

一种基于 RIPv2 的轻量级 IP/SDN 混合互联架构

舒兆港, 林甲祥, 杨长才, 张振昌, 景 林

(福建农林大学计算机与信息学院 福建 福州 350002)

摘要: 分析了现有 IP/SDN 混合互联架构方案的优缺点, 提出一个基于 RIPv2 路由协议的轻量级 IP/SDN 互联架构. 其主要原理是将 SDN 网络抽象成 RIPv2 协议中的单个虚拟路由节点, 在控制器内部实现 RIPv2 路由表和 Openflow 流表信息的转换. 从 RIPv2 的协议角度来看, 无论 SDN 网络内部包含多少个交换机和子网, 网络报文穿过该 SDN, 其跳数只经过 1 跳. 在开源控制器 Floodlight 上实现了该架构并进行了试验验证和性能评估. 与现有架构相比, 该架构不需要增加额外的协议转换设备, 能够实现与传统 IP 网络的多出口节点互联, 且具有较低的性能开销, 适用于中小型园区网的 IP/SDN 互联场景.

关键词: 软件定义网络; 混合互联架构; 路由协议

中图分类号: TP391 **文献标识码:** A **文章编号:** 1671-5470(2016) 06-0719-07

DOI: 10.13323/j.cnki.j.fafu(nat.sci.).2016.06.018

A lightweight hybrid interconnection architecture of IP/SDN based on RIPv2

SHU Zhaogang, LIN Jiaxiang, YANG Changcai, ZHANG Zhenchang, JING Lin

(College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou, Fujian 350002, China)

Abstract: Basing on advantages and disadvantages of existing hybrid interconnection architectures of IP/SDN, a lightweight hybrid interconnection architecture of IP/SDN was proposed based on RIPv2. Its main principle is to abstract single virtual router from the whole SDN network, in which RIPv2 routing table can be transformed from/into flow table by the SDN controller. Theoretically, hop count will only increase by one when network packet gets through SDN domain, despite of the number of switches and subnets SDN domain contains according to RIPv2 protocol. The architecture was successfully implemented based on an open source controller Floodlight, then its functionality and performance were evaluated. Compared with existing architecture, the new architecture can be connected with multiple traditional IP network in the same topology without further conversion equipment, as well as being in low performance overhead, which makes it suitable for small or medium-sized IP/SDN interconnection scenarios.

Key words: software-defined network; hybrid interconnection architecture; routing protocol

软件定义网络(software defined network, SDN)作为新一代网络技术架构,其核心思想是分离网络系统的转发面和控制面,使得网络用户能够在不需要升级网络设备或改变网络拓扑结构的前提下,通过软件编程的方式自定义网络功能,从而加快网络的创新应用^[1].如果要采用全新的 SDN 架构,则原有的传统网络设备将被废弃,这在短期内不具有可行性,在相当长的时期内将维持两者共存互通的局面^[2].

有关 SDN 与传统网络的互联互通技术方案,目前有 3 种类型,即混合拓扑互联^[3-6]、混合流量转发^[7-9]及传统设备的 SDN 功能模拟^[10-11].混合拓扑互联方案^[3-6]的主要特点是借助 BGP/OSPF 协议的原有特性,屏蔽 SDN 域内的设备连接结构, BGP/OSPF 的协议规则不需要大量修改.但由于运行 BGP/OSPF 协议对于控制器而言是一个较大的负担,且需要额外的边界转换设备,提高了网络部署成本.混合流量转发方案^[7-9]的优点是可以利用 SDN 灵活编程的方式主动监控用户感兴趣的流量,实现细粒度的流量调控和提高服务质量的需求,而这些功能在传统 IP 中可能很难实现;同时也可以减少全 SDN 网络中控制器的负载,当控制器失效时 IP 控制方式也可以起到有效的备份作用.但是,这种路由器节点在软件架构实现上与传统路由设备差别较大,需要对原有设备进行大规模软件升级,在实际网络中推广有较大的难度.传统设备的 SDN 功能模拟方案^[10-11]虽然很好地解决了传统设备短期内无法更换的问题,但这种模拟的功

收稿日期:2016-04-24 修回日期:2016-06-10

基金项目:国家自然科学基金资助项目(41401458);福建省科技厅资助项目(2016J05148、2014J05045);福建省教育厅资助项目(JA14119、JA14125).

作者简介:舒兆港(1981-),男,高级工程师,博士.研究方向:软件定义网络、云计算等.Email: zhaogang.shu@gmail.com. 通讯作者:景林(1957-),男,教授.研究方向:计算机网络技术等.Email: jinglin@fafu.edu.cn.

能只能局限于 SDN 的基础功能,对于一些高级的 SDN 创新应用很难实现,而且会导致传统设备的各种 ACL 配置变得非常复杂。

鉴于当前研究存在的问题,本文提出一种轻量级 IP/SDN 混合互联架构,属于混合拓扑互联类型,通过在开源控制器 Floodlight 中集成 RIPv2 路由处理机制,在 SDN 网络边缘设备实现与 IP 网络的互通。与现有架构相比,本架构不需要增加额外的中间转换设备,且能够实现与传统 IP 网络的多出口节点互联,同时具有较低的性能开销,适用于中小型园区网的 IP/SDN 互联场景。

1 轻量级 IP/SDN 混合互联架构的设计

为了达到轻量级的效果,本文选取 RIPv2 路由协议作为 IP/SDN 互连的路由协议。RIPv2 采用距离矢量算法,路由器之间通过发送组播地址为 224.0.0.9 的组播报文,获取到达目的网络的跳数以及下一跳地址,从而生成路由表。由于每个路由器并不需要维护全网的路由拓扑信息,只需要根据最小跳数确定最优路由,因此该算法具有简单轻便的特点,适合中小型网络应用场景。SDN 的报文转发是通过控制器维护整网拓扑,实时下发 Openflow 流表到交换机进行转发,无论是控制器还是交换机,并不存在 RIPv2 路由协议。为了将基于 RIPv2 的 IP 网与 SDN 进行混合互联,SDN 在不改变原有运行规则的前提下,必须兼有 RIPv2 的路由特征。

1.1 设计原理

本设计的主要原理是将 SDN 网络抽象成 RIPv2 协议中的单个虚拟路由节点,而虚拟路由节点的接口,可能分布在不同的 Openflow 交换机上。不管 SDN 网络内部包含多少个交换机和子网,从 RIPv2 的协议角度来看,网络报文穿过该网络,其跳数只经过 1 跳,其原理如图 1 所示。

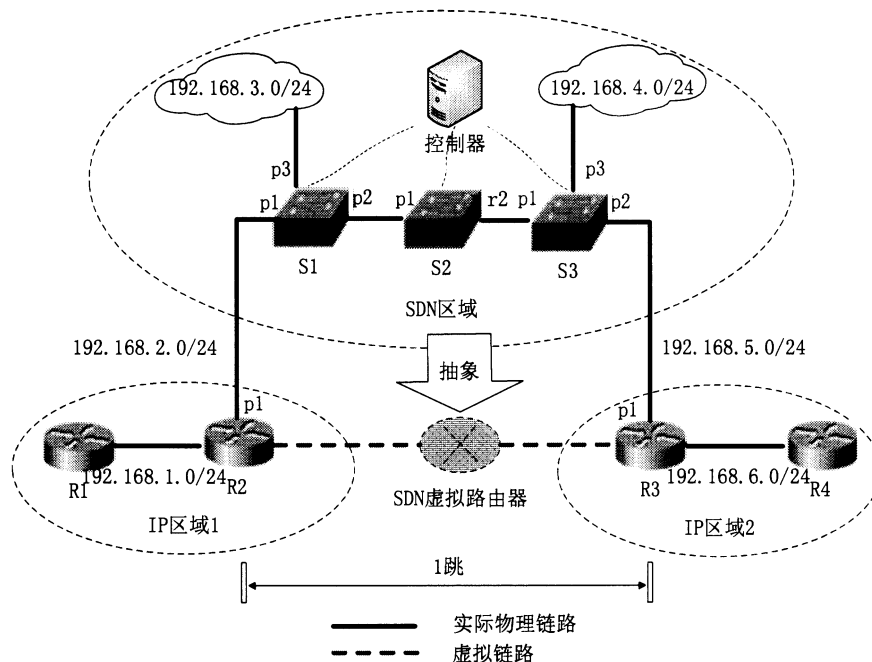


图 1 基于 RIPv2 的 IP/SDN 混合互联原理

Fig.1 Principle of hybrid interconnection IP/SDN based on RIPv2

图 1 中,SDN 区域有 S1/S2/S3 个交换机和 1 个控制器,其中 S1 的 p2 端口与 S2 的 p1 端口连接,S2 的 p2 端口与 S3 的 p1 端口连接。而 S1 的 p1 端口与 IP 区域 1 中路由器 R2 的 p1 端口连接,S3 的 p2 端口与 IP 区域 2 中路由器 R3 的 p1 端口连接,构成一个 IP/SDN 混合拓扑。该拓扑中包含 6 个子网网段(分别为 192.168.1.0/24、192.168.2.0/24、192.168.3.0/24、192.168.4.0/24、192.168.5.0/24 和 192.168.6.0/24),其中 SDN 内部包含 2 个网段(192.168.3.0/24 和 192.168.4.0/24)。当 R2 发送报文到 R3 时,其报文转发实际物理路径为 R2→S1→S2→S3→R3,但从 RIPv2 协议逻辑上看,其路由路径为 R2→Rv→R3,其中 Rv 表示 SDN 网络区域抽象出来的一个虚拟路由器,从 R2 到 R3 只经过一跳,而不需要考虑 SDN 网络内部的具体

结构和网段数量,而且,IP区域路由器把SDN内部网段识别为与上述虚拟路由器直连相连的网段,因此,IP区域的路由器工作方式基本不变,而唯一需要改变的是SDN区域。

1.2 SDN区域内的RIPv2路由架构

在SDN区域内部,拓扑发现、转发决策和流表下发全由控制器来完成,控制器内部保存了当前所有SDN交换机的端口连接信息和网段分布信息。为了和原有的转发决策和流表操作机制相配合,RIPv2路由决策也应该在SDN控制器内进行处理,本文提出SDN区域内的路由架构如图2所示。

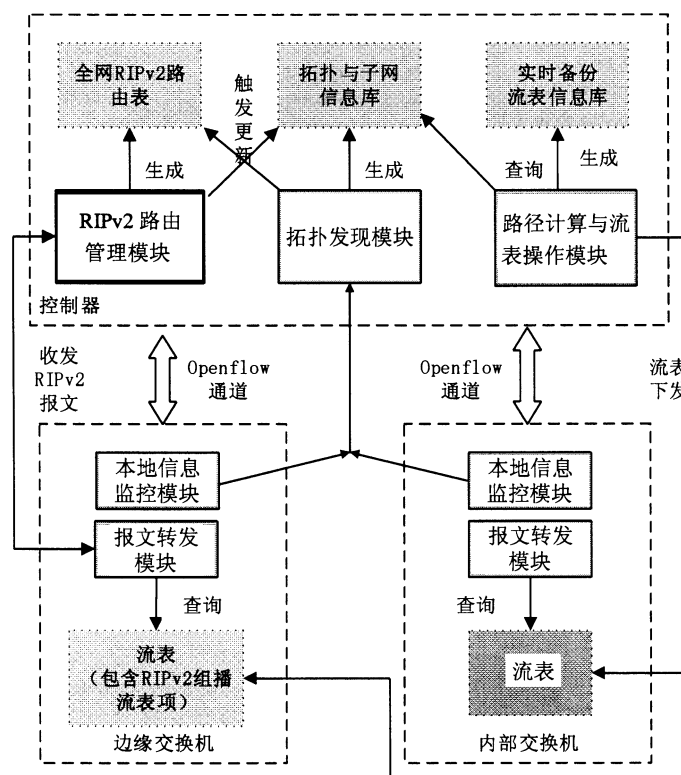


图2 SDN区域内RIPv2路由架构

Fig.2 Routing architecture of RIPv2 in the scope of SDN

如图2所示,控制器中主要包含3个功能模块,即RIPv2路由管理模块、拓扑发现模块、路径计算与流表操作模块,其中只有RIPv2路由管理模块是本文新增的模块,其他2个模块是控制器已有的功能模块,只需要进行少量修订。与以上3个模块密切相关的信息表分别是全网RIPv2路由表、拓扑与子网信息库和实时备份流表信息库。而所有交换机分为边缘交换机和内部交换机。边缘交换机是直接和外部IP网络具有直接连接接口的交换机,如图1中的R1和R3。下面分别描述控制器和交换机中各个功能模块与信息表的作用及其在RIPv2路由处理过程中的交互机制。

1.2.1 控制器的初始化与拓扑发现 控制器初始化时,上述3个信息表都为空。在和邻居IP区域的路由器进行RIPv2报文交互之前,控制器首先必须探测SDN区域内部的设备连接拓扑与子网信息,该功能由拓扑发现模块完成。根据Openflow的规范,SDN交换机相互发送LLDP报文,以探测与本设备连接的相关信息(如接口状态、对端链路信息以及连接子网信息等)。如果这些信息发生变化,则向控制器通告,这些功能由交换机的本地信息监控模块完成。控制器汇总所有交换机的通告信息,动态生成拓扑和子网信息表。以图1的网络拓扑为例,拓扑表包含的信息格式为:{(S1, p2, S2, p1), (S2, p2, S3, p1)},表示S1的p2端口与S2的p1端口连接,S2的p2端口与S3的p1端口连接。子网信息表包含的信息格式为:{(S1, p3, 192.168.3.0/24), (S3, p3, 192.168.4.0/24)},表示与交换机S1的p3端口连接的网段有192.168.3.0/24,与交换机S3的p3端口连接的网段有192.168.4.0/24。当拓扑与子网信息库产生之后,控制器才能响应交换机的转发请求,根据拓扑和子网信息进行路径计算,将相关流表下发到交换机,同时在控制器保存这些流表信息备份。该功能由路径计算与流表操作模块完成。

1.2.2 RIPv2 协议报文收发策略 当系统初始化与拓扑发现完成后,SDN 区域的交换机可以根据流表正常进行报文转发.控制器开始进行 RIPv2 协议报文的收发,该功能主要由控制器中的 RIPv2 路由管理模块完成.

(1) RIPv2 报文接收策略.边缘交换机收到 IP 区域路由器发送的 RIPv2 协议报文后,统一将其送到控制器进行处理.因此,在边缘交换机上应该添加 1 条流表项完成该功能.匹配域为目的 IP,即 RIPv2 的组播地址(如 224.0.0.9),Action 动作是将报文传给控制器.该流表项可以由网络管理员手动添加,也可以由流表下发模块自动下发.自动下发的过程和交换机处理未知名报文的方式类同,当边缘交换机第一次收到 RIPv2 报文时,找不到匹配的流表项进行转发,则其默认处理方式是将报文通过 Openflow 通道送到控制器.控制器解析报文发现是 RIPv2 协议报文,则为该交换机安装一条对应的表项.同时,控制器也得知该报文从哪个交换机的哪个端口进入,由此可判定该交换机端口必定与外部 IP 区域的路由器直接相连接,并由此判定该交换机为边缘交换机.

(2) RIPv2 报文发送策略.根据 RIPv2 协议原理,由于整个 SDN 区域被当成一个虚拟路由器,由 RIPv2 路由管理模块产生协议报文后,需要将该报文从所有的边缘端口(与 IP 区域路由器连接的端口)发送出去.因此,在发送报文之前,需要遍历当前所有边缘交换机,通过 Openflow 通道把协议报文发送到这些交换机.当交换机从控制器接收到这些报文后,则转发到本地所有边缘端口.交换机可以自己保存本地的边缘端口信息,也可以由控制器通告其边缘端口信息.

1.2.3 RIPv2 路由和流表更新机制 RIPv2 协议报文能够顺利收发,必须更新 RIPv2 路由,以及与流表机制进行协作互动.当收到 IP 区域路由器发送的 RIPv2 协议报文时,RIPv2 路由管理模块需要按顺序完成下列动作.

(1) 更新全网 RIPv2 路由表.提取报文携带的目的网络字段作为新增路由条目的目的网络字段;提取该报文的源 IP 地址,源 IP 作为新增路由条目的下一跳地址字段;提取对应目的网络的跳数,将跳数加 1,作为新路由表条目的跳数字段.如果跳数 ≥ 16 ,则放弃添加本路由条目.例如,在图 1 中,收到路由器 R2 发送的协议报文包含路由信息{(192.168.1.0/24,direct)},表示路由器 R2 与网段 192.168.1.0/24 直连,即距离为 0 跳.控制器添加 1 条路由信息{(192.168.1.0/24,192.168.2.1,1 跳)},表示 SDN 虚拟路由器到达网段 192.168.1.0/24 下一跳的地址为 192.168.2.1(该地址为路由器 R2 的 p1 接口地址),距离为 1 跳.

(2) 更新拓扑与子网信息库.当控制器更新 RIPv2 路由表的同时,也会触发更新拓扑与子网信息库.更新方法是:提取报文携带的目的网络字段作为新增子网字段,控制器判断该子网与边缘路由器直接相连.如图 1 中,增加的子网信息条目为{(S1_p1,192.168.1.0/24),(S1_p1,192.168.2.0/24)}.这样,当内部交换机请求去往目的网络为 192.168.1.0/24 网段时,控制器将根据拓扑与子网信息库启动内部路径计算,最后出口是 S1 交换机的 p1 端口.一旦报文出了 SDN 区域,按传统的 RIPv2 进行路由,实现报文路径从 SDN 区域到 IP 区域的跨越.

(3) 向 IP 路由器通告内部子网.为了在 IP 区域也能找到 SDN 内部的主机或网络,控制器内的 RIPv2 路由处理模块,也需要将 SDN 区域内的子网全部通告邻接的 IP 区域路由器.首先,拓扑与子网信息变动时,需要由拓扑发现子模块触发更新全网 RIPv2 路由表.图 1 拓扑中,当发现 192.168.3.0 网段与 S1 的 p3 端口连接,则需要在 RIPv2 路由表中添加 1 条信息{192.168.3.0/24,direct},表示 SDN 虚拟路由器与 192.168.3.0/24 直连.然后 RIPv2 路由处理模块需要将该路由信息发送给所有邻接的 IP 区域路由器,即 R2 和 R3.当 R2 或 R3 收到该信息后,在自己的路由表中添加 1 条路由信息{192.168.3.0/24,p1,1 跳},表示达到 192.168.3.0/24 网段,从本地的 p1 端口转发,距离为 1 跳.这样 IP 区域的主机能够顺利找到 SDN 区域内的 192.168.3.0 网段的主机,实现报文路径从 IP 区域到 SDN 区域的跨越.

2 试验验证

2.1 架构实现与试验拓扑

为了验证上述轻量级 IP/SDN 互联架构的可行性,本文在开源控制器 Floodlight 的基础上,开发了以上架构的功能.主要新增的功能为 RIPv2 路由模块,并通过调用 Restful 接口与其他模块进行联动,如拓扑

发现模块、流表下发模块等.试验环境主要通过 Virtualbox 虚拟平台实现,创建各种不同用途的虚拟机(所有虚拟机安装 Ubuntu 12.04 系统),用来仿真 IP 路由器(安装开源路由软件 Quagga,开启 RIPv2 协议)和主机,SDN 区域网络通过在 Ubuntu 12.04 系统上安装 Mininet 和 Floodlight 来仿真,仿真试验环境的网络拓扑如图3所示.该环境中,共有5个虚拟机,VM1为SDN网络仿真环境,内部有控制器、3个SDN交换机以及1个内部主机,VM2和VM3分别仿真IP区域路由器,VM4和VM5代表IP区域主机.整个混合互联网络包含5个子网网段,设备相关接口用 eth0/eth1/eth2 标识,并配置相应的 IP 地址.

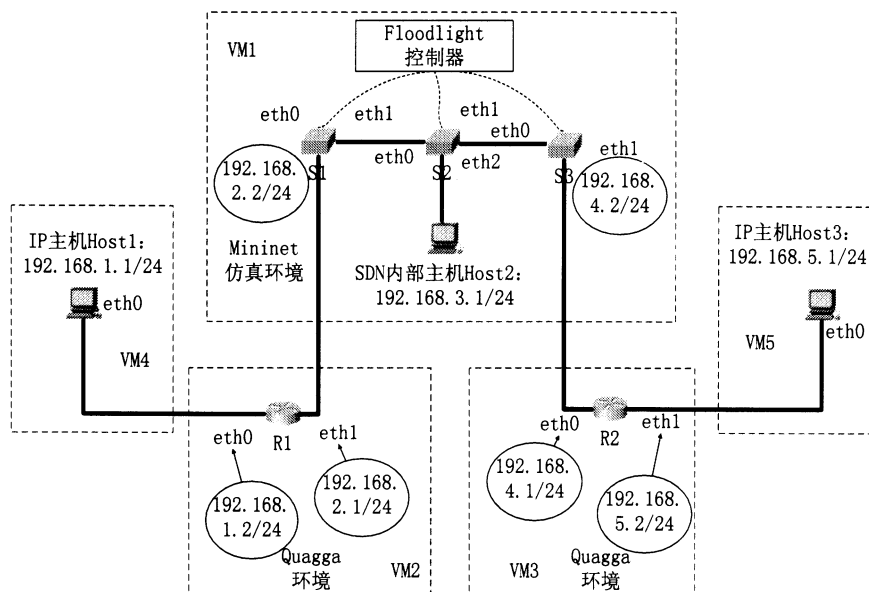


图3 仿真实验环境的网络拓扑

Fig.3 Network topology of simulation experiment environment

2.2 验证结果

为了验证 IP 区域和 SDN 网络区域的互通性,首先进行 ping 通测试,然后查看路由器的路由表和交换机的流表是否符合预期结果.验证步骤如下.

(1) Floodlight 完成拓扑检测后,查看 R1 和 R2 的路由表,发现其路由表中包含了 SDN 内部网段 192.168.3.0/24 的路由条目(图4).说明拓扑检测完成后触发更新了 RIPv2 的全网路由表.但此时在交换机 S1/S2/S3 上并未发现到达目的网络 192.168.1.0/24 和 192.168.5.0/2 的流表项.

<pre>localhost.R1> show ip route Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF, I - ISIS, B - BGP, > - selected route, * - FIB route K>* 0.0.0.0/0 via 192.168.2.2, eth1 C>* 127.0.0.0/8 is directly connected, lo R>* 192.168.3.0/24 [120/1] via 192.168.2.2, eth1, 00:20:16 R>* 192.168.4.0/24 [120/1] via 192.168.2.2, eth1, 00:23:12 R>* 192.168.5.0/24 [120/2] via 192.168.2.2, eth1, 00:28:11 C>* 192.168.1.0/24 is directly connected, eth0 C>* 192.168.2.0/24 is directly connected, eth1</pre>	<pre>localhost.R2> show ip route Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF, I - ISIS, B - BGP, > - selected route, * - FIB route K>* 0.0.0.0/0 via 192.168.4.2, eth0 C>* 127.0.0.0/8 is directly connected, lo R>* 192.168.1.0/24 [120/2] via 192.168.4.2, eth0, 00:29:12 R>* 192.168.2.0/24 [120/1] via 192.168.4.2, eth0, 00:31:19 R>* 192.168.3.0/24 [120/1] via 192.168.4.2, eth0, 00:23:15 C>* 192.168.4.0/24 is directly connected, eth0 C>* 192.168.5.0/24 is directly connected, eth1</pre>
--	--

图4 R1 和 R2 的路由表

Fig.4 Routing tables of R1 and R2

(2) 从 Host2 ping Host1 测试,可以 ping 通,再次查看 S1/S2/S3 的流表项,发现 S1 和 S2 增加了到达 192.168.1.0/24 网段的流表项,而 S3 没有增加(图5~7).控制器根据当前 RIPv2 获取的子网拓扑信息计算得到:从 Host2 到达 Host1 的内部通路为 Host2→S2→S1,而没有经过 S3,因此给 S1 和 S2 下发了相应的流表.

(3) 从 Host2 ping Host3 测试,可以 ping 通.同理,再次查看 S1/S2/S3 的流表项,发现 S2 和 S3 增加了到达 192.168.3.0/24 网段的流表项,而 S1 没有看到.控制器根据当前 RIPv2 获取的子网拓扑信息计算得

到: 从 Host2 到达 Host3 的内部通路为 Host2→S2→S3, 而没有经过 S1, 因此给 S2 和 S3 下发了相应的流表。

```
root@mininet-vm:~# dpctl dump-flows tcp:127.0.0.1:6634
stats_reply (xid=0xc1a9c432): flags=none type=1(flow)
cookie=0, duration_sec=8s, duration_nsec=525000000s, table_id=0, priority=65535, n_packets=8, n_bytes=784, idle_timeout=60, hard_timeout=0, icmp, in_port=eth0, dl_vlan=0xffff, dl_src=3a:0b:e2:87:70:a5, dl_dst=ea:70:ee:59:20:11, nw_src=192.168.1.1, nw_dst=192.168.3.1, nw_tos=0x00, icmp_type=0, icmp_code=0, actions=output:eth1
cookie=0, duration_sec=7s, duration_nsec=525000000s, table_id=0, priority=65535, n_packets=7, n_bytes=686, idle_timeout=60, hard_timeout=0, icmp, in_port=eth1, dl_vlan=0xffff, dl_src=ea:70:ee:59:20:11, dl_dst=3a:0b:e2:87:70:a5, nw_src=192.168.3.1, nw_dst=192.168.1.1, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:eth0
```

图 5 交换机 S1 的流表

Fig.5 Flow table of switch S1

```
root@mininet-vm:~# dpctl dump-flows tcp:127.0.0.1:6635
stats_reply (xid=0x4d8c2db9): flags=none type=1(flow)
cookie=0, duration_sec=10s, duration_nsec=525000000s, table_id=0, priority=65535, n_packets=8, n_bytes=784, idle_timeout=60, hard_timeout=0, icmp, in_port=eth0, dl_vlan=0xffff, dl_src=3a:0b:e2:87:70:a5, dl_dst=ea:70:ee:59:20:11, nw_src=192.168.1.1, nw_dst=192.168.3.1, nw_tos=0x00, icmp_type=0, icmp_code=0, actions=output:eth2
cookie=0, duration_sec=9s, duration_nsec=525000000s, table_id=0, priority=65535, n_packets=7, n_bytes=686, idle_timeout=60, hard_timeout=0, icmp, in_port=eth2, dl_vlan=0xffff, dl_src=ea:70:ee:59:20:11, dl_dst=3a:0b:e2:87:70:a5, nw_src=192.168.3.1, nw_dst=192.168.1.1, nw_tos=0x00, icmp_type=8, icmp_code=0, actions=output:eth0
```

图 6 交换机 S2 的流表

Fig.6 Flow table of switch S2

```
root@mininet-vm:~# dpctl dump-flows tcp:127.0.0.1:6636
stats_reply (xid=0xa535d954): flags=none type=1(flow)
```

图 7 交换机 S3 的流表

Fig.7 Flow table of switch S3

(4) 从 Host1 ping Host3 测试, 可以 ping 通。同理, 再次查看 S1/S2/S3 的流表项, 发现 S2 和 S3 增加了到达 192.168.3.0/24 网段的流表项, 而 S1 没有看到。控制器根据当前 RIPv2 获取的子网拓扑信息计算得到: 从 Host2 到达 Host3 的内部通路为 Host2→S2→S3, 而没有经过 S1, 因此给 S2 和 S3 下发了相应的流表。至此, 从 IP 区域 SDN 区域互联互通得到充分验证, 路由器路由表项和交换机流表添加的结果符合设计预期。

3 性能评估

为了进行性能评估, 拟将未修订的控制器(Floodlight)和加入 RIPv2 路由功能的控制器(Floodlight)在时间轴上的 CPU 进程利用率进行对比。启动完成后, 对其频繁进行拓扑变化操作(比如 S1 和 S2 链路切断与重连, 触发拓扑计算与路由更新), 观察 60 s 内控制器 CPU 进程利用率的变化, 对比结果如图 8 所示。从图 8 可知, 控制器的 CPU 进程占用率为 2.5%~5%, 加上 RIPv2 路由处理功能, 其变化率也没有超出这个范围。由此可见, 基于 RIPv2 的 IP/SDN 互联架构对原有控制器性能影响很小, 体现了其轻量级的特性。

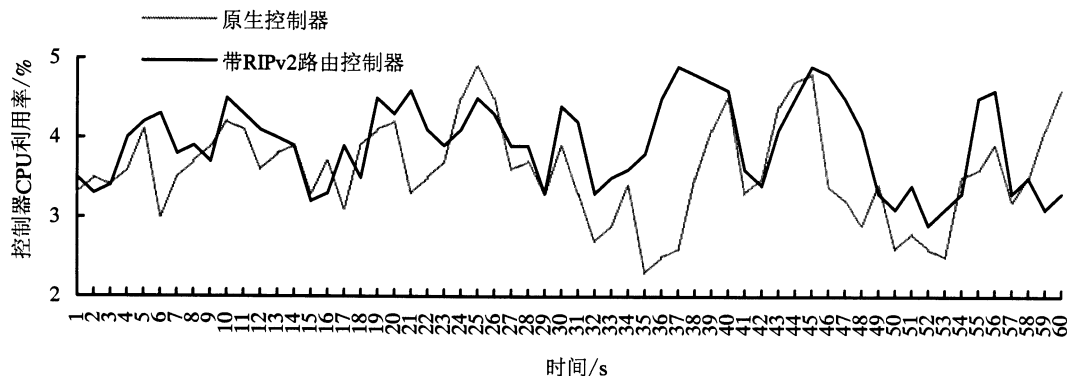


图 8 控制器(Floodlight)进程 CPU 利用率的变化

Fig.8 CPU utilization of controller (Floodlight)

4 小结

本文研究了基于 RIPv2 协议的 IP/SDN 互联机制, 通过在 Floodlight 控制器中增加 RIPv2 路由管理模块, 使得 IP 网络和 SDN 相互学习路由条目, 实现 IP 网络和 SDN 互联互通。结果表明, 该机制不需要增加

额外的协议转换设备,且具有较低的性能开销,解决了中小型园区网的IP/SDN混合互联问题.虽然文中只讨论了IP网络跨越SDN网络访问IP网络的情况,但对于多SDN网和多IP网络混合互联的场景,其原理和架构不需要任何改变都可以适用.由于本架构设计简洁,不需要额外的部署成本,且具有良好的性能,对解决当前IP和SDN网络混合互联问题具有重要意义.传统IP网络与SDN的巨大差异性,实现两者的互联互通不仅仅考虑其报文转发,还需要考虑通信安全、服务质量以及流量工程等方面,这将是本文下一步的研究工作.

参考文献

- [1] LARA A, KOLASANI A, RAMAMURTHY B. Network innovation using openflow: a survey [J]. Communications Surveys & Tutorials, IEEE, 2014, 16(1): 493-512.
- [2] VISSICCHIO S, VANBEVER L, BONAVENTURE O. Opportunities and research challenges of hybrid software defined networks [J]. ACM SIGCOMM Computer Communication Review, 2014, 44(2): 70-75.
- [3] LIN P, HART J, KRISHNASWAMY U, et al. Seamless interworking of SDN and IP [J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 475-476.
- [4] 徐宾伟. SDN与传统IP网络互联架构的设计与实现 [D]. 成都: 电子科技大学, 2015.
- [5] KUNIHIRO I. Quagga software routing suite documentation [EB/OL]. (2005-09-01) [2015-06-12] <http://www.nongnu.org/quagga/docs/quagga.html>.
- [6] 史衍伟, 曹争. 自治系统内IP子网和SDN子网的互联机制 [J]. 通信学报, 2014, 35(z2): 1-6.
- [7] SALSANO S, VENTRE P L, PRETE L, et al. OSHI-open source hybrid IP/SDN networking (and its emulation on mininet and on distributed SDN testbeds) [C]. Budapest: IEEE Computer Society, 2014: 13-18.
- [8] SALSANO S, VENTRE P L, LOMBARDI F, et al. Mantoo—a set of management tools for controlling SDN/NFV experiments [C]. Bilbao: IEEE, 2015: 1-2.
- [9] AARON C, ANUPAM C, DANIEL R, et al. Open vSwitch 2.5.0 documentation [EB/OL]. (2016-03-29) [2016-04-10] <http://openvswitch.org/support/dist-docs-2.5/>.
- [10] HAND R, ERIC K. ClosedFlow: OpenFlow-like control over proprietary devices [C]. Chicago: ACM SIGCOMM, 2014: 7-12.
- [11] FENG T, BI J. OpenrouteFlow: enable legacy router as a software-defined routing service for hybrid SDN [C]. Las Vegas: IEEE Communication Society, 2015: 1-8.

(责任编辑: 叶济蓉)