Conexão entre redes distintas em ambiente SDN-IP e BGP

Aluno: Fernando Henrique Santorsula
Tópicos Em Engenharia de Computação VII
Software-Defined Networking e Network Functions Virtualization

Orientador: Prof. Dr. Christian Esteve Rothenberg
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas, Campinas, Brasil

Abstract - SDN-IP is an ONOS application that allows a software-defined network to connect to external networks through a gateway protocol (BGP). From the BGP point of view, the SDN Network is displayed as an Autonomous System (AS) that behaves like any traditional AS. Within the AS, an SDN-IP application provides the integration mechanism between BGP and ONOS. At the protocol level, the SDN-IP behaves like a normal BGP speaker. From the ONOS perspective, it is just an application that uses its services to install and update the SDN data plane routing state.

I. INTRODUÇÃO

Este artigo técnico será abordado o funcionamento do laboratório SDN-IP, utilizando o controlador SDN ONOS, que será executado em ambiente virtual, efetuando o processo de emulação de uma rede simples com o SDN-IP, com auxílio do emulador Mininet e da suíte de roteamento código aberto, denominado de Quagga, que irá utilizar o protocolo BGP para efetuar comunicação entre redes distintas.

II. OBJETIVO

Um determinado nó, denominado de "bgp" é o comunicador interno, onde a rede SDN faz o trabalho de investigar de forma rigorosa todos os roteadores BGP externos para aprendizado de suas devidas rotas, com o intuito de efetuar a transmissão de dados para a aplicação SDN-IP, sendo executada pelo controlador SDN ONOS, para que os roteadores que residem em outras redes possam receber seu devido tráfego.

Os outros quatro nós, denominados de r1 a r4, são os roteadores BGP externos, os mesmos são uma fronteira entre roteadores que residem em outras redes que desejam trocar tráfego, sendo que sempre atrás de cada roteador irá possuir um host, estes hosts são denominados de h1 a h4 que trocaram tráfego entre si, através do emulador Mininet.

O controlador SDN ONOS não consegue acessar diretamente estes hosts, devido os mesmos estarem em redes distintas que não são controladas pelo ONOS.

III. TOPOLOGIA

A topologia do ambiente SDN-IP deste artigo técnico, está ilustrada na FIGURA 1, contendo todos os componentes do laboratório que será desenvolvido e testado utilizando a aplicação SDN-IP e suas devidas aplicações.

IV. FUNCIONAMENTO

Neste artigo técnico foi essencial o uso do emulador Mininet, que tornou o ambiente simples e funcional, no ambiente SDN-IP, que contém switches OpenFlow para compor a rede SDN, que são conectados nas bordas da rede SDN em roteadores emulados. Os roteadores poderão utilizar um software denominado de Quagga, que é uma suíte de roteamento de código aberto, porém não é obrigatório usar o Quagga neste ambiente, mas em projetos mais complexos, o uso desta suíte de software de roteamento é extremamente eficiente. Este ambiente poderá ser utilizado outro software ou hardware capaz de se comunicar com um roteador BGP, sendo possível sua devida implantação neste ambiente de rede, sendo assim, neste artigo técnico, executamos o BGP do Quagga, para simular roteadores BGP externos pertencentes a outros domínios administrativos. O objetivo do SDN-IP é poder se comunicar com o BGP desses roteadores para trocar tráfego entre diferentes AS externos.

V. ARQUITETURA

Este artigo técnico faz a adoção de uma arquitetura que é totalmente capaz de integrar a redes SDN e computação em nuvem, permitindo que cada centro de dados em nuvem, baseado sobre uma arquitetura SDN possa conectar-se com redes externas através da Internet, usando o protocolo BGP (Border Gateway Protocol). Neste caso, externamente, a partir de uma perspectiva BGP, a rede SDN aparece como um único Sistema Autônomo que se comporta como qualquer AS tradicional dentro do AS, a aplicação SDN-IP fornece um mecanismo de integração entre BGP e o controlador ONOS, ao nível de protocolo, onde o ambiente SDN-IP se comporta como um BGP speaker, do ponto de vista do ONOS, o SDN-IP é apenas uma aplicação que usa seus serviços para instalar e atualizar o estado de encaminhamento no plano de dados SDN.

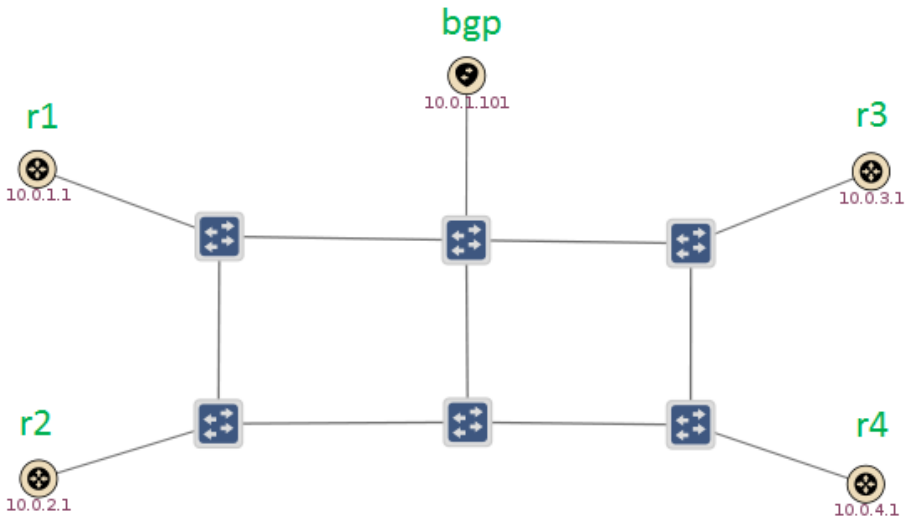
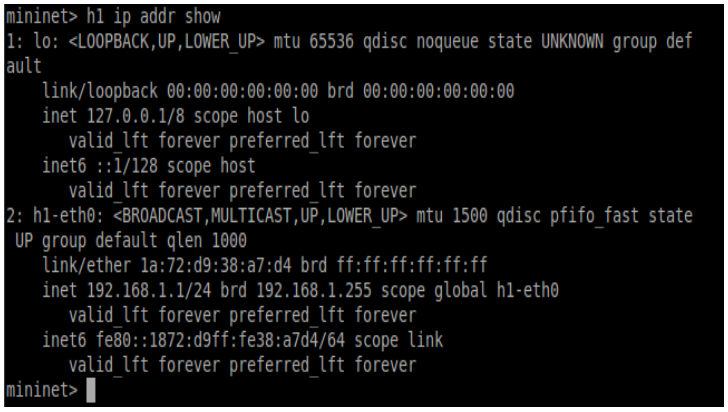


FIGURA 1

VI. RESETANDO CONFIGURAÇÕES

Para iniciar o ambiente é preciso aplicar um "reset" nas configurações para zerar as mesmas existentes, em seguida iremos verificar os endereços de IP pelo Mininet, para isso é preciso clicar duas vezes no ícone SDN-IP Mininet e digitar os comandos no emulador do Mininet, que é o software que irá auxiliar no ambiente SDN-IP, conforme a FIGURA 2:

FIGURA 2

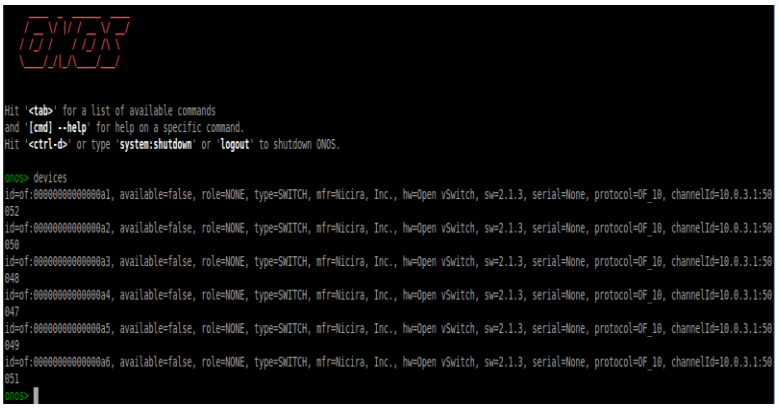


```
mininet> h1 ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: h1-eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
 UP group default qlen 1000
    link/ether 1a:72:d9:38:a7:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global h1-eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1872:d9ff:fe38:a7d4/64 scope link
       valid_lft forever preferred_lft forever
mininet>
```

VII. INSTALANDO APLICAÇÃO

Após efetuar o reset nas configurações, é preciso iniciar o controlador SDN ONOS e verificar se os devices estão ativos e operantes para podermos prosseguir com o ambiente SDN-IP, efetuando este procedimento é possível verificar qual rede foi iniciada e conectada ao ONOS, conforme ilustra na FIGURA 3:

FIGURA 3



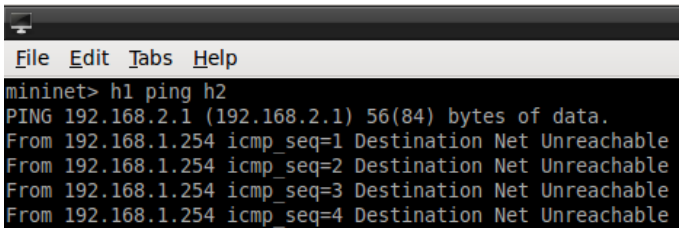
```
Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.

onos> devices
id=of:00000000000000a1, available=false, role=NONE, type=SWITCH, mfr=Nicira, Inc., hw=Open vSwitch, sw=2.1.3, serial=None, protocol=OF_10, channelId=10.0.3.1:50
052
id=of:00000000000000a2, available=false, role=NONE, type=SWITCH, mfr=Nicira, Inc., hw=Open vSwitch, sw=2.1.3, serial=None, protocol=OF_10, channelId=10.0.3.1:50
050
id=of:00000000000000a3, available=false, role=NONE, type=SWITCH, mfr=Nicira, Inc., hw=Open vSwitch, sw=2.1.3, serial=None, protocol=OF_10, channelId=10.0.3.1:50
048
id=of:00000000000000a4, available=false, role=NONE, type=SWITCH, mfr=Nicira, Inc., hw=Open vSwitch, sw=2.1.3, serial=None, protocol=OF_10, channelId=10.0.3.1:50
047
id=of:00000000000000a5, available=false, role=NONE, type=SWITCH, mfr=Nicira, Inc., hw=Open vSwitch, sw=2.1.3, serial=None, protocol=OF_10, channelId=10.0.3.1:50
049
id=of:00000000000000a6, available=false, role=NONE, type=SWITCH, mfr=Nicira, Inc., hw=Open vSwitch, sw=2.1.3, serial=None, protocol=OF_10, channelId=10.0.3.1:50
051
onos>
```

VIII. EXECUTANDO SDN-IP

Primeiro precisamos efetuar o procedimento de troca de pacotes icmp, sendo assim iremos utilizar o software ping para efetuar a comunicação entre dois hosts e percebemos que não é possível chegar até o destino solicitado pelo software ping, conforme a FIGURA 4:

FIGURA 4



```
File  Edit  Tabs  Help
mininet> h1 ping h2
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
From 192.168.1.254 icmp_seq=1 Destination Net Unreachable
From 192.168.1.254 icmp_seq=2 Destination Net Unreachable
From 192.168.1.254 icmp_seq=3 Destination Net Unreachable
From 192.168.1.254 icmp_seq=4 Destination Net Unreachable
```

IX. INSTALANDO APLICAÇÕES

Primeiro precisamos instalar alguns aplicativos auxiliares nos quais o SDN-IP irá utilizar neste ambiente. Esses aplicativos permitem ONOS ler em vários arquivos de configuração e responder a solicitações ARP entre os roteadores externos e internos do BGP, conforme a FIGURA 5, em seguida iremos verificar quais redes o controlador ONOS conhece, assim podermos fazer testes de comunicação e aplicar devidas rotas, conforme a FIGURA 6:

FIGURA 5

```
LXTerminal
File  Edit  Tabs  Help
onos> app activate org.onosproject.config
onos> app activate org.onosproject.proxyarp
onos> app activate org.onosproject.sdnip
onos>
```

Com o aprendizado das redes existentes o controlador ONOS programou programa algumas dessas rotas nos switches, usando a API de intenção. Para verificar na prática, é possível ver este processo ilustrado na FIGURA 7, que lista o resumo da intenção, onde é possível ver as diferentes intenções que SDN-IP está usando.

FIGURA 6

```
onos> routes
   Network            Next Hop
   192.168.1.0/24     10.0.1.1
   192.168.2.0/24     10.0.2.1
   192.168.3.0/24     10.0.3.1
Total SDN-IP IPv4 routes = 3

   Network            Next Hop
Total SDN-IP IPv6 routes = 0
```

FIGURA 7

```
onos> intents -s
Connectivity              total=            27   installed=        27
Connectivity              withdrawn=         0   failed=            0
Connectivity              installReq=        0   compiling=         0
Connectivity              installing=        0   recompiling=       0
Connectivity              withdrawReq=       0   withdrawing=       0
HostToHost                total=             0   installed=         0
HostToHost                withdrawn=         0   failed=            0
HostToHost                installReq=        0   compiling=         0
HostToHost                installing=        0   recompiling=       0
HostToHost                withdrawReq=       0   withdrawing=       0
PointToPoint              total=            24   installed=        24
PointToPoint              withdrawn=         0   failed=            0
PointToPoint              installReq=        0   compiling=         0
PointToPoint              installing=        0   recompiling=       0
PointToPoint              withdrawReq=       0   withdrawing=       0
MultiPointToSinglePoint   total=             3   installed=         3
MultiPointToSinglePoint   withdrawn=         0   failed=            0
MultiPointToSinglePoint   installReq=        0   compiling=         0
MultiPointToSinglePoint   installing=        0   recompiling=       0
MultiPointToSinglePoint   withdrawReq=       0   withdrawing=       0
SinglePointToMultiPoint   total=             0   installed=         0
SinglePointToMultiPoint   withdrawn=         0   failed=            0
SinglePointToMultiPoint   installReq=        0   compiling=         0
SinglePointToMultiPoint   installing=        0   recompiling=       0
SinglePointToMultiPoint   withdrawReq=       0   withdrawing=       0
Path                      total=             0   installed=         0
Path                      withdrawn=         0   failed=            0
Path                      installReq=        0   compiling=         0
Path                      installing=        0   recompiling=       0
Path                      withdrawReq=       0   withdrawing=       0
LinkCollection            total=             0   installed=         0
LinkCollection            withdrawn=         0   failed=            0
LinkCollection            installReq=        0   compiling=         0
```

X.TESTE DE CONECTIVIDADE SDN-IP

Após instalar as devidas aplicações no controlador ONOS, iremos simular um teste de icmp com o software ping e conferir o resultado após o procedimento de instalação de aplicações no ONOS, conforme é ilustrado na FIGURA 8:

FIGURA 8

```
LXTerminal
File  Edit  Tabs  Help
mininet> h1 ping h2 -c 10
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=62 time=0.702 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=62 time=0.178 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=62 time=0.158 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=62 time=0.207 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=62 time=0.189 ms
64 bytes from 192.168.2.1: icmp_seq=6 ttl=62 time=0.178 ms
64 bytes from 192.168.2.1: icmp_seq=7 ttl=62 time=0.128 ms
64 bytes from 192.168.2.1: icmp_seq=8 ttl=62 time=0.213 ms
64 bytes from 192.168.2.1: icmp_seq=9 ttl=62 time=0.508 ms
64 bytes from 192.168.2.1: icmp_seq=10 ttl=62 time=0.181 ms

--- 192.168.2.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9001ms
rtt min/avg/max/mdev = 0.128/0.264/0.702/0.177 ms
mininet>
```

Durante o teste de icmp com o software ping é possível verificar que a comunicação foi efetuada com sucesso, porém uma nota deve ser tomada a respeito do ttl (time to live), onde corresponde 62 e não a 64, como é padrão de sistemas Linux, isso ocorre devido o roteador que hospeda o host de origem e o roteador do host de destino possuírem a versão 1.0 do OpenFlow, sendo assim o ambiente SDN-IP atualmente, utiliza o padrão de TTL 62, o que não tem certa importância ao ambiente de rede SDN, devido os switches OpenFlow 1.0 não suportarem o TTL 64, com isso o mesmo efetua o processo de decremento de TTL, tornando de 64 para 62, agindo de forma mais flexível, durante a troca de pacotes icmp, conforme ilustrado na FIGURA 8.

XI. ANÚNCIO DE ROTA

Após termos o ambiente SDN-IP em funcionamento operacional, o próximo processo, é fazer com o que um dos roteadores externos anuncie uma nova rota, que nos permitirá efetuar comunicação com um novo host. Neste momento o r4 não está anunciando nenhuma rota, então não podemos obter comunicação com o host h4, antes será efetuado um novo teste de icmp com o ping, para verificar a tentativa de comunicação com o host h4, através do host h1 que está conectado em outra rede distinta, conforme a FIGURA 9:

FIGURA 9

```
LXTerminal
File  Edit  Tabs  Help
mininet> h1 ping h4
PING 192.168.4.1 (192.168.4.1) 56(84) bytes of data.
From 192.168.1.254 icmp_seq=1 Destination Net Unreachable
From 192.168.1.254 icmp_seq=2 Destination Net Unreachable
From 192.168.1.254 icmp_seq=3 Destination Net Unreachable
From 192.168.1.254 icmp_seq=4 Destination Net Unreachable
```

Nesta próxima etapa deste artigo técnico, será aplicar uma rota para a rede 192.168.4.0/24, a qual o host h4 pertence, assim o roteador irá conseguir se comunicar com qualquer host dentro da mesma, para isso é preciso abrir o um novo terminal, onde o processo de configuração será feito na suíte de roteamento Quagga, software de código aberto, que prove serviços de roteamento dinâmico com diversos protocolos de rede, o processo de configuração será ilustrado nas FIGURAS 10 e 11:

FIGURA 10

```
"Node: h4"
root@mininet-vm:~# telnet 10.0.4.1 2605
Trying 10.0.4.1...
Connected to 10.0.4.1.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.23).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
r4>
```

FIGURA 11

```
"Node: h4"
root@mininet-vm:~# telnet 10.0.4.1 2605
Trying 10.0.4.1...
Connected to 10.0.4.1.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.23).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
r4> enable
r4# conf t
r4(config)# router bgp 65004
r4(config-router)# network 192.168.4.0/24
r4(config-router)# exit
r4(config)# exit
r4# exit
Connection closed by foreign host.
root@mininet-vm:~#
```

Depois da conclusão da configuração de roteamento, vamos verificar no controlador ONOS se o mesmo já conhece a rede do host h4, 192.168.4.0/24, para uma possível comunicação através do protocolo icmp com o nó h4, conforme esta ilustrado na FIGURA 12:

FIGURA 12

```
onos> routes
   Network            Next Hop
   192.168.1.0/24     10.0.1.1
   192.168.2.0/24     10.0.2.1
   192.168.3.0/24     10.0.3.1
   192.168.4.0/24     10.0.4.1
Total SDN-IP IPv4 routes = 4

   Network            Next Hop
Total SDN-IP IPv6 routes = 0
onos>
```

Além do aprendizado da nova rede que esta diretamente conectado ao host h4, 192.168.4.0/24, o ambiente SDN-IP instalou uma nova intenção MultiPointToSinglePoint na rede, conforme esta ilustrado na FIGURA 13:

FIGURA 13

```
MultiPointToSinglePoint    total=          4   installed=      4
MultiPointToSinglePoint    withdrawn=      0   failed=         0
MultiPointToSinglePoint    installReq=     0   compiling=      0
MultiPointToSinglePoint    installing=     0   recompiling=    0
```

XII.TESTE DE CONECITIVIDADE

Devemos nos atentar com o número de intenções que temos no ambiente SDN-IP, onde o mesmo recebeu mais uma nova rede, com isso a intenção MultiPointToSinglePoint aumentou para 4, sendo assim, já é possível efetuar comunicação com a rede do host h4, partindo do host h1, conforme está ilustrado na FIGURA 14:

FIGURA 14

```
LXTerminal
File  Edit  Tabs  Help
mininet> h1 ping h4 -c 10
PING 192.168.4.1 (192.168.4.1) 56(84) bytes of data.
64 bytes from 192.168.4.1: icmp_seq=1 ttl=62 time=0.960 ms
64 bytes from 192.168.4.1: icmp_seq=2 ttl=62 time=0.173 ms
64 bytes from 192.168.4.1: icmp_seq=3 ttl=62 time=0.228 ms
64 bytes from 192.168.4.1: icmp_seq=4 ttl=62 time=0.221 ms
64 bytes from 192.168.4.1: icmp_seq=5 ttl=62 time=0.252 ms
64 bytes from 192.168.4.1: icmp_seq=6 ttl=62 time=0.230 ms
64 bytes from 192.168.4.1: icmp_seq=7 ttl=62 time=0.202 ms
64 bytes from 192.168.4.1: icmp_seq=8 ttl=62 time=0.223 ms
64 bytes from 192.168.4.1: icmp_seq=9 ttl=62 time=0.222 ms
64 bytes from 192.168.4.1: icmp_seq=10 ttl=62 time=0.278 ms

--- 192.168.4.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9000ms
rtt min/avg/max/mdev = 0.173/0.298/0.960/0.223 ms
mininet>
```

XIII. CONCLUSÃO

Neste procedimento denominado de SDN-IP foi possível demonstrar de forma prática a comunicação entre redes distintas utilizando o controlador SDN ONOS, trabalhando em paralelo com o emulador Mininet, o que proporcionou excelente estabilidade e agilidade no processo de roteamento utilizando o protocolo de roteamento BGP, através da suíte de código aberto de roteamento Quagga, testes efetuados com o programa ping, utilizando o protocolo icmp, comprovaram a eficácia do ambiente SDN-IP, onde concluo que o ambiente testado em ambiente virtual é apto para ser colocado em produção em ambiente físico.

XIV.REFERÊNCIAS.

[1] Open Network Operating System (ONOS) Wiki!, SDN-IP Tutorial (https://wiki.onosproject.org/display/ONOS/SDN-IP+Tutorial), (https://drive.google.com/file/d/1JcGUJJDTtbHNnbFzC7SUK52RmMDBVUry/view)

[2] Serviço Federal de Processamento de Dados - Uma Arquitetura SDN para Computação em Nuvem orquestrada pelo ONOS (http://intra.serpro.gov.br/tema/tematec/arquitetura-sdn-computacao-nuvem)

[3] FEEC - Faculdade de Engenharia Elétrica e de Computação da Unicamp (https://intrig.dca.fee.unicamp.br:8840/owncloud/index.php/s/ilUsYFLcVcPmOMv)