

面向未来网络的白盒交换机体系综述

张成林¹, 宋玲玲²

(1.东南大学 网络空间安全学院, 江苏 南京 211189; 2.网络通信与安全紫金山实验室, 江苏 南京 211111)

摘要: 面向 2030 的未来网络需要承载自动驾驶、虚拟现实等新型复杂业务, 云技术发展倒逼现有网络云化升级, 要求网络设备具备可编程性并能够开放化。而传统封闭固定的网络设备难以满足需求, 亟需开放可编程的白盒交换机。通过概述白盒交换机体系的发展历程和主要架构, 围绕设备开放化和网络可编程两大原则, 梳理和比较白盒交换机在体系架构方面的研究, 分析白盒交换机当前面临的技术和产业挑战, 并指出白盒交换机未来可能的发展方向, 以期对白盒交换机以及未来网络的发展做出贡献。

关键词: 白盒交换机; 设备开放化; 网络可编程; 未来网络

中图分类号: TP393

文献标识码: A

DOI: 10.19358/j.issn.2096-5133.2022.03.001

引用格式: 张成林, 宋玲玲. 面向未来网络的白盒交换机体系综述[J]. 信息技术与网络安全, 2022, 41(3): 2-8.

A review on architecture of white box switches for future networks

Zhang Chenglin¹, Song Lingling²

(1.School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China;

2.Purple Mountain Laboratories, Nanjing 211111, China)

Abstract: According to the forecast, future networks for 2030 need to carry new complex services such as autonomous driving and virtual reality, while the development of cloud technology forces the current network to be upgraded in the cloud, which requires network devices with programmability and openness. However, the traditional closed and fixed network devices can no longer meet the demand, so there is an urgent need for open and programmable white box switches. This paper summarizes the development history and main structures of white box switch architecture, combs and compares the research conducted on the architecture of white box switches around two major principles, analyzes the current technical and industrial challenges faced by white box switches, and points out the possible future directions of white box switches, excepting to contribute to the development of white box switches and future networks.

Key words: white box switch; device opening; network programming; future network

0 引言

根据预测, 至 2030 年, 未来网络将需要承载自动驾驶、虚拟现实、全息通信、远程医疗、智能制造、智慧城市等新型复杂业务^[1]。这些业务要求网络满足不同用户的差异化服务、数据传输的高精度通信以及服务器的高带宽响应, 网络设备要满足可编程、确定性、高性能等需求。同时, 随着近十年来用户规模和流量的爆炸式增长, 云计算、云原生等云化技术获得充分发展, 使得数据中心应用的服务器、存储器等资源逐渐虚拟化, 倒逼应用通信相关的网络设备也进行虚拟化, 网络需要云化升级, 网络设备需要开放化。然而, 当前网络依然采用封闭

的、供应商垄断式的专用交换机, 这种交换机采用一体化的设计: 定制的专用集成电路 (Application-Specific Integrated Circuit, ASIC) 芯片、特定于 ASIC 芯片的软件开发工具包 (Software Development Kit, SDK)、特定于设备的闭源操作系统, 交换机的各软件和硬件组件紧密地捆绑在一起, 只能提供指定的网络服务与功能。显然, 这些设备既无法满足未来网络灵活业务的需求, 又难以适应当前网络云化的要求, 严重阻碍着未来网络的创新发展。

作为未来网络开放创新的关键设备, 白盒交换机采用支持设备开放化以及网络可编程的开放架构, 契合了新型业务和网络发展的亟需。白盒交换

机突破传统设备的软硬件一体化设计,解耦底层网络硬件与上层操作软件,同时控制平面支持网络协议的软件可编程,数据平面支持转发芯片的硬件可编程,从而提升了网络设备的开放性以及可编程性,对未来网络建设和研究具有重要意义。

本文概述了白盒交换机的体系,介绍发展历程和主要架构,梳理和比较白盒交换机在体系架构方面的研究,分析白盒交换机现存的挑战,并指出未来可能的发展方向。

1 白盒交换机体系概述

相对于封闭式黑盒设计的传统交换机,白盒交换机采用开放式白盒设计模式,设备硬件由原始设计制造商(Original Design Manufacturer, ODM)提供,适配的操作系统软件可以自由开放选择。此外,白盒交换机由于硬件采用 ODM 生产方式,并没有具体的品牌,又被称为白牌交换机。

1.1 发展历程

如图 1 所示,白盒交换机在过去三十年尤其是近十年间得到了快速发展。首先, Linux 于 1994 年开源发布,允许用户按需修改,定制网络协议和功能。由此,1998 年 IBM、Compaq 等公司陆续开始商用 Linux, Linux 技术生态开始快速发展。2008 年, Linux 开始与高性能 ASIC 芯片结合而构建数据中心,以应对网络流量的爆炸式增长。同年,为应对网络体系僵化、管控困难,开放网络基金会(Open Networking Foundation, ONF)发起人 Nick McKeown 团队提出 OpenFlow^[2],支持以软件定义的新方式控制交换机。2010 年,互联网工程任务组(Internet Engineering Task Force, IETF)也制定了自己的软件定义规范“转发和控制元素分离”(Forwarding and Control Element Separation, ForCES)^[3],不过由于缺乏供应商支持而失败。2014 年,开放计算项目(Open Compute Project, OCP)发布开放网络安装环境(Open Network Install Environment, ONIE)^[4]规范,支持交换机硬件和系统的解耦,同时联合 Edgcore 公司发布首款白盒交换机,由此 OCP 开始推动白盒交换机的发展。同年, Nick McKeown 团队针对 OpenFlow 的协议相关性缺陷设计可编程语言 P4 (Programming Protocol-independent Packet Processors)^[5], Linux 基金会开始将 switchdev (Ethernet switch device driver model)^[6]引入内核进行网络转发的硬件卸载。2015 年, OCP 陆续接纳开放网络操作系统(Open Network Linux, ONL)^[7]、交换机

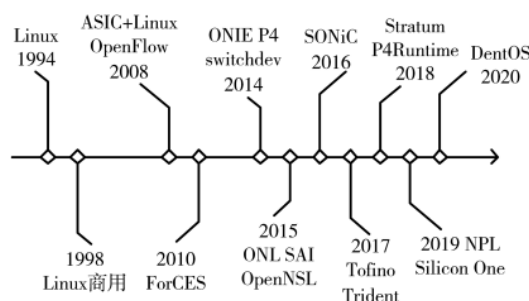


图 1 白盒交换机发展历程图

抽象接口(Switch Abstraction Interface, SAI)^[8]、开放网络交换机库(Open Network Switch Library, OpenNSL)^[9],增强白盒交换机的模块化、标准化和开放化。之后,2016 年 OCP 接纳的 SONiC(Software for Open Networking in the Cloud)^[10]、2018 年 ONF 推出的 Stratum^[11]、2020 年 Linux 推出的 DentOS^[12]等操作系统逐步完善了交换机的白盒生态。数据平面可编程方面,2017 年 Barefoot 公司首先设计出支持 P4 的可编程 ASIC 芯片 Tofino^[13],2018 年 Tofino 的芯片接口标准 P4Runtime^[14]随即出现。随后,传统交换机芯片巨头博通以及思科也分别设计 Trident^[15]和 Silicon One^[16]与之竞争,并于 2019 年推出 Trident 芯片的网络编程语言(Network Programming Language, NPL)^[17]。

1.2 主要架构

围绕设备开放化和网络可编程两大体系原则,白盒交换机开放化解耦底层网络硬件与上层操作软件,同时向上层应用开放底层芯片的可编程能力。大体上白盒交换机的体系架构具有裸机硬件、芯片接口、引导系统、操作系统和应用几部分,如图 2 所示。

1.2.1 裸机硬件

白盒交换机的裸机硬件主要包括 CPU、ASIC 以及电源、风扇、存储和相关接口模块,其中核心是 ASIC 交换芯片,负责数据包的高速转发。ASIC 芯片可分为固定功能芯片和可编程芯片两大类。固定功能芯片一般只支持基本的网络协议算法以及一些定制功能,对于白盒交换机而言,其侧重点在于实现控制平面可编程的芯片接口。可编程芯片则具有通用的数据处理流水线,可以利用芯片接口灵活高效地设计各种网络协议算法,成为白盒交换机数据平面可编程的关键。

1.2.2 芯片接口

白盒交换机的芯片接口抽象底层芯片的功能

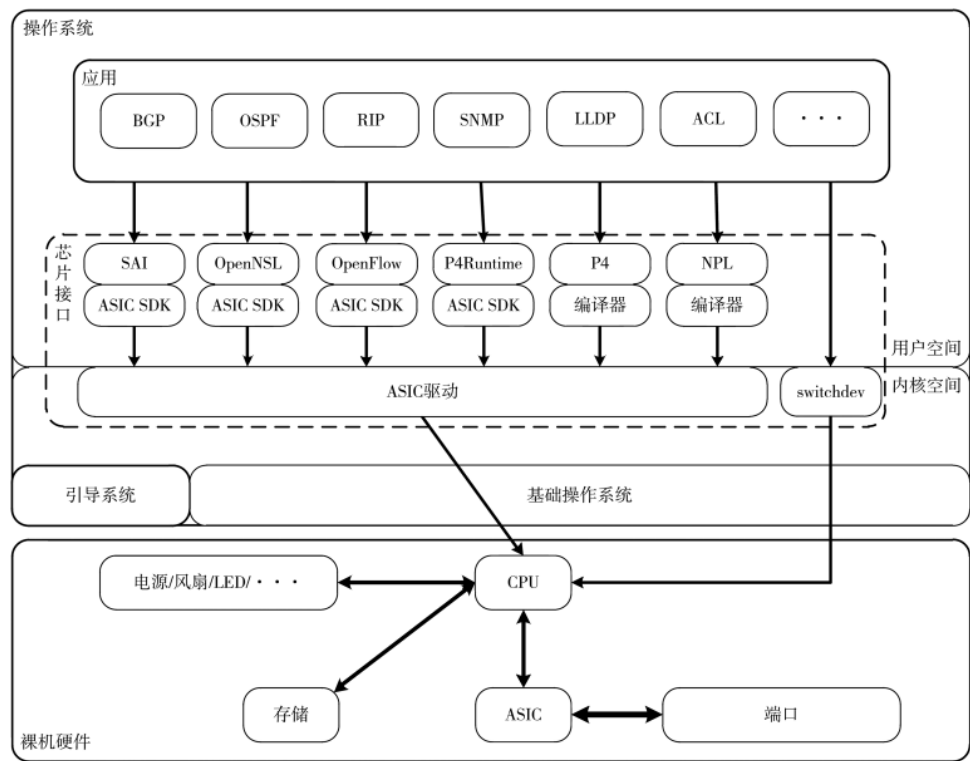


图2 白盒交换机体系架构图

并向上层软件提供一组编程接口。这里的芯片接口是个宽泛的概念,不仅指狭义上传统的 SDK 接口,广义上还可以包括软件定义网络 (Software-Defined Networking, SDN) 概念里基于 SDK 所封装的具备标准性和可编程性的抽象接口。狭义的芯片接口一般是硬件厂商提供的芯片 SDK,目前有形成统一标准的趋势;广义的芯片接口按照网络可编程性的实现位置可以分为控制平面的南向接口和数据平面的芯片可编程语言。

1.2.3 引导系统

引导系统是存储在白盒交换机主板固件中的一个启动引导程序,可以引导安装不同的操作系统。传统软硬件一体化交换机的引导系统是厂商绑定的,只能安装特定的操作系统,而白盒交换机的则支持统一开放地引导安装自定义操作系统。

1.2.4 操作系统和应用

操作系统是运行于白盒交换机并控制交换机硬件行为的基础软件。操作系统本身一般不包含太多的网络功能,但是可以基于芯片接口开发各种网络应用,比如常见的路由交换、网络管理、访问控制等应用。同时,白盒交换机一般包含一个基础操作系统,可以用作基础平台层去模块化开发,作为一

部分而构建新的操作系统。

2 白盒交换机体系研究及比较

白盒交换机追求白盒开放式设计,要求能够打破供应商垂直垄断以及开放固定的网络功能,所以近年来白盒交换机体系的研究主要围绕着设备开放化以及网络可编程两大体系原则。

2.1 设备开放化

设备开放化是白盒交换机开放式架构设计的重要原则,能够有效打破供应商垂直垄断。传统交换机采用软硬件一体化的设计,底层芯片和上层系统紧密地捆绑在一起。相较之下,白盒交换机利用标准化的芯片接口解耦底层芯片和上层应用,利用开放化的引导系统解耦裸机硬件和操作系统,同时利用开源化的操作系统进一步促进软硬件的解耦,充分实现了设备的开放化,打破了供应商的垂直垄断。

2.1.1 标准化芯片接口

芯片接口的标准化可以解耦底层芯片和上层应用,目前呈现出两个方向的趋势。一个方向是统一芯片接口,由开放组织制定接口标准,各硬件厂商负责具体 SDK 的实现。如微软向 OCP 贡献的 SAI^[8]定义了一组交换机 ASIC 芯片的抽象接口,并由各硬件厂商提供各自芯片的 SDK 以适配,使得以一

种统一的方式控制不同厂商、不同芯片的交换机成为可能。随着白盒交换机市场和产业生态的发展,传统交换机芯片巨头博通也开源了自己的统一芯片接口 OpenNSL^[9]。OpenNSL 与 SAI 相似,也是一组芯片抽象接口的定义,需要具体 SDK 的实现支持,只不过 OpenNSL 目前只支持博通的 ASIC 芯片架构。

另一个方向是取代芯片接口,以 Linux 内核驱动直接控制底层芯片,取消接口各异的芯片 SDK。switchdev^[6]是一个 Linux 内核态交换设备驱动模型,能够平等抽象所有底层基础设施,并辅助卸载数据平面转发到交换机 ASIC 芯片上,同时允许用户使用 Linux 内核公开的标准化 API 来控制底层芯片,直接以内核态驱动取代用户态的 SDK。

表 1 标准化芯片接口对比

芯片接口	SAI ^[2]	OpenNSL ^[3]	switchdev ^[4]
推动者	OCP	OCP	Linux 基金会
类型	应用 API	应用 API	应用 API
运行空间	用户空间	用户空间	内核空间
芯片 SDK	需要	需要	不需要
芯片无关	是	是	是
协议无关	否	否	否
应用无关	是	是	是

2.1.2 开放化引导系统

引导系统的开放化允许自定义安装操作系统,实现裸机硬件和操作系统的解耦。当前白盒交换机采用 OCP 推出的统一开放化引导系统 ONIE^[4]。传统软硬件一体化的网络设备直接利用引导程序安装操作系统,而 ONIE 则运行在两者之间。设备启动时,由开放的 ONIE 引导操作系统安装器,完成相应系统的安装。ONIE 为网络设备提供了一个开放的安装环境,软件和硬件实现了解耦,同一裸金属设备可以安装不同的操作系统。

2.1.3 开源化操作系统

操作系统的开源化大大加剧了设备开放化的程度。一方面操作系统本身的开源让其有了更多的选择,促进了裸机硬件和操作系统的解耦;另一方面开源的操作系统推出了自己的抽象芯片接口,促进了芯片接口的标准化以及底层芯片和上层应用的解耦。

操作系统开源化第一个里程碑是 ONL^[7]。ONL 是 OCP 推荐的一款参考性网络操作系统,为操作白

盒交换机提供了必要的软件集合。ONL 具备两个特点:一是兼容 ONIE^[4],推动产业界裸机硬件和操作系统的解耦;二是组件化,既可以单独运行,也可以作为组件构建新的操作系统。

以 ONL^[7]为基础组建新的操作系统的第一个重要节点是 SONiC^[10]。SONiC 也有两个特点:一是采用 SAI^[8]芯片标准化接口,促进底层芯片和上层应用的解耦;二是充分利用现代软件技术,支持 ONL^[7]模块化、上层应用容器化以及交换机状态数据库化。但是,SONiC 实际上并不是直接可使用的操作系统,是需要和 ONL^[7]、SAI^[8]、ASIC 芯片驱动打包才能安装到交换机上的操作系统镜像。

而 Stratum^[11]则意在构建一个生产就绪的操作系统,或作为嵌入式系统组件最小化部署。Stratum 的主要特点是公开了一组面向下一代 SDN 的网络编程接口,包括 P4Runtime^[14]和 P4^[15],实现了网络设备的可交换性和转发行为的可编程性。

DentOS^[2]是 ONL^[7]作为基础操作系统的又一个重要节点。与 SONiC^[10]不同,DentOS 旨在以 switchdev^[6]取代各异的芯片接口。作为一个 Linux 基金会项目,DentOS 利用 Linux 内核、switchdev 和其他 Linux 项目力图构建一个开放标准化的操作系统。

表 2 开源化操作系统对比

操作系统	ONL ^[6]	SONiC ^[7]	Stratum ^[8]	DentOS ^[9]
推动者	OCP	OCP	ONF	Linux 基金会
芯片接口	无	SAI ^[2]	P4Runtime	switchdev ^[4]
基础平台	无	ONL ^[6]	无	ONL ^[6]
生产就绪	是	否	是	否
兼容 ONIE ^[5]	是	是	是	是
设备无关	是	是	是	是

2.2 网络可编程

网络可编程是白盒交换机开放式功能设计的重要原则,能够有效开放网络固定功能。传统交换机一般采用不可编程的固定功能芯片,同时 SDK 芯片接口功能有限,只能提供有限且固定的网络功能。而白盒交换机支持控制平面和数据平面可编程,在控制平面利用南向接口实现网络的软件可编程,在数据平面利用可编程芯片和编程语言实现网络的硬件可编程,充分开放了固定的网络功能。

2.2.1 控制平面可编程

控制平面利用芯片接口编程控制数据平面底

层芯片,当底层芯片功能固定而不可编程时,可编程的芯片接口(一般为南向接口)便成为关键。

第一个比较成熟的南向接口是 IETF 制定的 ForCES^[3]协议。ForCES 协议初次尝试在以太网封装网络设备控制逻辑,形成控制平面,集中管理控制芯片的数据转发,但因为缺乏供应商的支持而没有被采用。

与 ForCES^[3]不同,ONF 支持的 OpenFlow^[2]获得了业界的广泛接受。OpenFlow 创造性地提出流(Flow)和匹配动作表(Match-Action Table)的概念,简化统一了数据包处理流程,进一步实现了芯片功能的 OpenFlow 模式封装,使得控制平面可以很容易控制芯片转发。同时,OpenFlow 支持扩展现有固定功能交换机,对当前交换机体系改动有限,获得了广泛支持。

在此基础上更进一步,P4Runtime^[14]真正做到了 OpenFlow^[2]追求的 SDN 承诺。OpenFlow 流的概念本身是依赖于协议所支持的字段的,这就使得 OpenFlow 是协议相关的。同时,OpenFlow 各版本的协议字段也在不断扩展,无法后向兼容。P4Runtime 则和 P4^[5]结合,使用协议无关的 P4 定义交换机转发逻辑,并用 P4Runtime 配置和填充可编程芯片流水线,由此真正实现控制平面对数据平面的软件定义。

表 3 控制平面可编程南向接口对比

南向接口	ForCES ^[3]	OpenFlow ^[2]	P4Runtime ^[14]
推动者	IETF	ONF	ONF
类型	协议规范	协议规范	协议规范
芯片 SDK	需要	需要	需要
芯片无关	是	是	是
协议无关	否	否	是
应用无关	否	否	否

2.2.2 数据平面可编程

数据平面利用专用编程语言对可编程芯片的控制来实现可编程。即数据平面的可编程不仅依赖可编程语言的芯片接口,更依赖可编程芯片,这一点与控制平面可编程常见的固定功能芯片是不同的。

第一个通用的可编程 ASIC 芯片是 Tofino^[13]。Tofino 系列专为 P4^[5]语言而设计,是首款支持 P4 数据平面可编程的高性能 ASIC 芯片,在开源开放生态支持上具有绝对优势。

随着可编程网络生态的快速发展,交换机芯片

巨头博通和思科也分别推出了自己的可编程芯片。Trident^[15]系列是博通推出的一款可编程 ASIC,凭借产业优势以及基础广泛也获得了一席之地,但其支持的编程语言不是业界广泛接受的 P4^[5],而是自家专门研发的 NPL^[17]。面对行业竞争以及白盒网络的发展,思科也发布了 Silicon One^[16]网络可编程芯片,支持 P4^[5]语言编程,并采用统一架构,支持多规格和多场景部署,获得了产业界的广泛关注。

专用编程语言方面,按照现有的可编程芯片,当前有 P4^[5]和 NPL^[17]支持数据平面可编程。P4^[5]是随着 SDN 发展率先设想的一种芯片可编程语言。相比于 OpenFlow^[2]模式,P4 语言更加抽象和通用。P4 可以任意定义数据平面处理逻辑,不受目标设备、现有协议的限制,用于白盒交换机的可编程芯片,还可以和 P4Runtime^[14]结合实现运行时动态可编程。

NPL^[17]则是随着博通可编程芯片而配套出现的一款编程语言。NPL 以一组适当的原语构造块来描述数据平面包处理行为,构造块涵盖从单个控制信号到高级的硬件接口连接。NPL 具备 P4^[5]的主要特性,但不如 P4 成熟,多种控制块原语过于低级,抽象程度不如 P4。

表 4 数据平面可编程芯片及编程语言对比

芯片/语言	Tofino ^[13]	Trident ^[15]	Silicon One ^[16]	P4 ^[5]	NPL ^[17]
推动者	Barefoot	博通	思科	ONF	博通
类型	ASIC	ASIC	ASIC	编程语言	编程语言
编程语言	P4 ^[5]	NPL ^[17]	P4 ^[5]	\	\
芯片无关	\	\	\	是	是
协议无关	是	是	是	是	是
应用无关	是	是	是	是	是

3 现有挑战和未来方向

3.1 现有挑战

对新型业务和网络发展的需要,白盒交换机在设备开放化和网络可编程方面进行了标准制定、开源开放、产研创新诸多努力,但仍然面临一些由技术标准和产业发展造成的挑战。

(1)可编程 ASIC 芯片价格昂贵,数据平面可编程并不广泛。

以 Tofino^[13]为代表的商业可编程 ASIC 芯片不仅具备 FPGA、NPU 的网络可编程优势,而且不会降低 ASIC 芯片的高性能,是未来网络创新发展的一

个里程碑式节点。正因如此,可编程 ASIC 芯片投入巨大,市场价格昂贵,只有大规模数据中心才会批量配备。同时,当前网络演进的重心在于逻辑的集中控制,数据平面可编程还在学术探索阶段,大多数的基础设施网络依然采用固定功能芯片,利用 OpenFlow^[2]控制平面可编程的白盒交换机来降低成本和开放网络。

(2)芯片接口规范各异,产业界芯片接口并未融合统一。

对于传统芯片接口而言,SAI^[8]受到 OCP 标准化支持,同时被 SONiC^[10]原生配备,受到交换机下游互联网服务商的大力支持;OpenNSL^[9]来自交换机上游芯片巨头博通,产业渗透率高,基础牢固;switchdev^[6]是 Linux 内核组件,受到 Linux 基金会和 CentOS^[12]操作系统支持,意在直接以内核态驱动取代各种传统芯片接口。对于网络可编程的芯片接口而言,OpenFlow^[2]、P4Runtime^[14]以及 P4^[5]都出自 ONF,在网络可编程学术发展中占有重要地位,其中又以 OpenFlow 取得了学术创新和产业发展的平衡,当前获得产业界的较大认同;而 NPL^[17]依然只能算是博通可编程芯片的配套语言。总之,当前芯片接口规范较多,各有千秋,并未在产业界形成融合统一。

白盒交换机产业发展落后,网络设备并未全面白盒化。一方面,OpenFlow 对交换机改动较小,为主流供应商所接受,加之可编程芯片价格昂贵,当前的白盒交换机主要是搭载固定功能芯片的 OpenFlow 交换机。另一方面,固定功能芯片使得 OpenFlow 交换机主要适用于功能简单、带宽高的交换机场景,而难以满足数据中心逻辑复杂功能多的路由器和防火墙等场景。

3.2 未来方向

当前,新型业务和网络发展提出了对白盒交换机的迫切需求,学术界和产业界也大力推动白盒交换机技术与产业生态的发展,按照现有发展趋势和存在的挑战,可以得出如下未来发展方向。

(1)网络编程异构化

如果可编程 ASIC 芯片价格居高不下,白盒交换机可以采用多种芯片组合的异构架构进行高性能的网络可编程。除 CPU 芯片负责计算之外,为了加速数据转发,专用的交换机多配备了一颗 ASIC 芯片。同样地,白盒交换机也可以考虑组合 FPGA、NPU 等可编程芯片进行异构的网络可编程。

(2)芯片接口语言化

当 ASIC 芯片像 CPU 一样形成自己的网络指令集时,各种芯片接口也可能像通用编程语言一样抽象为编程语言而林立着。当前芯片接口规范繁杂,ASIC 还没有形成一致的网络指令集,导致芯片接口抽象程度表现各异。未来开发者利用芯片接口语言定义网络功能,再经过目标特定的编译器编译转换为网络指令程序,送到 ASIC 芯片中执行指令,完成交换机的配置或数据的转发。

(3)网络设备通用化

传统网络设备有交换机、路由器、防火墙等多种类别,白盒交换机可以实现网络设备的通用化,单一白盒交换机可以作为多种网络设备使用。随着 OpenFlow^[2]的出现,白盒交换机已经可以表现得像路由器、交换机、防火墙或者其他网络中间件^[18],同时可编程 ASIC 芯片可以使得白盒交换机既保持高性能又具备多功能,能充分适应各种网络设备场景。这样,未来白盒交换机会具有更强大的通用化,逐步广泛部署在各种网络基础设施中。

4 结论

本文对白盒交换机体系的技术研究、现有挑战以及未来方向进行了梳理和综述。白盒交换机是开放化的网络设备,利用引导系统解耦裸机硬件和操作系统,利用芯片接口解耦底层芯片和上层应用,利用多种开源操作系统促进设备的开放。同时,白盒交换机在控制平面利用南向接口实现固定功能芯片的软件级网络可编程,在数据平面利用编程语言实现可编程芯片的硬件级网络可编程。当前阶段,由于可编程 ASIC 芯片价格昂贵、芯片接口规范各异以及白盒交换机产业发展落后,白盒交换机还面临着技术标准和产业发展造成的挑战。但按照现有发展趋势,白盒交换机有可能向着网络编程异构化、芯片接口语言化以及网络设备通用化而发展。

参考文献

- [1] YASTREBOVA A, KIRICHEK R, KOUCHERYAVY Y, et al. Future networks 2030: architecture & requirements[C]// 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2018: 1-8.
- [2] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review,

- 2008, 38(2): 69–74.
- [3] DORIA A, SALIM J H, HAAS R, et al. Forwarding and control element separation(ForCES) protocol specification[J]. RFC, 2010, 5810: 1–124.
- [4] Open Compute Project. Open network install environment[S/OL]. 2013[2021–11–15]. <https://opencompute-project.github.io/onie/>.
- [5] BOSSHART P, DALY D, GIBB G, et al. P4: programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87–95.
- [6] Linux Foundation. Ethernet switch device driver model (switchdev)[S/OL]. 2014[2021–11–15]. <https://www.kernel.org/doc/Documentation/networking/switchdev.txt>.
- [7] Open Compute Project. Open network linux: a reference network operating system for OCP[EB/OL]. 2014[2021–11–15]. <http://opennetlinux.org/>.
- [8] Open Compute Project. Switch abstraction interface(SAI): a reference switch abstraction interface for OCP[S/OL]. (2015–xx–xx)[2021–11–15]. <https://www.opencompute.org/documents/switch-abstraction-interfaceocp-specification-v0-2-pdf>.
- [9] Broadcom Inc. OpenNSL: library of open networking APIs[EB/OL]. (2015–09–27)[2021–11–15]. <https://www.broadcom.com/products/ethernet-connectivity/software/opennsl>.
- [10] Open Compute Project. SONiC: software for open networking in the cloud website[EB/OL]. 2016[2021–11–15]. <https://azure.github.io/SONiC/>.
- [11] Open Networking Foundation. Stratum[EB/OL]. 2019[2021–11–15]. <https://opennetworking.org/stratum/>.
- [12] The Linux Foundation. DentOS[EB/OL]. 2020[2021–11–15]. <https://dent.dev/dentos/>.
- [13] Barefoot Networks Inc. Intel® Tofino™ 2: second-generation P4-programmable ethernet switch ASIC[EB/OL]. 2018[2021–11–15]. <https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-2-series.html>.
- [14] Open Networking Foundation. P4Runtime specification: version 1.3.0[S/OL]. [2021–11–15]. <https://p4.org/p4-spec/p4runtime/main/P4Runtime-Spec.html>.
- [15] Broadcom Inc. Trident4/BCM56880 series: high-capacity StrataXGS® trident4 ethernet switch series[EB/OL]. (2019–xx–xx)[2021–11–15]. <https://www.broadcom.com/products/ethernet-connectivity/switching/strataxgs/bcm56880-series>.
- [16] Cisco Systems, Inc. Cisco Silicon One: meet the first unified silicon architecture for web scale and service provider networks[EB/OL]. (2020–xx–xx)[2021–11–15]. <https://www.cisco.com/c/en/us/solutions/silicon-one.html>.
- [17] Broadcom Inc. NPL – Network programming language specification v1.3[S/OL]. (2019–xx–xx)[2021–11–15]. <https://nplang.org/npl/specifications/>.
- [18] FEAMSTER N, REXFORD J, ZEGURA E. The road to SDN: an intellectual history of programmable networks[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(2): 87–98.

(收稿日期: 2022–01–10)

作者简介:

张成林(1996–), 通信作者, 男, 硕士研究生, 主要研究方向: 软件定义网络、数据平面可编程、白盒交换机。E-mail: chenglinzhang@seu.edu.cn。

宋玲玲(1983–), 女, 硕士研究生, 工程师, 主要研究方向: 网络操作系统、白盒交换机。



版权声明

经作者授权，本论文版权和信息网络传播权归属于《信息技术与网络安全》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《信息技术与网络安全》编辑部
中国电子信息产业集团有限公司第六研究所