

OpenDaylight 应用指南

唐宏 刘汉江 陈前锋 李鹏 等◎编著

OpenDaylight是目前备受瞩目的主流开源控制器之一，
它旨在建设一个开放的SDN网络系统平台，
推动SDN和NFV技术的创新实施和透明化。

APPLICATION GUIDELINES
OF OPENDAYLIGHT



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

OpenDaylight 应用指南

唐宏 刘汉江 陈前锋 李鹏 等◎编著

APPLICATION GUIDELINES
OF OPENDAYLIGHT

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

OpenDaylight应用指南 / 唐宏等编著. — 北京 :
人民邮电出版社, 2016. 1
ISBN 978-7-115-41427-4

I. ①O… II. ①唐… III. ①软件开发—指南 IV.
①TP311.52-62

中国版本图书馆CIP数据核字(2015)第316557号

内 容 提 要

本书分为 OpenDaylight 基础概述、重点项目介绍和开发实践三部分。介绍了 OpenDaylight 社区背景、组织架构与项目概况; 分析与阐释了 OpenDaylight 部分主要项目的设计思路、原理和开发技术; 探讨了基于 OpenDaylight 进行业务控制平台开发的实践经验。本书可以作为网络技术人员和 IT 系统开发人员的中初级读物, 帮助他们快速了解 OpenDaylight 社区及其大型通用 SDN 控制器的开发方法。

-
- ◆ 编 著 唐 宏 刘汉江 陈前锋 李 鹏 等
责任编辑 吴娜达
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>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 23.5 2016 年 1 月第 1 版
字数: 587 千字 2016 年 1 月北京第 1 次印刷
-

定价: 98.00 元

读者服务热线: (010) 81055488 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

关于 SDN 的著述已经很多，一直以基于 OpenFlow 为核心的控制与转发技术介绍为主，这也是早期 SDN 技术研究的一个主要特点。随着工业界对 SDN/NFV 技术的理解和研究的不断深化，各种开放性的 SDN 技术、开源社区和技术标准不断涌现，OpenDaylight 开源控制器社区就是其中的典型代表。自 2014 年 4 月发布第一个版本以来，OpenDaylight 改变了 SDN 控制器技术长期以来局限于 OpenFlow 框架的格局，很好地体现了传统网络技术与 SDN 技术的共存架构，也体现了通信技术与 IT 技术在软件实现架构上的充分融合，目前已经成为工业界最受关注的 SDN 控制器项目之一。但由于 OpenDaylight 涉及广泛的网络基础概念，且社区成立时间较短，目前系统性的介绍较少。

本书的写作目的非常明确，一是为那些 SDN 的初学者和中级技术人员，提供一个了解 OpenDaylight 开源社区的快速参考工具，需要编者提供一个基本忠实于社区内容的项目概述和用例介绍；另一个目的就是反映我们对 OpenDaylight 技术的理解和应用实践，这需要编者从实用的角度针对一个具体的应用场景，介绍从编排、控制器到转发平面的端到端设计与实现。本书尽最大可能保留这两个特点，以帮助网络设计人员和控制器开发人员快速了解 OpenDaylight 社区及其控制器实现思路，便于后续深入学习。

本书共分三个部分：第一部分主要帮助读者快速理解 OpenDaylight 社区概况，包括社区的构成、管理方法和不同版本的特点；第二部分以社区项目为章节，重点介绍社区内开展的各类项目，鉴于目前（锂版本）社区的内部研发项目已达数十个之多，我们只能选择存在于多个版本中的一些基础性项目，以及诸如 SFC 等前沿性的热门研究项目。上述两部分以社区提供的资料编译为主，部分章节补充了一些必要的技术背景。第三部分重点介绍我们在 OpenDaylight 上的实践应用，从编排层、控制器和转发平面三个方面介绍了一个宽带运营网络中的典型解决方案及其实现方法。

成稿过程遇到很多困难，最大的困难来自于 OpenDaylight 社区本身，在社区建立初期，各项目文档和代码均尚未稳定，排除代码质量本身的因素，各个发布版本间的实现框架和项目实现思路会发生较大变化，如从 AD-SAL 向 MD-SAL 的转变，而这些变化本身往往体现了 OpenDaylight 设计思路的变迁过程，能准确地反映这些变化和过程需要良好的 ICT 专业背景；其次，社区项目增长非常迅速，成稿过程中恰逢“锂”版本发布，建立了大量新增项目和补充文档，编者不得不根据新增内容进行修订；另外，OpenDaylight 潜在的应用场景非常

广泛，从 IDC 网络到运营商广域网，专业类型也横跨宽带、传输、移动网络以及运营支撑系统等多个 SDN/NFV 领域，横贯这些知识点需要较高的专业能力和较牢固的技术基础。

尽管如此，我们仍然努力完成此书的编著，鉴于成稿仓促，OpenDaylight 社区仍在发展的初级阶段，内容纰漏和观点偏差难以避免。我们相信，SDN 是一个众人拾柴的过程，每个人都有自己的理解，只有勇于讲出自己的视角和故事，我们最终才能完成 SDN 这个巨大的拼图。因此，编者希望平实地介绍 OpenDaylight 相关技术，而非像以往的 SDN 书籍那样，花较多篇幅去辨析什么是真正的 SDN 架构，业界现阶段的发展重点都已聚焦于各类开源实现。

参与本书撰写以及开源方案实现的人员，既有长期从事电信运营网络研究的技术专家，也有新近加入 SDN 研究的 IT 新兵，这是一个有趣的技术组合，也是一种有益的技术转型尝试，SDN 时代将是一个 ICT 跨界人才辈出的年代。

中国电信广州研究院的刘汉江、陈前峰、李鹏、罗雨佳、欧亮、王勇、莫博奇等参与了本书各章节的编写工作，对他们的辛勤付出表示衷心的感谢！

唐 宏

2015 年 10 月于广州

第一部分 基础概述

第 1 章	SDN/NFV 概述	3
1.1	SDN/NFV 介绍	3
1.1.1	SDN 架构和技术特点	3
1.1.2	NFV 架构和技术优势	5
1.1.3	对 SDN 与 NFV 关系的理解	6
1.2	控制器概述	7
1.2.1	商用控制器	8
1.2.2	开源控制器	8
1.2.3	OpenDaylight 控制器概述	10
第 2 章	OpenDaylight 社区	11
2.1	OpenDaylight 社区介绍	11
2.1.1	OpenDaylight 开源社区	11
2.1.2	OpenDaylight 社区管理	12
2.1.3	OpenDaylight 社区运营	12
2.2	OpenDaylight 版本介绍	14
2.2.1	Hydrogen 版本简介	14
2.2.2	Helium 版本简介	15
2.2.3	Lithium 版本简介	16
2.3	OpenDaylight 项目简介	17

第二部分 OpenDaylight 重点项目介绍

第 3 章	OpenDaylight Controller 项目	25
3.1	项目介绍	25

3.1.1	基础框架	25
3.1.2	OSGi	26
3.1.3	Karaf	27
3.1.4	Maven	28
3.1.5	SAL	29
3.1.6	RESTCONF	34
3.2	安装与配置	34
3.2.1	安装运行	34
3.2.2	Mininet 安装	38
3.3	开发指南	38
3.3.1	环境准备	38
3.3.2	创建项目	41
3.3.3	加载组件	43
3.3.4	定义启动配置	46
3.3.5	定义数据模型	51
3.3.6	实现自定义 RPC	55
3.3.7	数据读写	58
3.3.8	通知事件	60
3.3.9	总结	61
第 4 章	YANG Tools 项目	62
4.1	YANG 语言介绍	62
4.1.1	NETCONF 中的 YANG	62
4.1.2	OpenDaylight 中的 YANG	63
4.1.3	YANG 语法	64
4.2	项目介绍	70
4.2.1	YANG Tools 项目结构	71
4.2.2	YANG Tools 的 Java 映射介绍	71
4.3	安装与运行	84
4.3.1	安装	84
4.3.2	配置	85
4.3.3	运行	86
4.4	开发指南	86
第 5 章	OpenFlow 项目	88
5.1	项目介绍	88
5.1.1	总体框架	88
5.1.2	OpenFlow 协议简介	89
5.1.3	OpenFlow 协议库	91
5.1.4	OpenFlowPlugin	91

5.2 安装与使用	92
5.2.1 安装	92
5.2.2 配置	93
5.2.3 使用	94
5.3 开发指南	97
5.3.1 环境准备	97
5.3.2 编译项目	97
5.3.3 代码分析	98
第 6 章 L2Switch 项目	101
6.1 传统 L2Switch 简介	101
6.2 L2Switch 项目介绍	101
6.2.1 L2Switch 项目架构	102
6.2.2 L2Switch 工作流程	103
6.3 安装与使用	104
6.3.1 安装 L2Switch 项目	104
6.3.2 L2Switch 的功能验证	105
6.3.3 配置文件说明	108
6.4 开发指南	110
6.4.1 包处理服务代码说明	111
6.4.2 环路消除服务代码说明	111
6.4.3 ARP 处理服务代码说明	111
6.4.4 地址跟踪服务代码说明	112
6.4.5 主机跟踪服务代码说明	113
6.4.6 L2Switch 主服务代码说明	113
第 7 章 Topology Processing Framework 项目	114
7.1 项目简介	114
7.1.1 项目架构	115
7.1.2 工作原理	115
7.1.3 处理流程	118
7.2 安装使用	119
7.2.1 安装 Topoprocessing 项目	119
7.2.2 添加拓扑处理规则	120
7.2.3 添加 underlay 拓扑	120
7.2.4 添加 underlay 节点	121
7.2.5 删除 underlay 节点方法	121
7.2.6 查看拓扑的方法	121
7.2.7 删除拓扑的方法	122
7.3 拓扑聚合和拓扑过滤示例	122

7.3.1 拓扑聚合示例——equality	122
7.3.2 拓扑聚合示例——unification	124
7.3.3 拓扑过滤示例——filtration	126
第 8 章 BGPCEP 项目 (BGP 插件)	129
8.1 BGPCEP 项目简介	129
8.1.1 项目间依赖关系	130
8.1.2 项目组件结构	130
8.2 BGP 介绍	131
8.2.1 BGP 消息	132
8.2.2 BGP 路径属性	132
8.2.3 BGP Linkstate	133
8.3 BGP 和 PCEP 协议消息处理过程	135
8.3.1 消息生命周期	135
8.3.2 通道处理器	136
8.3.3 注册	136
8.3.4 解析	137
8.3.5 序列化	137
8.4 开发指南	138
8.4.1 BGP 插件实现	138
8.4.2 BGP 消息处理	141
8.5 安装与使用指南	142
8.5.1 安装	142
8.5.2 配置与使用	143
8.6 BGP 插件在 ODL-SDNi 项目中的应用	156
第 9 章 BGPCEP 项目 (PCEP 插件)	158
9.1 PCEP 协议介绍	158
9.1.1 PCE 产生原因	158
9.1.2 PCEP 介绍	159
9.1.3 Segment Routing	160
9.2 开发指南	161
9.2.1 PCEP 插件实现	161
9.2.2 PCEP 消息处理	164
9.3 安装与使用指南	165
9.3.1 安装	165
9.3.2 配置与使用	166
第 10 章 OVSDB Plugin 项目	180
10.1 项目介绍	180

10.1.1	OVS	180
10.1.2	OVSDDB	181
10.1.3	OVSDDB 管理协议	183
10.1.4	OVSDDB Plugin 架构	184
10.1.5	OVSDDB Plugin 的北向接口	184
10.2	安装与使用	186
10.2.1	安装	186
10.2.2	使用	186
第 11 章	Virtual Tenant Network 项目	188
11.1	VTN 概述	188
11.2	VTN 项目介绍	188
11.2.1	虚拟网络层	188
11.2.2	虚拟网络结构	189
11.2.3	物理网络资源映射	190
11.2.4	vBridge 功能	191
11.2.5	vRouter 功能	191
11.2.6	流过滤器功能	192
11.2.7	多控制节点间的协作	193
11.2.8	OpenFlow 网络与传统 L2/L3 网络间的协作	193
11.2.9	VTN 接口介绍	193
11.3	VTN 项目架构介绍	194
11.3.1	软件功能架构	194
11.3.2	VTN 协调器架构	195
11.3.3	VTN 管理器架构	196
11.4	安装与使用	197
11.4.1	总览	197
11.4.2	安装 VTN 协调器	197
11.4.3	安装 VTN 管理器	201
11.5	应用案例	202
11.5.1	单控制器场景下的配置示例	202
11.5.2	多控制器场景下的配置示例	204
第 12 章	业务链项目	206
12.1	概述	206
12.1.1	业务链技术概述	206
12.1.2	业务链结构	207
12.2	SFC 项目介绍及使用	210
12.2.1	SFC-UI 模块	210
12.2.2	SFC 南向 REST 插件	212

12.2.3	SFC-OVS 插件	213
12.2.4	SFC 分类器	217
12.2.5	SFC OpenFlow 二层映射器插件	217
12.2.6	SF 选择算法	220
12.3	SFCOFL2 使用案例介绍	221
第 13 章	Group Based Policy 项目	226
13.1	技术背景	226
13.1.1	策略模型	227
13.1.2	策略管理技术与平台	228
13.1.3	策略一致性问题	229
13.1.4	承诺理论	230
13.2	GBP 项目与基本架构	231
13.2.1	项目概况	231
13.2.2	GBP 策略模型	232
13.2.3	GBP 策略解析过程	234
13.2.4	GBP 总体架构	238
13.3	GBP 用户配置界面	240
13.3.1	Policy Expression 视图	241
13.3.2	Governance 视图	242
13.3.3	Renderer Configuration 视图	243
13.3.4	使用 Access Model Wizard	244
13.4	GBP 安装与测试	245
13.4.1	Karaf 集成安装	245
13.4.2	编译安装	245
13.4.3	Demo 安装与测试	245

第三部分 开发实践

第 14 章	IP 网络边缘业务链技术方案	253
14.1	方案背景	253
14.1.1	SDN 控制器应用场景	253
14.1.2	项目提出背景	254
14.2	IP 网络边缘业务链总体方案	256
14.2.1	总体思路与功能架构	256
14.2.2	业务编排器功能	258
14.2.3	ODL 控制器及其业务链相关插件功能	259
14.2.4	BRAS 与 NFV 业务链设备功能	259
14.3	IP 网络边缘业务链主要功能	259

14.3.1	业务控制策略管理	260
14.3.2	可编程转发功能	260
14.3.3	业务链管理与调度	262
第 15 章	SCP 架构和实现	264
15.1	总体方案	264
15.1.1	SCP 架构	264
15.1.2	业务编排流程	265
15.1.3	系统接口	266
15.2	项目管理模块	269
15.2.1	实现思路	269
15.2.2	模块结构	269
15.2.3	处理流程	270
15.2.4	主要数据结构	270
15.3	策略管理模块	271
15.3.1	实现思路	271
15.3.2	模块结构	272
15.3.3	处理流程	273
15.3.4	主要数据结构	274
15.4	资源监控模块	275
15.4.1	实现思路	275
15.4.2	模块结构	276
15.4.3	处理流程	276
15.4.4	主要数据结构	278
15.5	用户管理模块	279
15.5.1	实现思路	279
15.5.2	模块结构	279
15.5.3	处理流程	280
15.5.4	主要数据结构	281
第 16 章	ODL 控制器实现	282
16.1	策略管理插件	283
16.1.1	策略管理插件功能	283
16.1.2	策略管理插件实现	283
16.2	业务链管理插件	288
16.2.1	业务链管理插件功能	288
16.2.2	业务链管理插件实现	289
16.3	NETCONF 插件	294
16.3.1	NETCONF 插件功能及二次开发	294
16.3.2	NETCONF 插件使用	295

16.4	SFC 项目路径选择算法增强	298
16.4.1	SFC 项目路径选择算法模块介绍	298
16.4.2	新增路径算法介绍	299
16.4.3	在 SFC 项目中添加 TLO 算法	300
第 17 章	软件转发层功能增强	305
17.1	转发面背景技术	305
17.1.1	DPDK 技术	305
17.1.2	OVS 介绍	310
17.1.3	OVDK 介绍	312
17.2	PF 方案与实现	314
17.2.1	PF 功能架构	314
17.2.2	可编程交换机的实现	319
17.3	软件转发层性能监控	321
17.3.1	NETCONF 协议介绍	321
17.3.2	LibNetConf 介绍	323
17.3.3	Netopeer 介绍	323
17.3.4	采集程序介绍	324
第 18 章	应用案例	334
18.1	基于 PF 的 SDN 协议网关	334
18.1.1	场景与需求描述	334
18.1.2	部署与配置	335
18.1.3	验证业务流程	341
18.2	IP 边缘业务链部署	343
18.2.1	功能要求	343
18.2.2	安装部署	344
18.2.3	测试验证	347
参考文献		358
缩略语		361

第一部分 基础概述

简要介绍SDN/NFV技术和OpenDaylight开源社区，帮助读者了解OpenDaylight的技术背景、社区状况、版本发布情况、项目组成等。

SDN/NFV 概述

1.1 SDN/NFV 介绍

互联网产业的迅猛发展不仅催生了云计算等新技术的涌现,也促使大量互联网创新应用取得巨大成功。在此期间,互联网业务及其核心收益,也由以往的以网络运营商管道接入为主,向着以用户体验为核心的应用价值进行转变,传统管道服务的日益低值化,促进了运营商不断寻求适应新型业务发展的网络新技术。

然而,当前建立在尽力而为 IP 网络及其运营支撑系统之上的运营商承载网络,由于智能化承载控制平面的缺失,以及业务与网络设备的紧耦合造成的系统封闭性,存在着网络资源利用率低、新业务部署周期长、感知和差异化保障不足、网络维护复杂困难等诸多问题,严重影响了移动互联网时代的竞争力。

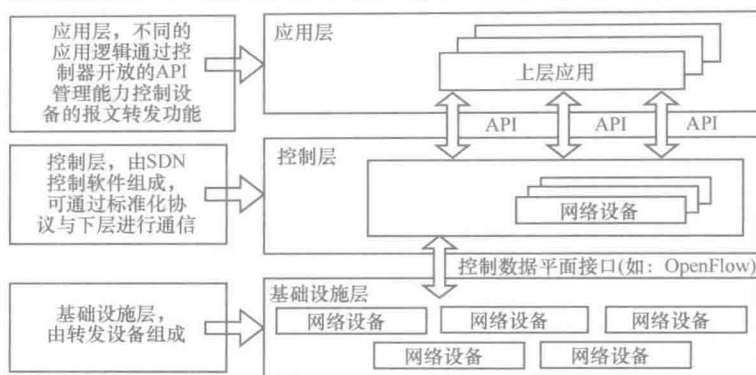
为此,积极开拓信息和网络应用新技术,建设开放、高扩展、敏捷、智能化新型网络,为客户提供便捷、丰富、个性化、高体验的网络服务,已成为通信业和网络运营商的最新发展目标。在这个背景下,以研究新技术、新架构为目标的 SDN 和 NFV 技术应运而生,相关的网络技术变革不仅已成为业界共识,而且将逐步成为未来运营商业务与网络转型的核心技术。

本章将简要阐述 SDN/NFV 技术的架构和特点以及这两者之间的关系。

1.1.1 SDN 架构和技术特点

软件定义网络(Software Defined Networking, SDN)起源于斯坦福大学的校园网络,是一种新型的网络架构和技术,它的核心设计理念是将传统 IP 网络的数据转发和路由控制进行分离,实现集中控制,分布转发,通过软件可编程的方式来简化和优化网络^[1]。

图 1-1 所示为目前业界理解的 SDN 一般架构,主要由以下三个层面组成:基础设施层、控制层、应用层。

图 1-1 SDN 的逻辑架构^[2]

- 基础设施层由转发设备组成，主要负责数据处理、转发和状态收集等功能。
- 控制层由 SDN 控制软件组成，可通过 OpenFlow (OF)、PCEP (Path Computation Element Communication Protocol, 路径计算单元通信协议) 等标准化协议与下层进行通信，控制底层网络设备的转发行为，处理数据平面的资源编排，维护网络拓扑、状态信息等。控制层还将网络能力进行抽象，向应用层提供开放的 API (应用程序接口) ^[3]。
- 应用层由众多业务和应用软件构成，这些软件可根据控制层抽象的网络信息执行相应的控制算法，形成网络策略，然后将控制策略通过控制层转化为流量控制命令下发给转发设备。

SDN 技术自出现以来，形成了众多流派，可以大致将其分为两大类：SDN 过渡技术和 OpenFlow。其中，I2RS (Interface to the Routing System) 和 overlay 是最具代表性的 SDN 过渡技术。I2RS 是 IETF 成立的 SDN 工作组，其出发点是维持现有路由体系架构不变，在网络设备上提供专用接口供用户开发应用。overlay 是 Big Switch、VMware 等厂商主导的虚拟化解决方案，其核心思路是采用 VxLAN (Virtual extensible Local Area Network, 虚拟可扩展局域网)、NVGRE (Network Virtualization using Generic Routing Encapsulation, 网络虚拟化通用路由封装)、STT (Stateless Transport Tunneling) 协议等隧道技术在物理网络上叠加虚拟的逻辑网络。用户可以灵活定义逻辑网络、调整网络策略。而 ONF 定义的 OpenFlow 遵循控制与转发分离的设计原则，其转发层由 OpenFlow 交换机组成，控制层与转发层之间通过标准化的 OpenFlow 协议 (南向接口) 进行通信，控制器可通过“流匹配”的方式定义网络转发规则，向交换机下发相应的流表，OpenFlow 交换机执行相应流表的动作即可高速转发数据包 ^[4]。

从上述 SDN 技术的几种不同实现方式，可以总结出以下几大技术特征。

- **控制与转发分离**：将传统 IP 设备的控制面和转发面进行分离，转发设备抽象为一个通用的受控的网络设备，只关注转发和存储能力，与业务特性彻底解耦。控制面则通过控制协议控制转发面的流量行为。
- **集中控制**：将网络信息进行抽象处理构成统一的控制平台，实现逻辑上的集中式控制，从而获取网络资源的全局信息，并根据全局状态实现资源的全局调度和优化，提高网络控制的灵活性。
- **开放可编程**：通过开放的 API，应用层可以告知控制层如何进行网络资源操作才能更