

## 软件定义网络(SDN)研究进展\*

张朝昆, 崔勇, 唐嵩祎, 吴建平

(清华大学 计算机科学与技术系, 北京 100084)

通讯作者: 崔勇, E-mail: cuiyong@tsinghua.edu.cn

**摘要:** 网络抽象促使软件定义网络(software-defined networking, 简称 SDN)的产生. SDN 将数据平面与控制平面解耦合, 简化了网络管理. 首先从 SDN 诞生发展的背景入手, 梳理了 SDN 的体系结构, 包括数据层、控制层和应用层, 并按照 SDN 的层次结构深入阐述其关键技术, 特别分析了一致性、可用性和容错性等特性. 然后, 论述了 SDN 在不同应用场景下的最新研究成果. 最后, 展望未来研究工作.

**关键词:** 软件定义网络; 接口; 交换机; 控制器; 网络抽象

**中图法分类号:** TP393

中文引用格式: 张朝昆, 崔勇, 唐嵩祎, 吴建平. 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>

英文引用格式: Zhang CK, Cui Y, Tang HY, Wu JP. State-of-the-Art survey on software-defined networking (SDN). Ruan Jian Xue Bao/Journal of Software, 2015, 26(1): 62-81 (in Chinese). <http://www.jos.org.cn/1000-9825/4701.htm>

### State-of-the-Art Survey on Software-Defined Networking (SDN)

ZHANG Chao-Kun, CUI Yong, TANG He-Yi, WU Jian-Ping

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

**Abstract:** Network abstraction brings about the naissance of software-defined networking. SDN decouples data plane and control plane, and simplifies network management. The paper starts with a discussion on the background in the naissance and developments of SDN, combing its architecture that includes data layer, control layer and application layer. Then their key technologies are elaborated according to the hierarchical architecture of SDN. The characteristics of consistency, availability, and tolerance are especially analyzed. Moreover, latest achievements for profiled scenes are introduced. The future works are summarized in the end.

**Key words:** SDN (software-defined networking); interface; switch; controller; network abstraction

传统网络的层次结构是互联网取得巨大成功的关键. 但是随着网络规模的不断扩大, 封闭的网络设备内置了过多的复杂协议, 增加了运营商定制优化网络的难度, 科研人员无法在真实环境中规模部署新协议. 同时, 互联网流量的快速增长(预计到 2018 年, 全球流量将达到  $1.6 \times 10^{21}$  字节<sup>[1]</sup>), 用户对流量的需求不断扩大, 各种新型服务不断出现, 增加了网络运维成本.

SDN 起源于 2006 年斯坦福大学的 Clean Slate 研究课题<sup>[2]</sup>. 2009 年, Mckeown 教授正式提出了 SDN 概念<sup>[3]</sup>. 利用分层的思想, SDN 将数据与控制相分离. 在控制层, 包括具有逻辑中心化和可编程的控制器, 可掌握全局网络信息, 方便运营商和科研人员管理配置网络和部署新协议等. 在数据层, 包括哑的(dumb)交换机(与传统的二层交换机不同, 专指用于转发数据的设备). 交换机仅提供简单的数据转发功能, 可以快速处理匹配的数据包, 适应流量日益增长的需求. 两层之间采用开放的统一接口(如 OpenFlow<sup>[4]</sup>等)进行交互. 控制器通过标准接口向交

\* 基金项目: 国家自然科学基金(61120106008, 61161140454); 国家高技术研究发展计划(863)(2013AA010401); 国家下一代互联网示范工程(CNGI-12-03-003)

收稿时间: 2014-02-28; 修改时间: 2014-04-19; 定稿时间: 2014-07-19; jos 在线出版时间: 2014-08-19

CNKI 网络优先出版: 2014-08-19 14:16, <http://www.cnki.net/kcms/doi/10.13328/j.cnki.jos.004701.html>

交换机下发统一标准规则,交换机仅需按照这些规则执行相应的动作即可。因此,SDN 技术能够有效降低设备负载,协助网络运营商更好地控制基础设施,降低整体运营成本,成为最具前途的网络技术之一。因此,SDN 被 MIT 列为“改变世界的十大创新技术之一”<sup>[5]</sup>,SDN 相关技术研究迅速开展起来,成为近年来的研究热点。2013 年, SIGCOMM 会议收录了多篇相关文章,甚至将 SDN 列为专题来研讨,带动了 SDN 相关研究的蓬勃发展。

本文第 1 节首先论述 SDN 诞生与发展的背景,以此提出 SDN 分层的体系结构,并讨论 SDN 开放式接口。第 2 节和第 3 节分别对 SDN 数据层和控制层的关键技术进行详细阐述。第 4 节论述 SDN 在不同场景下的应用。第 5 节针对未来工作提出见解。最后给出本文结论。

## 1 SDN 体系结构

SDN 是当前最热门的网络技术之一,它解放了手工操作,减少了配置错误,易于统一快速部署。本节首先论述了 SDN 诞生发展的背景,指出 SDN 的产生及快速发展的必然性定律。接下来探讨了 SDN 主流的体系结构。由于 SDN 标准接口机制确保层次之间既保持相对独立,又能正常通信,因此,标准接口设计的好坏是 SDN 成功设计的关键。

### 1.1 SDN 诞生发展的背景

随着网络的快速发展,传统互联网出现了如传统网络配置复杂度高等诸多问题<sup>[6]</sup>,这些问题说明网络架构需要革新,可编程网络的相关研究为 SDN 的产生提供了可参考的理论依据<sup>[7]</sup>。主动网络<sup>[8,9]</sup>允许数据包携带用户程序,并能够由网络设备自动执行。用户可以通过编程方式动态地配置网络,达到了方便管理网络的目的。然而由于需求低、协议兼容性差等问题,并未在工业界实际部署。4D 架构<sup>[10,11]</sup>将可编程的决策平面(即控制层)从数据平面分离,使控制平面逻辑中心化与自动化,其设计思想产生 SDN 控制器的雏形<sup>[12]</sup>。

借鉴计算机系统的抽象结构,未来的网络结构将存在转发抽象、分布状态抽象和配置抽象这 3 类虚拟化概念<sup>[13]</sup>。转发抽象剥离了传统交换机的控制功能,将控制功能交由控制层来完成,并在数据层和控制层之间提供了标准接口,确保交换机完成识别转发数据的任务。控制层需要将设备的分布状态抽象成全网视图,以便众多应用能够通过全网信息进行网络的统一配置。配置抽象进一步简化了网络模型,用户仅需通过控制层提供的接口对网络进行简单配置,就可自动完成沿路径转发设备的统一部署。因此,网络抽象思想解耦了路径依赖,成为数据控制分离且接口统一架构(即 SDN)产生的决定因素。

此外,众多标准化组织已经加入到 SDN 相关标准的制订当中。专门负责制订 SDN 接口标准的著名组织是开放网络基金会(Open Networking Foundation,简称 ONF)<sup>[14]</sup>,该组织制订的 OpenFlow 协议业已成为 SDN 接口的主流标准,许多运营商和生产厂商根据该标准进行研发。互联网工程任务组(Internet Engineering Task Force,简称 IETF)的 ForCES 工作组<sup>[15]</sup>、互联网研究专门工作组(Internet Research Task Force,简称 IRTF)的 SDNRG 研究组<sup>[16]</sup>以及国际电信联盟远程通信标准化组织(ITU Telecommunication Standardization Sector,简称 ITU-T)的多个工作组<sup>[17]</sup>同样针对 SDN 的新方法和新应用等展开研究。标准化组织的跟进,促使了 SDN 市场的快速发展。据悉,SDN 市场已于 2013 年达到约 2 亿美元的产值,预计到 2016 年将达到 20 亿美元<sup>[18]</sup>,市场需求确保 SDN 有足够的发展空间。由此可见,SDN 具有广阔的发展前景和巨大的研究价值。

### 1.2 体系结构概述

针对不同的需求,许多组织提出了相应的 SDN 参考架构。SDN 架构<sup>[19]</sup>最先由 ONF 组织提出,并已经成为学术界和产业界普遍认可的架构。除此之外,欧洲电信标准化组织(European Telecommunications Standards Institute,简称 ETSI)提出的 NFV 架构<sup>[20]</sup>随之发展起来,该体系结构主要针对运营商网络,并得到了业界的支持。由各大设备厂商和软件公司共同提出了 OpenDaylight<sup>[21]</sup>,目的是为了具体实现 SDN 架构,以便于实际部署。

ONF 组织最初在白皮书中提到 SDN 体系结构<sup>[19]</sup>,并于 2013 年底发布最新版本<sup>[22]</sup>,其架构如图 1 所示。SDN 由下到上(或称由南向北)分为数据平面、控制平面和应用平面。数据平面与控制平面之间利用 SDN 控制数据平面接口(control-data-plane interface,简称 CDPI)进行通信,CDPI 具有统一的通信标准,目前主要采用 OpenFlow 协

议<sup>[4]</sup>.控制平面与应用平面之间由 SDN 北向接口(northbound interface,简称 NBI)负责通信,NBI 允许用户按实际需求定制开发.

数据平面由交换机等网络元素组成,各网络元素之间由不同规则形成的 SDN 网络数据通路形成连接.控制平面包含逻辑中心的控制器,负责运行控制逻辑策略,维护着全网视图.控制器将全网视图抽象成网络服务,通过访问 CDPI 代理来调用相应的网络数据通路,并为运营商、科研人员及第三方等提供易用的 NBI,方便这些人员订制私有化应用,实现对网络的逻辑管理.应用平面包含着各类基于 SDN 的网络应用,用户无需关心底层设备的技术细节,仅通过简单的编程就能实现新应用的快速部署.CDPI 负责将转发规则从网络操作系统发送到网络设备,它要求能够匹配不同厂商和型号的设备,而并不影响控制层及以上的逻辑.NBI 允许第三方开发个人网络管理软件和应用,为管理人员提供更多的选择.网络抽象特性允许用户可以根据需求选择不同的网络操作系统,而并不影响物理设备的正常运行.

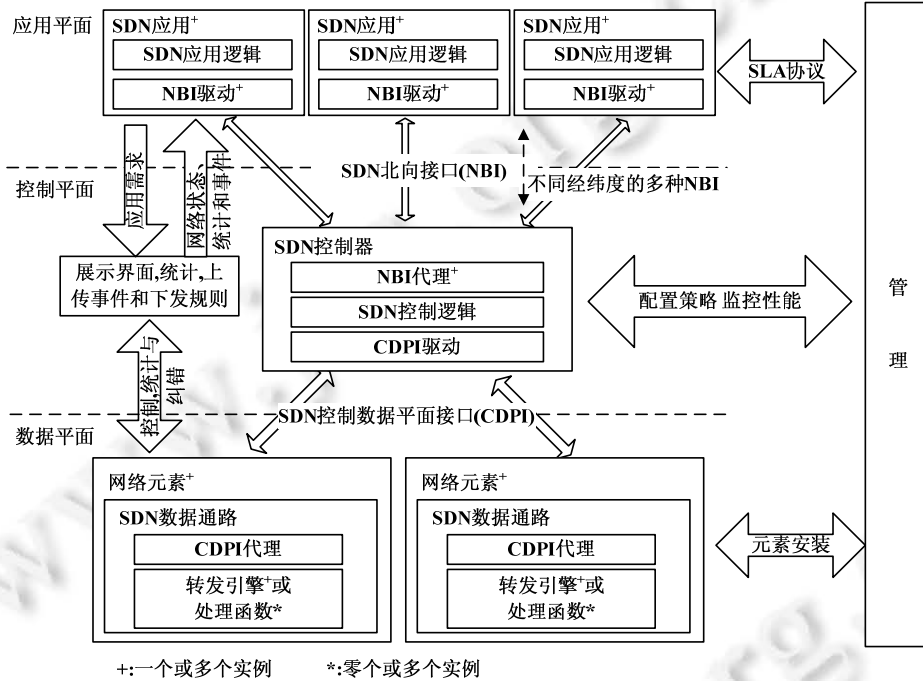


Fig.1 SDN architecture

图 1 SDN 体系结构

NFV<sup>[20]</sup>是针对运营商网络出现的问题而提出的 SDN 解决方案.网络运营商的网络由专属设备来部署,随着各种新型网络服务的产生,这些专属设备功能变得繁杂,而管理这些繁杂的硬件设备造成运营成本及能耗的增加,从而导致运营商网络的发展遇到瓶颈.针对上述问题,NFV 将传统网络设备的软件与硬件相分离,使网络功能更新独立于硬件设备.为此,NFV 采用了资源虚拟化的方式,在硬件设备中建立一个网络虚拟层,负责将硬件资源虚拟化,形成虚拟计算资源、虚拟存储资源和虚拟网络资源等,运营商通过软件来管理这些虚拟资源.由于采用的是通用硬件设备,NFV 降低了设备成本,减少了能耗,缩短了新网络服务的部署周期,从而适应网络运营商的发展需求.在接口设计方面,NFV 既可以基于非 OpenFlow 协议,又能与 OpenFlow 协同工作,同时还支持 ForCES<sup>[16]</sup>等多种传统接口标准化协议,以便适应网络运营商对设备的不同需求,并与 ONF 的 SDN 保持相对独立的发展.

OpenDaylight<sup>[21]</sup>的目标是通过 SDN 的开源开发,推进业界可部署方案具体实施,其架构由设备厂商提出并得到众多 IT 软件厂商的支持.考虑到兼容性问题,OpenDaylight 继承了 SDN 架构形式,同时又结合了 NFV 的特

点.架构共分为 3 个层次,分别是网络应用与业务流程(即应用层)、控制平台(即控制层)和物理与虚拟网络设备(即数据层).OpenDaylight 的控制平台直接由自带的 Java 虚拟机实现.针对不同的网络任务,控制器自身携带了一系列可插入模块,并兼容第三方模块以增强 SDN 的功能.与 ONF 的 SDN 架构最大的不同在于:OpenDaylight 控制器的南向接口除了支持 OpenFlow 协议之外,还支持 NETCONF<sup>[23]</sup>等配置协议和 BGP<sup>[24]</sup>等路由协议,并支持生产厂商的专有协议(如思科的 OnePK 协议<sup>[25]</sup>).为了能够处理不同的标准协议,OpenDaylight 增加了服务抽象层 SAL,它负责将不同的底层协议标准转换成 OpenDaylight 控制层所理解的请求服务,保持了底层协议的透明性,并提高了整体架构的可扩展性.

SDN,NFV 和 OpenDaylight 的对比见表 1.由于 NFV 与 ONF 的 SDN 分别负责不同的网络,两种架构的协同工作能够获得更好的网络体验,将两者结合可以降低设备成本.通过利用通用交换机等设备和软件代替原有设备,使得设备的升级与网络应用的拓展相对独立.OpenDaylight 具有开源性,因此,它可以兼容 SDN,NFV 以及未来与 SDN 并行的体系结构.总之,无论是哪个组织提出的 SDN 体系结构,实现的目标是一致的.SDN 使得数据控制相分离的网络具有开放性和可编程性,科研人员及运营商可以通过 PC 机、手机、Web 网页或未来可能出现的各种途径进行网络部署,而部署工作也仅是应用软件的简单开发或配置.可以预见:针对 SDN 并行架构的研究,是未来研究进展的重要趋势之一.

Table 1 Comparison of SDN, NFV and OpenDaylight

表 1 SDN,NFV 和 OpenDaylight 的对比

体系结构	接口标准	与 SDN 兼容性	相关背景介绍			特点
			应用领域	发起人	组织者	
ONF SDN <sup>[22]</sup>	OpenFlow	-	校园网,企业网,数据中心	高校科研人员	ONF	强调控制与数据分离等
NFV <sup>[20]</sup>	多种接口协议,支持可扩展	可协同工作	运营商网络	运营商	ETSI NFV 工作组	强调网络功能虚拟化
OpenDaylight <sup>[21]</sup>	多种接口协议,支持可扩展	完全支持 OpenFlow	未来的互联网	设备生产厂商和 IT 软件厂商	OpenDaylight (Linux 基金)	强调软件开源及实现

### 1.3 开放式接口与协议设计

SDN 中的接口具有开放性,以控制器为逻辑中心,南向负责与数据层通信,北向负责与应用层通信.此外,由于单一控制机制容易造成控制节点失效,严重影响性能,可采用多控制器方式<sup>[26]</sup>,此时,多控制器之间将采用东西向通信方式.开放式接口的研究,必然进一步推动 SDN 的深入发展.

在这些开放式接口研究中,控制器南向接口作为数据与控制分离的核心被广泛研究,成为业界关注的焦点.由于控制层与数据层解耦,使得针对这两层的改进相对独立,在层与层之间仅需提供标准南向接口即可.南向接口是 SDN 分层架构的关键元素,然而逻辑上,它既要保证数据层与控制层之间的正常通信,又要支持两层独立更新;物理上,设备生产厂商需要开发支持这种标准接口的设备,因为传统网络设备是不能在 SDN 网络之中运行的.因此,研发南向标准接口成为 SDN 基础研究中的重要内容之一.

许多组织着手制订南向标准接口.ONF 提出的 CDPI 成为了主流南向接口,它采用 OpenFlow<sup>[4]</sup>协议.OpenFlow 是 SDN 中第一个广泛使用的数据控制层接口协议,得到学术界普遍关注<sup>[27-29]</sup>,它将单一集成和封闭的网络设备成为灵活可控的通信设备.OpenFlow 协议是基于流的概念来匹配规则的,因此,交换机需要维护一个流表(flow table)来支持 OpenFlow,并按流表进行数据转发,而流表的建立、维护及下发均由控制器来完成.

为了便于设备生产厂商开发支持 OpenFlow 的设备,ONF 开始提供 OpenFlow 协议标准<sup>[30]</sup>.OpenFlow 1.0.0 规定流表头为 12 元组(如源/目的 IP 地址、源/目的 MAC 地址等),在一定程度上满足了用户对 SDN 网络的需求.然而,1.0.0 版本还不完善,如支持的规则和动作过少、仅支持单表、无关联动作的组合容易造成组合爆炸等问题.因此,OpenFlow 1.1.0 增加了部分规则,并开始支持多级流表、群组表和动作集等功能.IPv6 是下一代互联网的核心元素,因此从 1.2 版本开始,OpenFlow 增加了对 IPv6 源/目的地址的支持.网络拥塞一直是传统互联网需要面临的问题之一,在 SDN 网络中也是如此,因此,从 1.3 版本开始支持流控机制.在 1.4.0 版本中,OpenFlow



































