

Lab 4 : Endereçamento IP e Roteamento Estático

Objetivos

O objetivo inicial será reproduzir a antiga topologia física do laboratório no ambiente do emulador para em seguida realizar as atividades práticas de roteamento IP como indicado.

- Familiarizar-se com a criação de topologias de rede através do Miniedit;
- Exercitar a configuração do endereçamento IP e roteamento estático dentro de uma topologia criada no Mininet.

Introdução

Laboratório: Antiga topologia física no LE-25 na FEEC.

O laboratório LE-25 no bloco B da FEEC possui 20 estações de trabalho, 6 switches (comutadores Ethernet de camada 2 / camada de enlace ou layer 2 - L2) e 6 roteadores (camada de rede, ou L3), organizados conforme a figura 1. Cada 4 (quatro) estações de trabalho estão conectadas a um único switch SW2 (Layer 2).

Lembrar que uma estação com endereço 10.0.0.X ($101 \leq X \leq 120$) pode possuir várias máquinas virtuais conectadas à rede física do laboratório. Caso seja necessário definir endereços IP para as máquinas virtuais interconectadas à rede física (endereços laranja na figura 1), os respectivos endereços devem seguir a seguinte norma: 10.0.0.(X+20); 10.0.0.(X+40); 10.0.0.(X+60); ..., ou seja, os endereços devem ser criados em sequência com um acréscimo de 20 ao valor anterior do último campo do endereço IP.

Os roteadores estão conectados a um switch SW2 e a outros três roteadores, com exceção do roteador no meio da topologia que se interconecta a quatro outros. O switch 5 está conectando todos os outros switches ao switch SW3, que também conecta todos os roteadores. O objetivo desta interconexão através do switch SW3 é permitir que os roteadores e os switches façam parte de uma mesma sub-rede com endereço de rede igual a 10.0.0.0/23.

A figura 2 mostra a organização da topologia simplificando as conexões existentes para formar a sub-rede 10.0.0.0/23. A figura não define endereços IP para as outras interfaces dos roteadores já que elas não se encontram habilitadas. Os endereços IP dessas interfaces dependem de cada experimento e aluno. Observe que cada conexão envolvendo dois roteadores corresponde a uma nova sub-rede, e essas conexões ponto a ponto são normalmente configuradas com máscara /30. A camada de enlace com a interconexão entre os switches pode ser abstraída para o barramento mostrado na figura 3, formando a sub-rede 10.0.0.0/23, que segue essa representação simplificada ao longo do experimento e roteiro.

MiniEdit

Figura 1:

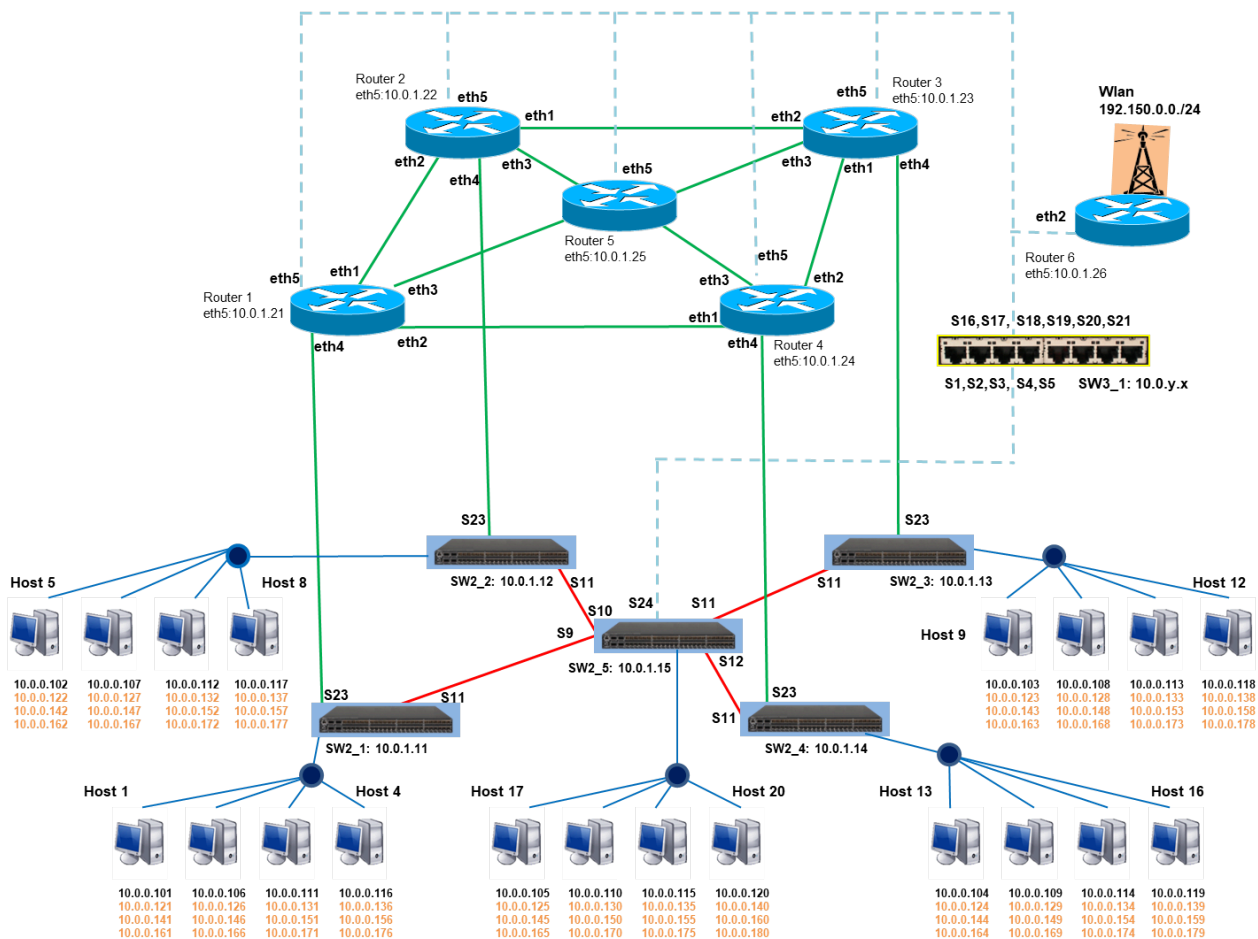
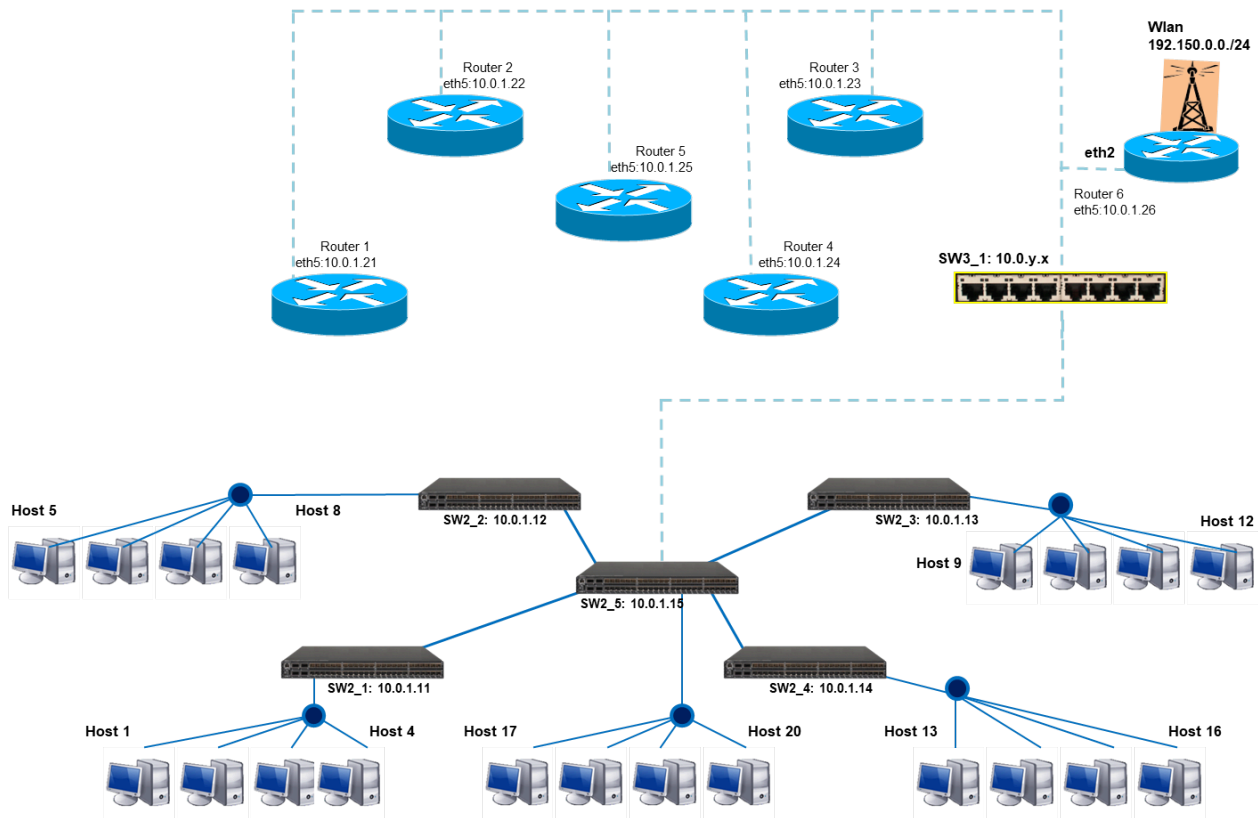


Figura 2:



A criação de topologias pode ser feita de duas maneiras:

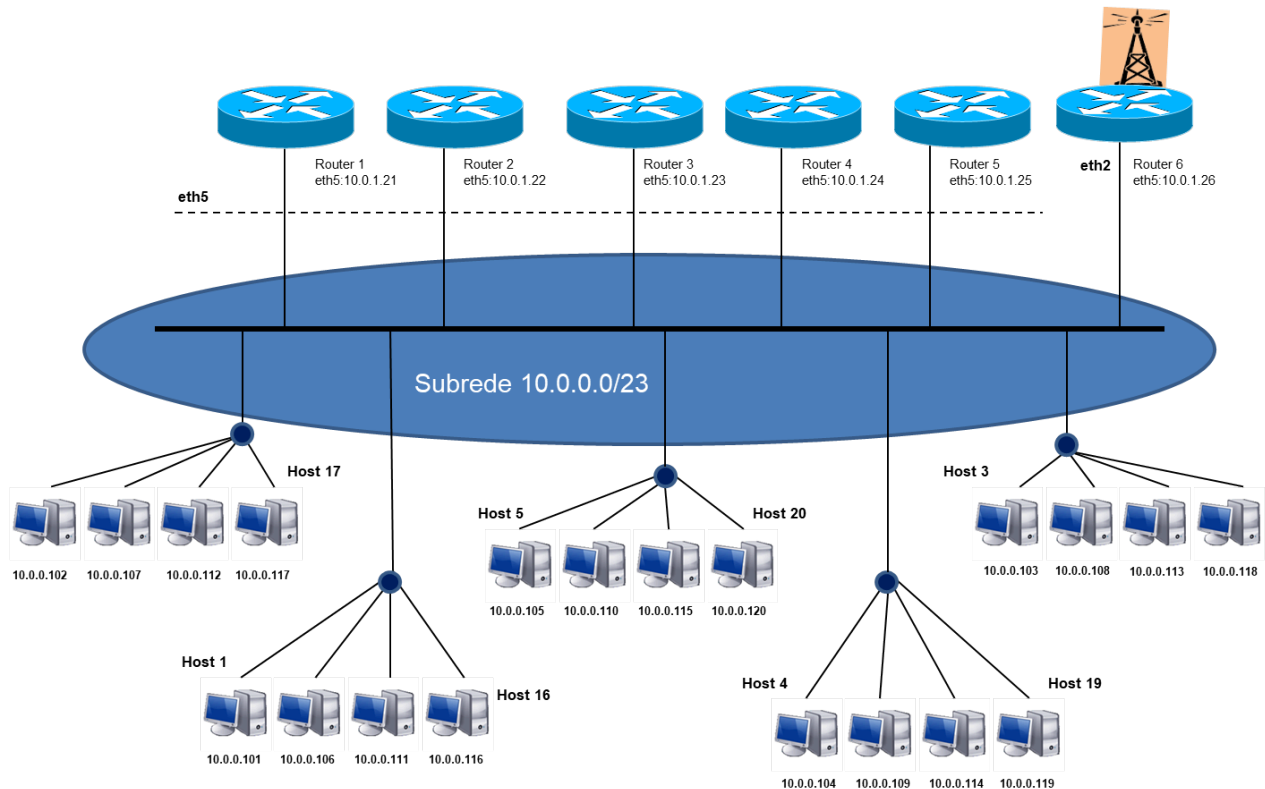
- manualmente, através dos scripts em Python já vistos nos laboratórios anteriores ou;
- através da ferramenta Miniedit (como visto no exercício opcional do Laboratório 1).

O Miniedit é um script que fornece uma interface gráfica para criar a topologia do Mininet. Recomenda-se a utilização do MiniEdit, pelo menos nessa primeira atividade, para a resposta a algumas perguntas e um contato inicial. Caso não tenha utilizado no primeiro laboratório, há um bom tutorial disponível em <http://www.brianlinkletter.com/how-to-use-miniedit-mininets-graphical-user-interface/> (<http://www.brianlinkletter.com/how-to-use-miniedit-mininets-graphical-user-interface/>). Recomenda-se a leitura antes de iniciar as atividades.

Utilizaremos os switches do tipo LegacySwitch no Miniedit, que representa um switch Ethernet padrão que não precisa ser configurado ou de controlador.

Obs.: o script miniedit.py (<http://miniedit.py>) está localizado dentro do diretório mininet-wifi/examples

Figura 3:



Preparação

Neste laboratório, cada aluno precisará de um máquina Windows ou Linux que funcionará como um host para uma máquina virtual :

- VirtualBox
- VM do Mininet-WiFi

Atenção: A VM que utilizaremos neste laboratório é a do Mininet-WiFi cuja instalação foi detalhada em laboratórios anteriores.

Login: wifi

Password: wifi

Atenção: Antes de começar verifique se o Wireshark está instalado, caso não esteja execute o seguinte comando:

```
sudo apt-get install wireshark-qt
```

Atividades práticas

Atividade 1

1. Execute o MininetEdit e desenhe a topologia do laboratório da figura 2. Não é necessário colocar todos os 4 hosts em cada switch, somente 1 é suficiente.

```
$ cd mininet-wifi/examples  
$ sudo python miniedit.py
```

Obs.: Para evitar um erro ao criar a topologia no MiniEdit, o primeiro link a ser criado tem que começar em um switch.

2. Exporte a topologia na forma de script em Python (File>Export Level 2 Script) e verifique o arquivo gerado.

Obs.: Se desejar ter endereço similar à figura utilize ipBase='10.0.0.100/23'.

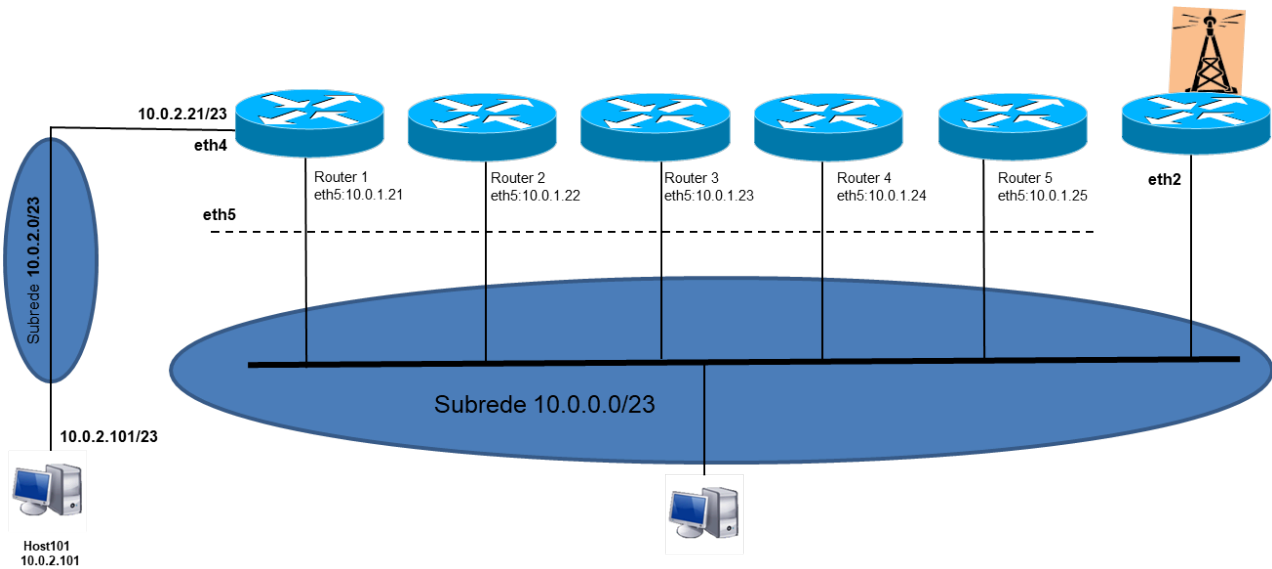
3. Execute o script e verifique a conectividade entre hosts utilizando o comando ping. Verifique também as tabelas ARP à medida em que realiza os pings.

Obs.: Se algum erro ocorrer durante a execução da topologia, tente eliminar a topologia anterior com o comando `sudo mn -c`

4. Verifique a tabela de roteamento dos hosts. Há alguma informação referente aos roteadores da topologia? Por que esse comportamento é observado?
5. Como os roteadores são representados no script? Qual comando foi necessário para definir seu comportamento como roteador e por que ele é necessário?
6. O que está faltando na configuração dos roteadores?

Atividade 2

Topologia para o exercício 2:



1. Analise a Figura e responda:

- quantos domínios de broadcast existem na topologia?
- informe a faixa de endereços de cada uma das sub-redes incluindo e identificando o endereço IP de broadcast.

2. Crie a topologia da Figura.

No host adicionado (Host101), configure o IP da sua interface na nova sub-rede 10.0.2.0/23 (subrede à esquerda na figura).

Vale lembrar que as interfaces são criadas pelos enlaces conectados em cada nó, e podemos especificar a interface em que um enlace será conectado através do script (caso não especificado, o enlace conecta na primeira interface disponível):

```
net.addLink('n1','n2',intfName1='n1-eth0',intfName2='n2-eth4')
```

3. Configure o roteador (gateway) padrão no Host101. Existem diferentes opções para definir qual é o roteador padrão de um sistema final.

- Acessando o host e executando os comandos de Linux como mostrado a seguir.

pacote net-tools:

```
route add default gw f.g.h.i
```

ou pacote iproute2:

```
ip route add default via f.g.h.i
```

- Dentro do script do Mininet, as outras opções para definir o gateway:
 - direto na adição do nó:

```
net.addHost('node1', ip='a.b.c.d/e', defaultRoute='via f.g.h.i')
```

- enviando comandos específicos para os nós

```
node.setDefaultRoute('via f.g.h.i')
```

4. Configure as interfaces do roteador R1 (Router 1) para definir o endereçamento, e após isso explique o conteúdo da tabela de roteamento.

Dentro do script, podemos definir o endereço IP de várias maneiras:

- direto na adição do nó, com máscara da subrede, sendo atribuído para a primeira interface no caso de múltiplas:

```
node = net.addHost('node1', ip='a.b.c.d/e')
```

- atribuindo um IP para uma interface:

```
node.setIP('a.b.c.d/e', intf='node1-eth0')
```

- enviando comandos específicos no linux para os nós:
pacote net-tools:

```
node.cmd("ifconfig node1-eth0 a.b.c.d netmask x.y.w.z")
```

ou pacote iproute2:

```
node.cmd("ip addr add a.b.c.d/e dev node1-eth0")
```

Comandos no Linux para visualizar o conteúdo da tabela de rotamento:
pacote net-tools:

```
route -n
```

ou pacote iproute2:

```
ip route show
```

Cabe destacar que toda interface de rede configurada com um endereço IP implica na inserção automática de uma rota referente àquela rede diretamente conectada, porque a interface passa a ser membro da rede.

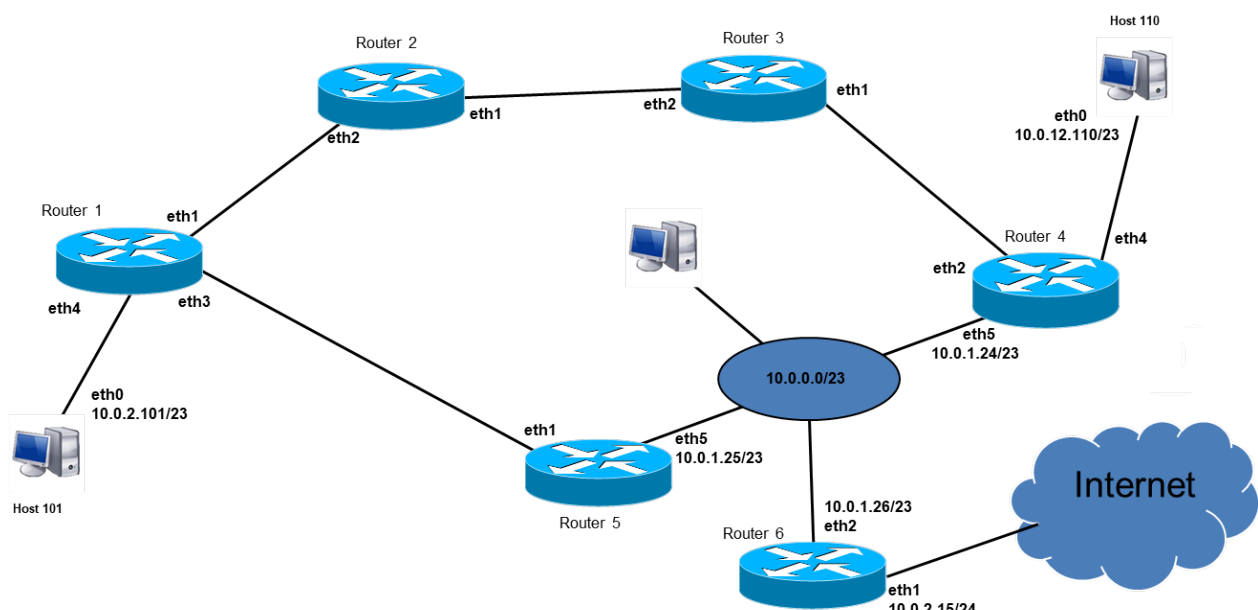
5. Realize ping e tracepath (traceroute) para testar a conexão com um host do restante da topologia. Funcionou?

Obs.: O uso do Wireshark pode ser útil para ver o que está acontecendo e resolução do problema.

6. Após a solução do problema, teste novamente a conexão e observe o valor do TTL (Time To Live). Compare com o TTL observado na atividade 1 e justifique o resultado.
7. Apresente e explique as tabelas de roteamento do roteador R1 e dos hosts utilizados. Por que usar opção -n em route? Obs.: teste sem ela

Atividade 3

Topologia para o exercício 3:



1. Instanciaremos a topologia da Figura, note o novo host conectado no roteador R4 em relação à atividade anterior.

2. Configure as rotas estáticas de modo que os hosts possam “pingar” um ao outro e a algum host da rede 10.0.0.0/23.
3. Apresente as tabelas de roteamento dos roteadores e o resultado dos testes de ping e tracepath (traceroute). Discuta o número de saltos (ou seja roteadores no caminho) conforme observado no campo TTL nos testes do ping e o número de saltos conforme a saída do tracepath (traceroute).

Atenção: As atividades a seguir são OPCIONAIS e podem somar no total até 3,0 (três) pontos adicionais na nota do relatório.

4. (Opcional) Estenda a solução do item anterior para que os hosts da rede 10.0.0.0/23 possam pingar uma interface externa à rede do laboratório (ex: 143.106.150.123 ou 8.8.8.8).

Dica: Inclusão de um nó como NAT (Network Address Translation) com ligação com a interface com acesso externo da máquina virtual. Um exemplo de nó com interface NAT encontra-se no apêndice.

5. (Opcional) Insira um host malicioso na rede 10.0.0.0/23 para tentar personificar o gateway padrão (10.0.1.26 do R6) que faria a ligação à Internet e rede pública.

Explique os passos desse tipo de ataque (arp spoofing) usando gratuitous ARP e implemente um exemplo mostrando o tráfego redirecionado para a máquina maliciosa.

Dicas:

- Utilize o comando arping para geração de gratuitous ARP (vide Apêndice).
- Pesquise sobre o assunto: <http://lmgty.com/?q=ARP+spoofing+portugues>
(<http://lmgty.com/?q=ARP+spoofing+portugues>).
- ARP Spoofing: https://en.wikipedia.org/wiki/ARP_spoofing
(https://en.wikipedia.org/wiki/ARP_spoofing).
- Exercising the "arp" and "arping" commands:
http://homepage.smc.edu/morgan_david/cs75/labs/arp-and-arping.html
(http://homepage.smc.edu/morgan_david/cs75/labs/arp-and-arping.html).

Atividade EXTRA: Network Address Translation (NAT)

Para realizar o experimento para analisar o mecanismo de tradução de endereços de rede (Network Address Translation – NAT), faça os seguintes passos e responda as seguintes perguntas:

A ideia deste laboratório é analisar dois arquivos contendo traces de pacotes. Um destes arquivos (NATCliente.pcap) contém pacotes capturados do lado do cliente em que um roteador faz NAT para um cliente se comunica com um servidor. Enquanto o outro arquivo (NATInternet.pcap) corresponde a captura de pacotes na interface de comunicação do roteador com a rede externa, contendo pacotes com a operação de NAT já realizada.

Utilizando o programa Wireshark, abra o arquivo do lado do cliente, NATCliente.pcap. Na opção de filtro do Wireshark insira a seguinte expressão “http && ip.addr==64.233.169.104” e tecle enter. Com base nas informações mostradas, responda:

- a) Qual o endereço IP do cliente?
- b) Quais os endereços fonte e destino, e as portas fonte e destino da primeira operação HTTP GET?
- c) Quais os endereços fonte e destino, e as portas fonte e destino da primeira resposta HTTP OK? Neste mesmo arquivo, no menu superior do Wireshark clique em “Analyze → Follow TCP Stream”.
- d) Quais as portas fonte e destino das operações SYN, SYN/ACK e ACK mostradas?

Feche o programa Wireshark e abra o segundo arquivo. NATInternet.pcap. Como anteriormente, aplique o filtro “http && ip.addr==64.233.169.104” a esta captura.

e) Quais os endereços fonte e destino, e as portas fonte e destino da primeira operação HTTP GET?

f) Quais os endereços fonte e destino, e as portas fonte e destino da primeira resposta HTTP OK? Neste mesmo arquivo, no menu superior do Wireshark clique em “Analyze → Follow TCP Stream”.

g) Quais as portas fonte e destino das operações SYN, SYN/ACK e ACK mostradas?

h) Estas informações diferem em quais campos das respostas dadas na análise do arquivo NATCliente.pcap?

i) Com base nas informações respondidas, qual a tabela NAT do roteador intermediário às duas capturas de pacotes?

Apêndice

Adicionando rotas

Na atividade 2 vimos o comando para adicionar uma rota padrão, entretanto é útil também adicionar rotas específicas para outras subredes. Para isso podemos usar os comandos abaixo:

pacote net-tools:

```
route add -net ${address}/${mask} gw ${next hop}
```

ou pacote iproute2:

```
ip route add ${address}/${mask} via ${next hop}
```

LinuxRouter

Na criação de uma topologia neste laboratório diretamente através do script em python pode ser útil a utilização de uma classe para definir o roteador. O mininet tem um exemplo localizado em mininet/examples/linuxrouter.py, e abaixo a classe e um exemplo de utilização está reproduzida:

```

class LinuxRouter( Node ):
    "A Node with IP forwarding enabled."

    def config( self, **params ):
        super( LinuxRouter, self ).config( **params )
        # Enable forwarding on the router
        self.cmd( 'sysctl net.ipv4.ip_forward=1' )

    def terminate( self ):
        self.cmd( 'sysctl net.ipv4.ip_forward=0' )
        super( LinuxRouter, self ).terminate()

# Na criação de um nó para ser roteador
defaultIP = '192.168.1.1/24' # IP address for r0-eth0
router = self.addNode( 'r0', cls=LinuxRouter, ip=defaultIP )

```

NAT

Para conseguir comunicação com redes externas ao mininet (a partir da rede privada emulada), pode-se configurar um NAT com ligação com uma interface de rede da VM que tenha o acesso desejado.

No script, basta utilizar um nó com classe NAT e passar os parâmetros desejados. Assim cria-se um nó com comportamento NAT que faz a ligação entre a rede pública (internet) e um nó da rede privada do mininet (e.g. um roteador).

```

inetIntf = 'eth0' # interface da VM com acesso externo
localIP = '10.1.2.2/24' # IP da interface local do nó NAT
localSubnet = '10.0.0.0/23' # rede local para regras de NAT
nat = net.addNode('nat1', cls=NAT, subnet=localSubnet,
inetIntf=inetIntf, ip=localIP, inNamespace=False)

# ligação da interface NAT com algum elemento da topologia
# e.g. um roteador
rot = net.addNode('rot1', cls=LinuxRouter, ip='a.b.c.d/e')
net.addLink(nat, rot)
# e adicionar rota dentro do nó nat para subrede desejada
nat.cmd('route add -net 10.0.0.0/23 gw f.g.h.i')

```

Gratuitous ARP

O comando arping tem uma função semelhante ao ping, a diferença entre ambos é que o ping é baseado no ICMP, tratando-se de um comando que opera na camada de redes, enquanto o arping é baseado no protocolo ARP, operando

no nível da camada de enlace. Uma das funcionalidades oferecidas pelo arping é a possibilidade de mensagens ARP não solicitadas (Gratuitous ARP), tanto Gratuitous ARP Reply, quanto Gratuitous ARP Request, permitindo a atualização das tabelas ARP além de outras situações.

Provavelmente será necessário a instalação do pacote que contenha a ferramenta arping na VM com o mininet. Existem dois pacotes disponíveis, iputils-arping e arping, com algumas diferenças de sintaxe mas que realizam essencialmente a mesma tarefa. Fica a sugestão da utilização do último citado.