

Relatório do EP3

MAC0352 – Redes de Computadores e Sistemas Distribuídos – 2s2017

Cesar Cano (8536169)
Lucas Romão (8536214)
Victor Sprengel (9298002)

1 Passo 0

Na definição do protocolo OpenFlow, o que um switch faz toda vez que ele recebe um pacote que ele nunca recebeu antes?

O comportamento do switch ao receber um pacote não identificado depende da configuração das tabelas de fluxo do OpenFlow. O comportamento padrão é enviar o pacote para o controlador por mensagem através do canal de controle. Uma outra opção é descartar o pacote.

2 Passo 2

Com o acesso à Internet funcionando em sua rede local, instale na VM o programa `traceroute` usando `sudo apt install traceroute` e escreva abaixo a saída do comando `sudo traceroute -I www.inria.fr`. Pela saída do comando, a partir de qual salto os pacotes alcançaram um roteador na Europa? Como você chegou a essa conclusão?

```
traceroute to www.inria.fr (128.93.162.84), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  3.980 ms  3.835 ms  3.739 ms
 2  10.61.0.1 (10.61.0.1)  16.533 ms  21.874 ms  22.779 ms
 3  c9062081.virtua.com.br (201.6.32.129)  23.178 ms  23.639 ms  24.326 ms
 4  c9062905.virtua.com.br (201.6.41.5)  24.941 ms  30.769 ms  31.640 ms
 5  embratel-T0-0-0-1-uacc03.spoph.embratel.net.br (200.212.132.1)  32.390 ms
 6  ebt-T0-3-0-0-tcore01.spo.embratel.net.br (200.230.159.86)  35.918 ms  18.75
 7  ebt-B11151-intl01.atl.embratel.net.br (200.230.230.32)  145.076 ms  131.500
 8  * * *
 9  ae-1-3514.edge2.Atlanta4.Level3.net (4.69.150.197)  138.561 ms  134.929 ms
10  gtt-level3.Atlanta4.level3.net (4.68.63.158)  137.605 ms  142.098 ms  146.
11  xe-1-3-1.cr0-par7.ip4.gtt.net (89.149.185.65)  240.778 ms  228.628 ms  245
12  renater-gw-ix1.gtt.net (77.67.123.206)  245.475 ms  234.145 ms  239.995 ms
13  193.51.177.107 (193.51.177.107)  231.548 ms  232.486 ms  230.774 ms
14  inria-rocquencourt-tel-4-inria-rtr-021.noc.renater.fr (193.51.184.177)  23
15  unit240-reth1-vfw-ext-dc1.inria.fr (192.93.122.19)  230.431 ms  229.849 ms
16  ezp3.inria.fr (128.93.162.84)  229.359 ms  234.286 ms  230.867 ms
```

Pesquisando pelos endereços IP no site <http://geoiplookup.net/>, chegou-se à conclusão de que os pacotes chegaram a um roteador na Europa no passo 11 onde chegou a um endereço de IP localizado na Alemanha.

3 Passo 3 - Parte 1

Execute o comando `iperf`, conforme descrito no tutorial, antes de usar a opção `--switch user`, 5 vezes. Escreva abaixo o valor médio e o intervalo de confiança da taxa retornada (considere sempre o primeiro valor do vetor retornado).

Resultado de 5 execuções do `iperf`

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['28.6 Gbits/sec', '28.6 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['28.1 Gbits/sec', '28.2 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['28.1 Gbits/sec', '28.2 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['29.4 Gbits/sec', '29.4 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['28.5 Gbits/sec', '28.6 Gbits/sec']
```

Valor médio: 28.54 Gbits/sec Intervalo de confiança de 95%: (27.88, 29.20)

4 Passo 3 - Parte 2

Execute o comando `iperf`, conforme descrito no tutorial, com a opção `--switch user`, 5 vezes. Escreva abaixo o valor médio e o intervalo de confiança da taxa retornada (considere sempre o primeiro valor do vetor retornado). O resultado agora corresponde a quantas vezes menos o da Seção anterior? Qual o motivo dessa diferença?

Resultado de mais cinco execuções do comando `iperf` com a opção `--switch-user`:

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['1.28 Mbits/sec', '1.36 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['1.45 Mbits/sec', '1.50 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['1.44 Mbits/sec', '1.49 Mbits/sec']
```

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['1.28 Mbits/sec', '1.36 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['1.44 Mbits/sec', '1.49 Mbits/sec']
```

Resultado médio: 1.38Mbits/sec Intervalo de confiança de 95%: (1.27, 1.49)

O resultado do segundo experimento é 20711 menor em relação ao da seção anterior isso se deve ao fato de que, quando usada a opção `--switch-user`, os pacotes precisam passar entre o espaço de usuário e o espaço do kernel e voltar para o espaço de usuário a cada salto ao passo que na seção anterior cada pacote ficava apenas no espaço de kernel.

5 Passo 4 - Parte 1

Execute o comando `iperf`, conforme descrito no tutorial, usando o controlador `of_tutorial.py` original sem modificação, 5 vezes. Escreva abaixo o valor médio e o intervalo de confiança da taxa retornada (considere sempre o primeiro valor do vetor retornado). O resultado agora corresponde a quantas vezes menos o da Seção 3? Qual o motivo para essa diferença? Use a saída do comando `tcpdump`, deixando claro em quais computadores virtuais ele foi executado, para justificar a sua resposta.

Resultado de cinco execuções do comando `iperf` usando o controlador: `of_tutorial.py`:

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['19.9 Mbits/sec', '22.6 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['18.6 Mbits/sec', '21.1 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['15.3 Mbits/sec', '16.9 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['18.1 Mbits/sec', '20.5 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['20.0 Mbits/sec', '24.3 Mbits/sec']
```

Valor médio: 18.38 Mbits/sec Intervalo de confiança de 95%: (16.01, 20.75)

O novo resultado é 1552 vezes menor do que o resultado visto na seção 1. Isso se deve ao fato de os pacotes irem para o controlador e esse por sua vez manda os pacotes para todas as portas ao invés de mandar para apenas uma.

6 Passo 4 - Parte 2

Execute o comando `iperf`, conforme descrito no tutorial, usando o seu controlador `switch.py`, 5 vezes. Escreva abaixo o valor médio e o intervalo de confiança da taxa retornada (considere sempre o primeiro valor do vetor retornado). O resultado agora corresponde a quantas vezes mais o da Seção anterior? Qual o motivo dessa diferença? Use a saída do comando `tcpdump`, deixando claro em quais computadores virtuais ele foi executado, para justificar a sua resposta.

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['20.9 Mbits/sec', '24.1 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['18.3 Mbits/sec', '20.8 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['19.2 Mbits/sec', '21.9 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['19.8 Mbits/sec', '22.5 Mbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['19.2 Mbits/sec', '22.0 Mbits/sec']
```

Valor médio: 19.48 Mbits/sec Intervalo de confiança de 95%: (18.64, 20.32)

Quase não houve diferença apesar de agora estarmos usando o `resend packet` quando a porta destino é conhecida, talvez a diferença entre os computadores que foram usados desse passo pro anterior tenha algo a ver, mas de qualquer forma a mudança não deveria ser muito grande dado que os pacotes ainda são reenviados várias vezes e o switch não cria os fluxos para evitar passar pelo controlador.

7 Passo 4 - Parte 3

Execute o comando `iperf`, conforme descrito no tutorial, usando o seu controlador `switch.py` melhorado, 5 vezes. Escreva abaixo o valor médio e o intervalo de confiança da taxa retornada (considere sempre o primeiro valor do vetor retornado). O resultado agora corresponde a quantas vezes mais o da Seção anterior? Qual o motivo dessa diferença? Use a saída do comando `tcpdump`, deixando claro em quais computadores virtuais ele foi executado, e saídas do comando `sudo ovs-ofctl`, com os devidos parâmetros, para justificar a sua resposta.

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['28.9 Gbits/sec', '28.9 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['28.6 Gbits/sec', '28.7 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
```

```

*** Results: ['29.9 Gbits/sec', '30.0 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['29.6 Gbits/sec', '29.7 Gbits/sec']
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['27.4 Gbits/sec', '27.5 Gbits/sec']

```

Valor médio: 28.8 Gbits/sec Intervalo de confiança de 95%: (27.94, 29.66)

O resultado é 1571 vezes maior do que o da seção anterior, pois agora os fluxos são adicionados conforme as portas são descobertas, evitando alguns caminhos que passam pelo controlador após fluxos serem adicionados.

Uso de `ovs-ofctl` antes e depois de executar um pingall:

```

mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=2.755s, table=0, n_packets=1, n_bytes=98, idle_timeout=3
  cookie=0x0, duration=2.681s, table=0, n_packets=2, n_bytes=196, idle_timeout=3
  cookie=0x0, duration=2.717s, table=0, n_packets=1, n_bytes=98, idle_timeout=3
  cookie=0x0, duration=2.489s, table=0, n_packets=0, n_bytes=0, idle_timeout=30
  cookie=0x0, duration=2.641s, table=0, n_packets=1, n_bytes=98, idle_timeout=3
  cookie=0x0, duration=2.609s, table=0, n_packets=2, n_bytes=140, idle_timeout=3
mininet@mininet-vm:~$

```

Após o pingall, a tabela de fluxos está preenchida.

8 Passo 5

Explique a lógica implementada no seu controlador `firewall.py` e mostre saídas de comandos que comprovem que ele está de fato funcionando (saídas dos comandos `tcpdump`, `sudo ovs-ofctl`, `nc`, `iperf` e `telnet` são recomendadas)

9 Configuração dos computadores virtual e real usados nas medições (se foi usado mais de um, especifique qual passo foi feito com cada um)

Para seção 6 e 7, foi utilizado um computador com processador i7-4510U, 6GB de RAM, Ubuntu 16.04 64 bits. e uma maquina virtual Mininet-VM 64 bits com 1GB de RAM.

E para as outras seções: Processador i7, 8gb de RAM, 1 TB HD, Antergos Linux.

10 Referências

- www.comp.nus.edu.sg/tbma/teaching/cs5229y14_past/Tutorial.pdf