

Lab 1 : Introdução ao ambiente de emulação de redes Mininet

Esta atividade prática introduz o emulador de redes Mininet. Nós criaremos topologias de rede e definiremos parâmetros como largura de banda, latência e taxa de perda de pacotes. Os objetivos principais deste laboratório incluem:

- Compreender o funcionamento do emulador de redes Mininet.
- Exercitar a criação de topologias de rede utilizando o emulador

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

Note que similar ao lab 0, caso você execute o laboratório nos computadores do LE-25, a vm já está instalada e configurada.

user: wifi

password: wifi

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

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

Atenção: Durante as atividades práticas erros comuns ou pequenos bugs podem acontecer. A Seção Problemas Frequentes apresenta alguns destes erros e o passo a passo para sua solução. Além disso, caso você encontre algum problema não reportado, pode reporta-lo como comentário no próprio Hackmd.

Atividades Práticas

Atividade 1: Aquecendo

Em um terminal da máquina virtual do Mininet (ou dentro da conexão SSH), digite:

```
sudo mn --mac
```

Este comando dará início ao Mininet e irá configurar uma pequena rede com dois hosts e um switch. Use também a opção `-topo` do comando `mn` e descubra mais topologias disponíveis.

No terminal do Mininet (`mininet-wifi>`), execute o comando `nodes` para observar a rede criada. Então execute `intfs` para verificar todas as interfaces da rede, e `pingall` para testar a conectividade entre os hosts.

```
nodes
intfs
pingall
```

Digite `xterm h1` e `xterm h2` na CLI do Mininet para abrir o terminal (console) de ambos os hosts h1 e h2. Agora você pode executar comandos como se estivesse dentro da máquina Linux destes hosts.

```
xterm h1
xterm h2
```

Siga estes passos e observe:

Acesse o Wireshark digitando `wireshark` no terminal de h2. Use-o para monitorar a interface deste host.

```
h2> sudo wireshark
```

Teste o ping entre h1 e h2, usando no terminal de h1:

```
h1> ping -c1 10.0.0.2
```

Para cada host (h1 e h2) e switch (s1):

- verifique o endereço IP de cada interface
- verifique o endereço MAC de cada interface
- verifique a tabela de roteamento IP
- verifique a tabela ARP

Dica: utilize as ferramentas do linux `ifconfig`, `arp`, `route`, `ip route`, `netstat` etc...

Você pode terminar a seção no Mininet digitando `quit`, `exit` ou `Ctrl+D`.

Atividade 2: Criando topologias de rede

Primeiro temos que baixar e descompactar os scripts de python com as topologias, digitando (no diretório home):

```
git clone https://github.com/intrig-unicamp/EA080-2S2021.git
cd EA080-2S2021/lab1
```

Agora, execute o script python, digitando (no diretório home):

```
sudo python pratica-1-II.py
```

O mininet irá iniciar a topologia desejada

Execute o comando pingall (obs: falha de testes, por exemplo sem resposta de ping, são representados com X)

Faça as modificações necessárias no arquivo pratica-1-II.py (<http://pratica-1-II.py>), para que todos os hosts tenham conectividade com todos os outros.

Explique as modificações necessárias e exiba os resultados obtidos.

Atividade 3: Definindo largura de banda, latência e taxa de perda de pacotes nos enlaces da rede

Execute o script python pratica-1-III.py (<http://pratica-1-III.py>), dentro do diretório lab1. Além disso, analise o código fonte do script e observe como os parâmetros dos enlaces são especificados (largura de banda em Mbps, latência em ms e taxa de perda em %).

Verifique o tempo de resposta e taxa de perda de pacotes entre h1 e h2, e h1 e h4 utilizando o comando ping:

```
h1> ping h2
h1> ping h4
```

Para medir o desempenho da rede em termos de largura de banda, use o comando iperf (vide documentação em: <https://iperf.fr/> (<https://iperf.fr/>)).

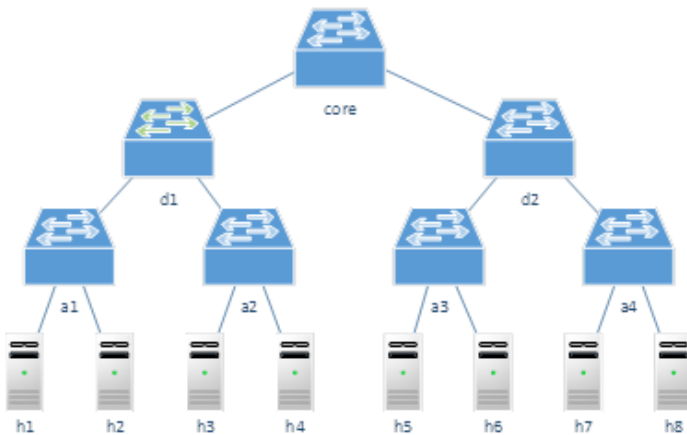
```
iperf h1 h2
iperf h1 h4
```

Discuta os resultados.

Atividade 4: Experimentando uma topologia mais complexa

a) Editando o arquivo da atividade anterior (ou em um novo arquivo), cria uma rede (estilo data center) ilustrada na figura abaixo com os seguintes parâmetros de enlace: (Obs.: acrescente um número ao nome do switch 'core')

- Core para Distribuição (d1, d2): 10 Gbps, 1 ms
- Distribuição para Acesso (a1, a2, a3, a4): 1 Gbps, 3 ms
- Acesso para Hosts: 100 Mbps, 5 ms



b) Faça modificações no arquivo de topologia para inserir uma taxa de perda de 15% no enlace entre h8 e o switch a4.

c) Como você mediria a perda de pacotes para h8 usando ping? Qual a taxa de perda de pacotes que você observa na prática?

d) Meça a largura de banda (usando iperf) entre:

- (i) h1 e h2
- (ii) h1 e h3
- (iii) h1 e h5

Atividade 5: Impacto da variação de parâmetros de enlace

a) Edite o arquivo da atividade anterior com os seguintes parâmetros de enlace:

- Core para Distribuição (d1, d2): 1 Gbps, 2 ms
- Distribuição para Acesso (a1, a2, a3, a4): 100 Mbps, 2 ms
- Acesso para Hosts: 10 Mbps, 2 ms

b) Execute medidas de atraso e vazão usando ping e iperf desde diferentes hosts para avaliar o desempenho da rede. Explique e discuta os resultados observados (comparados com a atividade anterior) com respeito ao efeito das novas características dos enlaces.

c) Execute uma captura de pacotes (salve como nome_sobrenome.pcap) para ilustrar a conectividade da rede e performance da topologia. Use informações disponíveis através do analisador de pacotes do Wireshark, e explique as métricas de desempenho observadas no item b).

d) (Opcional) Execute o script miniedit (ferramenta GUI para Mininet localizada em /mininet-wifi/examples/mininedit.py) para criar uma topologia de rede. Faça algumas capturas de tela e adicione ao relatório, comentando quais aspectos da topologia e parâmetros de enlace podem ser vistos e definidos no miniedit.

FAQ

1. \$ sudo mn -mac não é executado

A principal causa para este problema é o fato de um controlador da rede estar rodando na porta tcp:6553 quando a VM é iniciada.

Para solucioná-lo basta matar o processo rodando na porta indicada utilizando o comando a seguir.

```
$ sudo fuser -k tcp:<port>
```

Outra causa para este problema é o serviço de suporte ao Open vSwitch (switch padrão do Mininet) não ter sido iniciado corretamente.

Para solucionar esse problema basta iniciar ou reiniciar o serviço de suporte ao ovs.

```
$ systemctl stop openvswitch-switch.service  
$ systemctl start openvswitch-switch.service
```

Caso o erro persista, tente instalar novamente o Open vSwitch e, então, reinicie o serviço.

```
$ sudo ~/mininet-wifi/util/install.sh -v  
$ systemctl stop openvswitch-switch.service  
$ systemctl start openvswitch-switch.service
```

2. Wireshark não está instalado

As VMs deste laboratório não vêm instaladas com o programa wireshark. Para instalá-lo basta executar o seguinte comando.

```
$ sudo apt-get install wireshark
```