基于 OpenFlow 技术的流量分发管理系统

王同同，曹争

（东南大学信息科学与工程学院，南京，210096）

摘　要： SDN（软件定义网络）成为时下新的网络架构的潮流，本文首先介绍当前网络发展和面临的问题，然后介绍 SDN 网络的基本架构和特性。随后在 Mininet 仿真工具搭建的试验环境上，基于 Ryu 控制器开发一个 SDN 网络流量分发管理系统，该系统不仅能将网络拓扑和网络中元素的信息实时展现给用户，更可以允许用户根据当前网络流量状况下发路径策略，实现用户可编程网络。

关键词： SDN 网络；Ryu 控制器；策略下发；可编程网络

The Traffic Distribution Management System based on OpenFlow Technology

WangTongTong, CaoZheng

(School of Information Engineering, Southeast University, Nanjing, 210096)

Abstract: Nowadays, SDN (Software Defined Network) has become the trend of new network architecture. This paper describes the current problems of network development, and presents the basic architecture and features of SDN network. Then, a distribution network traffic management system is designed based on Ryu controller, so that users can not only view the network information, but also issued network policies according to their needs.

Key words: SDN network; Ryu controller; Issue policies; Programmable network

计算机网络的发展历经从独立的一台计算机到以分组交换网为中心的计算机网络，再到遵循开放互联参考模型实现计算机网络互联，最后发展成全球范围的互联网。基于 TCP/IP 协议的 OSI 模型将各种网络设备连接在一起，实现了信息通信、数据共享、分布式处理等功能。

随着网络业务和应用的不断增加，我们不断在网络设备中集成新的功能，从在路由器上实现各种路由协议到增设防火墙或者进行流量过滤乃至服务质量（QoS）、Vlan 通信等等，虽然我们解决了业务需求，却同时使得原本简单的网络设备变得越来越臃肿、复杂，管理控制变得越来越弱。虽然设备的性能提升和功能扩展是厂家关注的焦点，但出于市场竞争的缘故，设备的对外开放能力还比较弱，研发人员也就难以在真正网络环境中部署和实验。

作者简介： 王同同，（1990-），男，硕士研究生， E-mail: ttwang@njnet.edu.cn；曹争，（1958-），男，副教授，E-mail: zcao@njnet.deu.cn

如今，随着云计算、大数据的兴起，相应业务对网络实时配置的频率、速度提出了更高的要求，能按需调度网络资源变得至关重要，为了解决网络瓶颈，世界各国也展开各种研究。在这样的背景下，2007 年斯坦福大学的 NickMcKeown[1]在他的文章中首次提到了 OpenFlow 技术，自此 OpenFlow 开始为世人所关注。OpenFlow[2]的思想是将设备的控制平面和数据转发层面完全分离，控制平面负责网络全局的决策，而数据层面专注于实现高速数据转发，两者之间的通信就是 OpenFlow 协议的事情。随后在 ONF[3]（开放网络基金会）的推动下提出软件定义网络的概念。基于 OpenFlow 的 SDN[4]技术,通过软件平台来打造弹性化的可控互联网,在给网络的发展带来巨大冲击的同时,也为未来互联网的发展提供了一种新的解决思路.

1 系统目标

1.1 技术背景

SDN 被评为当今“十大前沿技术之一 ”，其主

要框架[5]由上到下可分为应用平面、控制平面和数据平面以及右侧的管理平面。其主要特性：一是数据控制分离，二是实现网络可编程性。数控分离是SDN 的核心思想之一，不仅使用户获得全局网络视图，实现网络集中管控，而且可以保证数据的高速转发能力。网络的可编程性在于 SDN 为开发者提供了强大的 API，这主要体现在北向接口上。随着网络需求的增多，传统网络架构的僵化已很难适应业务对网络资源调配提出的高灵活性和快速性。而 SDN 的开放性，特别是其北向接口却为业务创新带来了机会。

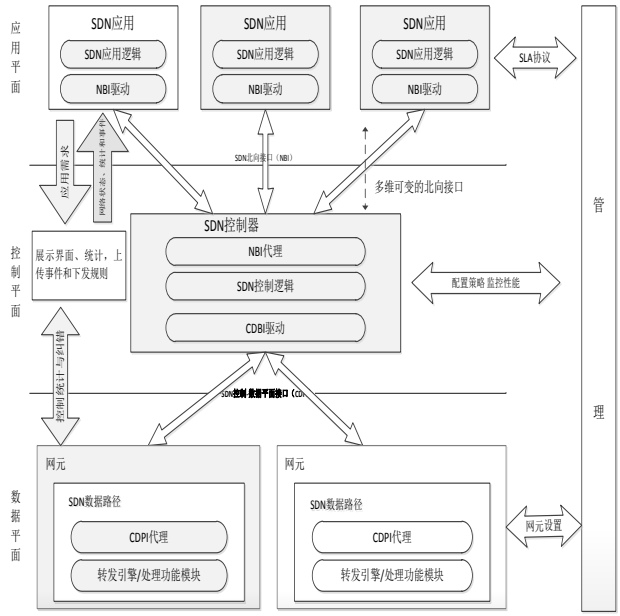

图 1 SDN 框架结构图

1.2 研究内容

SDN 的核心技术集中在负责数据转发的交换机及南向接口、核心位置的控制器及北向接口、应用编排和资源管理三个方向。

OpenFlow 和 OF-CONFIG[6]是南向接口领域的典型代表。然而，处于网络核心地位的控制器有多种，也意味着北向接口的多样性，SDN 之所以能推动网络业务创新，主要推动力在于其北向接口。虽然很多网络厂商为了应对开放协议选择在传统设备上提供一定的对外编程接口，但仍然是一个封闭的解决方案。本文基于当前主流控制器 Ryu[7]，利用其提供的 REST API[8]开发一个面向用户的 Web 应用，通过该系统用户不仅可以方便的获知网络资源的工作状态，还能快速的实现对网络资源的调度。

通过实验不仅体现北向接口开放性为上层应用开发带来的便捷，也更加明确北向接口设计合理性和网络能力开放的充分性会直接影响上层业务开发的便捷与否，也影响着 SDN 控制技术和产品的前景。

2 系统设计

2.1 系统结构

整个系统大致上可分为 2 大模块，系统总体设计如图 2，根据图 2 所示，一个模块是前端部分，主要有三个组成部分：系统日志，数据处理，界面显示。另一个模块就是后台模块，主要部分是请求处理和控制器端。系统的关键部分就是数据处理和请求处理两大部分，将用户需求数据传递到控制器并保证控制器执行。

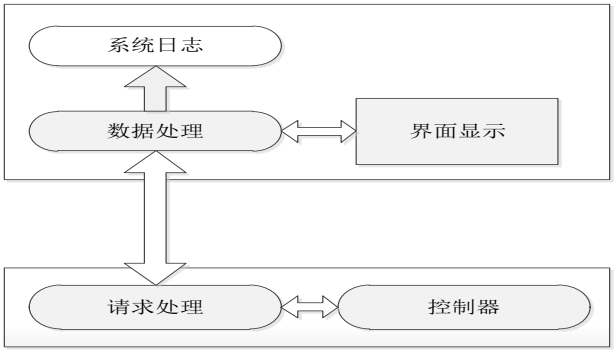

图 2 系统总体设计

控制器端是底层网络拓扑，实验环境包括一台 ryu 控制器，控制器控制着三台 OVS[9]交换机，其中有两台分别部署一台 PC，实验网络拓扑如图 3。

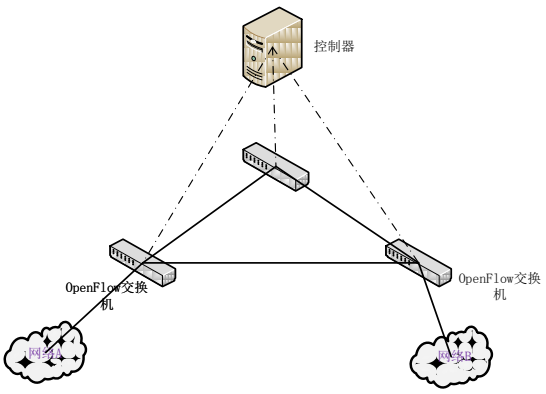

图 3 网络拓扑结构

2.2 功能模块

2.1.1 前台模块

1) 数据处理

该模块是前台的关键部分，负责检查参数，将用户请求转发到后台，接收响应数据，并将处理后的数据交给界面部分来显示或者存为日志。

2) 界面显示

界面显示部分是面向用户的，要将网络实时状态直观的显示给用户，同时，既要允许用户查询各个网元设备信息，又满足用户可以下发规则进行资源调度的需求。

3） 系统日志

一方面负责记录用户执行的重要操作，另一方面，交换机向控制器发送一些重要的事件信息，需要将这些事件提取上来，存为日志。

2.1.2 后台模块

1) 请求处理

该模块是后台的主要部分，负责监听请求，接收请求后分析消息，处理发往控制器；将控制器的响应数据返回前台。

2) 控制器

所有的动作（增加、删除流表等）由控制器下发到其下的交换机，所有的网元信息（流表、端口信息等）都从控制器获取。

3 关键技术设计

3.1 后台实现

实现用户通过网页来操作控制器，需要在后台搭建一个 web 服务器接收 http 请求，接收到请求后解析请求再将其发往注册的 application 进行处理，最后将处理结果返回客户端。Ryu 控制器是基于 python 开发的，ryu 使用 eventlet 库来提供 WSGI 对应的 web 服务器，整个后台就是通过 wsgi 和 webob 以及 routes 来实现的。

3.1.1 wsgi 服务

eventlet[10]是 python 的一个高并发网络编程库，提供了协程、wsgi 等实用功能。Wsgi(Web Server Gateway Interface)是一种接口规范，统一 web server 和 application 之间的接口，当服务器接收到 web 请求后把请求转给 application。

导入 eventlet 中的 wsgi，对其进行抽象和封装成应用中的 WSGIServer 服务类，在 wsgi.py 中使用该类启动服务，同时将类 WSGIApplication 注册为接收请求的应用。

3.1.2 webob 封装请求和响应

Webob 是一个 python 库，它主要是对 WSGI 应用请求的环境变量（request environment）封装成 Request，同时创建一个 Response 对象来返回响应。封装了 environ 的 Request 提供不少方法和属性以方便操作数据，比如获取 http 请求的方法等。

3.1.3 routes 映射路由

Routes 是用 python 重新实现的 Rails routes 系统，用来映射 url 与应用程序的系统。通过 routes 可以在 utl 和应用函数之间建立映射关系，当接收到用户请求时，可以匹配到对应函数进行处理请求。

3.2 界面功能

界面展示了用户可以执行的具体操作，直观显示内容包括：动态显示整个底层网络拓扑、OVS 交换机基本信息（dpid、ip、端口号）、交换机端口的流量统计信息；能够查询的信息：OVS 交换机的流表信息；能够执行的操作：根据参数下发流表。

3.2.1 网络拓扑界面

首先要和服务器建立连接，使用 WebSocket 协议，WebSocket 协议是一种双向通信协议，它建立在 TCP 之上，同 http 一样通过 TCP 来传输数据，但是它和 http 最大的不同有两点：1.WebSocket 是一种双向通信协议；2.WebSocket 需要通过握手连接，类似于 TCP 它也需要客户端和服务器端进行握手连接，连接成功后才能相互通信。

服务器端应用的返回数据使用 JSON 格式，JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，易于阅读和编写。数据的可视化使用 D3.js 来实现，D3.js 是一个用于网页作图、生成互动图形的 JavaScript 函数库，也是最流行的可视化库之一。

3.2.2 OVS 交换机操作

OVS 交换机的静态信息（如 dpid、ip 等）在显示网络拓扑的请求响应数据中可以获取。交换机的端口流量统计数据获取：创建 ryu 应用，利用 eventlet 的 greenlet 创建一个线程，每隔一段时间向 OVS 交换机发送端口状态请求事件查询统计数据，利用之前的 websocket 连接将数据传送到客户端，显示给用户。

另一个重要信息就是流表，流表是交换机的转发规则，用户可以根据交换机的 dpid 或者 ip 来实时获取对应的流表，可以添加过滤条件（如源地址、端口号等）来查看部分流表。

流表下发是用户对网络流量调度的主要手段，用户指定交换机 dpid 或者 ip，指定操作（增加、删除等），指定匹配域内容（数据报文入端口、源地址协议类型等），指定 action（丢弃、OUTPUT 等），最后将数据发往控制器，由控制器向交换机下发对应流表，之后可以通过查看流表功能判断是否成功添加流表。

4 结束语

SDN（软件定义网络）是针对当前网络发展遇到的瓶颈所提出的一种新的网络架构。本文是以 Ryu 控制器为网络核心，开发的一个网络流量管理系统，通过用户界面实现对整个网络的观测，用户可以自定义转发策略调度网络流量。透过该系统，也可以体现出 SDN 架构下对网络设备和资源的调度、配置更为方便和快捷。

参考文献

[1] Nick Mckeown, Tom Anderson, Hari Balakrishnan,et al. OpenFlow:enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008. 69-74.

[2] 左青云, 陈鸣, 赵广松等. 基于 OpenFlow 的 SDN 技术[J]. 软件学报,2013. http://www.jos.org.cn/1000-9825/43-90.htm.

[3] Open networking foundation. https://www.opennetworking.org/.

[4] Open Networking Foundation. Software-Defined Network-Ing:The New Norm for NetWorks[S]. Apri.13,2012.

[5] 张朝昆, 崔勇, 唐翯祎等.软件定义网络研究进展[J]. 软件学报,2014. http://www.jos.org.cn/1000-9825/4701.htm

[6] Open Networking Foundation. OpenFlow Configuration and Management Protocol Version 1.1.1[S]. March. 23,2013.

[7] 张卫峰. 深度解析 SDN:利益、战略、技术、实践[M]. 电子工业出版社, 2014.

[8] 雷葆华, 王峰, 王茜等. SDN 核心技术剖析和实战指南[M]. 北京: 电子工业出版社, 2013.

[9] Open Networking Foundation. Openflow Switch Specifi-cAtion Version 1.3.2[S]. April 25,2013.

[10] RYU SDN FRAMEWORK[EB].