

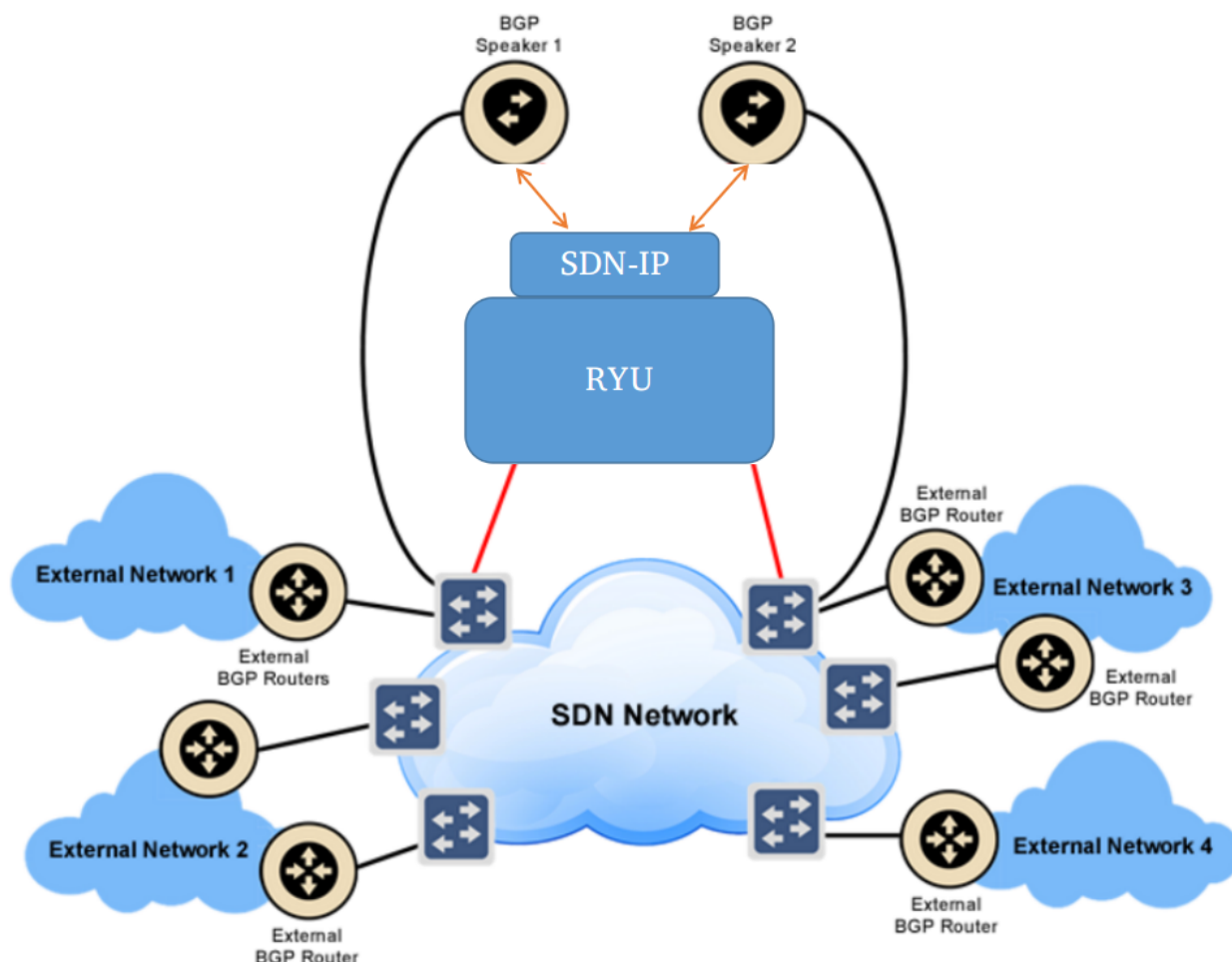
# SDN-IP Based On Ryu

## 一.需求

普通 IP 网与 SDN 网络互联,全部设备用电脑仿真,交换机用主机搭载 ovs 实现,BGP 边界路由器用主机通过运行 Quagga 软件实现,与搭载控制器的主机共同运行 IBGP 协议,控制器可以得知全局路由信息,两个 AS 之间运行 EBGp 协议,控制器使用 RYU, SDN 协议版本为 Openflow1.3,最终实现 sdn 网内外的主机可以互通互联。

## 二.总体设计

### 1.模型



Topology模型：

- 中间的网络为SDN网络，由Ryu控制器控制。
- SDN网络中运行一个或多个BGP Router作为BGP Speaker，BGP Router位于SDN网络内部（而不是边界）。
- 外部的AS域通过SDN网络边缘的OpenFlow交换机与SDN网络相连。

- 每个相邻域的边界BGP Router与SDN域中的一个或多个BGP Router组成EBGP对等体交换路由信息。
- Ryu控制器上运行SDN-IP应用，该应用启动一个软BGP Speaker与SDN网络中的所有BGP Router全相连组成IBGP对等体(在数据平面外相连)，通过BGP协议获取其它域路由信息。

流量模型：

SDN中的流量有下面几种：

- 穿越流量，流量从一个相邻域进入，通过SDN域到达另一个相邻域；
- 内部流量：SDN域内部主机间通信产生的流量；
- 内外通信流量：SDN域内外主机通信产生的流量。

## 2.模块设计

分为四大模块：

- 配置模块。
- SDN\_IP模块；
- FWD\_UTIL模块；
- ARP\_PROXY模块；

(1)配置模块

负责读取配置文件，生成BGP配置参数，供其他模块读取。主要包括：

- 本地网络地址；
- 本AS域的AS号；
- SDN-IP应用中的BGP Speaker监听BGP协议的端口号；
- 本地BGP Speakers的IP地址。

(2)SDN\_IP模块

负责建立EBGP对等体和IBGP对等体，监听路由信息变化，根据掌握的路由信息和全局拓扑信息为主机通信建立数据路径。

- 使用BGP配置参数，实例化IBGP Speaker，设置对等实体参数；
- 建立BGP Router间的数据路径；
- 监听路由信息变化，并存储路由信息；
- 监听路由信息变化，建立穿越流量通过SDN网络的数据路径。
- 建立SDN网络内外主机通信的数据路径。
- 建立SDN网络内部主机间访问数据路径。

(3)FWD\_UTIL模块

提供一些操作流表、建立数据路径的公共方法。

- 生成全局视图。
- 计算最短路径。
- 提供下发流表的方法。
- 提供在SDN内部两个接入点间按照最短路径建立数据路径的方法。
- 提供Packet-out方法。

## 3.实现要点

### 3.1IBGP对等体的建立。

运行在Ryu上的SDN-IP应用中会运行一个BGP Speaker，它需要与SDN域内的其他BGP Speaker相连，组成IBGP对等体。

- 物理连接：  
将运行Ryu的主机与BGP Speaker直接相连。
- 代码：

SDN-IP应用启动一个BGPSpeaker的实例，它将模仿真实的BGPSpeaker的行为。在BGPSpeaker实例建立以后，再建立与SDN域内的BGP Speaker的连接。

```
1.         # 在Ryu上运行一个iBGPSpeaker
2.         self.bgp_speaker =\
3.             BGPSpeaker(self.cfg_mgr.as_number,          #制定本域AS号
4.                         str(self.cfg_mgr.router_id),      #iBGP的router id
5.                         bgp_server_port=self.cfg_mgr.listen_port,          #监听BGP协
6.                         best_path_change_handler=self.best_path_change_handler,
7.                         peer_down_handler=self.peer_down_handler,          #BGP对等体
8.                         peer_up_handler=self.peer_up_handler)          #BGP对等体上线
9.
10.        speaker_ids = self.cfg_mgr.get_all_speaker_id()
11.        # 建立iBGPSpeaker与SDN域内的BGPSpeaker的连接
12.        for speaker_id in speaker_ids:
13.            self.bgp_speaker.neighbor_add(speaker_id,
14.                                           self.cfg_mgr.as_number,
15.                                           is_next_hop_self=True)
```

### 3.2 EBGPG对等体间数据路径的建立。

与传统网络不同，SDN域的边界路由器在并没有真正的位于网络的边界，而是位于网络内部，通过SDN网络中的OpenFlow交换机间接与邻域的边界路由器相连。

因此，SDN-IP应用启动后的第一个任务是要建立SDN域中的BGP Speaker和邻域的BGP Speaker的数据通道，这样彼此间才能建立BGP通信。

SDN-IP建立这个数据路径的方式为被动方式，我们以AS1中的边界路由器r1向AS0(SDN)中的BGP路由器bgp发起BGP连接为例。

- 交换机s1的r1端口收到r1发往bgp的数据包，不知如何处理，将其上报给Ryu控制器；
- Ryu控制器上的SDN-IP应用收到这个数据包的处理请求，进行分析，结果如下：

eth_type	dst_mac	src_ip	dst_ip
IPV4	bgp的mac地址 (00:00:00:00:00:01)	r1的ip地址(10.0.1.0)	bgp的ip地址 (10.0.1.101)

- SDN-IP根据分析结果判断出这个数据包的目的地是bgp，并根据掌握的全局视图对数据包要到达目的主机的路径进行分析，得到数据包到达目的地要通过的路径：

- 根据得到的数据包路径，SDN-IP需要向路径沿途的交换机下发处理规则，以使得沿途的交换机知道怎样转发此类数据包。在这个例子中SDN-IP下发的处理规则是这样的：
  - 交换机s1上下发的规则：
    - 规则匹配：eth\_type=IPV4，eth\_dst=00:00:00:00:00:01，ipv4\_dst=10.0.1.101
    - 处理方式：将数据包从端口2转发出去。
  - 交换机s2上下发的规则：
    - 规则匹配：eth\_type=IPV4，eth\_dst=00:00:00:00:00:01，ipv4\_dst=10.0.1.101
    - 处理方式：将数据包从端口1转发出去。
- 经过上述步骤，后续从r1发往bgp的BGP协议数据包就可以顺利沿着这条路径到达bgp了。这样就建立了从r1---->bgp的通道，类似的，bgp发往r1的数据包也按照这样的方式会建立相应的数据路径。

### 3.3访问域外主机的单向数据路径的建立。

访问域外主机的数据包的来源有两种：

- 来自SDN域内的主机；
- 来自SDN域邻域。

但是他们的目的地是同类的，都是SDN域外的主机。因为相邻AS域与SDN域直接相连的都是它们的边界路由器。因此，对于目的地是域外主机的数据包，只要找到目的主机所在的域，然后将其交给其边界路由器即可，剩下的转发工作就是目的域内部的工作了。

这里引申出了SDN-IP应用的一个工作要点，记录到达目的网络的下一跳地址。例如，AS1中的网络为190.168.1.0/24，它与SDN域相连的边界路由器的IP地址是10.0.1.0。这也就是说，我们去往域外网络192.168.1.0/24的下一跳地址为10.0.1.0，只要将这些数据包交给路由器10.0.1.0，它们就可以顺利到达目的地。那么，我们可以从哪里得到到达一个相邻网络的下一跳地址信息的呢？

#### • BGP路由表

答案就是：SDN域中的BGP路由器。

仍然以域AS0和AS1为例。上一部分，我们顺利建立了AS1中边界路由器r1与AS0中的BGP路由器bgp的之间的数据路径。这之后，他们将通过这条数据路径建立BGP通信，具体的通信协议为BGP(边界网关协议)，彼此成为EBGP对等体。

再之后，r1将依据BGP协议的规定，向bgp宣告自己掌握的本域内的路由情况。具体情形是，r1知晓自己内部的网络为192.168.1.0/24后，向bgp宣告，你以后将发往这个网络的数据包交给我好了。bgp就会根据r1的宣告，更新自己的BGP路由表：

1.	Network	Next Hop
2.	*> 192.168.1.0/24	10.0.1.0

#### • 建立访问域外主机的单向数据路径

上一步，SDN域内的BGP路由器成功获取了到达邻域的路由信息。但在SDN网络中，BGP路由器并不负责数据包的转发，负责数据包转发的是我们的SDN-IP应用。因此，对于SDN-IP应用来

说，此时仍然不知道该怎么要访问的域外主机到底在哪里。那么，SDN-IP应用怎么得到bgp中的路由信息呢？

答案就是：SDN-IP应用中的BGP Speaker。

参考前面的设计方案，我们在SDN-IP应用中运行一个BGP Speaker，然后将它与SDN域中的BGP路由器建立IBGP对等体。这样，SDN域中的BGP路由器得到的路由信息后会紧接着将自己得到的信息宣告给IBGP对等体。继续上面的例子，bgp在得到网络192.168.1.0/24的下一跳地址为10.0.1.0后，紧接着就将这条路由信息宣告给SDN-IP中的IBGP Speaker，大致如下：

```
1.  r1(192.168.1.0/24, 下一跳在本地)--->bgp(192.168.1.0/24, 下一跳为10.0.1.0)--->SDN-IP(192.168.1.0/24, 下一跳为10.0.1.0)
```

在得到这条对外的路由信息后，SDN-IP马上采取行动，向所有下发转发规则：目的地为网络192.168.1.0/24的数据包全部发往10.0.1.0。这样就建立了访问目的网络192.168.1.0/24中所有主机的单向数据路径。至于域外主机发往SDN本网络的数据包，处理方法与普通域内转发相同，不再赘述。

### 3.4SDN-IP应用配置

SDN-IP应用的配置文件为config.json。

下面的配置为例说明配置文件的使用。

```
1.  {
2.    "local": {
3.      "as_number": 65000,      /*SDN域的AS号*/
4.      "router_id": "127.0.0.1", /*SDN-IP中IBGP Speaker的router id(IP Address)*/
5.      "listen_port": 2000,    /*SDN-IP中IBGP SpeakerBGP协议的监听端口*/
6.      "networks": [          /*本地的网络*/
7.        "192.168.0.0/24"
8.      ]
9.    },
10.   "speakers": [             /*SDN域内的BGP Speaker配置*/
11.     {
12.       "dpid": "00:00:00:00:00:00:a3", /*BGP Speaker连接的交换机ID*/
13.       "port": 1, /*BGP Speaker连接的交换机的端口号*/
14.       "speaker_ids": [
15.         "10.10.10.1" /*BGP Speaker相对IBGP Speaker的IP地址*/
16.       ],
17.       "mac": "00:00:00:00:00:01"/*BGP Speaker的mac地址*/
18.     }
19.   ]
20. }
```