

Simulações de Rede com Mininet

Rafael Menezes Barboza¹, Jonas Felipe Alves²

¹ Universidade Tecnológica Federal do Paraná (UTFPR)

1. Introdução

O seguinte trabalho tem o objetivo de apresentar de maneira detalhada, os passos seguidos para a elaboração da APS proposta na disciplina de Redes 1. A APS consiste em implementar a simulação de dois cenários distintos de redes. Sendo o primeiro formado por um servidor conectado a um switch, este switch interliga quatro hosts, dois configurados com IP fixo e dois através do protocolo DHCP. Já o segundo cenário, é formado por duas redes executando em um mesmo roteador Linux, assim, teremos dois switches (um para cada rede) e dois hosts para cada switch, todos estes hosts configurados com IP fixo.

2. Tecnologias utilizadas

SSH: Projetado para substituir o telnet, o SSH fornece um canal seguro para conectar aplicações cliente-servidor.

Mininet: Desenvolvido por Eric Biederman, mininet é o responsável por simular e virtualizar o ambiente de rede pretendido.

VirtualBox: máquina virtual utilizada sob o mininet, no processo de virtualização utilizou-se de um sistema operacional distribuição Linux.

Python: Linguagem de programação utilizada para a escrita dos scripts referentes a cada cenário.

3. Cenários propostos

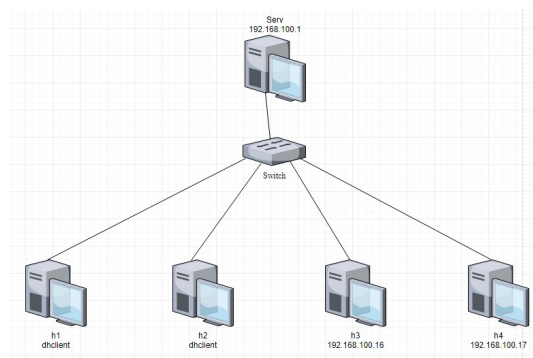


Figura 1. Representação do cenário 1, formado por um servidor, um switch e quatro hosts, sendo dois com ips fixos e dois via DHCP.

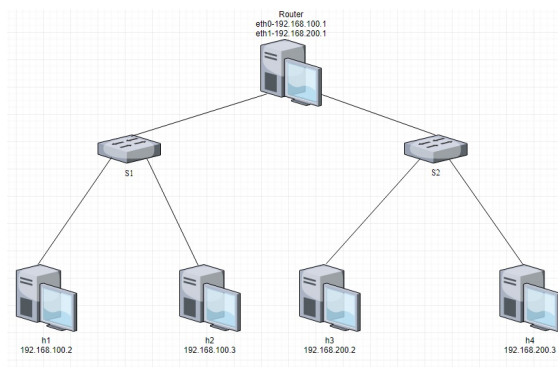


Figura 2. Representação do cenário 2, formado por duas redes, dois switches e quatro hosts de IP fixo, sendo dois para cada rede e switch.

4. Tutorial

Para reproduzir ambos os cenários aqui abordados primeiramente realize o download de uma Máquina Virtual (VM) Mininet / Ubuntu. Esta VM inclui o Mininet, todos os binários OpenFlow e ferramentas pré-instaladas, e ajustes na configuração do kernel para suportar as simulações, é possível realizar o download no próprio site do Mininet [Mininet 2017].

Para executar esta Máquina virtual precisamos de um software de virtualização, foi utilizado neste trabalho o Virtual Box [Oracle 2017] ele permite a instalação e utilização de um sistema operacional dentro de outro, assim como seus respectivos softwares, como dois ou mais computadores independentes, mas compartilhando fisicamente o mesmo hardware. Após realizar o download é só iniciar a instalação da imagem virtual (VM) do mininet com as configurações auto recomendadas.

Afim de tornar esta experiencia mais agradável iremos configurar o acesso remoto via Secure Shell (SSH) da maquina virtual assim podemos utilizar recursos do sistema operacional nativo como o sistema de janelas, porem nada impede de que se instale uma interface gráfica na própria Máquina Virtual (VM) porem não será abordado neste tutorial. Para permitir o acesso via Secure Shell (SSH) iremos alterar as configurações de rede do VirtualBox clicando no botão de configurações e na sessão de redes, ativar o adaptador 2 com a configuração "Placa de rede exclusiva de hospedeiro (host-only)" permitindo que o sistema operacional nativo reconheça a Máquina Virtual como um Host acessível, instale o servidor DHCP através do comando "sudo apt-get install isc-dhcp-server" é necessário também alterar o arquivo de configuração do servidor DHCP através do comando 'sudo nano /etc/dhcp/dhcpd.conf', iremos inserir ao final do arquivo as configurações da Figura 3 que dará suporte a criação das redes no mininet e a utilização do DHCP. Após isto já podemos nos conectar com a maquina virtual, pra isto realizamos o comando "ifconfig" na própria (VM), com isto em mãos iremos acessar a maquina virtual pelo comando "sudo ssh mininet@'ip-da-vm' -Y" partindo do sistema nativo, o usuário e a senha são os mesmos da maquina virtual portanto Usuário: mininet, Senha: mininet.

```

subnet 192.168.100.0 netmask 255.255.255.0{
    range 192.168.100.10 192.168.100.20;
    option routers 192.168.100.1;
    option broadcast-address 192.168.100.255;
}

```

Figura 3. Cria uma Subnet com mascara de rede, range de IPs, endereço de broadcast e routers

4.1. Cenário 1

Para construção deste ambiente utilizaremos o código em Python abaixo, nas primeiras linhas (2 a 8) é realizado alguns imports e instanciações de ferramentas do mininet necessárias para construção do ambiente, as linhas (10 a 16) é realizado a criação das maquinas necessárias no caso "h1, h2, h3, h4", um servidor DHCP com nome de "serv", e um Switch que realiza a ligações entre eles, as linhas (19 a 24) realizam o link de cada host com o meu Switch "s1", ao final do comando "net.start" temos topologia de rede semelhante a Figura 1.

Agora, é necessário atribuir o IP do servidor DHCP e inicia-lo para que atue como um servidor DHCP, isto é realizado nas linhas (26 e 27) respectivamente. para este ambiente metade dos hosts de tal LAN devem ter os IPs, gateway padrão e configurados de forma estática/manual, as linhas (29 a 32) atribuem ao hosts (h3 e h4) os Ips manuais e setam o gateway padrão que no caso é ("192.168.100.1") a outra metade dos hosts no caso "h1 e h2" recebem seus IPs do servidor DHCP pelo comando 'dhclient' realizado para ambas as maquinas nas linhas (33 e 34).

A configuração do do DNS pode ser feita manualmente a partir do terminal de qualquer host, é só alterar o arquivo localizado em "/etc/hosts" inserir o IP do host e o nome correspondente exemplo "192.168.100.2 h1".

```

1  #!/usr/bin/python
2  from mininet.topo import Topo
3  from mininet.net import Mininet
4  from mininet.util import dumpNodeConnections
5  from mininet.log import setLogLevel
6  from mininet.cli import CLI
7
8  net = Mininet()
9  # Adiciona os hosts
10 serv = net.addHost('serv')
11 h1 = net.addHost('h1')
12 h2 = net.addHost('h2')
13 h3 = net.addHost('h3')
14 h4 = net.addHost('h4')
15 s1 = net.addSwitch('s1')
16 c0 = net.addController('c0')
17
18 # Adiciona os links

```

```

19 net.addLink(serv, s1)
20 net.addLink(h1, s1)
21 net.addLink(h2, s1)
22 net.addLink(h3, s1)
23 net.addLink(h4, s1)
24 net.start()
25
26 serv.cmd( 'ifconfig serv-eth0 192.168.100.1')
27 serv.cmd( 'dhcpd -cf /etc/dhcp/dhcpd.conf')
28
29 h3.cmd( 'ifconfig h3-eth0 192.168.100.16' )
30 h3.cmd( 'route add default gw 192.168.100.1')
31 h4.cmd( 'ifconfig h4-eth0 192.168.100.17' )
32 h4.cmd( 'route add default gw 192.168.100.1')
33 h1.cmd( 'dhclient')
34 h2.cmd( 'dhclient')
35 CLI(net)
36 net.stop()

```

4.2. Cenário 2

Para construção do ambiente 2 iremos utilizar o código abaixo, nas primeiras linhas (2 a 8) é realizado alguns imports e instâncias de ferramentas do mininet, como teremos duas redes conectadas por um roteador, iremos criar os hosts "h1, h2, h3, e h4" os switches das duas redes "s1 e s2", e o roteador que no nosso caso se trata também de um host "router", realizado nas linhas (10 a 17) respectivamente.

É realizado os links de cada host com seu switch nas linhas (20 a 23) por tanto temos o a seguinte ligação "h1 - sw1, h2 - sw2" e "h3 - sw2, h4 - sw2", nas linhas (24 e 27) é feita a ligação dos switches "s1 e s2" com o "router" ao final do comando "net.start" temos uma rede semelhante a Figura 2.

As linhas (29 a 36) do script configuram os IPs e o gateway de cada host, sendo que "h1 e h2" estão na mesma rede por tanto recebem o mesmo endereço de gateway "192.168.100.1", caracterizado como Rede 1, os hosts "h3 e h4" recebem o gateway "192.168.200.1" e estão na Rede 2.

A configuração do roteador, ocorre nas linhas (37 a 39) configuramos a Rede 1 na placa eth0 com o IP "192.168.100.1" e da Rede 2 na placa eth1 com o IP "192.168.200.1", assim a máquina router está ligada as 2 redes por através de duas placas de redes, para que a host "router" realize o roteamento de pacote entre as redes é necessário atribuir o valor 1 ao arquivo "ip_forward" vemos isto acontecer na linha 39.

Para comprovar o roteamento de pacotes entre as redes foi montado um simples servidor HTTP rodando no host "h1" o comando da linha 40 inicia o servidor HTTP e se torna acessível a qualquer host das redes 1 e 2 através de seu IP "192.168.100.2".

```

1 #!/usr/bin/python
2 from mininet.topo import Topo
3 from mininet.net import Mininet

```

```

4 from mininet.util import dumpNodeConnections
5 from mininet.log import setLogLevel
6 from mininet.cli import CLI
7
8 net = Mininet()
9 #Criando os hosts/Sw/routers
10 h1 = net.addHost('h1')
11 h2 = net.addHost('h2')
12 h3 = net.addHost('h3')
13 h4 = net.addHost('h4')
14 s1 = net.addSwitch('s1')
15 router= net.addHost('router')
16 s2 = net.addSwitch('s2')
17 c0 = net.addController( 'c0')
18
19 #Criando Links
20 net.addLink(h1, s1)
21 net.addLink(h2, s1)
22 net.addLink(h3, s2)
23 net.addLink(h4, s2)
24 net.addLink(s1, router)
25 net.addLink(s2, router)
26
27 net.start()
28
29 h1.cmd('ifconfig h1-eth0 192.168.100.2')
30 h1.cmd('route add default gw 192.168.100.1')
31 h2.cmd('ifconfig h2-eth0 192.168.100.3')
32 h2.cmd('route add default gw 192.168.100.1')
33 h3.cmd('ifconfig h3-eth0 192.168.200.2')
34 h3.cmd('route add default gw 192.168.200.1')
35 h4.cmd('ifconfig h4-eth0 192.168.200.3')
36 h4.cmd('route add default gw 192.168.200.1')
37 router.cmd( 'ifconfig router-eth0 192.168.100.1')
38 router.cmd( 'ifconfig router-eth1 192.168.200.1')
39 router.cmd("echo 1 > /proc/sys/net/ipv4/ip_forward")
40 h1.cmd('sudo /etc/init.d/apache2 restart' )
41 CLI(net)
42 net.stop()

```

5. Resultados

5.1. Ambiente 1

Podemos observar na Figura 4 que todos os hosts receberam um IP sendo ele via DHCP ou estático, A Figura 5 mostra que estão conectados, pois é possível realizar o comando 'ping' entre todos os hosts da rede, para os hosts com IP fixo foi possível incluir seu nome no arquivo de DNS, podendo assim realizar o seguinte comando "ping h3"ou "ping h4", como a outra metade dos hosts recebem um IP via DHCP não podemos

adivinhar qual IP sera atribuído para inserir no arquivo DNS por tanto conseguimos pingar estes com seus determinados Ips como ilustra na Figura 4.

The figure displays four terminal windows, each representing a host in a mininet environment. Each window shows the output of the 'ifconfig' command for the host's primary interface (h1-eth0, h2-eth0, h3-eth0, h4-eth0) and the loopback interface (lo). The configuration includes the IP address, netmask, broadcast address, and various statistics for received and transmitted packets, errors, and collisions. For example, h1-eth0 has IP 192.168.100.12, and h3-eth0 has IP 192.168.100.15.

Figura 4. Imagem dos hosts com seus determinados IPs

The figure displays four terminal windows showing the results of ping tests between the hosts. Each window shows the output of the 'ping' command from a specific host to another. The results include the IP address of the destination, the size of the packet, the time taken for the packet to reach the destination (time), and the sequence number (seq=1). For example, h1 pings h3 at 192.168.100.15, and h2 pings h4 at 192.168.100.17.

Figura 5. Pings entre todos os hosts

5.2. Ambiente 2

Podemos observar na Figura 6 que todos os hosts da rede receberam um IP e que foi incluído no arquivo de DNS pois conseguimos realizar o comando 'ping' através do

nomes de de cara host, na Figura 7 temos a demonstração do servidor HTTP configurado na maquina h1, e sendo executado em todas as maquinas da rede, através de um navegador de linha de comando.

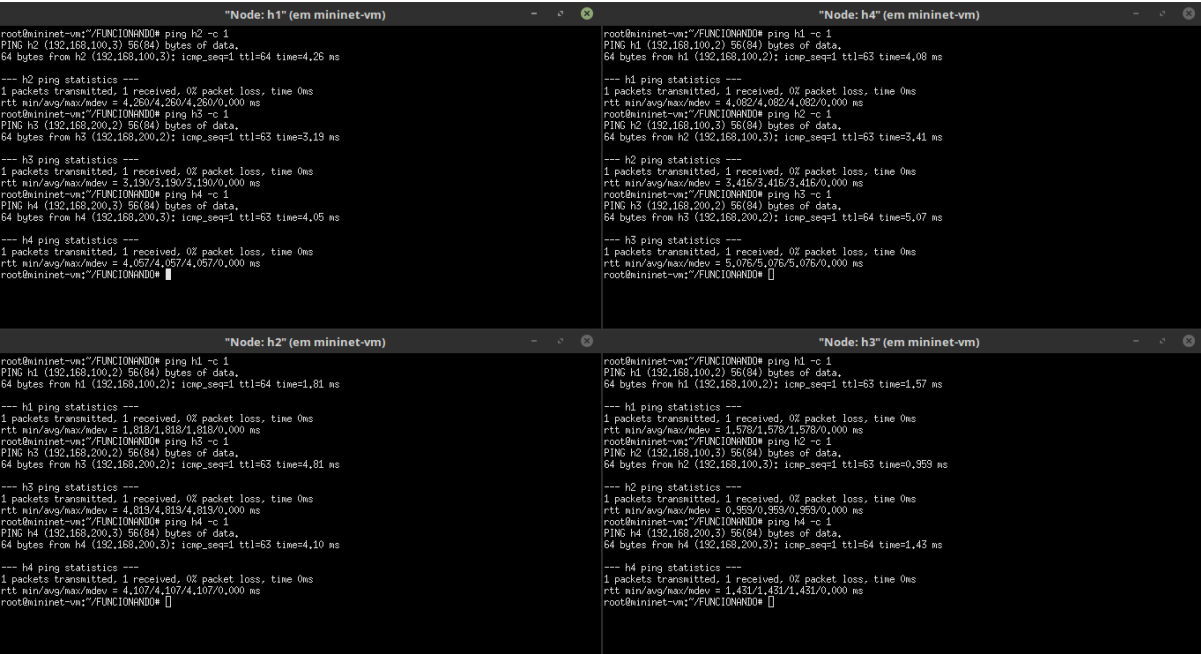


Figura 6. Pings entre todos os hosts



Figura 7. Serviço HTTP funcionando em todos os hosts

6. Dificuldades enfrentadas

A maior dificuldade enfrentada nesse trabalho foi configurar o servidor DHCP e fazer com que funcione através do comando "dhclient".

7. Conclusão

Este trabalho foi muito importante para entender na pratica as teorias aprendidas em sala de aula. Onde foi possível implementar e observar o funcionamento das redes.

Mininet (2017). Mininet vm images.

Oracle (2017). Virtualbox.