OpenFlow 网络中组播技术的实现

张敏[1]，曹争[1]

（1.东南大学计算机科学与工程学院，南京，211189）

摘　要：在传统网络向软件定义网络过渡的过程中，不可避免地会出现新旧网络共存的情况。然而传统网络中的协议在软件定义网络中是不支持的。组播技术能有效解决单点发送、多点接收的问题，从而实现网络中点到多点高效的数据传输，是应用非常广泛的技术。本文的研究目标是在不改变现有传统网络中组播协议的情况下，在 OpenFlow 网络中实现组播流的转发。

关键词：OpenFlow；组播

The Realization of Multicast Technology in OpenFlow Network

Zhang Min[1], Cao Zheng[1]

(1. School of Computer Science and Engineering, Southeast University, Nanjing, 211189)

Abstract: In the process of transition of traditional network to Software Defined Network (SDN), the inevitable coexistence of them would happen. However, traditional protocols under the environment of SDN are not supported. Multicast technology can effectively solve the problem of single point to send, multi point to receive, so as to realize the data transmission effectively. It's a widely used technology. The aim of this article is whithout changing the traditional multicast protocol, to implement forwarding of multicast flow in OpenFlow network.

Key words: OpenFlow; Multicast

从上个世纪 70 年代到现在，新的业务和新的应用的不断出现，网络从最初的端到端模型发展到现在如云计算等复杂场景，网络协议从最初的 OSI 7 层参考模型和简单的 TCP/IP 协议发展到现在的 TCP/IP 协议簇以及数千种协议。从技术发展来看，网络本身的架构几乎没有突破性的变化，一直是以“打补丁”的形式来解决新的需求。这样致使整个网络越来越复杂。但是设备制造和关键技术都掌握在少数大公司的手中，这就制约了新技术和新协议的发展。于是数据与控制分离的思想被逐渐提出来。2008 年，Nick McKeown 在 SIGCOMM 会议上发表文章“OpenFlow: Enabling Innovation in Campus Networks”[1]，首次提出将 OpenFlow 协议应用于校园网络的创新试验。OpenFlow 协议是一种数据平面与控制平面之间的交互协议。它使得网络具有高度的灵活性和可编程能力，是 SDN（Software Defined Networking， 软件定义网络）架构中南向的核心技术。从此 SDN 被越来越多的人所关注。

SDN [2]是一种数据与控制分离、可编程的新型网络架构。从传统架构到新型架构的转变，不可避免要经历一个过渡时期，即传统网络与新型网络共存的时期。然而传统网络中的协议在 OpenFlow 网络中是不支持的。所以传统网络中实现的服务在 OpenFlow 网络中要重新思考。本文的目标是在传统网络与 OpenFlow 网络互连的情况下，在 OpenFlow 网络中实现组播技术。

在传统网络中的路由器与主机之间存在 OpenFlow 网络时，通过控制器实现 IGMPv3 报文的转发，同时控制器可以获取 OpenFlow 网络中的全局信息来制定组播在 OpenFlow 网络中的转发路径。本文最后通过实验来证明该方案的可行性。

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

1 技术背景

1.1 OpenFlow 概述

OpenFlow[3]不是技术，而是 SDN 架构里主流的南向接口协议之一。OpenFlow 引入了“流”的概念。网络通信中产生的具有共同特征（如 MAC 地址，IP 地址，IP 协议等）的数据分组可以抽象成一个“流”，这使得交换设备可以统一处理这些具有共同特征的数据分组。

运行 OpenFlow 协议的交换机称为 OpenFlow 交交换机（简称 OF 交换机）。OF 交换机有流表，流表中的各个表项是针对不同特征的“流”的处理规则。OF 交换机构成的网络称为 OpenFlow 网络。

1.2 组播概述

组播[4]（Multicast）技术能够有效解决单点发送、多点接收的问题，从而实现网络中点到多点的高效的数据传输，能够节约大量网络带宽、降低网络负载。

组播协议可以按其工作在网络层还是数据链路层分为“三层组播协议”和“二层组播协议”。三层组播协议又包括组播组管理协议和组播路由协议。组播组管理协议运行于主机与其直连的路由器之间。IPv4 网络中的组播组管理协议是 IGMP（Internet Group Management Protocol， 互联网组管理协议）。

1.3 Ryu

SDN 控制器[5]在数控分离的 SDN 架构中具有举足轻重的地位。它是连接底层交换设备与上层应用的桥梁。控制器通过南向接口对底层交换设备进行集中管理、状态监测和转发决策；通过北向接口向上层应用开放 API，提供可编程能力。

开源的控制器有很多，目前被 SDN 业界的广泛采用的有 NOX/POX[6]、Ryu、Floodlight 和 OpenDaylight。目前支持 OpenFlow 1.3 的只有 Ryu 和 Opendaylight。本论文采用的控制器是 Ryu。

Ryu[7]是有日本电报电话公司（NTT）主导开发的一个开源 SDN 项目，其字面意思就是日语中“流”。它有大量的组件和库函数供 SDN 应用开发。这些库函数可以被组件直接调用，并且组件之间是相互独立的，用户可以灵活开发自己的组件。

2 系统设计

组播控制器在功能上可以分成三大部分，分别是：（1）底层信息获取部分：拓扑发现模块和链路状态获取模块。这两个模块获取网络拓扑和链路信息，这些信息提供给组播转发路径计算模块。（2）组播处理部分：组播报文解析模块对 Packet-In 的组播报文进行解析，如果是 REPORT 报文，则交给组播组成员管理模块处理。组播组管理成员模块发现是新组播成员或者是组播成员离开，则交给组播转发路径计算模块处理。（3）流表下发模块：组播转发路径计算模块把路径计算好后交给流表下发模块。流表下发模块通过调用系统的 API 下发流表到相应的 OF 交换机上。系统框架见图 1。

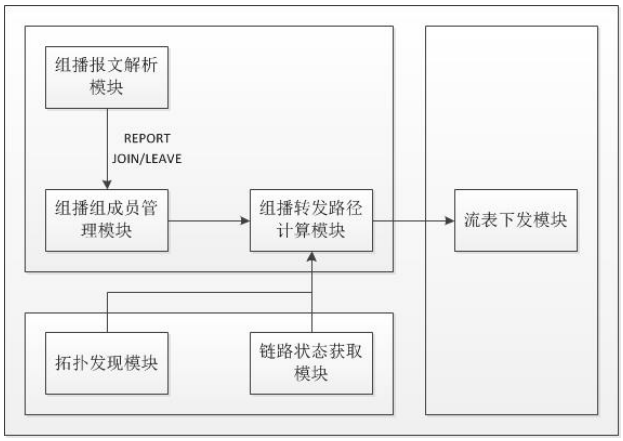

图 1 系统模块设计

2.1 拓扑发现模块

控制器使用 LLDP[8]（Link Layer Discovery Protocol， 链路层发现协议）作为链路发现协议。Ryu 的 rest_topology 组件已经实现了该功能，可以直接调用 rest_topology 组件的 API。另外 ofctl_rest[9]组件提供了获取 OF 交换机信息以及配置 OF 交换机的 API。这两个组件提供的是 REST API。本系统将它们的 API 封装为 python 可以直接调用的 rest_api。

2.2 链路状态获取模块

目前没有方法能够直接得到链路的利用率。所以采用定时获取端口的统计信息来间接计算。控制器创建一个线程来定时读取 OF 交换机各端口的统

计数据，其中 tx_bytes（Number of transmitted bytes）可以用来计算链路带宽利用率。

2.3 组播报文解析模块

IGMPv3[10]协议有两种组播报文，分别是查询报文（0x11）和报告报文（0x22）。通过解析 IGMPv3 报文的 Type 字段来确定是哪种报文。如果是查询报文，则不作处理，直接 Packet-Out 给与主机相连的 OF 交换机上。如果是报告报文，则继续解析。IGMPv3 报告报文里有 N 个 Group Record，逐一解析每个 Group Record。Group Record 的 Record Type 表明是 Join（4）还是 Leave（3）。最后把 Join 还是 Leave 的消息通知组播组成员管理模块。

2.4 组播组成员管理模块

组播组成员管理模块有组成员表。如果是 Join，则查组成员表，如果在组成员表里，则不做处理，如果不在，则添加到组成员表中，并通知组播转发路径计算模块有新组播成员；如果是 Leave，则从组成员表中删除，并通知组播转发路径计算模块有组成员离开。

2.5 组播转发路径计算模块

有新组成员加入时，组播转发路径计算模块去取链路状态获取模块计算得到各链路的利用率。该模块运用 Dijkstra 算法来计算生成最优转发树。边的权值是链路利用率。最优转发树 path 的格式：

path = {dpid1: port1, dpid2: port2, …}

其中 dpid 是 OF 交换机在 Ryu 控制器里的编号，port 是端口号，dpid1: port1 的意思是 dpid1 将数据报文发往 port1 给 dpid2。

最后该模块将 path 保存下来并提交给流表下发模块。

有组成员退出时，该模块通知流表下发模块删除 path 对应的流表，然后删除 path。

2.6 流表下发模块

流表下发模块接收到组播转发路径计算模块发过来的 path 后。如果是新组成员加入，则调用 rest_api 里的添加流表接口下发流表到相应的 OF 交换机，如果是组成员退出，则调用 rest_api 里的删除流表接口删除相应的 OF 交换机上相应的流表。

3 系统实现及测试

3.1 实验环境简介

OpenFlow 网络由三台 OF 交换机构成。因硬件设备约束，实验环境里的 OF 交换机是用 PC 安装 Open vSwitch[11]后的 OpenFlow 软件交换机。路由器（H3C MSR50-40）是传统网络中的路由器。组播源和控制器分别是两台 PC。控制器与 OF 交换机通过的普通交换机（H3C S5500EI）连接。实验拓扑见图 2。

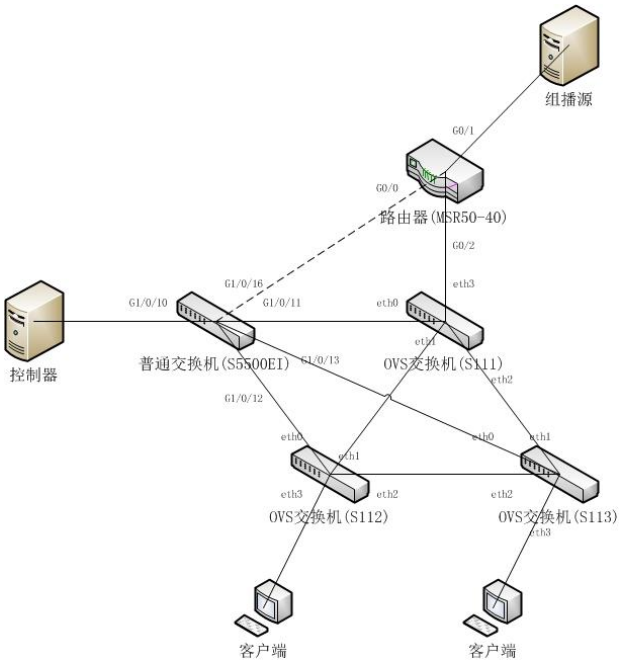

图 2 实验网络拓扑

3.2 组播报文的转发

控制器启动后做出以下动作：首先清空所有 OF 交换机中的流表；然后初始化 OF 交换机流表。初始化过程下发的是针对组播报文如何转发的流表项。如图 2 中，S111 转发报告报文给 eth3 口（与路由器直连的端口）；S112 转发查询报文给 eth3 口（与客户端直连的端口）。

客户端发出加入组播（239.1.1.4）的报告报文后，经过控制器处理后到达路由器。在路由器上查看组播成员，见图 3，可以看出路由器现已知道端口 G0/2 上有组播组成员（192.168.0.101）。

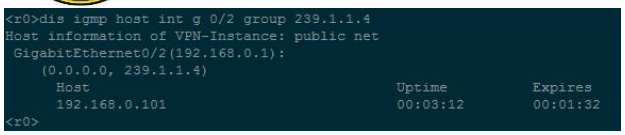

图 3 查看组播组成员

3.3 组播流的转发

控制器收到 Packet-In 的报告报文后，会计算生成最优转发树，并下发流表到相应的 OF 交换机。以 S112 为例，查看其流表，见图 4，可以看出对“IP，UDP，目的地址是 239.1.1.4”的流的处理动作是转发到端口 4（eth3）。

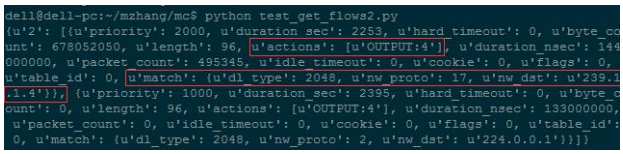

图 4 查看 S112 的流表

最后成功在客户端（192.168.0.101）上看到组播视频。

4、结论

本论文的主要工作是在不改变现有组播协议的情况下，在 OpenFlow 网络中实现组播转发。从实验室结果来看，通过组播控制器来转发 IGMP 报文，生成组播转发树并下发流表到相应的 OF 交换机上，最终实现组播转发是可行的。并且与传统组播路由协议相比，组播控制器具有可以获取全网信息的优势，能够从全局出发生成组播转发树。

参考文献

[1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[2] NADEAU T D, GRAY K. 软件定义网络: SDN 与 OpenFlow 解析[M]. 毕军等译. 北京: 人民邮电出版社, 2014.

[3] Specification, OpenFlow Switch. Version 1.0.0 (Wire Protocol 0x01) [S]. 2009.

[4] 谢希仁. 计算机网络（第 5 版）. 北京: 电子工业出版社. 2010.

[5] MONACO M, MICHEL O, KELLER E. Applying operating system principles to SDN controller design[A]. Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks[C]. ACM, 2013.

[6] GUDE N, KOPONEN T, PETTIT J, et al. NOX: towards an operating system for networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.

[7] Ryu[EB/OL]. http://osrg.github.io/ryu/, 2015.

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

[9] ryu development team. Ryu Documentation Release 3.14[EB/OL]. https://media.readthedocs.org/pdf/ryu/latest/ryu.pdf, 2014.

[10] Cain B, Deering S, Kouvelas I, et al. Internet Group Management Protocol, Version 3[S]. RFC3376. 2002.

[11] Open vSwitch[EB/OL]. http://openvswitch.org/, 2015.