



网络可编程技术研究

谢立军, 胡宇翔, 汪斌强, 兰巨龙, 程国振, 王志明

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

摘要: 网络可编程技术通过在网络设备提供开放的可编程接口, 增强了网络的灵活配置能力和资源整合能力。首先从网络可编程技术的诞生背景入手, 回顾了开放可编程思想、控制与转发分离架构以及目前典型的软件定义网络 (software defined networking, SDN) 和网络功能虚拟化 (network function virtualization, NFV) 架构。然后, 详细阐述了 SDN 架构下的协议无关性、流表实现、高级编程语言以及与 NFV 融合方面的研究成果。最后, 总结了网络可编程技术的研究趋势。

关键词: 网络可编程; 软件定义网络; 网络功能虚拟化; 协议无关性; 编程语言

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016038

Research on network programmable technologies

XIE Lijun, HU Yuxiang, WANG Binqiang, LAN Julong, CHENG Guozhen, WANG Zhiming

National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China

Abstract: Network programmable technologies enforce the capacity of flexible configuration and resource composition by providing network devices with programmable interfaces. Starting with the discussion on the background in the naissance and developments of network programmable technologies, the initial programmable technologies, frameworks of control and data plane and typical software defined networking (SDN) and network function virtualization (NFV) were reviewed. The research on protocol irrelevance, implementation of flow tables, high-level programming languages and integration with NFV were elaborated. Moreover, the future works were summarized in the end.

Key words: network programmability, software defined networking, network function virtualization, protocol irrelevance, programming language

1 引言

随着网络规模不断扩大, 用户数日益增多, 网络新技术、新应用日趋多样, 传统互联网的体系结构和设计理念

日益暴露出严重不足^[1]。首先, 网络服务多样化要求打破传统网络尽力而为的服务模式, 提高网络的服务能力, 以满足用户部署新型服务和协议的需求; 其次, 作为一个庞大的复杂系统, 各种协议的决策逻辑共同控制网络流量的机

收稿日期: 2015-06-29; 修回日期: 2015-12-17

基金项目: 国家重点基础研究发展计划 (“973”计划) 基金资助项目 (No.2012CB315901, No.2013CB329104); 国家自然科学基金资助项目 (No.61372121, No.61309019); 国家高技术研究发展计划 (“863”计划) 基金资助项目 (No.2013AA013505)

Foundation Items: The National Basic Research Program of China (973 Program) (No.2012CB315901, No.2013CB329104), The National Natural Science Foundation of China (No.61372121, No.61309019), The National High Technology Research and Development Program of China (863 Program) (No.2013AA013505)



制,使网络行为更加难以预测和控制;最后,网络设备种类多样且相互独立和封闭,使得网络管理员只能通过命令行接口(CLI)等方式,手动地将高层策略转化为设备底层配置命令来实现网络管控和运维,导致网络更新缓慢、运行成本膨胀。研究人员发现,存在上述问题的根本原因在于网络核心设备控制逻辑与数据转发之间紧耦合以及核心设备之间的分布式控制。

学术界和产业界普遍认为,要弥补网络日益凸显的不足,应该重新设计网络控制和管理结构,将控制逻辑与数据转发相分离,采用集中控制的方式,降低网络核心设备的复杂性,提高网络控制和管理的灵活性,增强对网络新技术、新协议的支持能力,从而提出了网络可编程的思想。网络可编程是对网络整体功能和行为的高度抽象以及软件编程的自定义,其核心思想是通过在网络节点提供开放可编程接口,利用编程语言向网络设备发送强大的编程指令,实现对网络功能和行为的按需管控和新业务的快速部署。随着国内外对网络可编程思想认识的不断加深,网络可编程技术的不断完善,该技术逐渐成为构建新型未来网络的核心技术。2008年,SDN (software defined networking, 软件定义网络)^[2]思想随着 OpenFlow^[3]的面世推广开来,其借用了电话网络的思想,将控制和转发平面严格分离,简化了网络管理和新服务的部署。随后,作为对可编程技术的进一步补充和应用,ETSI^[4]在2012年提出了网络功能虚拟化的思想,进一步推动了网络可编程思想落地的进程。

2 网络可编程技术发展历程

随着硬件/软件、网络操作系统、虚拟化等技术水平的不断提高,网络可编程技术逐渐成熟。整体来说,网络可编程技术分为3个发展阶段:早期开放可编程思想,将可编程功能的思想引入网络设计;“控制与转发分离”研究推进了网络控制平面与转发平面之间开放接口的进一步研究;对基于 OpenFlow 的 SDN 和 NFV 的研究不断深入和推广,提高了网络可编程技术实际部署的可行性。下面对网络可编程技术发展史进行详细介绍。

2.1 早期开放可编程思想

2.1.1 基于开放信令的思想

从1995年开始,OPENSIG工作组专注于研究“开放、可扩展、可编程的ATM、Internet和移动网络”,第一次真正将可编程思想加入网络设计^[5]。其主要思想是通过开放、可编程的网络接口访问网络硬件,将通信硬件和控制软件分

离并且使开放可编程接口标准化。但是,由于垂直、整合的交换机和路由器等网络设备具有封闭特性,导致“开放信令”思想在当时的技术条件下难以实现。不过,随着软硬件技术的不断发展,后续的IEEE P1520、MSF (Multiservice Switching Forum)^[6]、国际互联网组织IETF发布的GSMP^[7]、转发件与控制件分离ForCES工作组以及NPF (Network Processing Forum)^[8]都基于该思想进行研究。

2.1.2 基于动态代码的主动网络思想

同期,美国国防部高级研究计划署(DARPA)在“未来网络发展方向研讨会”上提出了主动网络(active network)^[9]这一新的网络体系架构,该架构首次提出“面向定制化服务可编程的网络基础设施”思想。随着美国的GENI^[10]、NSF FIND^[11]和欧盟的FIRE^[12]等项目不断推进,主动网络成为一系列新型网络架构方案中的研究热点。主动网络产生了两个主要模型:一是可编程路由/交换机模型^[13,14],即网络操作员利用“out-of-band”方式将程序代码放置到网络节点运行;二是封装模型^[15],即用户利用“in-band”方式,将程序代码通过数据分组的形式传递到网络节点运行。尽管主动网络的思想有效、可行,但是由于在当时缺乏大量的应用范例和工业部署需求,脱离硬件技术水平,难以广泛使用。

上述两种可编程思想对开放架构网络研究具有一定的互补性,其共同目标都是实现面向科研人员和终端用户的网络开放可编程,研究重点都主要围绕转发平面的可编程,由于受到软硬件技术的限制,仅仅能实现设备级的可编程。随后,关于网络开放可编程的研究工作,基本都采用了基于开放信令思想的控制平面(control plane)和转发平面(data plane)分离的基本体系结构。下面对这些研究工作进行详细介绍。

2.2 控制与转发分离技术

网络规模的不断扩大,对服务提供商提出了集中式控制和更高可靠性的需求,设备生产商将控制平面软件从分组转发硬件中分离,直接部署在整合的控制器硬件中。这种“控制与转发分离”的设计为新型网络创新带来新的机遇和挑战:控制平面和转发平面之间的开放可编程接口的设计以及网络在逻辑上的集中控制。围绕这两个方向比较典型的研究工作和项目包括DCAN (devolved control of ATM networks)^[16]、ForCES^[17]、4D (decision、dissemination、discovery、data)^[18,19]和NETCONF^[20]。

DCAN将大量网络设备的控制、管理功能与转发设备

本身解耦,并委托给第三方执行,硬件设施只执行简单的转发操作。与之不同,ForCES将网络设备的内部架构重新定义为两个逻辑实体:转发元件(FE)和控制元件(CE),FE使用底层硬件进行数据分组的处理,CE使用ForCES协议控制FE进行数据分组的处理。4D项目将网络分为决策平面、数据平面、分发平面和发现平面。其中,决策平面具有全局的网络视图,利用分发平面和发现平面分别提供通信和信息服务,数据平面负责转发分流。该4层架构解决了网络逻辑决策平面和分布式的硬件设备结合过紧的问题,实现了路由选择逻辑和管理网络元素之间交互协议的严格分离,使网络更加顽健、安全,并有利于异构网络的管理。NETCONF协议为网络设备提供了一个API(application programming interface,应用程序编程接口),利用该API可以发送和接收可扩展的配置数据^[21]。

该阶段的项目研究主要实现面向网络管理者的可编程技术,研究重点为控制平面编程技术,实现了网络级的视图和控制。简而言之,控制与转发分离的架构简化了网络管理和配置操作,实现了网络控制的集中化和顽健性,为网络设备提供了开放可编程接口,提高了网络可编程能力,为SDN的产生奠定了理论和实践基础。

2.3 软件定义网络

2006年,在控制与转发分离架构的基础上,美国斯坦福大学的Clean Slate研究组针对企业网的安全需求,提出了面向企业网的管理架构SANE^[22],首次实现了路由和接入在逻辑上的中央安全控制和更细粒度的流表转发策略。次年,在对SANE进行功能扩展的基础上提出了Ethane^[23],实现了基于流表进行报文转发的交换机。随后,在这两个项目研究成果的基础上,Nick McKeown教授等人提出了OpenFlow,并大力推广SDN的思想。

SDN技术的本质是网络转发功能的可编程性,通过将控制器作为可编程控制的切入点,网络管理员通过网络应用层软件能够动态、快速、按需地进行网络配置。OpenFlow作为第一个具有现实意义的SDN协议和原型实现,具有良好的实现性,使SDN更加实用。基于OpenFlow的SDN可编程架构的关键组件包括OpenFlow交换机和控制器。其中,OpenFlow交换机负责数据转发功能,由流表、安全信道和OpenFlow协议3部分组成。控制器中运行的网络操作系统(network operating system,NOS)实现了网络的集中式控制逻辑,为网络应用程序提供了面向网络设备的可编程接口。目前,支持OpenFlow协议的多种控制器软件已经得到开发和推广,第一个网络控制器平台NOX^[24]已经演化出了多个版本^[25]。随着研究的进展,Maestro^[26]、POX^[27]、Beacon^[28]、Floodlight^[29]、Ryu^[30]、SNAC^[31]、Trema^[32]、ONOS^[33]、RouteFlow^[34]、NOX-MT^[35]、Onix^[36]和OpenDaylight^[37]等多个控制器软件平台实现了更加完善的功能。下面以典型的OpenDaylight控制器为例,介绍控制器的组成结构。

OpenDaylight控制器架构分为3层^[37]:南向协议插件、SAL(service adaptation layer,服务适应层)和北向应用功能,如图1所示。其中,南向协议插件作为控制器与网络设备之间的接口,支持OpenFlow、BGP-LS、LISP等协议;SAL将南向协议插件功能转换为高层应用/服务功能,高层应用功能为应用提供北向API。应用功能包括拓扑输出、目录管理和OpenFlow统计管理等。OpenDaylight控制器支持不同的南向协议插件以及不同的服务和应用集合,允许程序开发者和网络研究人员更加专注于SDN的API,而不用考虑与网络设备之间的通信协议。

作为目前新型网络架构的研究热点之一,SDN既继承了前期控制与转发分离思想,也实现了可实际部署的网络可编程关键技术,对于网络革新具有重要的意义。

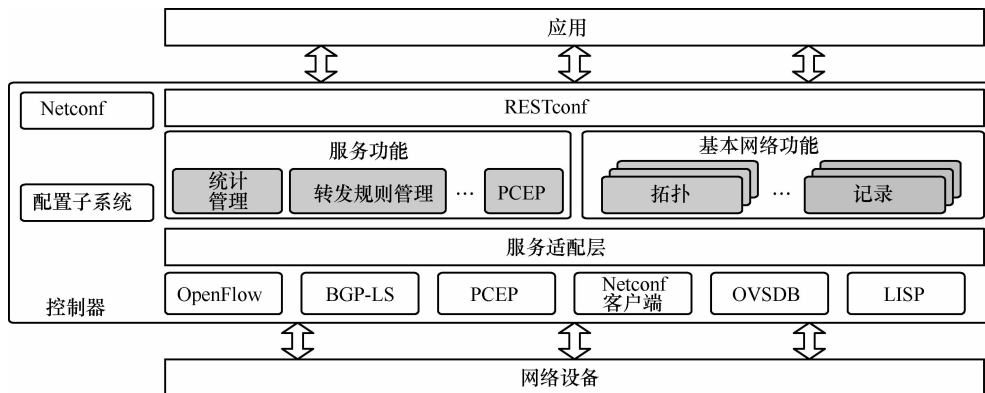


图1 OpenDaylight控制器架构



(1)统一了网络设备功能

传统的网络设备依靠不同的分组头字段进行流量控制,而基于 OpenFlow 协议的交换机则可以使用任意的分组头匹配域定义转发行为,实现了网络设备匹配域与转发行为的统一。同时,基于 OpenFlow 的 SDN 归一化了规则配置技术,将粗粒度规则预配置改进为反应式的细粒度规则配置。

(2)提供了网络操作系统视图

不同于主动网络中提出的节点操作系统,SDN 中的网络操作系统^[28]在逻辑上将网络分为 3 层^[29]:转发层、状态管理层和逻辑控制层,能够对网络状态进行全局抽象,为网络管理者提供了网络级的控制视图。

(3)实现了分布式网络状态管理技术^[30-32]

SDN 控制平面与转发平面分离的架构导致网络状态管理面临挑战。为了实现可扩展性、可靠性和高性能,需要同时运行多个逻辑上集中的控制器,确保其协同工作。

2.4 网络功能虚拟化

2012 年,为了应对电信网络部署的封闭性、私有性带来的可操作性和管理成本挑战,20 多个全球最大的电信网络服务商组成的 ISG(industry specification group)与欧盟电信标准协会 ETSI 提出了 NFV(network function virtualization,网络功能虚拟化)^[38,39]。NFV 在本质上基于云技术,通过虚拟和整合传统的专用网络设备功能,利用控制平面技术动态编程网络数据流路径,支持数据流按需经过功能/服务集合,使网络获得更高的灵活性,加速新服务部署,同时降低管理和资金成本^[40]。

与 SDN 观点不同,NFV 认为,路由器不应该仅仅是一个“简单的、哑的”转发设备,而应是一个连接多种智能模块、更加整合、支持定制化处理功能的设备,并对未来网络设备架构进行了描述^[41]:未来的转发平面架构应实现网络与通用处理硬件之间的深度整合;新型网络设备应允许第三方自由安装转发平面功能;应该同时支持传统网络任务和新型应用。但是,在硬件架构实现上还存在争议^[42]:一种观点认为应该用标准化、大容量设备完全取代现有的专用硬件,另一种观点认为应将专用硬件和通用线速卡混合使

用。这两种观点都有各自的优势和不足,硬件资源和性能之间的平衡是考虑的首要问题。由于其巨大的发展潜力,NFV 自诞生之初便获得了广泛关注,尤其是产业界对其前景表示十分乐观。

2.5 小结

综上所述,网络可编程基于网络抽象实现了集中的网络控制、自动化网络配置和定制化的网络服务,提高了网络的灵活性,使网络更加易于管理和更新。表 1 总结了上述各个阶段网络可编程技术研究的特点,主要从可编程范围、可编程粒度、实现复杂性、可编程面向对象等方面进行比较。

3 SDN 架构下的可编程关键技术

SDN 通过控制平面与转发平面分离,提供强大的可编程接口,赋予网络可编程能力^[65]。目前,有关 SDN 的可编程技术研究主要围绕控制平面和转发平面的可编程能力的提升。下面对这两方面的研究工作进行详细介绍。

3.1 转发平面的可编程技术

作为 SDN 转发平面的实现范例,OpenFlow 交换机以流表作为基本的数据分组处理单元,以报文解析、关键字匹配、报文操作组成多级流表的流水线处理结构,通过实现每个处理单元的标准化达到整个转发平面的完全可编程。但是目前交换机的设计和实现上存在的很多问题制约了转发平面的可编程性,比如报文解析中的协议相关性、流表实现不灵活等问题。下面就目前针对转发平面的这两方面的相关研究进行介绍。

3.1.1 协议无关性

在传统网络的 5 层体系结构中,对于协议的处理是分层次的,即位于不同层的网络设备只需解析本层数据分组头部。然而,SDN 为了实现数据分组的扁平化处理,其转发平面通过配置匹配域对不同层次、多种协议进行选择解析,增加可编程性的同时也带来了协议相关性问题。以 OpenFlow 交换机为例:首先,OpenFlow 交换机的分组头协议解析模块目前只实现了基本的 IPv4 和 IPv6 协议解析和目的 IP 地址抽取功能,其灵活性非常有限,只能适用于 IP

表 1 各阶段网络可编程技术对比

发展阶段	可编程范围	可编程粒度	实现程度	可编程面向对象
早期可编程思想	设备级可编程	设备可编程	没有实现	科研人员和终端用户
控制与转发分离	网络级可编程	协议可编程	实验室环境下实现	网络管理者
SDN	网络级可编程	报文、流可编程	数据中心环境下实现	网络管理者
NFV	网络级可编程	报文、流可编程	电信运营商网络环境下实现	网络提供商

地址查找操作,而无法实现非IP地址字段的分组头提取。其次,OpenFlow交换机中流表关键字的提取受到预先定义的匹配域限制,如果用户的新协议使用数据分组头的其他域,OpenFlow交换机将无法对其进行匹配查找。虽然OpenFlow标准不断扩充匹配字段,但是也进一步增加了OpenFlow交换机的硬件设计成本。研究人员针对该问题进行了大量的研究工作,具体情况如下。

(1)CAFE

针对当前转发平面中数据分组头解析模块的不灵活性,Lu等人^[43]提出了可编程的数据分组转发引擎CAFE。如图2所示,CAFE的分组头解析处理包括3部分。其中,PKtype Identifier模块负责识别数据分组类型,只有被识别的数据分组才会进入下一个步骤,丢弃不能识别的分组。第二部分包含多种类型的header field verifier、lookup key extractor和checksum verifier模块,根据识别的分组类型,从中选择相应类型的模块进行分组头验证和转发表查询。其中,header field verifier模块过滤掉硬件不转发的数据分组,checksum verifier验证分组的校验和。Lookup key extractor提取出查找关键字的分组头域集合,lookup engine负责使用关键字查表。Arbiter模块检查前两部分的处理结果——如果出现数据分组被过滤、校验、错误以及转发表匹配失败等情况,数据分组会被直接丢弃。由于数据分组不仅与分组类型相关,还与流表项相关,因此第三部分header modifier模块同时支持根据“类型”信息和“流”信息修改数据分组头。

CAFE的优势在于其分组头解析模块中加入了任意比特域提取器lookup key extractor,支持数据分组头部任意比特域的自由组合,实现了用户自定义的查找关键字提取和对新型协议的支持。然而,其复杂的关键字提取过程给分组头解析带来了性能方面的挑战。

(2)PPL

Attig等人^[44]设计了分组头解析流水线和数据分组头高级解析语言PPL,以面向对象方法描述了分组头解析算法。利用FPGA的并行处理能力,设计了定制化、可编程

的虚拟处理架构,能够在系统运行时动态地更新解析算法,满足了特殊的分组头解析算法需求,使分组处理性能大大提高。以类C++的风格定义了每个头部的格式和处理规则,方便用户对新协议分组头解析的自定义处理。PPL在Xilin Virtex-7 FPGA上运行的线速处理能力达到了400 Gbit/s,在解析性能上优于CAFE。

(3)PLUG

如图3所示,传统ASICs电路针对每种协议提供专用的查找模块,使得硬件内的功能定义比较困难,导致数据链路层和网络层的新协议部署往往需要升级硬件设备。为了便于新协议的部署,Lorenzo等人^[46]提出一个灵活的查表模块——PLUG(pipelined lookup grid),使用SRAM解决灵活的多级流表的问题,使得新协议的部署不需要改变硬件,增加了网络更新能力,如图4所示。

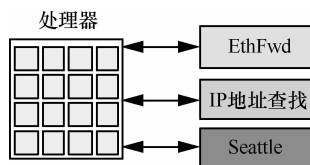


图3 传统线速查找模块

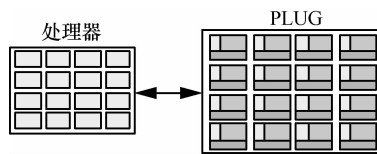


图4 流水线查找网格

PLUG设计了可全局使用、协同工作的资源“基片”,通过不同“基片”的组合,构成支持不同协议的处理单元。同时,通过资源的全局规划使用,PLUG避免多个处理单元之间存在存储单元或通信端口的冲突,完成了IPv4、Ethernet和Seattle等基于数据流的编程模块,并将它们映射到PLUG硬件。PLUG架构实现了可编程的查找模块,支持修改转发平面的协议;设计了一个基于数据流的编程模型,基于该模型的转发表项可灵活支持现有或新型协议的部署。但是,PLUG中大量的资源“基片”对芯片容量要求太高,而且基于散列表的查找无法进行任意字段的匹配,只

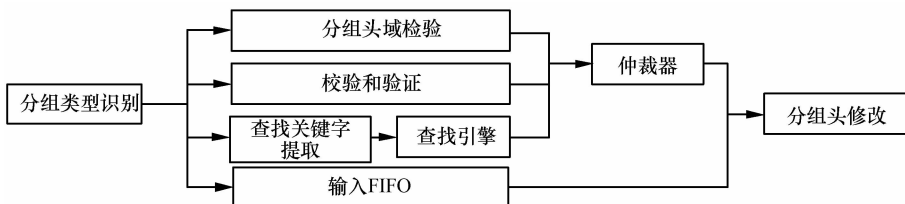


图2 CAFE分组头解析模块结构



能支持 Match-Action 模式的传统芯片功能。

(4)POF

针对 OpenFlow 协议中固定匹配字段的僵化,POF(protocol-oblivious forwarding)^[47] 基于偏移位和长度实现了用户自定义字段域。POF 由 search key、metadata 和 instruction 3 个模块构成,其中 search key 和 instruction 提取由软硬件结合实现,硬件实现机制实现了协议无关性。

3.1.2 实现灵活性

由于现有 OpenFlow 交换机和协议将流表匹配域限制在固定的 2~4 层分组头信息,导致任何使用高层信息的流必须转发到控制器来获得处理指令,降低了网络效率,增加了流处理时延。同时,多级流表交换机芯片只能完成 48 个流表,而且在芯片制造时,流表的宽度、深度和执行顺序都要预先指定,这都降低了 SDN 可编程技术的灵活性。针对以上问题,研究人员对软件平台、硬件设计、流规则更新、网络视图、数据分组格式等方面进行了大量研究。

(1)软件实现

基于 Open vSwitch 平台的灵活可编程性,Mekky H 等人^[48] 设计了一个应用感知的可扩展转发平面,以软件形式实现了多种应用层的流处理和决策功能(TCP 分片、NAT、L7 服务选择、防火墙等)。通过使用 2~4 层以外的分组头信息,将 OpenFlow 转发抽象归纳为有状态的动作,将大部分的流处理限制在转发平面。该软件架构如图 5 所示。

由于标准 OpenFlow 交换机中的未匹配流必须发送到控制器进行处理,导致控制器负载很高。Mekky H 在转发平面增加了“app-table”,并将“miss-match”流表项的“to controller”指令改为“to app-table”,若在“app-table”中仍然匹配失败,再将该流转发到控制器,由控制器下发流表,图 5 中步骤(6)~(8)表示此转发过程。其中,“app-table”的表项利用软件实现,由管理员预先编程,而且多个表项以流水线架构实现 app action 的任意组合,能够执行复杂的网络服务。这种方法通过将“app-table”预先下发到交换机中,允许交换机实现更多的流表匹配,加速流处理过程,降低

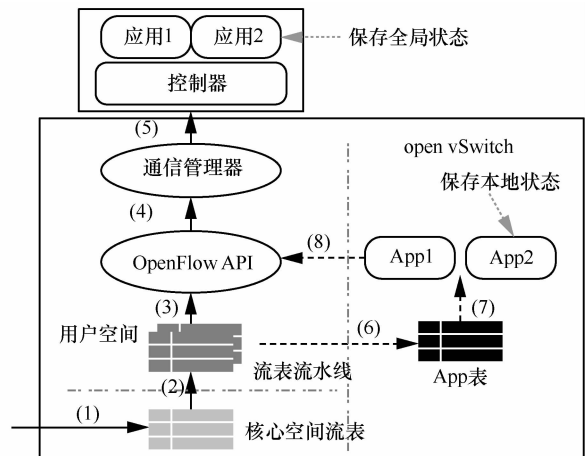


图 5 基于 OVS 的应用感知 SDN 系统架构

了控制平面的处理负载和通信开销。但是,在运行时刻动态增加转发平面“app-table”时,需要一种动态加载机制来保证,而且“app-table”实现的功能有限。

(2)硬件实现

与软件平台相比,利用硬件平台可以提高转发效率,但是同时存在编程不灵活的问题。针对此问题,参考文献[49]提出了一个新型的流水线结构——可重配置的匹配表(reconfigurable match table, RMT)模型,如图 6 所示。该模型主要由分组头解析器、逻辑块、可配置输出队列组成。其中,分组头解析器用于识别数据分组的协议类型,同时根据数据分组的协议类型进行数据分组解析,提取关键字段并将其组合成新的分组头域 header,之后将 header 和数据分组同时向后级逻辑块输出。逻辑块是最基本的数据分组处理单元,支持灵活的用户可配置的匹配域提取,利用匹配得到的动作字段来完成对数据分组的处理。RMT 模型中,灵活可编程的分组头解析器可以提取任意协议的匹配字段,而灵活可配置的逻辑块可以匹配任意字段,实现灵活的数据分组处理。但是该模型对硬件的容量要求很高,目前只能实现不多的逻辑块数量。

(3)流规则更新

由于大量的网络功能要求根据当前流状态改变流动

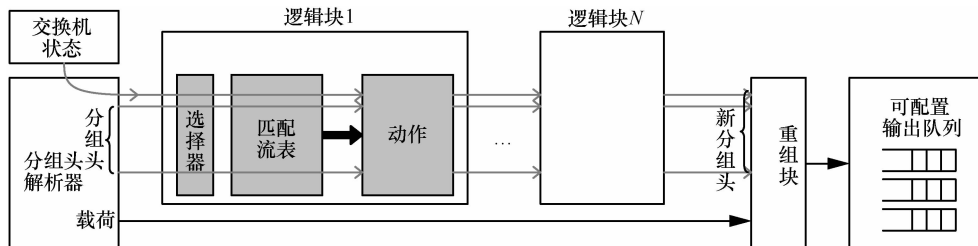


图 6 RMT 模型

作,导致基于流的规则更新过于频繁,降低了网络的可编程性。Masoud 等人^[50]提出了一个新的交换机抽象模型——FAST(flow-level state transition,流级别状态转换),将基于流或流集合的网络任务表示为本地状态机。FAST 模型包含 3 个部分:抽象,允许操作者为不同的应用编程状态机;FAST 控制器,将状态机翻译为转发平面 API,并管理本地状态机之间的交互;FAST 转发平面,基于商用交换机支持状态机的流水线处理。FAST 允许控制器预先编程状态转化,支持交换机基于本地信息动态地执行动作,并且易于部署商用交换机。针对各流表项之间存在的依赖关系导致表项更新速度缓慢而且开销较大的问题,参考文献[51]通过分析规则模式提出了 CacheFlow 算法,基于该算法获得的关系图表达转发平面规则。采用一种新的分片技术打破大量规则之间较长的关系链,并将其转换为新的规则组覆盖大量的不常用规则,提高了常用规则的存储空间。实验数据证明,CacheFlow 算法能有效地提高流表更新效率并减少更新开销。

(4) 网络视图

SDN 为了支持全局视图,控制器需要获取每个流表项的计数器信息。而大部分计数器都部署在 ASIC 硬件中,其不灵活性和有限资源导致控制器难以获得全局视图。参考文献[52]将 ASIC 与通用 CPU 连接,将传统计数器替换为规则匹配记录串,并转发到 CPU 进行处理。利用 CPU 强大的处理能力,能更灵活地处理与计数器相关的信息,降低了 ASIC 存储空间占用率和计算复杂性,使控制器易于获取计数信息。

由于控制器对流的可视性与流表的操作性之间的耦合,会造成流视图的缺失、冗余的流视图以及流表规模急剧膨胀等问题。针对该问题,参考文献[53]采用 FlowInsight 架构分离了流的可视性和可操作性。该架构包含 FlowView 和 FlowOps 两个流水线结构表,数据分组依次由 FlowView 和 FlowOps 进行处理,如图 7 所示。FlowView 定义了控制器可见的流,其中每个流都包含匹配域、优先级、视图域和计数器。视图域根据预设的触发条件向控制器发送相对应

的流视图,例如,see-on-first-packet 要求每个流的第一个数据分组发送至控制器,trigger-on 要求流量大于 1 Mbit/s 的流视图发往控制器。FlowOps 与现有流表工作方式相同,匹配数据分组并执行转发操作。

(5) 数据分组嵌入式可编程

针对目前转发平面功能单一、难以扩展的问题,参考文献[54]提出了一个快速增加网络转发平面功能的方法:终端用户将 TPP(tiny packet program,小数据分组程序)嵌入数据分组中,进行网络状态的主动查询和操作。其中,TPP 本质上是一个以太网数据分组,具有唯一可识别的头部,包含程序指令、额外空间(用来分组存储)和一个可选的封装载荷(IP 分组)。基于高速、一致、低耗的转发平面接口,TPP 为终端用户提供了空前的网络行为视图,将网络状态和数据分组信息之间进行关联,支持拥塞控制、测量、故障检测和验证等大量的网络管理功能。

3.1.3 小结

综上所述,由于 SDN 架构中将所有控制决策部署在控制器中,而转发平面只执行简单的转发任务,导致转发平面功能单一、可编程性不足。OpenFlow 协议标准发展到现在的 1.4 版本,通过不断增加流表项的匹配域长度实现转发平面功能的扩展,导致交换机硬件性能难以满足,成为转发平面可编程技术发展的壁垒。目前,实现转发平面可编程的主要技术包括协议无关性和流表实现的可编程,涵盖了数据分组头解析、关键字匹配查找、计数器、流表视图等方面,增强了转发平面的灵活性,减少了硬件性能带来的限制。

3.2 控制平面高级编程语言

SDN 控制器作为逻辑集中控制节点,既是 SDN 应用的运行平台,也是 SDN 应用的开发环境。由于目前 SDN 控制器通过在服务器上运行不同的控制软件来实现,且各种控制软件设计的逻辑不同,导致其与控制应用的编程接口呈现多样化和复杂化的特征,阻碍了新型网络应用和功能的开发。针对该问题,研究人员设计了多种控制平面高级编程语言。

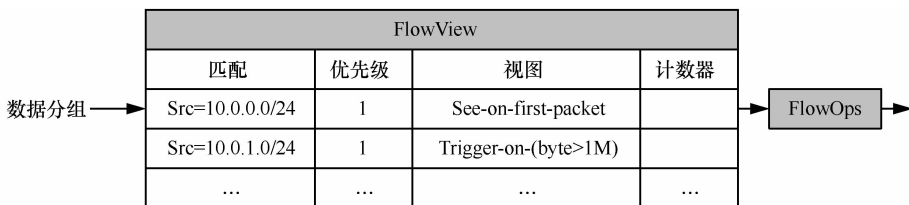


图 7 FlowInsight 架构



2011年,Foster N和Harrison R等^[55]提出面向分布式网络交换机的高层编程语言Frenetic,采用声明式和模块化设计准则,提供单层编程模型、无竞争语义和代价控制机制,基于函数响应编程(functional reactive programming, FRP)组合库管理网络报文转发策略,为描述高层报文转发策略提供了网络流量聚合和分类功能,实现了模块化推理和模块重用。2012年,在Frenetic基础上,Monsanto C和Foster N等人^[56]提出新的NetCore编程语言来表达SDN的转发策略,该语言提供了编译算法以及新的运行环境。Christopher等人^[57]提出新的模块化应用抽象语言Pyretic,允许程序员利用高层抽象定义并以不同的方式组合网络策略,并在抽象的网络拓扑上运行程序。同时Pyretic程序能够在运行系统上执行,实现了网络配置的可编程化。Andreas等人^[58]基于FRP模型提出了Nettle语言,采用了分层的设计,支持以事件流的方式获取控制消息,能够简便地表达动态负载算法。针对大部分OpenFlow原型系统缺少可配置接口的问题,Voellmy A等^[59]提出了一种新的SDN控制架构——Procera,基于FRP的策略语言,支持通配符规则、规则的预生成和自定义,进一步对语言抽象进行了优化。上述控制平面高级编程语言对比情况见表2。

表2 控制平面编程语言

接口语言	技术特点
Frenetic	提供状态查询、规则组合和一致性更新的API抽象
Pyretic	提供转发和查询策略库,支持规则的串行和并行组合
NetCore	支持通配符规则、规则的预生成和自定义
Nettle	基于函数响应式编程(FRP)框架,提供离散事件流和连续属性的网络控制API
Procera	应用FRP框架,支持事件流的组合和自定义

4 SDN与NFV融合

SDN控制平面是可以部署任意网络控制程序的开放可编程平台,允许不同的参与者对转发决策进行细粒度的控制。但是,由于SDN将网络决策全部放置在控制平面,由于流表中的“action”仅支持数据分组的转发、丢弃以及修改分组头字段等,转发平面只是一个“简单的、哑巴的”交换设备,目前只能实现这些功能组合叠加,而无法真正实现深度分组检测、防火墙、网络地址转换、Web缓存等4~7层的网络功能^[60]。为了增强SDN的网络功能可编程,Qazi Z A等^[62]提出了一个基于SDN的策略实施层SIMPLE,将中间件作为黑盒处理,通过策略的动态调整,自动化配置中间件功

能,满足了NFV的基本要求,但是该方法会导致中间件大量的上下文信息丢失,降低了安全性。

与SDN不同,NFV利用运行在虚拟化标准服务器上的软件实现各种网络中间件功能,实现了灵活的网络功能可编程性,缓解了网络所面临严峻的成本、可靠性和复杂性等挑战。针对中间件状态面临的描述、操作和控制挑战,Gember A等^[63]提出了一个软件定义的中间件网络框架,实现了中间件的状态抽象描述、动态扩展和在线迁移,简化了复杂、多样的网络功能管理。ClickOS^[64]则提供了网络功能虚拟化的实施方案,采用快速启动和低时延的小型虚拟机(只有5 MB大小)方案,每个虚拟化可以承载一个中间件。但是,传统基于硬件设备的网络中间件具有静态、固定的地址,网络中的流量可以准确地引导至各个中间件进行处理;而NFV基于软件的网络功能具有动态、灵活的特性,其位置、地址根据需求而变,当高层策略要求流量经过多个网络功能序列时,流量引导成为下一步需要解决的问题。

但是,若将两者的优势相互结合,则可以弥补对方的不足,产生1+1>2的效果。表3对SDN和NFV的优缺点进行了详细的对比和分析。

表3 SDN和NFV分析

发展阶段	分离架构	流量引导	高级网络功能	可编程粒度
SDN	支持	支持	不支持	每流
NFV	支持	不支持	支持	每流
SDN+NFV	支持	支持	支持	每流

最近,ONF提出了“SDN+NFV”的思想,并发布了NFV与SDN融合方案^[61],采用SDN控制与转发分离以及集中控制的架构,为NFV提供了高效的网络连接方式。通过SDN技术可以将网络节点功能进一步拆分,使转发功能和业务功能分离,其中业务功能采用NFV的大容量服务器实现,转发功能采用OpenFlow交换机实现。该方案采用SDN思想构建可扩展、弹性的基础设施,在SDN架构的应用层引入了NFV编排功能,构建了电信级的可编程网络,实现了网络资源利用率最大化、网络配置管理简单化、网络升级更新快速化以及网络功能开放可编程化。

另外,OpenNF^[64]为网络中间件流量的重分配提供了高效的协同控制,其控制平面架构如图8所示。控制应用通过北向接口控制OpenNF控制器,同时监测网络功能运行状态,将流引导至所需的网络功能进行处理,并为每个网络功能实例提供所需的状态。OpenNF控制器由网络功能状态管理器和流管理器构成,其中网络功能状态控制器通

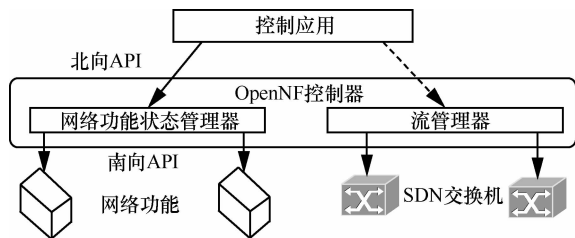


图 8 OpenNF 控制平面架构

过南向接口控制虚拟化网络功能,流管理器控制 SDN 交换机。当控制应用制定网络策略后,形成一条“网络功能+SDN 交换机”的服务链。网络功能状态管理器将策略规定的网络功能实例状态信息下发至网络功能,流管理器将流表下发至 SDN 交换机。SDN 交换机接收流后,根据流表规则判断流发送到网络功能还是发送到下一个 SDN 交换机,网络功能处理完后将流发送到下一个 SDN 交换机。直至网络策略规定的所有网络功能完成全部的流量处理。该架构在 SDN 控制器中增加了虚拟化网络功能状态管理的功能,并通过南向 API 连接虚拟化网络功能,充分利用了 SDN 控制平面与转发平面分离的优势;另一方面,在转发平面加入虚拟化网络功能,提供了高级处理功能,实现了网络功能的灵活部署和升级。

5 研究展望

网络可编程对网络架构和协议的实现产生重大的革命性影响。当前,实现网络可编程最具代表性的技术是 SDN 和 NFV,学术界和工业界对 SDN 南向接口的协议、北向接口的高级编程语言和网络操作系统都进行了大量的研究工作,但是网络可编程技术仍然面临很多困难和挑战。

(1) 灵活可编程的转发平面抽象机制

SDN 对网络的可控性归根到底取决于转发平面提供的编程灵活性。目前以 OpenFlow 为代表的 SDN 转发平面抽象机制仅以“Match-Action”方式对数据分组进行转发,无法支持更灵活的转发平面操作,比如基于逻辑运算的分组转发机制、有状态的分组计算等。如何进行更加合理、更加灵活的转发平面抽象,是 SDN 值得研究的重要问题。

(2) 控制逻辑一致性问题研究

控制逻辑集中和分布式状态管理提高网络的可编程性的同时,也带来了控制逻辑一致性问题。在控制平面,为了解决大规模网络中控制器单点性能瓶颈问题,采用多控制器管控的分布式控制机制,控制器之间同步全网络视图信息成为保证控制逻辑一致性的重要手段。在数据平面,

交换机失效和流表项频繁更新是分布式状态管理需要解决的问题。

(3) 高性能转发平面技术与实现

对于转发平面而言,不同高速硬件平台具有不同优势:如 FPGA 适合于简单、高性能的转发计算;TCAM 支持高速查找;而多核处理器具有灵活性强、支持复杂转发处理等特点。在 SDN 转发设备中,为了更好地满足不同网络应用的性能需求,硬件平台趋于异构化。如何利用异构芯片的性能实现高效灵活的数据转发,既需要支持高效查找、快速更新、低存储开销的算法,也需要基于流表特征提升实现技术。

(4) 大范围部署问题

目前,虽然 SDN 已经在部分数据中心网络中实践部署,但是其只能实现简单的分组转发控制,无法实现更加复杂的网络功能和应用。同样,尽管 NFV 思想使网络功能可编程的前景更加明朗,但其还处于理论探讨阶段,实际部署还需要研究探讨。因此,在大范围的网络中实现完全的可编程还面临很多严峻的挑战。

(5) 网络可编程带来的安全问题

网络可编程技术支持网络参与者(包括运营商、管理员、用户)定制网络应用的思想打破了网络本身的完整性,会增加网络安全危险,必须考虑如何控制用户的权限和行为。

6 结束语

网络可编程技术是实现网络定制化服务、集中化管理、递进式革新的关键技术之一,其内涵伴随网络技术的发展而不断变换,并在未来网络研究中占据了主要地位。作为目前网络可编程技术的典型代表,SDN 和 NFV 受到学术界和产业界的共同关注,学术界深入研究了 SDN 控制平面和转发平面的可编程关键技术以及与 NFV 的融合技术,产业界也正在筹划一场网络运营商市场的变革。要想实现网络可编程技术的广泛应用和部署,不仅需要克服很多技术难题,包括控制逻辑一致性、可编程灵活性、网络高级编程语言以及网络安全管理等关键性问题,同时需要虚拟化、网络操作系统、硬件开发等辅助性技术的支撑,更需要网络运营商、设备生产商、服务提供商的全力协同合作。

参考文献:

- [1] FEAMSTER N, REXFORD J, ZEGURA E. The road to SDN[J]. Queue, 2013, 11(12): 20-27.



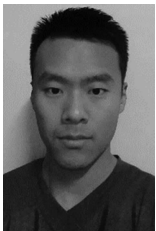
- [2] NUNES B A A, MENDONCA M, NGUYEN X N. A survey of software-defined networking: past, present and future of programmable networks [J]. IEEE Communications Surveys & Tutorials, 2014, 16(3): 1617-1634.
- [3] OpenFlow switch specification version 1.3.0 [EB/OL]. [2015-02-28]. <http://www.Open-network-ing.org>.
- [4] Network functions virtualisation: an introduction, benefits, enablers, challenges & call for action [EB/OL]. [2015-03-22]. http://portal.etsi.org/NFV/NFV_White_Paper.pdf.
- [5] CAMPBELL A T, KATZELA I, MIKI K, et al. Open signaling for ATM, internet and mobile networks [J]. ACM SIGCOMM Computer Communication Review, 1999, 29(2): 97-108.
- [6] Multiservice switching forum[EB/OL]. [2015-05-18]. <http://www.msforum.org/>.
- [7] DORIA A, HELLSTRAND F, SUNDELL K, et al. General switch management protocol [S/OL]. [2015-02-10]. <https://tools.ietf.org/html/draft-worster-gsmp-00>.
- [8] Optical internetworking forum [EB/OL]. [2015-05-14]. <http://www.oiforum.com/>.
- [9] TENNENHOUSE D L, SMITH J M, SINCOSKIE W D, et al. A survey of active network research [J]. Communications Magazine IEEE, 1997, 35(1): 80-86.
- [10] Elliott C. GENI: opening up new classes of experiments in global networking[J]. IEEE Internet Computing, 2010, 14(1): 39-42.
- [11] NSF future internet design [EB/OL]. [2015-05-08]. <http://www.nets-find.net/>.
- [12] GAVRAS A, KARILA A, FDIDA S, et al. Future internet research and experimentation: the FIRE initiative [J]. ACM SIGCOMM Computer Communication Review, 2007, 37(3): 89-92.
- [13] BHATTACHARJEE S, CALVERT K L, ZEGURA E W. An Architecture for Active Networking [M]. New York: Springer US, 1997.
- [14] SMITH J M, FARBER D J, GUNTER C A, et al. SwitchWare: accelerating network evolution (white paper) [R/OL]. [2015-05-29]. http://repository.upenn.edu/cis_reports/211.
- [15] WETHERALL D J, GUTTAG J V, TENNENHOUSE D L. ANTS: a toolkit for building and dynamically deploying network protocols [C]//The IEEE 1st Conference on Open Architectures and Network Programming, April 3-4, 1998, San Francisco, CA, USA. New Jersey: IEEE Press, 1998: 117-129.
- [16] Devolved control of ATM networks[EB/OL]. [2015-05-20]. <http://www.cl.cam.ac.uk/research/srg/netos/old-projects/dcan/#pub>.
- [17] RFC 5810 (Proposed Standard) [S/OL]. [2014-10-30]. http://wenku.baidu.com/link?Url=4kZCAq87BmnGJcjYDZnn751D2qLjzrHj0kKy9VhLSjraFEu00B9_RJQYB2Z-gm8jjcjJfhoe8EI4NTWrrJb8_sCCBCjyuYKKqFgJ449pNi.
- [18] REXFORD J, GREENBERG A, HJALMTYSSON G, et al. Network-wide decision making: toward a wafer-thin control plane [C]//The Third Workshop on Hot Topics in Networks, November 15-16, 2004, San Diego, CA, USA. [S.l.:s.n.], 2004: 59-64.
- [19] GREENBERG A, HJALMTYSSON G, MALTZ D A, et al. A clean slate 4d approach to network control and management [J]. ACM SIGCOMM Computer Communication Review, 2005, 35(5): 41-54.
- [20] RFC4741 (proposed standard)[S/OL]. [2015-04-30]. <http://www.rfc-editor.org/rfc/rfc4741.txt>.
- [21] RFC1157[S/OL]. [2015-05-30]. <http://www.faqs.org/rfcs/rfc1157.html>.
- [22] CASADO M, GARFINKEL T, AKELLA A, et al. SANE: a protection architecture for enterprise networks [C]//The 15th Conference on USENIX Security Symposium, July 3-August 4, 2006, Vancouver, B.C., Canada. New York: ACM Press, 2006: 137-151.
- [23] CASADO M, FREEDMAN M J, PETTIT J, et al. Ethane: taking control of the enterprise [J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4): 1-12.
- [24] GUDE N, KOPONEN T, PETTIT J, et al. Nox: towards an operating system for networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.
- [25] NOX[EB/OL]. [2015-03-15]. <http://noxrepo.org>.
- [26] The preliminary design and implementation of the maestro network control platform [EB/OL]. [2008-01-30].https://www.researchgate.net/publication/241342434_The_Preliminary_Design_and_Implementation_of_the_Maestro_Network_Control_Platform.
- [27] POX [EB/OL]. [2014-11-18]. <http://www.noxrepo.org/pox/about-pox/>.
- [28] Beacon[EB/OL]. [2015-01-10]. <http://www.beaconcontroller.net>.
- [29] Floodlight[EB/OL]. [2015-01-10]. <http://floodlight.openflowhub.org>.
- [30] Ryu[EB/OL]. [2015-03-15]. <http://osrg.github.com/ryu/>.
- [31] SNAC[EB/OL]. [2015-02-18]. <http://www.openflow.org/wp/snac>.
- [32] Trema[EB/OL]. [2015-04-13]. <http://trema.github.com/trema>.
- [33] BERDE P, GEROLA M, HART J, et al. ONOS: towards an open, distributed SDN OS [C]//The Third workshop on Hot Topics in Software Defined Networking, August 22, 2014, Chicago, IL, USA. New York: ACM Press, 2014: 1-6.
- [34] NASCIMENTO M R, ROTHENBERG C E, SALVADOR M R, et al. Virtual routers as a service: the RouteFlow approach leveraging software-defined networks[C]//The 6th International Conference on Future Internet Technologies, June 13-15, 2010, New York, NY, USA. New York: ACM Press, 2011: 34-37.
- [35] TOOTOONCHAIN A, GORBUNOV S, GANJALIY, et al. On controller performance in software-defined networks [C]//The 9th USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, April 24, 2012, Berkeley, CA, USA. New York: ACM Press, 2012:10.

- [36] KOPONEN T, CASADO M, GUDE N, et al. Onix: a distributed control platform for large-scale production networks [C]//The 9th USENIX Conference on Operating Systems Design and Implementation, October 4-6, 2010, Vancouver, BC, Canada. New York: ACM Press, 2010: 351-364.
- [37] MEDVED J, VARGA R, TKACIK A, et al. OpenDaylight: towards a model-driven SDN controller architecture [C]//The IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), June 19, 2014, Sydney, Australia. New Jersey: IEEE Press, 2014: 1-6.
- [38] Network functions virtualisation infrastructure architecture; architecture of infrastructure network domain [EB/OL]. [2015-05-10]. <http://www.etsi.org/standards-search>.
- [39] Network functions virtualisation (NFV): network operator perspectives on industry progress[EB/OL]. [2015-01-14]. http://portal.etsi.org/NFV/NFV_White_Paper2.pdf.
- [40] Network functions virtualisation infrastructure architecture; overview [EB/OL]. [2014-12-01]. <http://www.etsi.org/standards-search>.
- [41] Network functions virtualisation (NFV); virtual network functions architecture [EB/OL]. [2014-11-02]. <http://www.etsi.org/standards-search>.
- [42] RISSO F, MANZALINI A, NEMIROVSKY M. Some controversial opinions on software-defined data plane services[C]//IEEE SDN for Future Networks and Services (SDN4FNS), October 3-4, 2011, Trento, Italy. New Jersey: IEEE Press, 2013: 1-7.
- [43] LU G, SHI Y, GUO C, et al. CAFE: a configurable packet forwarding engine for data center networks [C]//The 2nd ACM SIGCOMM Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO), August 21, 2009, Barcelona, Spain. New York: ACM Press, 2009: 25-30.
- [44] ATTIG M, BREBNER G. 400Gb/s programmable packet parsing on a single FPGA[C]//The Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), October 3-4, 2011, Brooklyn, NY, USA. New Jersey: IEEE Press, 2011: 12-23.
- [45] LV G F, SUN Z G, LI T, et al. LabelCast: a general abstraction for the forwarding plane of SDN [J]. Chinese Journal of Computers, 2012, 35(10): 2037-2047.
- [46] DE CARLI L, PAN Y, KUMER A, et al. PLUG: flexible lookup modules for rapid deployment of new protocols in high-speed routers [J]. ACM SIGCOMM Computer Communication Review, 2009, 39(4): 207-218.
- [47] SONG H Y. Protocol oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane [C]//ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN), August 12-16, 2013, Hong Kong, China. New York: ACM Press, 2013: 127-132.
- [48] MEKKY H, HAO F, MUKHERJEE S, et al. Application-aware data plane processing in SDN[C]//The Third Workshop on Hot Topics in Software Defined Networking, August 22, 2014, Chicago, IL, USA. New York: ACM Press, 2014: 13-18.
- [49] BOSSHART P, GIBB G, KILM H S, et al. Forwarding metamorphosis: fast programmable match-action processing in hardware for SDN [C]//ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN), August 12-16, 2013, Hong Kong, China. New York: ACM Press, 2013: 99-110.
- [50] MOSHREF M, BHARGAVA A, GUPTA A, et al. Flow-level state transition as a new switch primitive for SDN [C]//The 2014 ACM Conference on SIGCOMM, August 17-22, 2014, Chicago, IL, USA. New York: ACM Press, 2014: 377-378.
- [51] KATTA N, ALIPOURFARD O, REXFORD J. Infinite CacheFlow in software-defined networks [C]//The 2014 ACM Conference on SIGCOMM, August 17-22, 2014, Chicago, IL, USA. New York: ACM Press, 2014.
- [52] MOGUL J C, CONGDON P. Hey, you darned counters!: get off my ASIC [C]//The First Workshop on Hot Topics in Software Defined Networks, August 13-17, 2012, Helsinki, Finland. New York: ACM Press, 2012: 25-30.
- [53] JEYAKUMAR V, ALIZADEH M, GENG Y, et al. Millions of little minions: using packets for low latency network programming and visibility [C]//The 2014 ACM Conference on SIGCOMM, August 17-22, 2014, Chicago, IL, USA. New York: ACM Press, 2014: 17-22.
- [54] LI Y, YAO G, BI J. Flowinsight: decoupling visibility from operability in SDN data plane [C]//The 2014 ACM Conference on SIGCOMM, August 17-22, 2014, Chicago, IL, USA. New York: ACM Press, 2014: 137-138.
- [55] FOSTER N, HARRISON R, FREEDMAN M J, et al. Frenetic: a network programming language [J]. ACM Sigplan Notices, 2011, 46(9): 279-291.
- [56] NetCore[EB/OL]. [2015-01-05]. <http://www.jgn.nict.go.jp/jgn2archive/english/jgn2inde.html>.
- [57] MONSANTO C, REICH J, FOSTER N, et al. Composing software defined networks [C]//The 10th USENIX Symposium on Networked Systems Design and Implementation, April 2-5, 2013, Lombard, IL, USA. New York: ACM Press, 2013: 1-13.
- [58] Nettle: functional reactive programming for OpenFlow networks [EB/OL]. [2015-01-30]. <http://haskell.cs.yale.edu/wp-content/uploads/2011/03/NettlePADL2011.pdf>.



- [59] VOELLMY A, KIM H, FEAMSTER N. Protera: a language for high-level reactive network control[C]//The first Workshop on Hot Topics in Software Defined Networks, June 15, 2012, Helsinki, Finland. New York: ACM Press, 2012: 43-48.
- [60] Software-defined networking: the new norm for networks [EB/OL]. [2015-04-25]. [https:// www. Opennetworking.org](https://www.opennetworking.org).
- [61] OpenFlow-enabled SDN and network functions virtualization[EB/OL]. [2014-10-09]. [https:// www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-sdn-nfv-solution. pdf](https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-sdn-nfv-solution.pdf).
- [62] QAZI Z A, TU C C, CHIANG L, et al. Simple-fying middlebox policy enforcement using SDN [C]//The ACM SIGCOMM 2013 Conference on SIGCOMM, August 12-16, 2013, Hong Kong, China. New York: ACM Press, 2013: 27-38.
- [63] GEMBER A, PRABHU P, GHADIYALI Z, et al. Toward software-defined middlebox networking [C]//The 11th ACM Workshop on Hot Topics in Networks, October 29, 2012, Seattle, WA, USA. New York: ACM Press, 2012: 7-12.
- [64] GEMBER A, VISWANATHAN R, PRAKASH C, et al. OpenNF: enabling innovation in network function control [C]//ACM Conference on Sigcomm, August 19, 2014, Chicago, USA. New York: ACM Press, 2014: 163-174.
- [65] MARTINS J, AHMED M, RAICIU C, et al. ClickOS and the art of network function virtualization [C]//The 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), April 2-4, 2014, Seattle, WA, USA. New York: ACM Press, 2014: 459-473.

[作者简介]



谢立军(1988-),男,国家数字交换系统工程技术研究中心博士生,主要研究方向为新型网络体系结构。



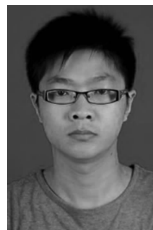
胡宇翔(1982-),男,博士,国家数字交换系统工程技术研究中心讲师,主要研究方向为新型网络体系结构。



汪斌强(1963-),男,博士,国家数字交换系统工程技术研究中心教授,主要研究方向为新型网络体系结构。



兰巨龙(1962-),男,博士,国家数字交换系统工程技术研究中心教授,主要研究方向为新型网络体系结构。



程国振(1986-),男,博士,国家数字交换系统工程技术研究中心讲师,主要研究方向为新型网络体系结构。



王志明(1986-),男,国家数字交换系统工程技术研究中心博士生,主要研究方向为新型网络体系结构。