

# 网络控制器 OpenDaylight 的研究与分析

许名广<sup>1</sup> 刘亚萍<sup>1</sup> 邓文平<sup>2</sup>

(国防科学技术大学并行与分布处理国家重点实验室 长沙 410073)<sup>1</sup>

(国防科学技术大学计算机学院 长沙 410073)<sup>2</sup>

**摘要** OpenDaylight 控制器(ODL)是一款全新的开源 SDN 控制器。它分为 3 层:南向协议层、控制层、北向接口层。南向协议层支持 Openflow、LISP 等多种协议;控制层具备完整的网络基础服务功能;北向接口层提供 REST API 以供网络应用程序调用。同时它支持集群工作模式,可应用于大规模网络。通过分析 ODL 的总体框架、工作流程、模块扩展,以及 ODL 的 LISP 服务功能,搭建一个关于 LISP 通信的实验系统,可以看出 ODL 架构清晰、功能合理,便于进行功能模块的扩展和实际部署。

**关键词** SDN, OpenDaylight, Openflow, LISP, 控制层

中图法分类号 TP393 文献标识码 A

## Research and Analysis of OpenDaylight Controller

XU Ming-guang<sup>1</sup> LIU Ya-ping<sup>1</sup> DENG Wen-ping<sup>2</sup>

(Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073, China)<sup>1</sup>

(College of Computer National University of Defense Technology, Changsha 410073, China)<sup>2</sup>

**Abstract** OpenDaylight controller(ODL) is a new open source SDN controller. It is divided into three layers: southbound protocol layer, control layer and northbound interface layer. Southbound protocol layer supports Openflow, LISP and other protocols. Control layer has complete base network services. Northbound provides REST API for web application to access these services. Meanwhile, it supports cluster mode and can be applied to large-scale networks. By analyzing the overall framework of the ODL, work processes, module extensions, and LISP services, an experiment on LISP communication system can be built. We can see that ODL has clear architecture and reasonable function. It is also easy to expand and actual deployment.

**Keywords** SDN, OpenDaylight, Openflow, LISP, Control layer

## 1 概述

软件定义网络(Software Defined Network, SDN)<sup>[1]</sup>分为 3 层:最底层为数据转发平面,提供物理连接网络并根据转发规则转发数据;中间是控制平面,它是一个软件层,负责提供网络基础服务,提升网络的可编程性,使网络变得敏捷、高效;最上层为应用层,主要负责网络的管理、优化和使用。

ODL 处在 SDN 的控制平面,具备拓扑管理、交换机管理、路径转发管理、主机管理、网络资源切片管理这 5 个基础功能,支持 Openflow<sup>[2]</sup>、LISP(Locator/ID Separation Protocol)<sup>[3]</sup>、BGP(Border Gateway Protocol)<sup>[4]</sup>等多种协议,多个控制器之间可以采用集群的模式进行工作。ODL 采用 OSGi 作为实现架构,使用 Maven 管理代码,允许在控制器运行时进行功能模块的安装、删除和更新。与其它控制器相比,ODL 在设计上灵活,扩展性好;在功能上,ODL 支持更多的协议,具备更为完备的服务功能。而且 ODL 得到微软、思科、IBM

等公司的支持,更有可能在实际中得到广泛的应用。

本文通过对 ODL 代码进行分析,具体描述了 ODL 的总体框架、工作流程以及功能模块扩展。并分析了 ODL 中的 Openflow 协议插件、LISP 协议插件和 LISP Service,以及控制层的网络基础服务的主要功能模块。

本文第 2 节分析 ODL 的实现机制,包括 ODL 总体框架、工作流程、模块扩展以及 ODL 支持的 LISP 服务功能;第 3 节使用 ODL 搭建了 LISP 实验系统,对 ODL 的 LISP 服务进行验证;最后是对本文的总结。

## 2 ODL 实现分析

### 2.1 总体框架

ODL 在设计上,自底向上分为 4 层:南向协议层、服务抽象层、控制层,以及北向接口层。框架视图如图 1 所示。

南向协议层包含 Openflow、LISP、GBP 等协议插件,负责与实际的网络设备进行通信、收集设备信息、监听相应端口、

本文受 973 项目:可重构基础网络的验证平台(2012CB315906),国家自然科学基金项目:抵御违反路由策略攻击的可验证安全路由协议的研究(61472438)资助。

许名广(1990—),男,硕士生,主要研究方向为 SDN 技术, E-mail:mdsmqh@126.com;刘亚萍(1973—),女,博士,研究员,主要研究方向为网络体系结构、路由技术、SDN 技术、无线网络技术等, E-mail:ypliu@nudt.edu.cn;邓文平(1982—),男,博士,助理研究员, E-mail:wsfdwp@163.com。

处理有关的控制和数据报文、向控制层提供配置网络设备和获取网络设信息的接口。

服务抽象层 (Service Layer, SAL) 作为整个框架设计的核心,屏蔽了不同南向协议插件之间的差异,负责接口路由和适配的工作,使得南向协议层呈现统一的北向接口供控制层调用。在本文分析的控制器的版本中,SAL 实现上又分为 AD-SAL (API-Driven SAL) 和 MD-SAL (Model-Driven SAL)。MD-SAL 实现了接口的动态匹配,并且网络应用程序可以通过 RESTCONF/NETCONF 模块提供的服务对 MD-SAL 中的数据节点进行操作,MD-SAL 会通知数据节点的监听者进行处理,以达到配置网络设备的目的。而 Base Network Service Functions 中的功能模块同 Openflow 协议插件的交互采用了 AD-SAL 的实现方式。

控制层主要包括在 Openflow 等协议插件的基础上形成的 6 个基础服务: Switch Manager、Statistics Manager、Topology Manager、Forwarding Rule Manager、ARP Handler 和 Host Tracker,分别负责交换机管理、信息统计、网络拓扑管理、流表管理、ARP 报文处理以及主机追踪,这 6 个模块之间相互联系,并为其它的功能模块提供服务接口。控制层中还包括一些与南向协议相对应的特有的功能模块,如 LISP Service、BGP Service 等模块。还有一些网络业务编排和服务功能管理的模块,如 OpenStack Service、VTN Manager 等。

北向接口层为网络应用程序提供访问的接口,ODL 采用 REST API。应用程序可通过 REST API 对流表、设备连接等事件进行操作。扩展的功能模块可以自己完成北向接口,也可以使用 MD-SAL 方式来提供应用程序访问的接口。

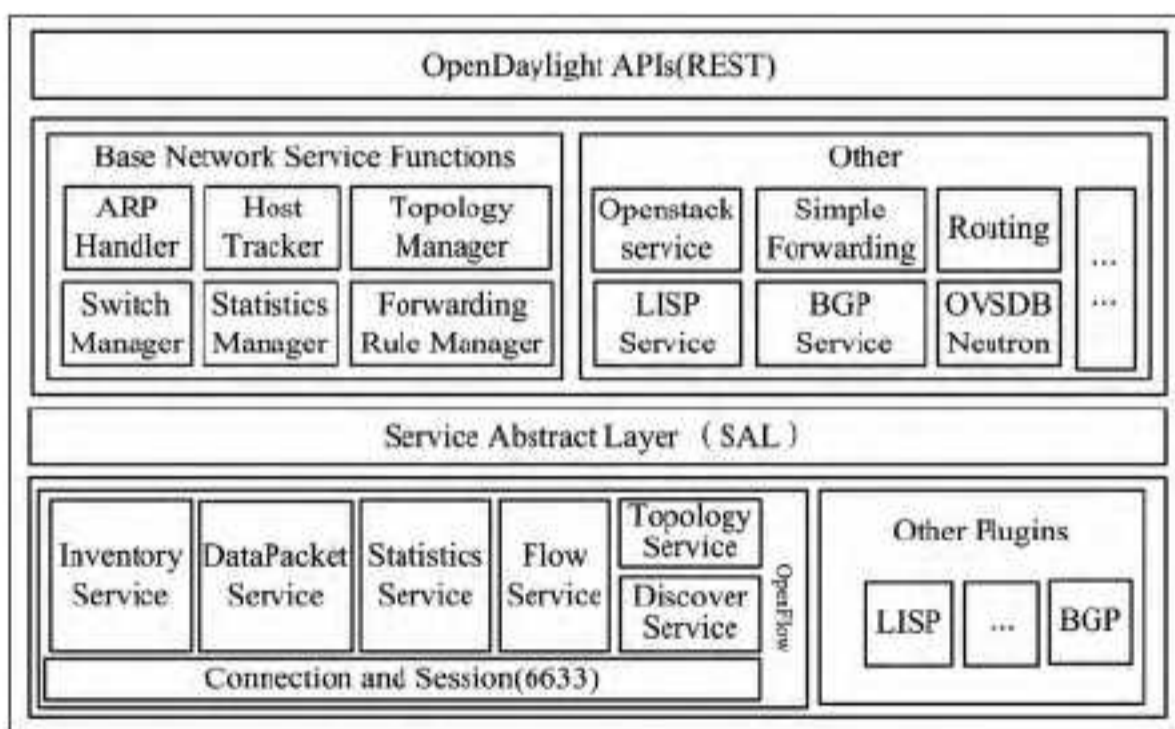


图 1 ODL 架构及功能图

## 2.2 工作流程分析

ODL 中,每个模块可以调用其它模块暴露的接口,一般上层模块调用下层模块的接口获取服务,下层模块调用上层模块的接口进行事件通知。

本文以网络拓扑的形成为例,分析 ODL 如何获取底层网络设备的信息,从而形成网络拓扑结构的流程。由于 SAL 主要负责接口路由和适配的工作。为了简单,图中将省略 SAL 及其它辅助的功能模块。网络拓扑形成过程如图 2 所示。

步骤 1—9 是 ODL 通过 Openflow 协议插件发送 LLDP<sup>[5]</sup> 报文获取交换机的连接拓扑,其中步骤 1—3 为 Discover Service 对新连接的交换机发送 LLDP 探测报文,步骤 4 为交换机收到报文后向与它相邻的交换机转发该报文,步骤 5—7 为交换机将 LLDP 报文打包发送给 Discover Service;步骤 8、9 为 Discover Service 形成一条新的边并通过 Topology Service 交给 Topology Manager;步骤 10—15 通过 ARP 报文获取主机与交换机链接的拓扑,其中步骤 10—13 为交换机收到主

机的 ARP 报文后将其打包转发给 ODL 并交由 ARP Handler 处理,步骤 14 为 ARP Handler 从收到的报文中提取出主机 IP 和与交换机的端口信息,并将这些信息交给 Host Tracker;步骤 15 为 Host Tracker 形成主机与交换机链接的边交给 Topology Manager。Topology Manager 再将这两种类型的边结合起来,形成整个网络的拓扑结构。

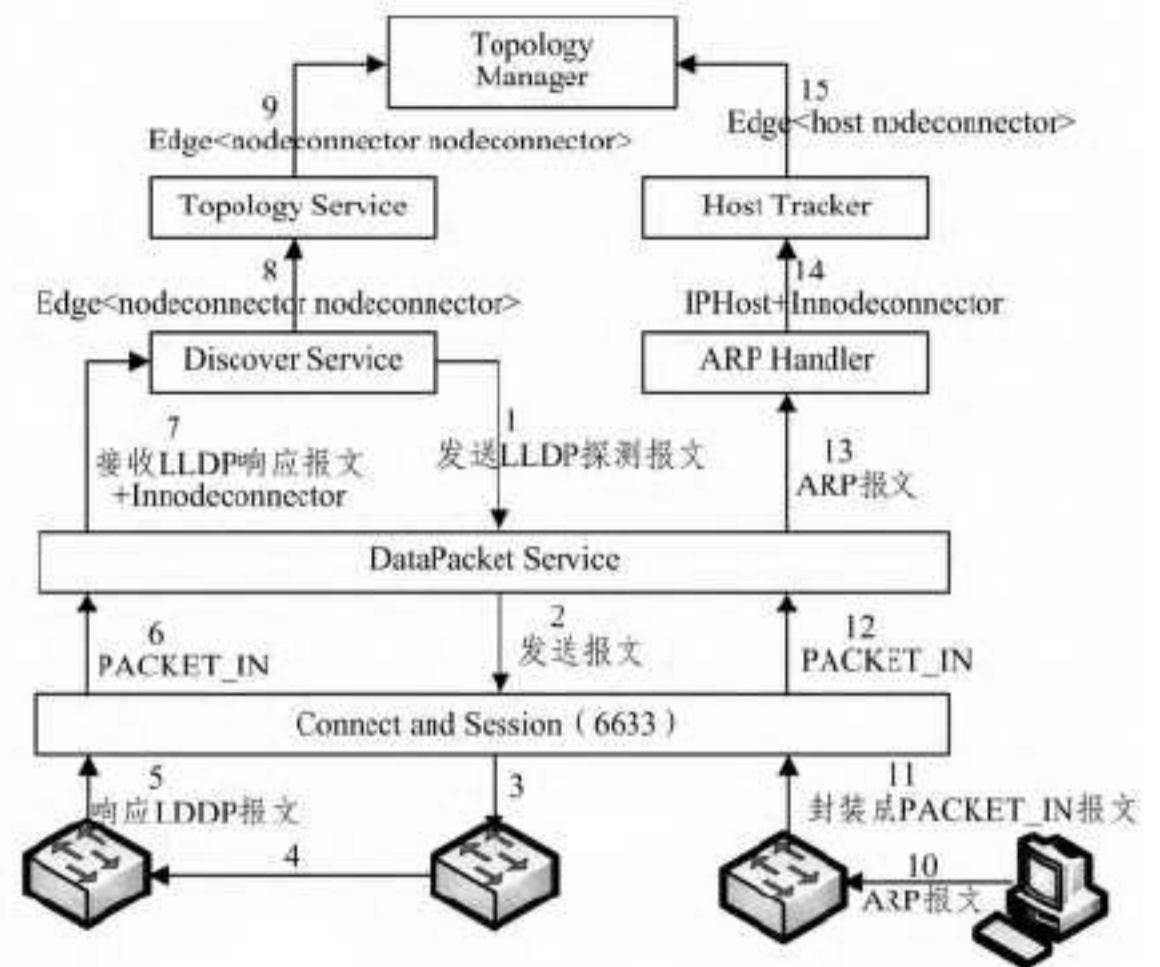


图 2 网络拓扑形成过程

## 2.3 ODL 功能扩展

对 ODL 功能扩展大致分为两种情况(见图 1):1) 在原有功能模块的基础上,只在控制层添加模块。比如 ODL 中的 Routing 模块,就是在 Topology Manager 提供的服务的基础上,采用 dijkstra 算法来获取两个交换机之间的最短路径;Simple Forwarding 模块使用 Routing 模块计算的路径,调用 Forwarding Rule Manager 的服务下发流表,形成一个简单的 IP 转发网络。2) 对协议的扩展,需要在南向协议层增加协议插件模块,负责监听协议端口、收发相关报文,并完成报文到相关数据结构的序列化与逆序列化;在控制层需要增加具体的处理模块,这两个模块之间通过 SAL 进行交互,建议使用 MD-SAL 的模式。如 ODL 的 LISP 功能,南向的 LISP 协议插件监听了 4342 (LISP 控制报文端口),然后将接收到的 LISP 控制报文经由 MD-SAL 交给控制层的 LISP Service 进行处理。

另外,如果扩展的模块需要对外部应用程序提供访问接口,需要实现北向的 REST API。

## 2.4 LISP 功能分析

ODL 中提供的 LISP 服务是基于 rfc-6830 标准实现的,功能视图如图 3 所示。

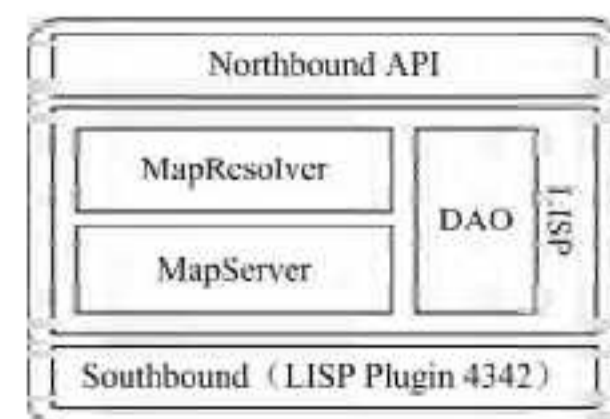


图 3 LISP 服务视图

南向层的 LISP 协议插件监听了 4342 端口,接收注册 MapRegister<sup>[3]</sup> 和查询 MapRequest<sup>[3]</sup> 控制报文,对报文进行简单处理后,将报文交给 LISP Service 处理。

LISP Service 主要分为两个模块,MapResolver 和 MapServer。MapResolver 负责处理 MapRequest 报文,MapServer

负责处理 MapRegister 报文,然后调用 LISP 协议插件提供的服务将处理的结果 MapReply<sup>[3]</sup>和 MapNotify<sup>[3]</sup>返回给网络设备。LISP Service 在北向接口层提供了 REST API,具备分配 ID、添加映射等功能。另外,DAO 模块提供访问数据库的接口。目前在 ODL 中,所有的映射数据存放在内存中。

ODL 中 LISP 主要存在以下两个不足:1)没有采用数据库系统,所有的映射存储在内存中,这难以满足大规模部署的需求。2)MapServer 和 MapResolver 集成到了一起,理论上它们可以分开实现,部署在不同的机器中,可以使用配置文件来决定启动哪一服务或者都启动。

### 3 LISP 功能验证

本文搭建如图 4 的实验系统对 ODL 中提供的 LISP 服务进行验证。

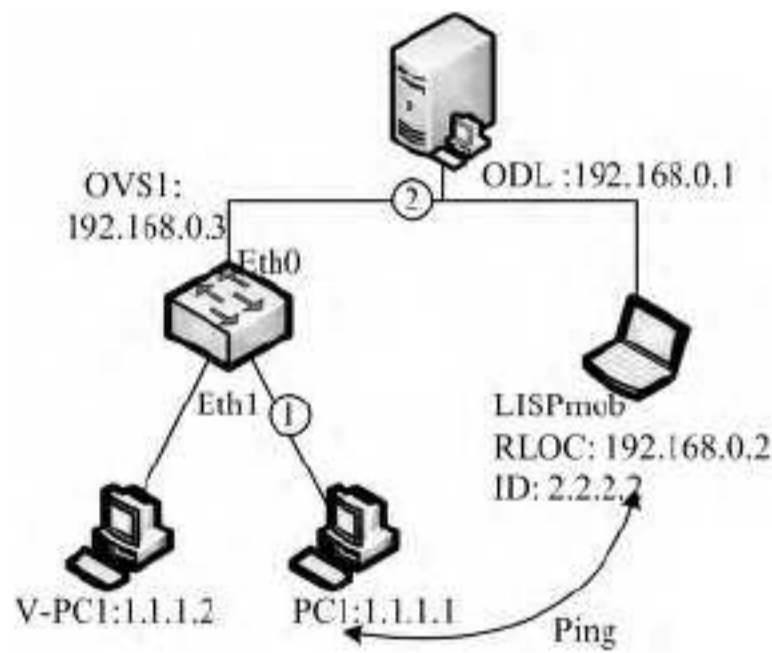


图 4 LISP 实验环境

实验环境说明:本实验环境使用了一台 IBM 服务器、一台双网卡主机、一台单网卡主机和一台笔记本电脑。其中,ODL 运行在 IBM 服务器上,OVS 运行在双网卡主机上,一个网卡和 ODL 连接,另一个和端主机连接,端主机除了本身和 OVS 连接外,还运行一台虚拟机 V-PC1,需要将 V-PC1 设置为 PC1 的默认网关,LISPmob<sup>[6]</sup>运行在笔记本电脑上。

实验步骤:1)配置并启动 ODL、OVS 以及 LISPmob;2)配置 PC1 的 LISP 转发流表到 OVS1 中;3)用 LISPmob ping PC1。具体步骤请参照:<https://wiki.OpenDaylight.org/view/OpenDaylight-Lisp-Flow-Mapping:User-Guide-for-Hydrogen>。

验证方法:1)在 OVS 中输入流表查看命令,看是否由 ODL 下发的基于 LISP 的转发流表;2)用 PC1 ping LISPmob,在图 4 中 1、2 处抓包,看是否存在 LISP 封装与解封装的操作后的报文以及 LISPmob 发给控制器的 LISP 控制报文。

实验结果:

(1)查看 OVS1 中的流表项,结果如图 5 所示。其中第 1、2 两条流表是有 Simple Forwarding 模块下发的 IP 转发流表,将目的地址为 1.1.1.1 和 1.1.1.2 的 IP 报文从端口 1 输出;第 3、4 条流表是手动配置的静态流表,分别是将目的地址为 2.2.2.2 的 IP 报文进行 LISP 封装,以及将从 LISP 隧道进入数据转发平面的报文添加一个以太网目的地址为 PC1 MAC 地址的报文头,并从端口 1 发出。

```
Foot@qh:/home/heh/openvswitch-1.11.0# ovs-ofctl dump-flows br0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=44.273s, table=0, n_packets=1, n_bytes=98, idle_age=43, pr
iority=1,ip,nw_dst=1.1.1.2 actions=mod_dl_dst:00:50:56:34:3a:93,output:1
 cookie=0x0, duration=311.47s, table=0, n_packets=0, n_bytes=0, idle_age=311, pr
iority=1,ip,nw_dst=1.1.1.1 actions=mod_dl_dst:00:1d:7d:50:b5:7c,output:1
 cookie=0x0, duration=592.73s, table=0, n_packets=0, n_bytes=592, idle_age=2, pr
iority=5,ip,in_port=1,vlan_tci=0x0000,nw_dst=2.2.2.2 actions=output:2
 cookie=0x0, duration=664.887s, table=0, n_packets=12, n_bytes=888, idle_age=2,
priority=10,dl_dst=02:00:00:00:00:00 actions=mod_dl_dst:00:1d:7d:50:b5:7c,output
:1
Foot@qh:/home/heh/openvswitch-1.11.0#
```

图 5 流表项

(2)在图 4 中 1、2 两个节点处进行报文捕获,结果如下:节点 1 的报文见图 6、节点 2 的报文见图 7(为了更清晰地展示通信过程,只将捕获的用于 LISP 通信的报文以文本的形式输出,并省略了 Wireshark 中的 Time 属性)。从图中可以看到整个 LISP 的通信过程,当 LISPmob ping PC1 时,会先向 ODL 发出 LISP Request 控制报文,ODL 收到后会返回 LISP Reply 报文,获得目的 RLOC(Routing Locator)<sup>[3]</sup>后,LISPmob 便可以将 ping 报文进行 LISP 封装并发给 OVS1。

No.	Src	Dst	Protocol	Info
38	1.1.1.1	2.2.2.2	ICMP	Echo(ping) request(id=0x0400, seq(be/le)=5632/22,ttl=64)
39	2.2.2.2	1.1.1.1	ICMP	Echo(ping) reply(id=0x0400,seq (be/le)=5632/22,ttl=64)
42	1.1.1.1	2.2.2.2	ICMP	Echo(ping) request(id=0x0400, seq(be/le)=5888/23,ttl=64)
43	2.2.2.2	1.1.1.1	ICMP	Echo(ping) reply(id=0x0400,seq (be/le)=5888/23,ttl=64)

图 6 节点 1 报文

No.	Src	Dst	Protocol	Info
43	192.168.0.2	192.168.0.1	UDP	Source port:lisp-control Destination port:lisp-control
44	192.168.0.1	192.168.0.2	UDP	Source port:lisp-control Destination port:lisp-control
46	192.168.0.2	192.168.0.3	UDP	Source port:lisp-data Destination port:lisp-data
49	192.168.0.3	192.168.0.2	UDP	Source port:52646 Destination port:lisp-data
55	192.168.0.2	192.168.0.3	UDP	Source port:lisp-data Destination port:lisp-data
56	192.168.0.3	192.168.0.2	UDP	Source port:52646 Destination port:lisp-data

图 7 节点 2 报文

实验表明 ODL 的 LISP 服务功能完整,可响应 LISPmob 发出的 LISP 控制报文。但整个实验系统存在以下缺点:OVS 端没有发送和处理 LISP 控制报文的能力,需要手动配置流表;ODL 端也没有数据库的支持。

结束语 本文主要分析了 ODL 这款全新的网络控制器实现机制,包括 ODL 基本设计架构、各层之间的关系、ODL 的工作原理以及如何 ODL 中扩展功能模块,并对 ODL 中的 LISP 功能介绍,同时搭建了一个验证 LISP 功能的实验环境。可以看到 ODL 设计上层次清晰,结构简单,具备良好的可扩展性。无论从功能上还是其南向协议层所支持的协议种类上,ODL 比其它控制器更为完整、全面,能够在更多的场合使用。ODL 支持集群模式,这也为 SDN 的大规模部署提供了一种方案。

在后续的工作中,将对 ODL 中一些重要的扩展功能的实现机制以及多个控制器之间如何协同工作进行具体的研究,并解决实验系统中所面临的问题,实现 LISP 自动通信的功能。

### 参考文献

[1] McKeown N. Software-defined networking[J]. INFOCOM keynote talk,2009,17(2):30-32  
 [2] McKeown N,Anderson T,Balakrishnan H,et al. Openflow:ena-

- bling innovation in campus networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74
- [3] Meyer D, Lewis D. The Locator/ID Separation Protocol (LISP) [J]. IETF RFC, 2013, 6830
- [4] Lougheed K, Rekhter Y. A border gateway protocol 3 (BGP-3) [R]. RFC 1267, cisco Systems, TJ Watson Research Center, IBM Corp, 1991
- [5] IEEE Standard. LLDP [OL]. <http://standards.ieee.org/get-ieee802/download/802.1AB-2009.pdf>
- [6] LISPmob 2014 [OL]. <http://lispmob.org/>
- [7] Kataoka K, Agarwal N, Kamath A V. Scaling a broadcast domain of Ethernet: Extensible transparent filter using SDN [C] // 2014 23rd International Conference on Computer Communication and Networks (ICCCN). IEEE, 2014: 1-8
- [8] 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
- [9] Medved J, Varga R, Tkacik A, et al. OpenDaylight: Towards a Model-Driven SDN Controller architecture [C] // 2014 IEEE 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE, 2014: 1-6
- [10] Pongrácz G, Molnár L, Kis Z L, et al. Cheap silicon: a myth or reality? picking the right data plane hardware for software defined networking [C] // Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013: 103-108
- [11] OpenDaylight 2014 [OL]. <http://www.opendaylight.org/>
- [12] 陈鸣, 赵广松, 邢长友, 等. 基于 Openflow 的 SDN 技术研究 [J]. 软件学报, 2013, 24(5): 1078-1097
- [13] Nadeau D T, Gray K, 等. 软件定义网络 [M]. 北京: 人民邮电出版社, 2014
- [14] 江国龙, 付斌章, 陈明宇, 等. SDN 控制器的调研和量化分析 [J]. Journal of Frontiers of Computer Science and Technology, 2014, 8(6): 653-664

(上接第 240 页)

图 2 中, 第三幅图像标注词“草”是根据计算标注词数据集中与“大象”和“树木”相关度之后添加的标注。

结束语 本文提出了一种利用标注词相关性改善标注结果的方法。首先, 我们通过  $N\text{-Cut}$  方法把图像分割成若干个同质区域, 然后用  $k\text{-means}$  方法对分割后的区域进行聚类。根据已标注图像计算出区域与标注词的相关性, 根据相关性对未知图像进行标注, 最后利用  $WUP$  方法计算标注词之间的相似度改善标注结果, 实验结果显示, 该方法能够有效地对图像进行自动标注。

### 参 考 文 献

- [1] Tmura H, Yokoya N. Image systems: A survey [J]. Pattern Recognition, 1984, 17(1): 29-43
- [2] Kato T. Database architecture for content-based image retrieval [C] // Proc. Of SPIE Int. Conf. on Image Storage and Retrieval System, San Jose, CA, USA, May. 1992: 112-123
- [3] Jeon J, Manamatha R. Automatic image annotation of news images with large vocabularies and low quality training data [C] // ACM Multimedia. New York, USA, Oct. 2004
- [4] Carneiro G, Chan A B, Moreno P J, et al. Supervised Learning of Semantic Classes for Image Annotation and Retrieval [C] // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007, 29: 394-410
- [5] Hollink L, Schreiber G, Wielemaker J, et al. Semantic Annotation of Image Collections [C] // Workshop on Knowledge Markup and Semantic Annotation (KCAP 2003). 2003
- [6] Jeon J, Lavrenko V, Manmatha R. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models [C] // Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2003
- [7] Escalante H J, Hernández C A, Gonzalez J A, et al. The segmented and annotated IAPR TC-12 benchmark [C] // Computer Vision and Image Understanding. 2010, 114: 419-428
- [8] Jeon J, Lavrenko V, Manmatha R. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models [C] // 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2003: 119-126
- [9] Ko B C, Byun H. Frisp: a region-based image retrieval tool using automatic image segmentation and stepwise Boolean and matching [C] // IEEE Trans. on multimedia. 2005, 7: 105-113
- [10] Liu Y, Zhang D, Lu G. Region-based image retrieval with high-level semantics using decision tree learning [J]. Pattern Recognition, 2008, 41(8): 2554-2570
- [11] Yining D, Manjunath B S, Shin H. Color image segmentation [C] // Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1999: 446-451
- [12] Jianbo S, Jitendra M. Normalized cuts and image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905
- [13] Çavuş, Aksoy S. Semantic Scene Classification for Image Annotation and Retrieval [J]. LNCS, 2008, 5342: 402-410
- [14] Wang L, Liu L, Khan L. Automatic Image Annotation and Retrieval using Subspace Clustering Algorithm [C] // 2nd ACM International Workshop on Multimedia Databases. 2004: 100-108
- [15] Li J, Wang J Z. Real-Time Computerized Annotation of Pictures [C] // ACM Multimedia Conference. 2006: 911-920
- [16] Z Xin, W Xian-bin, M Qing-xia. Research of Image Retrieval Method Based on Color Feature [J]. Computer Science, 2012, 39(11): 243-260
- [17] 王娟, 孔兵, 贾巧丽. 基于颜色特征的图像检索技术 [J]. 计算机系统应用, 2011, 20(7): 160-164
- [18] Mikolajczyk K, Cordelia S. A performance evaluation of local descriptors [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1615-1630
- [19] Deselaers T, Ferrari V. Visual and Semantic Similarity in ImageNet [C] // CVPR 2011. 2011
- [20] Wu Z, Palmer M. Verb Semantics and Lexical Selection [C] // Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL 1994). 1994: 133-138