Exercício Computacional (não é necessário entregar)

Este Exercício Computacional (EC) tem como objetivo que você se familiarize com o Kathará. Apresentamos três exercícios. O primeiro é uma rede entre dois hosts criada a partir de comandos do Kathará. No segundo, é um exercício de roteamento estático, são criados arquivos para o exercício, de maneira a salvarmos a topologia e configuração inicial da rede. Por fim, é apresentado um exercício de roteamento para que você exercite seus conhecimentos.

Configuração do Kathará:

No terminal digite

```
aluno@maquina:~$ kathara setting
```

Modifique a opção 10 para *Yes*

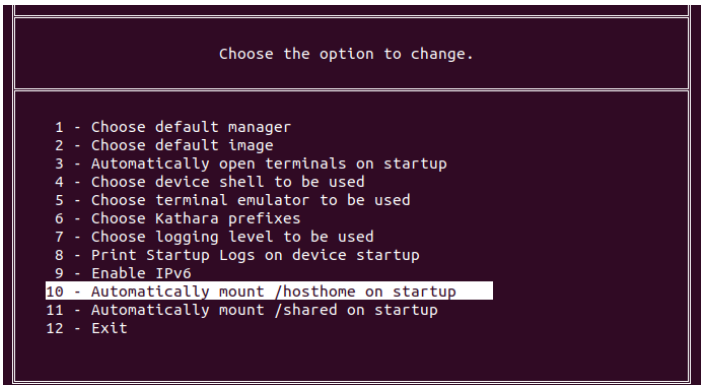


Figura 1: configuração do Kathará

Você também pode modificar o terminal emulado na opção 5. Se quiser usar o `gnome-terminal` do ubuntu altere para `/usr/bin/gnome-terminal`

Exercício 01:

Criaremos uma rede virtual bastante simples com dois nós.

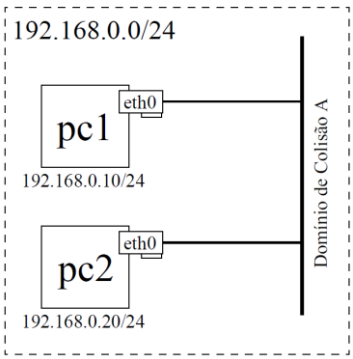


Figura 2: exemplo 1 -- uma rede simples com dois hosts conectados no mesmo domínio de colisão

Para criá-la executaremos os seguintes passos:

a) [REAL] Inicie as máquinas virtuais pc1 e pc2. No terminal digite:

```
aluno@maquina:~$ kathara vstart -n pc1 --eth 0:A

===================== Starting Device =====================

Deploying collision domains...
|##############################################################################################
##############################################################################################
##############################################| 1/1
```

```
Deploying devices...
|##################################################################################################
##################################################################################################
#########################################################| 1/1

aluno@maquina:~$ kathara vstart -n pc2 --eth 0:A

====================== Starting Device ======================

Deploying collision domains...
|##################################################################################################
##################################################################################################
#############################################| 1/1

Deploying devices...
|##################################################################################################
##################################################################################################
#########################################################| 1/1
```

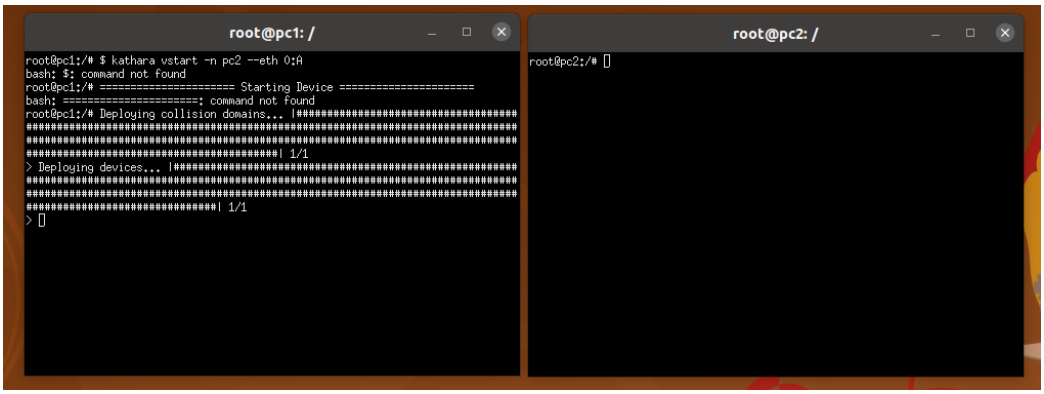


Figura 3: Duas máquinas (pc1 e pc2) instanciadas

b) [VIRTUAL] Configure as interfaces de rede das máquinas virtuais:
No pc1 digite:

```
pc1:~# ifconfig eth0 192.168.0.10 netmask 255.255.255.0
```

No pc2 digite:

```
pc2:~# ifconfig eth0 192.168.0.20 netmask 255.255.255.0
```

c) [VIRTUAL] Verifique com *ping* se os hosts estão interconectados.
No pc2 digite:

```
pc2:~# ping 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=64 time=0.230 ms
64 bytes from 192.168.0.10: icmp_seq=2 ttl=64 time=0.201 ms
64 bytes from 192.168.0.10: icmp_seq=3 ttl=64 time=0.205 ms
64 bytes from 192.168.0.10: icmp_seq=4 ttl=64 time=0.193 ms
64 bytes from 192.168.0.10: icmp_seq=5 ttl=64 time=0.180 ms
^C
--- 192.168.0.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.180/0.201/0.230/0.024 ms
       pc2:~#
```

d) [VIRTUAL] Capture alguns pacotes com *tcpdump* (sniffer).
No pc1 digite (use ctrl+C para interromper a captura):

```
pc1:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
02:16:58.586074 IP 192.168.0.20 > 192.168.0.10: ICMP echo request, id 770, seq 52, length 64
02:16:58.589262 IP 192.168.0.10 > 192.168.0.20: ICMP echo reply, id 770, seq 52, length 64
02:16:59.585897 IP 192.168.0.20 > 192.168.0.10: ICMP echo request, id 770, seq 53, length 64
02:16:59.585926 IP 192.168.0.10 > 192.168.0.20: ICMP echo reply, id 770, seq 53, length 64
02:17:00.575893 arp who-has 192.168.0.20 tell 192.168.0.10
02:17:00.576421 arp reply 192.168.0.20 is-at ee:b1:74:37:fa:bd (oui Unknown)
02:17:00.586252 IP 192.168.0.20 > 192.168.0.10: ICMP echo request, id 770, seq 54, length 64
02:17:00.586272 IP 192.168.0.10 > 192.168.0.20: ICMP echo reply, id 770, seq 54, length 64
```

```
02:17:01.595877 IP 192.168.0.20 > 192.168.0.10: ICMP echo request, id 770, seq 55, length 64
02:17:01.595906 IP 192.168.0.10 > 192.168.0.20: ICMP echo reply, id 770, seq 55, length 64
02:17:02.595871 IP 192.168.0.20 > 192.168.0.10: ICMP echo request, id 770, seq 56, length 64
02:17:02.595896 IP 192.168.0.10 > 192.168.0.20: ICMP echo reply, id 770, seq 56, length 64
02:17:03.595896 IP 192.168.0.20 > 192.168.0.10: ICMP echo request, id 770, seq 57, length 64
02:17:03.595924 IP 192.168.0.10 > 192.168.0.20: ICMP echo reply, id 770, seq 57, length 64
^C
14 packets captured
14 packets received by filter
0 packets dropped by kernel
       pc1:~#
```

e) [VIRTUAL] Verifique os pacotes capturados com o *Wireshark*.

No pc1 digite:

```
pc1:~# tcpdump -i eth0 -w /hosthome/pacotes-pc1.pcap
```

Obs.: isso irá salvar os pacotes capturados na sua máquina (real) em /home/aluno/pacotes-pc1.pcap

f) [REAL] Abra o arquivo com o *Wireshark*:

```
aluno@maquina:~$ cd  ~
aluno@maquina:~$ wireshark pacotes-pc1.pcap
```

Exercício 02:

Configuração de rotas estáticas

Neste exercício iremos emular uma rede cuja topologia é ilustrada na Figura 4. Detalhes, como endereços e máscara de cada rede, endereço IP das interfaces em cada nó, bem como os domínios de colisões, são ilustradas na Figura 5.

Para a configuração do Lab serão necessários criar os seguintes arquivos:

- `lab.conf`: descreve a topologia da rede.
- `pc1.startup`: contém comandos de shell script executados na inicialização do pc1.
- `pc2.startup`: contém comandos de shell script executados na inicialização do pc2.
- `r1.startup`: contém comandos de shell script executados na inicialização do r1.
- `r2.startup`: contém comandos de shell script executados na inicialização do r2.

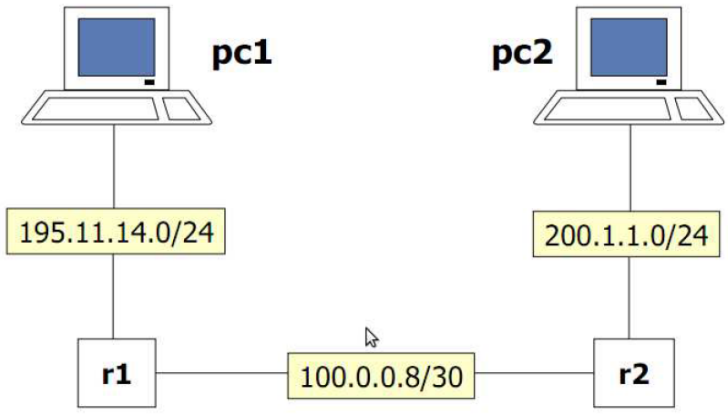


Figura 4: Topologia da Rede

a) [REAL] Crie uma pasta chamada lab01:

```
aluno@maquina:~$ mkdir lab01
aluno@maquina:~$ cd lab01
aluno@maquina:~/lab01$
```

b) [REAL] Dentro da pasta lab01 crie e o edite o arquivo *lab.conf* e insira as linhas abaixo:

```
r1[0]="A"
r1[1]="B"
r2[0]="C"
r2[1]="B"
```

```
pc1[0]="A"
pc2[0]="C"
```

A topologia é definida configurado a(s) interface(s) de cada dispositivo ao seu respectivo domínio. O índice [0] corresponde a interface 0 (eth0), índice [1] corresponde a interface 1 (eth1), e assim por diante.

c) [REAL] Será necessário configurar as interfaces de rede (com endereço IP, máscara de subrede e endereço de broadcast) quando os dispositivos iniciarem a execução. Para isso, edite os arquivos *.startup* da seguinte forma:
[REAL] Crie uma pasta chamada para cada nó:

```
aluno@maquina:~$ mkdir pc1 pc2 r1 r2
```

[REAL] Crie os respectivos arquivos *.startup*
Em pc1.startup insira:

```
ifconfig eth0 195.11.14.5 netmask 255.255.255.0 broadcast 195.11.14.255 up
```

Em pc2.startup insira:

```
ifconfig eth0 200.1.1.7 netmask 255.255.255.0 broadcast 200.1.1.255 up
```

Em r1.startup insira:

```
ifconfig eth0 195.11.14.1 netmask 255.255.255.0 broadcast 195.11.14.255 up
ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
```

Em r2.startup insira:

```
ifconfig eth0 200.1.1.1 netmask 255.255.255.0 broadcast 200.1.1.255 up
ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
```

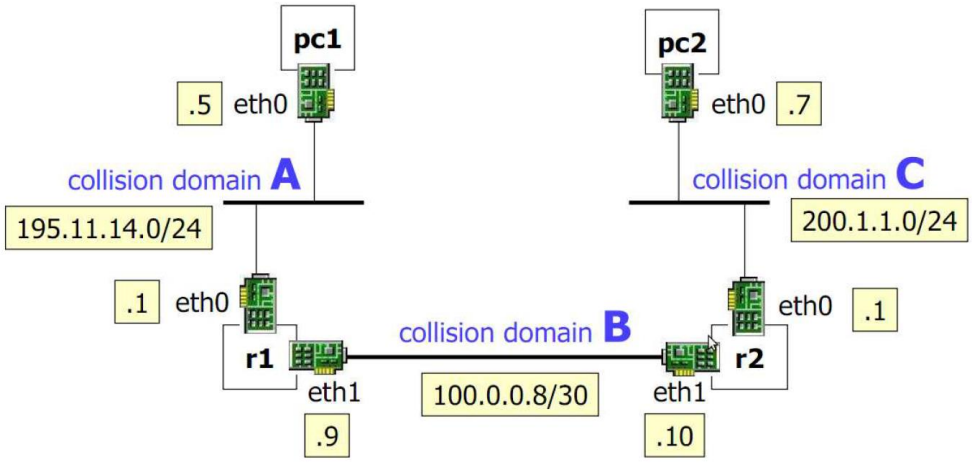


Figura 5: Topologia da Rede em detalhes

[REAL] Agora, dentro da pasta lab01, inicie a execução do lab com o comando:

```
aluno@maquina:~/lab01$ kathara lstart
```

Deverá aparecer 4 terminais, um para cada nó (pc1, pc2, r1 e r2).

e) [VIRTUAL] Será necessário configurar gateways as rotas entre as máquinas. Para tanto, utilize o comando `route` para e definir os gateways em pc1 e pc2, e adicionar as rotas de acesso nos roteadores r1 e r2.
No terminal do pc1 insira:

```
pc1:~# route add default gw 195.11.14.1 dev eth0
```

No terminal do pc2 insira:

```
pc2:~# route add default gw 200.1.1.1 dev eth
```

No terminal do r1 insira:

```
r1:~# route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
```

No terminal do r2 insira:

```
r2:~# route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
```

f) [VIRTUAL] A tabela de repasse (ou roteamento) nos nós pode ser verificada com o comando `route`.

Por exemplo, no terminal do r1, digite:

```
r1:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
100.0.0.8 * 255.255.255.252 U 0 0 0 eth1
200.1.1.0 100.0.0.10 255.255.255.0 UG 0 0 0 eth1
195.11.14.0 * 255.255.255.0 U 0 0 0 eth0
r1:~#
```

g) [VIRTUAL] Verifique a conectividade entre pc1 e pc2. No terminal do pc1 use o comando *ping* para verificar se pc1 atinge pc2, ou vice-versa.

h) [REAL] Finalizar a execução do lab, no terminal digite `kathara lclaen`

Como visto, a **rota padrão** (ou seja, a definição do Gateway) de uma máquina é configurada estaticamente com o comando:

```
route add default gw ip gateway dev iface
```

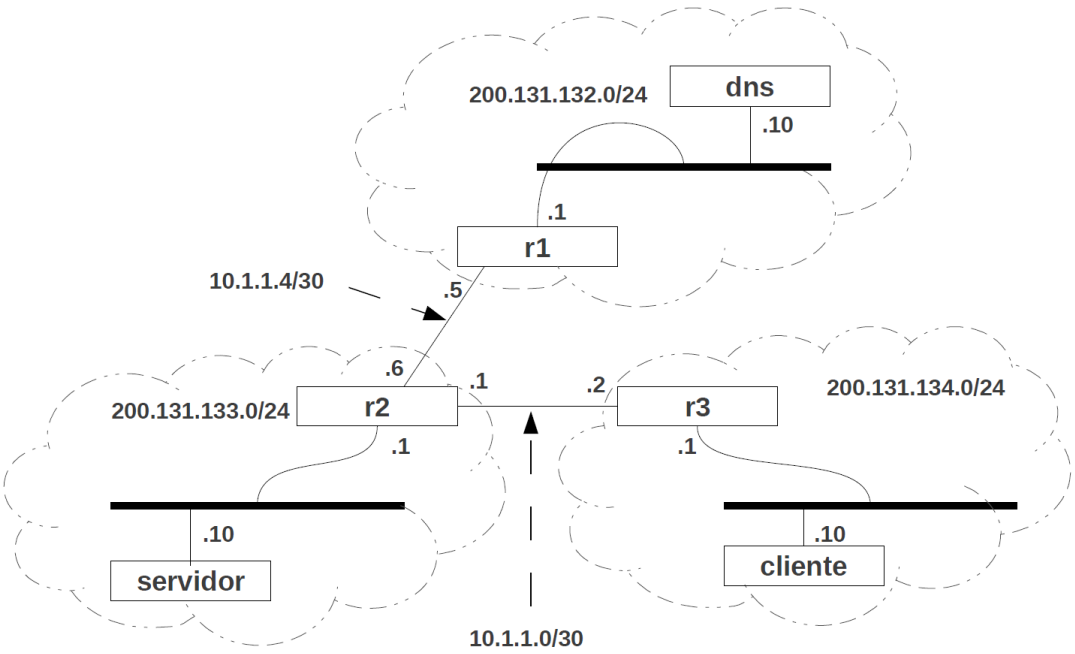
A outra forma equivalente de se configurar a rota padrão é definindo rotas estáticas para a rede de destino:

```
route add -net ip dst net netmask dst mask gw ip gateway dev iface
```

Se não conseguiu executar este exercício faça download do arquivo lab01 no moodle e compare os arquivos da pasta com os que você criou.

Exercício 03:

Implemente a topologia abaixo no Kathará. Com os comandos acima, faça com que seja possível as máquinas cliente, servidor e dns terem rotas de acesso, de modo que uma consiga atingir (*pingar*) a outra. Para manter a persistência, você pode inserir os comandos no arquivo de inicialização *.startup*



Algumas distribuições Linux deixaram de vir com a ferramenta *ifconfig*. Para maior padronização, as distribuições estão dando suporte ao comando *ip*. Exemplos de uso do comando você pode encontrar no site:

http://www.bosontreinamentos.com.br/linux/10-exemplos-de-configuracao-de-rede-com-comandos-ip-no-linux/