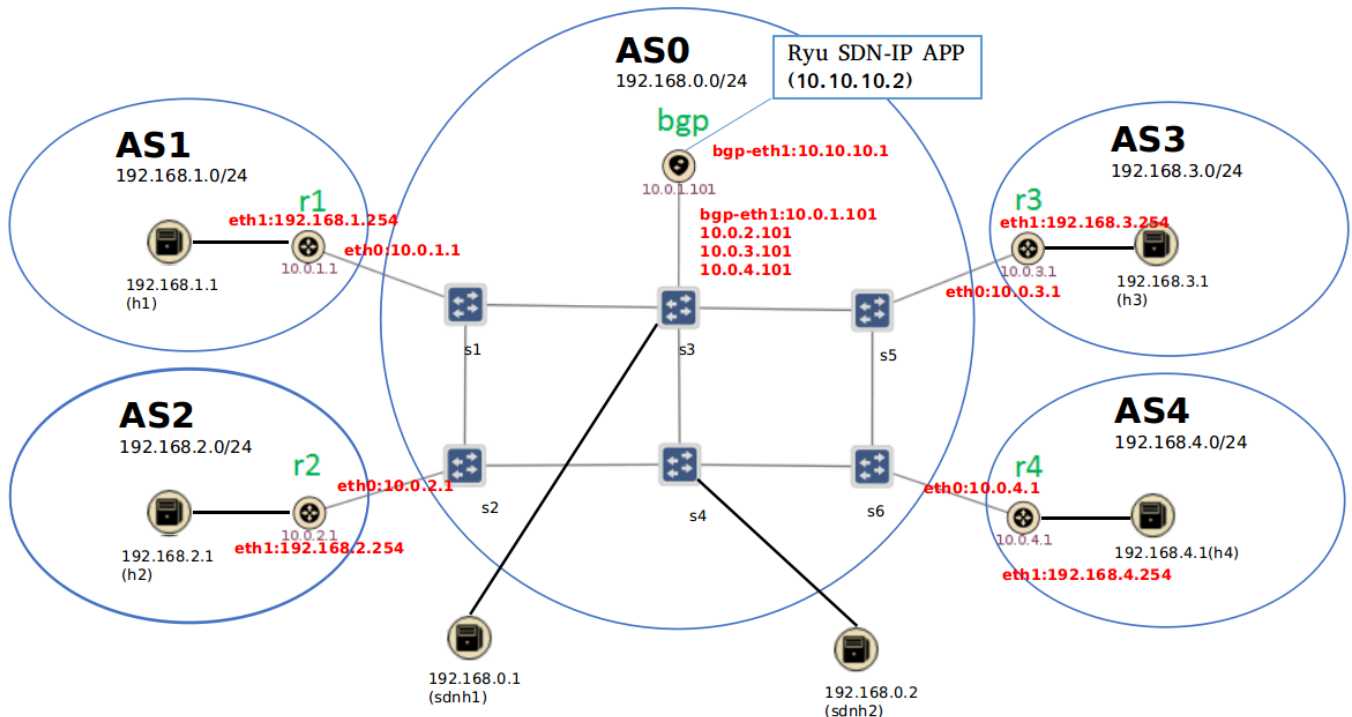


SDN-IP实验运行指南

1.网络结构图



上图为测试使用的网络拓扑，使用Mininet搭建。一共有 6 个OpenFlow交换机，5 个路由器，6个主机。

- "AS0"- "AS4"为 5 个AS域，其中AS0中的网络为SDN网络，其他的域为传统网络。
- "bgp","r1"- "r4"为 5 路由器，每个都是BGP Speaker。其中bgp与运行iBGP的的Ryu控制器直接连接。bgp与其他 4 个路由器相连，分别组成BGP对等体，学习它们的BGP路由信息，并把自己的BGP路由信息通报给它们。
- "r1"- "r4"每个路由器后面连接一个普通主机，每个主机将与其相连的Router设为默认网关。
- 我们在配置文件中分别设置r1,r2,r3宣告本地的静态路由信息，r4没有设置宣告路由信息，我们将在实验过程在手动设置r4宣告路由信息。

2.启动Ryu控制器和SDN IP应用

事先安装好Ryu控制器。另外，还需要安装networkx依赖。

```
1. sudo pip install networkx
```

在SDN IP项目根目录下运行下面命令，启动Ryu控制器和SDN IP应用。

```
1. ./run_sdnip.sh
```

启动信息：

```
1. loading app sdnip2.arp_proxy
2. Generating grammar tables from /usr/lib/python2.7/lib2to3/Grammar.txt
3. Generating grammar tables from /usr/lib/python2.7/lib2to3/PatternGrammar.txt
4. loading app sdnip2.sdn_ip
5. loading app ryu.topology.switches
6. loading app ryu.controller.ofp_handler
7. instantiating app None of FwdUtil
8. creating context fwd
9. instantiating app None of HopDB
10. creating context hop_db
11. instantiating app sdnip2.sdn_ip of SDNIP
12. API method core.start called with args: {'router_id': '127.0.0.1', 'label_range':
(100, 100000), 'waiter': <ryu.lib.hub.Event object at 0x7f327d70cd90>, 'local_as':
65000, 'bgp_server_port': 2000, 'refresh_max_eor_time': 0, 'refresh_stalepath_tim
e': 0}
13. API method neighbor.create called with args: {'connect_mode': 'both', 'remote_as':
65000, 'cap_mbgp_vpnv6': False, 'cap_mbgp_vpnv4': False, 'cap_mbgp_ipv6': False,
'is_next_hop_self': True, 'cap_mbgp_ipv4': True, 'is_route_server_client': False,
'cap_enhanced_refresh': False, 'peer_next_hop': None, 'password': None, 'ip_addres
s': u'10.10.10.1'}
14. instantiating app ryu.topology.switches of Switches
15. instantiating app sdnip2.arp_proxy of ArpProxy
16. instantiating app ryu.controller.ofp_handler of OFPHandler
```

3.启动网络拓扑

在拓扑项目跟目录下运行quagga-test1.py，启动虚拟网络。

```
1. sudo python quagga-test1.py
```

网络启动，等待片刻后，可以在Ryu的输出信息中看到路由信息更新的提示，说明BGP Speaker之间已经建立了对等体连接，并且交换了路由信息。

```
1. best path changed:
2. remote_as: 65000
3. route_dist: None
4. prefix: 192.168.1.0/24
5. nexthop: 10.0.1.1
6. label: None
7. is_withdraw: False
8.
9. best path changed:
10. remote_as: 65000
11. route_dist: None
12. prefix: 192.168.2.0/24
13. nexthop: 10.0.2.1
14. label: None
15. is_withdraw: False
16.
17. best path changed:
18. remote_as: 65000
19. route_dist: None
20. prefix: 192.168.3.0/24
21. nexthop: 10.0.3.1
22. label: None
23. is_withdraw: False
```

4.网络测试

4.1主机通信测试

在Mininet中分别测试Transit traffic、SDN域内间主机通信、SDN域内外主机通信。

```
1. mininet> h1 ping h2
2. PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
3. 64 bytes from 192.168.2.1: icmp_seq=1 ttl=62 time=0.419 ms
4. 64 bytes from 192.168.2.1: icmp_seq=2 ttl=62 time=0.063 ms
5. 64 bytes from 192.168.2.1: icmp_seq=3 ttl=62 time=0.065 ms
```

可以看到，AS1中的主机h1和AS2中的主机可以穿过SDN域AS0通信。

```
1. mininet> sdnh1 ping sdnh2
2. PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
3. 64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.349 ms
4. 64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.045 ms
5. 64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.047 ms
```

可以看到，SDN域AS0内的主机可以互相通信。

```

1. mininet> sdnh1 ping h1
2. PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
3. 64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=0.260 ms
4. 64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=0.059 ms
5. 64 bytes from 192.168.1.1: icmp_seq=4 ttl=63 time=0.057 ms

```

```

1. mininet> h2 ping sdnh2
2. PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
3. 64 bytes from 192.168.0.2: icmp_seq=2 ttl=63 time=4.09 ms
4. 64 bytes from 192.168.0.2: icmp_seq=3 ttl=63 time=0.071 ms
5. 64 bytes from 192.168.0.2: icmp_seq=4 ttl=63 time=0.063 ms

```

可以看到，SDN域内外的主机可以通信。

4.2查看BGP路由信息

从Minine中启动一个r1的xterm终端，并使用telnet连接Quagga的命令行工具查看r1的BGP路由信息，配置文件中设置的登录密码为sdnip。

```

1. mininet> r1 xterm &

```

```

1. root@cr:~# telnet localhost 2605
2. Trying 127.0.0.1...
3. Connected to localhost.
4. Escape character is '^]'.
5.
6. Hello, this is Quagga (version 1.0.20160315).
7. Copyright 1996-2005 Kunihiro Ishiguro, et al.
8.
9.
10. User Access Verification
11.
12. Password:
13. r1> show ip bgp
14. BGP table version is 0, local router ID is 10.0.1.1
15. Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
16.                  i internal, r RIB-failure, S Stale, R Removed
17. Origin codes: i - IGP, e - EGP, ? - incomplete
18.
19.      Network          Next Hop          Metric LocPrf Weight Path
20. *> 192.168.0.0        10.0.1.101         0           0 65000 i
21. *> 192.168.1.0        0.0.0.0            0          32768 i
22. *> 192.168.2.0        10.0.1.101         0 65000 65002 i
23. *> 192.168.3.0        10.0.1.101         0 65000 65003 i
24.
25. Displayed 4 out of 4 total prefixes

```

可以看到r1获得了除AS4之外其他AS域的路由信息，类似地可以查看其他BGP路由器的路由信息。

4.3宣告新的路由

在AS4域的BGP路由器r4的配置中，我们没有宣告AS4中的网络192.168.4.0/24，因此其他域的路由器不知道怎样到达网络192.168.4.0/24。现在我们通过配置r4手动宣告AS4中的网络。

首先我们通过Mininet的启动一个r4的xterm终端，并在终端中连接Quagga的CLI进行操作。

```
1. mininet> r4 xterm &
```

在终端中使用telnet连接到Quagga BGP CLI。

```
1. root@cr:~# r4 telnet localhost 2605
2. Trying 127.0.0.1...
3. Connected to localhost.
4. Escape character is '^]'.
5. Hello, this is Quagga (version 1.0.20160315).
6. Copyright 1996-2005 Kunihiro Ishiguro, et al.
7. User Access Verification
8. Password: sdnip
9. r4>
```

然后，我们对r4进行配置，宣告AS4中的网络192.168.4.0/24。

```
1. r4> enable
2. r4# configure terminal
3. r4(config)# router bgp 65004
4. r4(config-router)# network 192.168.4.0/24
5. r4(config-router)# exit
6. r4(config)# exit
7. r4# exit
8. Connection closed by foreign host.
```

为了验证这个路由信息是否被宣告到了其他路由器，我们连接r1的Quagga命令行进行查看。

```
1. mininet> r1 xterm &
```

```

1. root@cr:~# telnet localhost 2605
2. Trying 127.0.0.1...
3. Connected to localhost.
4. Escape character is '^]'.
5. Hello, this is Quagga (version 1.0.20160315).
6. Copyright 1996-2005 Kunihiro Ishiguro, et al.
7. User Access Verification
8.
9. Password:
10. r1> show ip bgp
11. BGP table version is 0, local router ID is 10.0.1.1
12. Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
13.                  i internal, r RIB-failure, S Stale, R Removed
14. Origin codes: i - IGP, e - EGP, ? - incomplete
15.
16.      Network          Next Hop          Metric LocPrf Weight Path
17. *> 192.168.0.0        10.0.1.101          0             0 65000 i
18. *> 192.168.1.0        0.0.0.0             0            32768 i
19. *> 192.168.2.0        10.0.1.101          0             0 65000 65002 i
20. *> 192.168.3.0        10.0.1.101          0             0 65000 65003 i
21. *> 192.168.4.0        10.0.1.101          0             0 65000 65004 i
22.
23. Displayed 5 out of 5 total prefixes
24. r1>

```

可以看到，在BGP路由表的最后一条，r1已经知道了去往192.168.4.0/24网络的下一条地址是10.0.1.101，r4的路由成功宣告了出去。然后我们再尝试AS1域AS4间主机的通信，使用主机h1 ping 主机h4。

```

1. mininet> h1 ping h4
2. PING 192.168.4.1 (192.168.4.1) 56(84) bytes of data.
3. 64 bytes from 192.168.4.1: icmp_seq=1 ttl=62 time=0.589 ms
4. 64 bytes from 192.168.4.1: icmp_seq=2 ttl=62 time=0.064 ms
5. 64 bytes from 192.168.4.1: icmp_seq=3 ttl=62 time=0.071 ms

```

h1与h4可以ping通。