myTunnel_ingress.

```
//测试用例实例:
CF_table_add⟨table name⟩⟨action name⟩⟨match fields⟩⇒⟨action parameters⟩[priority]
PC_The entity to be added should not be duplicate
TC_table_add ipv4_lpm myTunnel_ingress 0xffffffff/32⇒0x0 0x0
EO_Entry has been added with handle N
```

Fig.14 Test case example diagram

图 14 测试用例实例图

- 测试用例集示例(如图 15 所示)

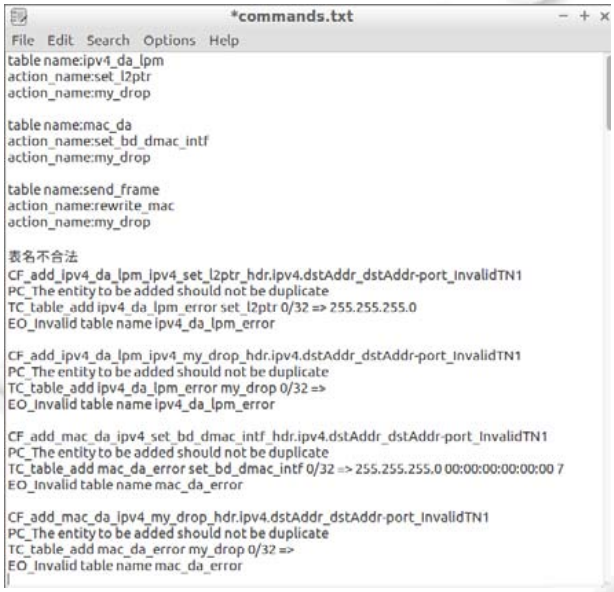


Fig.15 Test case set

图 15 测试用例集

4.4 测试用例自动生成工具测试

本节对我们已经实现的 SDN 数据平面软件一致性测试用例自动生成工具进行一致性测试用例生成测试实验,该实验选取了 3 个 P4 程序作为测试工具的输入,通过分析生成测试用例覆盖程度验证其有效性.

- demo1 含有 1 张表,名为 ipv4_lpm,表中有 3 个动作:ipv4_forward,drop,NoAction,因此,预期生成 3 条表项的插入命令、1 条表项的查询命令、1 条表项的删除命令、3 条表项的修改命令;
- demo2 含有 3 张表,分别是 ipv4_da_lpm,mac_da,send_frame.其中,表 ipv4_da_lpm 中有 2 个动作:set_l2ptr, my_drop,表 mac_da 中有 2 个动作:set_bd_dmac,my_drop,表 send_frame 中有 2 个动作:rewrite_mac, mydrop.预期生成 6 条表项的插入命令、3 条表项的查询命令、3 条表项的删除命令、6 条表项的修改命令;
- demo3 含有 2 张表,分别是 ipv4_lpm,myTunnel_exact.其中,ipv4_lpm 表中有 4 个动作:ipv4_forward, myTunnel_ingress,drop,NoAction,表 myTunnel_exact 中有 3 个动作:myTunnel_forward,myTunnel_egress,drop.预期生成 7 条表项的插入命令、2 条表项的查询命令、2 条表项的删除命令、7 条表项的修改命令.

如表 2 所示,3 个 P4 程序共 12 条被测控制命令通过一致性测试用例自动生成工具进行一致性测试用例生成测试.我们发现:生成测试用例的实际结果与预期结果完全一致,每条被测命令生成的测试用例集都覆盖了该控制命令字段所有可能出现的情况,实现了全覆盖.则以上实验验证了面向 P4 编程语言的 SDN 数据平面软件一致性测试用例自动生成工具具有有效性.

Table 2 Test case automatic generation tool test data table

表 2 测试用例自动生成工具测试数据表

测试用例		预期结果					实际结果				
		命令合法	表名不合法	动作不合法	匹配域不合法	参数不合法	命令合法	表名不合法	动作不合法	匹配域不合法	参数不合法
demo1	table_add	3	3	3	3	3	3	3	3	3	3
	table_delete	1	1	1	1	1	1	1	1	1	1
	table_modify	3	3	3	3	3	3	3	3	3	3
	table_dump	1	1	1	1	1	1	1	1	1	1
demo2	table_add	6	6	6	6	6	6	6	6	6	6
	table_delete	3	3	3	3	3	3	3	3	3	3
	table_modify	6	6	6	6	6	6	6	6	6	6
	table_dump	3	3	3	3	3	3	3	3	3	3
demo3	table_add	7	7	7	7	7	7	7	7	7	7
	table_delete	2	2	2	2	2	2	2	2	2	2
	table_modify	7	7	7	7	7	7	7	7	7	7
	table_dump	2	2	2	2	2	2	2	2	2	2

4.5 测试工具界面

测试用例生成工具的界面如图 18 所示,在用户选定.p4info 文件的后,点击 Write In 按钮,则生成.p4info 文件对应的全部测试用例.在选定.p4info 文件后,依次选择 table、actions 下拉框,随后读取 match_field 和 action_params,点击 Generate entry 按钮,即在下方空白处显示表项的信息.

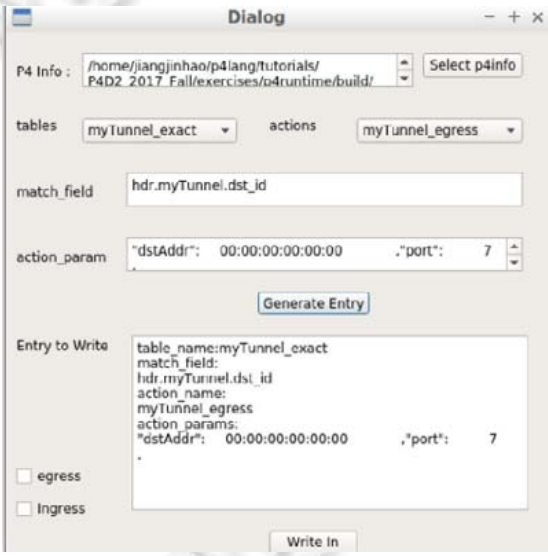


Fig.18 Test case generation tool interface

图 18 测试用例生成工具界面

5 总 结

面向 P4 编程语言的一致性测试本质上是在将测试用例发送给 P4 网络设备,通过比较实际输出和预期输出的一致程度,保证了 P4 网络设备在错综复杂的网络环境中的表现与预期表现一致性.测试用例是执行一致性测试的载体,传统的人工构造测试用例是一项繁琐耗时费力的工作.本文介绍了 SDN 架构、P4 编程语言特性,调研了 SDN 领域一致性测试现有工作情况,设计了用于面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成方法,实现了面向 P4 编程语言的 SDN 数据平面软件一致性测试用例自动生成工具.实验结果表明:该测试工具能够由用户自己选择或编辑生成单条测试用例,也可以按照预期生成任意 P4 程序的测试用例集.

除此之外,在实际应用中,本文工作还有进一步提升空间.

本文是面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成研究工作,在本文工作的基础上,可以针对 P4 编程语言的 SDN 数据平面软件进行真实网络环境下的一致性测试工作研究;

面向 P4 编程语言的 SDN 数据平面软件一致性测试过程需要包括测试环境搭建、测试系统初期部署、一致性测试用例生成、一致性测试等阶段.后续工作可以设计面向 P4 编程语言的 SDN 数据平面软件一致性测试一体化过程,可以基于 gRPC 框架对真实网络环境下的测试用例的发送、测试服务调用和测试结果接收过程进行研究.

交换机操作系统 Stratum 支持 P4 API 、P4Runtime API, 可以基于本文提出的面向 P4 编程语言的 SDN 数据平面软件一致性测试用例生成方法进行 Stratum 交换机一致性测试的研究.

References:

[1] Zhang CK, Cui Y, Tang W, Wu JP. Research progress of software defined network (SDN). Ruan Jian Xue Bao/Journal of Software, 2015,26(1):62−81 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4701.htm. [doi: 10.13328/j.cnki.jos.004701]

[2] Casado M, McKeown N, Shenker S. From ethane to SDN and beyond. ACM SIGCOMM Computer Communication Review. 2019, 49(5):92–95.

[3] Shenker S. The future of networking, and the past of protocols. In: Proc. of the Open Networking Summit. 2011.

[4] 1st P4 Workshop. P4 Official Website. Retrieved 1 August 2019.

[5] Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G, Walker D. P4: Programming protocol-independent packet processors. Computer Communication Review, 2014,44(3):87−95.

[6] OpenNetworking foundation. https://www.opennetworking.org/

[7] Zhou Y, Bi J. ClickP4: Towards modular programming of P4. In: Proc. of the SIGCOMM Posters and Demos. Los Angeles, 2017.

[8] Fan CZ, Bi J, Zhou Y. NS4: A P4-driven network simulator. In: Proc. of the SIGCOMM Posters and Demos. Los Angeles, 2017.

[9] ONF Connect (NG-SDN Track).https://www.opennetworking.org/wp-content/uploads/2019/09/2.00pm-Brian-OConnor-Stratum.pdf

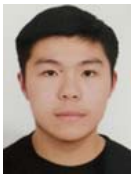
[10] p4pktgen: Automated test case generation for P4 programs. https://p4.org/assets/P4WS_2018/12_p4pktgen_Abstract.pdf.

附中文参考文献:

[1] 张朝昆,崔勇.软件定义网络(SDN)研究进展.软件学报,2015,26(1):62−81. http://www.jos.org.cn/1000-9825/4701.htm [doi: 10.13328/j.cnki.jos.004701]

张雯雯(1995－),女,学士,主要研究领域为软件定义网络.

章玥(1981－),女,博士,副教授,CCF 专业会员,主要研究领域为软件定义网络,网络管理,通信软件.



许天予(1995－),男,学士,主要研究领域为软件定义网络.

郑孝遥(1981－),男,博士,副教授,CCF 专业会员,主要研究领域为信息安全,个性化推荐.