

“十二五”
国家重点图书出版规划项目

学术中国·院士系列

ACADEMIC
CHINA

软件定义网络 核心原理与应用实践

(第二版) 上册

■ 黄韬 刘江 魏亮 张娇 杨帆 刘韵洁 著

SDN Core Principles and
Application Practice

(2nd Edition) Volume 1



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

“十二五”
国家重点图书出版规划项目

学术中国·院士系列

ACADEMIC
CHINA

软件定义网络 核心原理与应用实践

(第二版)上册

■ 黄韬 刘江 魏亮 张娇 杨帆 刘韵洁 著

SDN Core Principles and
Application Practice

(2nd Edition) Volume 1

人民邮电出版社
北京

图书在版编目 (C I P) 数据

软件定义网络核心原理与应用实践. 上册 / 黄韬等
著. — 2版. — 北京: 人民邮电出版社, 2016. 9
(学术中国·院士系列)
ISBN 978-7-115-43004-5

I. ①软… II. ①黄… III. ①计算机网络—研究
IV. ①TP393

中国版本图书馆CIP数据核字(2016)第194786号

内 容 提 要

本系列丛书包含核心原理和应用实践上下两册, 对软件定义网络(SDN)技术进行了全面剖析和深入解读。本书属于核心原理部分, 首先阐述了SDN的设计思想与体系架构, 详细分析了控制转发分离和开放可编程两个重要属性; 其次介绍了以OpenFlow为代表的SDN南向接口协议, 以及北向和东西向接口协议; 接下来根据SDN的层次化架构, 自下而上依次介绍了SDN数据平面、控制平面和应用案例; 最后梳理总结了SDN标准化和学术进展情况。

本书涉及内容广泛, 较为全面地阐述了SDN的核心原理和基本概念, 对从事SDN技术研发的专业人士、网络运营管理人员、相关专业的高校学生以及对SDN技术感兴趣的读者, 都具有一定的参考价值。

◆ 著 黄 韬 刘 江 魏 亮 张 娇 杨 帆

刘韵洁

责任编辑 代晓丽

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

固安县铭成印刷有限公司印刷

◆ 开本: 700×1000 1/16

印张: 15.5

2016年9月第2版

字数: 304千字

2016年9月河北第1次印刷

定价: 78.00元

读者服务热线: (010)81055488 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前 言

软件定义网络 (Software Defined Networking, SDN) 正在成为行业注目的焦点,越来越多的业界专家相信它将推动传统网络的发展与变革。但是,究竟什么是 SDN? 为什么需要 SDN? 它将对现有网络架构产生何种影响? 这些问题的背后,是大家对网络技术发展的思考与期许。

当前正在运行的互联网体系架构已经有超过 40 年的历史,随着网络规模的急剧膨胀和业务类型的不断丰富,互联网的结构和功能也日趋复杂,网络管控难度日渐增加,网络新功能难以快速部署。这促使人们重新思考网络体系架构的设计,而此时 SDN 的提出与兴起为未来网络的发展提供了一个新的可行方向。

SDN 之所以是一种革新性的技术,是因为它打破了传统网络架构的设计理念,一方面实现了控制平面与数据平面相分离;另一方面开放了网络可编程能力,从而提高了网络的灵活性和可管控性。此外,SDN 运营建立在开放软件的基础上,可以显著降低业务部署和维护成本。

本系列丛书期望对当前 SDN 的概念定义、核心原理、关键技术和部署应用几方面进行较为全面的介绍,具体包含核心原理和应用实践上下两册。本册属于核心原理部分,重点介绍了 SDN 的体系架构和主流南向接口协议 OpenFlow,并简要介绍了 SDN 的其他拓展思路和相关成果。具体章节内容介绍如下。

第 1 章通过讲述互联网的发展历程,介绍了 SDN 技术的产生背景,使读者对 SDN 的兴起有一个大致的了解。

第 2 章介绍了 SDN 的基本原理,重点阐述了 ONF 组织定义的 SDN 体系架构,通过回顾发展历史,详细论述了软件定义网络的数控分离和可编程性这两大突出属性,并解读了这两个属性对网络技术发展的重要意义。

第 3 章介绍了 SDN 南向、北向以及东西向接口等相关内容,重点分析了发展较为成熟的 OpenFlow 协议及其伴侣协议 OF-CONFIG,同时对 XMPP、PCEP、OVSDB 等南向接口协议和 SDN 北向接口进行了简单介绍,并对东西向接口的进展和发展趋势进行了分析。

第4~6章按照SDN层次化架构，分别从数据平面（交换设备）、控制平面（控制器）和业务应用（SDN应用案例）3个层面剖析了支撑SDN架构的关键元素。通过分析SDN交换设备架构，重点介绍了SDN芯片和SDN软/硬件交换机。在解读SDN控制器体系架构的同时，对开源和商用控制器进行了详细介绍。在此基础上，介绍了多个方面的SDN应用案例，阐释了在应用场景中引入SDN的策略和方法，帮助读者了解SDN多元的发展方向和前景。

第7章主要介绍了SDN在技术标准化和学术方面的进展，详细梳理了多个标准化组织对SDN相关技术标准的工作开展情况，并分类概述了SDN在相关领域的学术研究成果，使读者充分感受到SDN技术广阔的发展前景。

为便于读者检索，本书在附录中给出了SDN相关缩略语、名词索引以及有关SDN资源网址列表。

本书参与撰写和审校的人员包括：北京邮电大学的汪硕、张晨、李呈、丁健、王健、胡文博、俞淑妍、张健男、谢俊峰、肖海洋、杨潇、顾莹、于洁、张丽、刘娟、晏思宇、侯乐、张歌、胡晓露、蒋韵、曹睿、马宁宁、毛健炜、师圣蔓、王泽南、王领、张航、张然、辛远铭、吴理炫、耿直、李婕妤、吴畏虹、邹贵今、王彬、尹弼柏，以及江苏省未来网络研究院的方辉、冀烨、周洁、魏静波、邓晓涛、周洪利、张欣慰、陈俊霞、张小雅等。在此对大家表示衷心感谢。

感谢相关企业给予的大力支持与帮助，特别感谢中国联通研究院张云勇副院长、房秉毅博士、徐雷博士，盛科公司孙剑勇总经理、张卫峰总监，Pica8公司杨勇涛经理在本书编写过程中提出的诸多宝贵建议。

同时还要感谢北京邮电大学SDN试验平台小组高志鹏、赵钦、胡铮、李慧、乔秀全、孙咏梅等几位老师，正是由于大家的共同努力，为本书第6章试验平台部分提供了切实有力的支撑。

最后，感谢人民邮电出版社的大力支持和高效工作，使本书能尽早与读者见面。

本书内容是作者所在团队科研过程中一些实际经验的总结，希望能够对读者有所帮助。由于作者水平所限，同时SDN技术仍处于快速发展之中，因此书中难免存在不少疏漏，真诚地企盼读者批评指正。

作 者

2016年5月

目 录

第 1 章 SDN 概述	1
第 2 章 SDN 基本原理	8
2.1 SDN 定义与架构	8
2.1.1 SDN 定义	8
2.1.2 SDN 架构	10
2.2 数据控制分离	14
2.2.1 基本概念	14
2.2.2 数据控制分离历史	18
2.2.3 SDN 数据控制分离	21
2.3 网络可编程	25
2.3.1 基本概念	25
2.3.2 网络可编程历史	25
2.3.3 SDN 可编程	28
2.4 本章小结	33
参考文献	34
第 3 章 SDN 接口协议	36
3.1 OpenFlow 协议	36
3.1.1 OpenFlow v1.0 协议	37
3.1.2 OpenFlow 协议的演进	47
3.1.3 OpenFlow 协议面临的问题	57
3.2 OF-CONFIG 协议	59
3.2.1 协议框架	59

3.2.2 设计需求	60
3.2.3 数据模型	62
3.2.4 协议演进	64
3.3 其他 SDN 南向接口协议	64
3.3.1 XMPP	64
3.3.2 PCEP	66
3.3.3 I2RS	67
3.3.4 OpFlex	69
3.3.5 OVSDB Mgmt	70
3.3.6 POF	70
3.3.7 P4	75
3.4 SDN 北向接口	79
3.4.1 ONF 北向接口	80
3.4.2 SDN 其他北向接口	82
3.4.3 北向接口的趋势与前景	83
3.5 SDN 东西向接口协议	84
3.6 本章小结	88
参考文献	88

第4章 SDN 数据平面

4.1 数据平面架构	90
4.2 SDN 芯片	94
4.3 SDN 硬件交换机	98
4.3.1 基于 ASIC 芯片的 SDN 品牌交换机	99
4.3.2 基于 ASIC 芯片的 SDN 白盒交换机	106
4.3.3 基于 NP 的 SDN 交换机	110
4.3.4 基于 NetFPGA 的 SDN 交换机	111
4.4 SDN 软件交换机	113
4.4.1 Open vSwitch	113
4.4.2 Pantou	114
4.4.3 Indigo	114
4.4.4 LINC	115
4.4.5 OpenFlowClick	116
4.4.6 OF13SoftSwitch	117
4.5 其他 SDN 业界相关进展	117

4.5.1 DPDK	117
4.5.2 FD.io	119
4.6 本章小结	121
参考文献	122
第5章 SDN 控制平面	124
5.1 SDN 控制器	124
5.1.1 SDN 控制器体系架构	124
5.1.2 SDN 控制器评估要素	128
5.2 开源控制器	130
5.2.1 NOX/POX	131
5.2.2 Ryu	133
5.2.3 Floodlight	134
5.2.4 OpenDaylight	136
5.2.5 ONOS	139
5.2.6 OpenContrail	141
5.2.7 其他开源控制器	144
5.3 商用控制器	146
5.3.1 Big Network Controller	146
5.3.2 XNC	147
5.3.3 SDN Contrail	149
5.3.4 Agile Controller	150
5.3.5 ZENIC	151
5.4 本章小结	153
参考文献	153
第6章 SDN 应用案例	155
6.1 SDN 在网络试验平台中的应用	155
6.1.1 GENI OpenFlow	156
6.1.2 OFELIA	158
6.1.3 RISE	162
6.1.4 TWAREN	163
6.1.5 FINE	165
6.1.6 C-Lab	166
6.2 SDN 在网络虚拟化中的应用	170

6.2.1 基于 SDN 的网络虚拟化	170
6.2.2 网络虚拟化平台 FlowVisor	172
6.2.3 网络虚拟化平台 OpenVirteX	176
6.2.4 网络虚拟化平台 CNVP	180
6.3 SDN 在流量工程中的应用	184
6.3.1 B4 概述	185
6.3.2 B4 设计思路	186
6.3.3 B4 核心网络功能	189
6.3.4 基于 SDN 的流量工程实现	193
6.3.5 B4 的部署与成效	196
6.4 本章小结	199
参考文献	200
第 7 章 SDN 标准化与学术进展	202
7.1 SDN 标准化进展	202
7.1.1 开放网络基金会	202
7.1.2 互联网工程任务组	205
7.1.3 国际电信联盟	207
7.1.4 欧洲电信标准化协会	209
7.1.5 中国通信标准化协会	213
7.2 SDN 学术进展	216
7.2.1 控制平面可扩展性	216
7.2.2 交换机/数据平面设计	219
7.2.3 SDN 编程语言	220
7.2.4 测试、调试及管理	222
7.2.5 数据中心应用	223
7.2.6 校园网应用	224
7.2.7 其他	225
7.3 本章小结	226
参考文献	227
缩略语	232
名词索引	238
SDN 资源网站	240

第1章

SDN 概述

软件定义网络（Software Defined Networking, SDN）是近几年来受到广泛关注的一个技术理念。本章通过回顾整个互联网的发展历程来分析 SDN 诞生的背景，以及它之所以受到学术界和产业界如此强烈关注的原因，使读者对 SDN 技术有一个感性的认识；然后在第 2 章中再详细阐释 SDN 的具体定义，从而使读者能够对 SDN 有更深入的理解。

谈到互联网的发展历史，我们可以从任何一本关于计算机网络的课本中找到，下面仅对互联网发展历程中具有代表性的一些重要事件做简要介绍。在 20 世纪 50 年代，随着计算机技术和通信技术的快速发展，科学家们认识到通过组建连通性的网络，能够实现不同计算机用户之间的远距离通信，这促进了学术界开始对数据分组交换、分布式网络、排队论等一系列技术展开探索和研究。1962 年，Leonard Kleinrock 在麻省理工学院的博士论文中首次提出分组交换概念，该技术后来成为互联网的标准通信方式。在 20 世纪 60 年代特定的“冷战”历史背景下，美国国防部于 1969 年启动了计算机网络开发计划 ARPANET（Advanced Research Projects Agency Network），这一网络最初连接了加利福尼亚大学洛杉矶分校、斯坦福大学、加州大学圣塔芭芭拉分校以及犹他州大学的几台大型计算机，虽然开始时规模很小，但是 ARPANET 的成功运行使计算机网络的概念发生了根本变化，同时标志着现代互联网的诞生。1974 年，国际标准化组织（International Organization for Standardization, ISO）发布了著名的 ISO/IEC 7498 标准，它首次提出并定义了网络分层模型设计思想，也就是我们所熟知的 OSI（Open System Interconnection，开放式系统互连）7 层参考模型，同年 12 月，斯坦福大学的 Vinton G. Cerf 和 Robert E. Kahn 一起领导的研究小组提出了著名的 TCP/IP（Transmission Control Protocol/Internet Protocol），实现计算机网络互连的思想，以互连具有不同协议的网络，从而使得构建大规模数

据分组网络成为可能。1983 年，ARPANET 宣布将过去的通信协议——NCP（Network Control Protocol）向 TCP/IP 过渡。1984 年，欧洲粒子物理研究实验室（CERN）的 Tim Berners Lee 博士为解决由于 CERN 主机不兼容而无法共享文件的问题，提出了开发一个分布式系统的设想，1991 年夏，Tim 利用 HTML（HyperText Markup Language，超文本标记语言）、HTTP（HyperText Transfer Protocol，超文本传输协议）成功编制了第一个局部存取浏览器 Enguire，从此 Web 应用开始起飞，随后通过不断演进最终形成了著名的万维网（World Wide Web，WWW）技术。1996 年，随着万维网的大规模应用，Internet（互联网）一词广泛流传。之后 10 年，互联网基于细腰的设计理念成功容纳了各种不同的底层网络技术和丰富的上层应用，迅速风靡全世界。

可以看到，互联网最初设计目标是要把分散的计算机连接起来以达到资源共享的目的。在互联网发展的初期，一所大学、一个研究机构或者企业都能够把自己的计算机资源通过网络组织起来形成私有的网络，这个网络只是为了自己的小规模研究或者给员工提供辅助性服务。例如，在一些中小企业内部往往构建一个小规模专用机房，为员工提供邮件、数据库以及其他一些功能型应用服务，这些应用服务的主要服务对象是公司的内部人员，通常服务的规模很小，因此，各种设备只需要少量的运维管理人员就可以很好地进行管理。但是随着企业规模逐渐增大，以及对外提供服务的增加，原来专用的小型机房难以满足日益增长的需求了，这就需要企业对其进行扩容或扩建，当然这将会使得机房的运维管理成本急剧上升。为应对这一问题，互联网数据中心（IDC）应运而生，业界开始考虑把各种业务托管到数据中心进行统一管理，进而有效降低企业的 IT 运维成本。

随着业务的增长和数据中心的不断扩大，服务器的计算能力与最初的计算机相比提高了成百上千倍，但计算资源往往难以得到充分的利用，一台服务器如果只处理少量任务，或者有大量空闲时间没有操作系统的任务执行，显然是对计算资源的一种浪费，于是，有研究者提出了一种新的技术设想——虚拟化。虚拟化技术的核心思想是希望能够实现在一台真实的物理机器上创建多个虚拟的逻辑主机，每个逻辑主机有自己虚拟的网卡、主板等虚拟设备，这些虚拟设备能够高效的复用真实物理设备，从而极大地提高物理基础设施资源的利用率。同时，为便于用户使用各逻辑资源，还特别强调灵活性和隔离性，即各台逻辑主机可以支持不同的操作系统，能够并行运行且互不干扰。正是由于上述这些技术特点，在随后几年中，虚拟化技术凭借种种优势逐步成为了信息技术（IT）领域持续关注的热点之一。

事实上，虚拟化技术在充分利用物理资源的同时带来了许多其他好处。例如，IT 管理部门可以暂停虚拟机，通过复制文件把任意应用轻松地在一台物理主机上移动到另一台主机上，并可以通过拷贝一套虚拟机的文件来快速建立一台新的虚拟机。正是通过这种灵活性，使网络管理人员能够方便地根据用户访问需求、流

量负载情况以及电力制冷甚至经济杠杆等因素将虚拟机合理分配到不同的数据中心中，并能够通过虚拟化把所有活跃的主机集中起来，关掉或者睡眠一些不使用的设备来降低数据中心能耗。简单地说，网络管理员可以通过动态分配计算、存储、网络等 IT 基础设施资源来满足更灵活高效的需求。

正是虚拟化技术提供的这种灵活性，随后引发了一种新的商业模式。数据中心的网络管理人员意识到，为了满足用户对计算、存储资源不断增加的需求，考虑到峰值业务需求和服务体验保障，需要采购大量冗余的设备，这些冗余设备无论使用与否都会消耗大量的电能，占用大量的空间。亚马逊就陷入了这种困境，当时亚马逊的业务不断增长，迫使它增加更多的设备来满足业务需求，但是很多设备在业务高峰期之外的时间里通常是闲置的，造成了 IT 资源的巨大浪费。为了应对这种困境，亚马逊开发了著名的亚马逊云服务（Amazon Web Service, AWS），通过 AWS 将闲置的资源出租，从而提高了业务闲时的资源利用率。当亚马逊内部业务繁忙需要更多的资源时，AWS 就可以减少零售给用户的资源来保证内部对资源的需求；当内部不需要这些资源时，就可以接着把资源分配给零售用户，这种成功的商业模式直接推动了云计算技术的诞生。

从网络技术发展的角度来看，从 20 世纪 70 年代开始到现在，随着业务和应用的不断丰富与发展，用户对互联网提出的要求也在不断变化。网络最初设计的目标只是实现单纯的端到端数据传送，IP 数据分组主要包含源主机和目的主机的网络地址，互联网中几乎所有的流量都是建立在 TCP/IP 架构之上，尽管设备性能上有了飞跃性的提高，但网络本身的架构却没有什麼突破性的变化。然而随着网络中传输的流量越来越大，业务应用对网络的要求越来越高，传统的依靠端到端连接和尽力而为路由转发的网络体系架构，越来越难以满足日益出现的新需求。正如上面提到的亚马逊多租户数据中心应用场景，其对网络提出了更高的要求，它要求不同租户网络间的高度隔离性，以保证各租户差异性的服务质量（QoS），显然如果只依靠端到端的传统网络是难以满足这种需求的。为了解决租户之间的隔离性问题和更复杂的网络需求，在这些年中，各种新技术层出不穷。例如，VLAN（Virtual LAN，虚拟局域网）技术就是为了解决二层局域网内的隔离性问题而提出的，它主要采用 VLAN 标签的方法隔离网络中的不同用户，但是由于标签数量有限，因此在大规模的网络中，VLAN 技术不能很好的工作，于是近年来业界又进一步提出了 VXLAN（Virtual eXtensible LAN）、NVGRE 等改进的网络虚拟化技术。

从上述技术的发展思路可以看出，这些网络技术的创新往往都是在被动地进行修补，通常情况下是用户发现现有网络中的问题或者有新的业务需求后，由研究者在 TCP/IP 协议的基础上进行设计改进，提出新的协议，并逐步在网络设备中改造集成，尽管这些年出现了非常多新的协议帮助解决问题，但这种方法并不是系统性的解决方案，而且通过这种修补的方式，还将会使得原本

简单的网络设备变得越来越复杂、臃肿，从网络技术未来的整个发展来看无疑是饮鸩止渴，何况面临全新的网络应用场景，这种“打补丁”的解决办法是远远不够的。例如，在一个典型的多租户数据中心中，每个虚拟机都需要一个自己的 IP 地址来与外网相连，虚拟机迁移后，可以重新为虚拟机分配新的物理地址（MAC 地址）和 IP 地址。当虚拟机个数很少时，管理员可以手动进行配置，这不会引起太大问题，然而当虚拟机规模很大时，就很难手动配置各种策略了（如 QoS 策略、ACL 策略等）。针对这些新的需求，人们希望通过设计新的网络协议来完成新的功能。

但是，由于厂商设备的封闭性，人们想要添加新协议的行为在很大程度上受到了束缚，不够灵活。这是因为网络管理人员必须使用商用路由器和交换机提供的管理接口才能配置设备，虽然这些设备都提供了一些接口来方便网络管理人员对其进行配置，但是由于多方面原因它们都隐藏了底层的技术细节，网络管理人员的操作都要通过设备本身封闭的操作系统来完成，这极大地限制了网络管理的灵活性。当人们希望实现并加入一些新的协议时，通常需要设备厂商重新研发增加设备的相应功能，这往往需要很长的研发周期和高昂的研发成本，从而使得网络升级和新业务的部署变得非常困难。除了交换设备相对封闭不能使网络管理员快速开发协议外，事实上传统网络交换设备架构本身也限制了协议的创新，很多时候网络设备的更替往往只是单纯地提高转发速率性能和增加新协议的支持，而很少从系统架构上进行改进，这主要是由于传统的网络设备中负责智能决策的控制平面和负责高速转发的数据平面集中在一起，都是由硬件厂家进行控制，设备的可编程能力非常弱，研究人员和管理人员事实上根本无法添加并实现自己新的协议，由此推动了业界 ForCES（Forwarding and Control Element Separation，转发控制单元分离）和 RCP（Routing Control Platform，路由控制平台）两种新技术的产生（这两种技术将在第 2 章详细介绍）。这两种技术的核心理念是通过尝试分离网络设备的控制逻辑和转发逻辑，并采用标准的接口 API，以提高灵活性。但是，新协议的设计实现和推动完成是一个漫长的过程，可能需要花费几年的时间，并需要技术、标准、产业等多方面的配合，然而出于商业考虑，设备厂商不太愿意采用像 ForCES 那样的标准数据平面 API，因为开放的 API 会引入新的厂商与他们竞争，因此最终这两种协议并没有得到广泛的使用。

近些年来，为了加快推动网络技术的创新发展，研究者们建设了一系列大规模网络实验基础平台，如美国 PlanetLab 和 Emulab，希望在此之上运行新的协议进行网络创新。出于这些动机，2005 年，美国国家自然科学基金会（National Science Foundation，NSF）资助启动了全球网络创新实验环境（Global Environment Networking Innovations，GENI）计划。与此同时，为了更好地实现控制与转发分离并加快网络创新，2006 年由斯坦福大学主导，联合美国国家自然科学基金会以

及多个工业界厂商共同启动了 Clean-Slate Design for the Internet（以下简称 Clean-Slate）项目，该项目旨在重塑互联网，摒弃传统渐进叠加和向前兼容的原则，Clean-Slate 计划最初选择了网络体系结构、异构应用、异构物理层技术、安全、经济与政策 5 个关键研究方向。从 2006 年开始，斯坦福大学研究生 Martin Casado 参与了 Clean-Slate 项目，并着手领导了一个叫做 Ethane 的项目，该项目的最初目标是为了提出一个新型的企业网络架构，希望通过集中控制来简化管理模型，并为企业网提供更高的安全性。因此，Ethane 最初设计是允许网络管理员定义一个全网的安全策略，这些策略自动下发给各个交换机，以指导其处理网络流量，Ethane 项目最早部署了一台控制器和 19 台交换机组成的实验环境，用以管理 300 个有线用户和一些无线用户的流量。随后在 2007 年 SIGCOMM 会议上，Martin Casado 发表了一篇名为“Ethane: Taking Control of the Enterprise”的论文，引起了学术界的广泛关注，在 Ethane 的系统架构中，控制与转发完全解耦，控制器通过 Pol-Ethane 语言向交换机分发策略，可以说 Ethane 包含了 SDN 早期的思想，它实际上成为了 SDN 技术的雏形。

2007 年，Martin Casado 联合 Nick McKeown、Scott Shenker 等人共同创建了一个致力于网络虚拟化技术创新的公司——Nicira，并最早提出了 SDN 的概念。紧接着在 2008 年，Nick McKeown 在 SIGCOMM 会议上发表文章“OpenFlow: Enabling Innovation in Campus Networks”，首次提出了将 OpenFlow 协议用于校园网络的试验创新。OpenFlow 开始正式进入人们的视野，SDN 呼之欲出。OpenFlow 协议是为了简化 Ethane 项目中的交换机设计而被提出的，它是一个控制平面和数据平面之间的交互协议，它使得控制和转发完全分离，从而控制器专注于决策控制，而交换机完全专注于转发工作。OpenFlow 协议使得网络具有高度的灵活性以及强大的可编程能力，它蝉联了 2008 年和 2009 年的 SIGCOMM 最佳演示奖，被麻省理工学院和多家咨询机构评选为信息领域未来十大技术之一，为 SDN 早期的发展注入了强劲动力，并一举成为了后续 SDN 领域备受关注的核心技术。

2011 年初，在 Google、Facebook、Yahoo 等业界重量级企业的推动下，共同成立了 ONF（Open Networking Foundation，开放网络基金会），并正式提出了 SDN 的概念。ONF 组织致力于推动 SDN 架构、技术的规范和发展工作。同年 4 月，印第安纳大学、斯坦福 Clean-Slate 计划工作组与 Internet 2 联合发起了 NDDI（Network Development and Deployment Initiative，网络开发与部署行动计划），旨在创建一个基于 SDN 的网络创新试验平台，以支持全球科学研究，NDDI 将提供一项名为“开放科学、学问与服务交流（Open Science, Scholarship and Services Exchange, OS3E）”的 Internet 2 服务，并通过与加拿大的 CANARIE、欧洲的 GÉANT、日本的 JGN-X 以及巴西的 RNP 等国际合作伙伴的实验平台协作，实现了与北美、欧洲、亚洲以及南美洲的互联。

随着 SDN 大潮迅猛来袭，迫使传统网络设备商重新思考其未来的发展战略。2013 年 4 月，由 Cisco、Juniper、Broadcom、IBM 等传统网络设备、芯片、平台巨头主导的开源平台项目——OpenDaylight 引发了 IT 界一次巨大的轰动，OpenDaylight 旨在促进 SDN 技术交流和产业化开源架构，提供开源代码和架构以加快标准化、健壮性 SDN 平台的发展演进。OpenDaylight 项目的成立对于 SDN 技术的发展具有里程碑式意义，它代表了传统网络芯片、设备巨头对网络领域开源技术方向的认可。

2013 年 8 月的 SIGCOMM 会议上，Google 公司首次将其如何利用 SDN 技术解决数据中心之间流量问题的方案通过论文公之于众——“B4: Experience with a Globally-Deployed Software Defined WAN”，鉴于这篇论文在 SDN 实际应用场景分析上的重要贡献，最终获得了 SIGCOMM 2013 的最佳论文奖。Google 的 B4 网络作为 SDN 实际部署的一个成功的商业案例，引起了全球业界的广泛关注与思考，同时也使 SDN 技术名声大震。它向人们证明了，针对传统网络架构所面临一系列棘手问题，SDN 具有解决这些问题的能力，同时也增强了业界对于 SDN 技术可行性的信心。

2012 年，美国 AT&T、英国电信、德国电信等基础电信网络运营商在欧洲通信标准协会（ETSI）上提出了网络功能虚拟化（NFV）的概念。NFV 是一种新型网络架构概念，期望摆脱专用硬件的束缚，通过通用的 IT 硬件及虚拟化技术，实现网络功能的软件化，例如 NAT、防火墙、DNS、缓存等。NFV 与 SDN 既有联系又有区别：联系在于，它们的核心思想都是软件化、开放化、标准化，从而降低成本提高灵活性，也能够相互促进；区别在于，SDN 更注重网络系统的可编程性，即 OSI 参考模型的 3 层及以下，而 NFV 更注重网元层面的虚拟化以及网络上层功能的软件化，即 OSI 参考模型的 4~7 层。由于更接近运营商的业务需求和成本需求，NFV 一经提出就得到了以运营商为主的广泛关注。据 ETSI 的统计数据，截至 2016 年 2 月，ETSI 的 NFV 规范组（NFV ISG）已经拥有来自全球通信业界的 116 家成员单位以及 175 家参与单位。

随着 B4 在 Google 数据中心中的成功落地与应用，如何利用 SDN 技术改造运营商网络成为了新的探索目标。首先，美国运营商 AT&T 提出了 Domain 2.0 计划，到 2015 年底已完成该计划的 5.7%，2016 年将进一步增强网络软件化和云化的基础，进入加速推动软件化的高速通道，预计到 2020 年底实现 75% 的网络功能完全由软件构成，通过 SDN 和 NFV 将网络控制权放入云中和用户手中。目前，AT&T 已经成功发布了一个基于 SDN 的产品服务理念——按需服务（On Demand Service），用户可以自己来添加或改变网络服务类型，比如实时设定需求的网络速率，并付出相应的费用。按需服务从概念提出到试商用仅花了 6 个月时间。另外，我国三大内运营商分别初步进行了 SDN 的商用部署探索。2014 年 4 月，中国电信与华为联合宣布双方合作完成了全球首个运营商 SDN 商用部署，将 SDN 技术

成功应用于数据中心网络。2014年8月,中国移动与华为合作利用SDN技术完成了政企专线业务在分组传送网(PTN)中的改造,大幅提升了运维效率及带宽利用率。同年,中国联通首次成功实现了SDN IP-RAN的商用,初步实现了运维精简、业务可视、快速创新的目标。2015年9月,中国联通发布了新一代网络架构CUBE-Net 2.0白皮书,目标是基于SDN/NFV、云和超宽带技术实现网络重构。

在SDN网络操作系统方面,2014年12月,由斯坦福大学和加州伯克利分校SDN先驱共同创立的ON.Lab,推出了新的SDN开源操作系统——开放网络操作系统(ONOS)的首个版本。至此,形成了OpenDaylight与ONOS两大阵营。ONOS是业界首个面向运营商业务场景的开源SDN控制器平台,旨在满足用户高可靠性、高灵活性的使用需求。与OpenDaylight由设备商和互联网厂商主导不同的是,ONOS主要由运营商和设备商为主推动,主要致力于推动大规模组网场景下的应用,同时重点支持设备的白盒化,这在一定程度上代表了运营商的利益,可以帮助其降低开支、提高服务效率。

通过回顾整个互联网的发展历史,可以看到网络从最初的端到端模型已经发展到今天诸如多租户数据中心等复杂的应用场景;用户需求从最初的单纯可达性到现在对服务质量、流量工程等多方面的需求;网络协议从最初的TCP/IP模型发展到了现在TCP/IP协议簇和上千种补丁协议。整个网络越来越复杂,但设备架构还是一如既往的封闭,缺少灵活性,难以跟上日益变化的应用需求。同时,由于设备门槛很高,核心技术掌握在少数大公司手中,这就制约了新技术和新协议的部署与实施。在这一背景下,SDN技术应运而生,其使命就是加快网络创新,推动网络架构从注重标准到注重实现转型,打破设备的封闭性,使整个网络更加开放,以适应不断更新的业务和应用需求。可以说SDN技术是网络发展到这一阶段的必然产物,它的迅速发展自然也在情理之中。

介绍了这么多软件定义网络技术出现的背景,本章最后简要地总结一下它的核心思想。SDN系统架构分为一个集中式的控制平面(通常是控制器)和分布式的数据平面,控制器位于上层应用与物理设备之间,控制器首先负责把网络中的各种功能进行抽象,建立具体的操作模型,并向上层提供编程接口,上层应用着重根据业务需求通过控制器与物理设备进行交互,网络中的设备通过控制器向应用平面传递信息。可以看到,SDN的两个突出属性是数据平面和控制平面分离以及网络的可编程性,下一章将对此进行详细分析和介绍。

SDN技术目前仍处于发展初期,在协议和处理机制上仍有诸多需要改进和完善的地方,当前其概念和架构也还没有一个统一明确的定义和设计,甚至网络控制、网络管理以及业务编排之间界限也尚未定义清晰。第2章中我们将重点介绍ONF组织对SDN的定义与架构,并通过分析其几个基本属性,以期使读者对SDN基本原理有进一步的理解和认识。

第2章

SDN 基本原理

2.1 SDN 定义与架构

2.1.1 SDN 定义

顾名思义，SDN 与传统网络的最大区别就在于可以通过编写软件的方式来灵活定义网络设备的转发功能。在传统网络中，控制平面功能是分布式地运行在各个网络节点（集线器（Hub）、交换机（Switch）、路由器（Router））中的，因此新型网络功能的部署需要所有相应网络设备的升级，导致网络创新往往难以落地。而 SDN 将网络设备的控制平面与转发平面分离，并将控制平面集中实现，这样新型网络功能的部署只需要在控制节点进行集中的软件升级，从而实现快速灵活地定制网络功能。另外，SDN 体系架构还具有很强的开放性，它通过对整个网络进行抽象，为用户提供完备的编程接口，使用户可以根据上层的业务与应用个性化地定制网络资源来满足其特有的需求。由于其开放可编程的特性，SDN 有可能打破某些厂商对设备、协议以及软件等方面的垄断，从而使更多的人可以参与到网络技术的研发工作中来。

根据业界的通用性理解^[1,2]，可以将 SDN 定义为一种数据控制分离、软件可编程的新型网络体系架构，其基本架构如图 2-1 所示。SDN 采用了集中式的控制平面和分布式的转发平面，两个平面相互分离，控制平面利用控制—转发通信接口对转发平面上的网络设备进行集中式控制，并向上提供灵活的可编程能力，具备以上特点的网络架构都可以被认为是广义的 SDN 技术。