

# UC Redes de Computadores

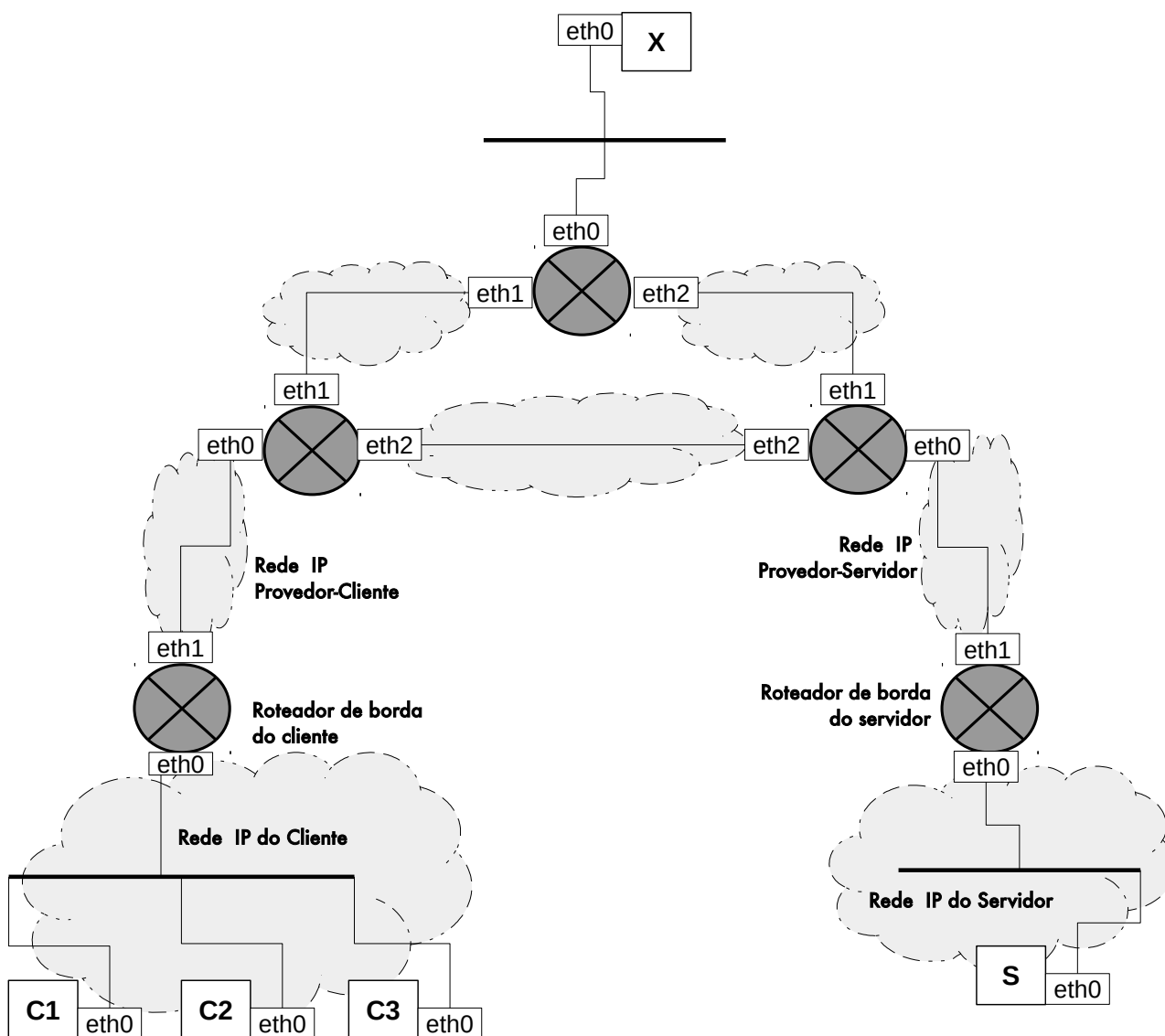
Prof. Bruno Kimura  
13/09/2018

## Trabalho 1 (Gargalos)

- O trabalho poderá ser realizado em dupla ou individualmente.
- Submeta no SEAD apenas um arquivo compactado contendo a resolução do trabalho. Verifique as observações em cada questão.
- A submissão no SEAD estará aberta até 20/09/18.

### 1) Implementação da Rede

Implemente a topologia abaixo no Netkit-NG. Verifique as instruções no ANEXO I para instalar o emulador.



Nesta questão é necessário:

- (a) Configurar a topologia física (domínios de colisão - enlaces) entre os nós.
- (b) Configurar a rede lógica (definição de endereçamento IP das redes e das interfaces dos nós). Note que há 4 redes IP na topologia acima. Verifique referência abaixo.
- (c) Definir roteamento da rede de forma estática, configurando as rotas manualmente nos roteadores. Verifique comandos no ANEXO II.

**Observações:**

- Crie uma pasta e defina os arquivos do Netkit-NG. A resolução deste trabalho deverá estar nos arquivos `lab.conf` e `*.startup`. Para entregar este trabalho, envie a pasta com todos arquivos EXCETO arquivos `.disk`.

**Referência:**

<http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html>

## 2) Configuração da Rede

Nesta topologia, há dois provedores que fornecem serviços de acesso para as redes dos clientes e do servidor. Os clientes fizeram o seguinte contrato com o provedor: *Upstream* de 5 Mbps e *Downstream* de 10 Mbps. Já o contrato do servidor especifica: *Upstream* de 50 Mbps e *Downstream* de 50 Mbps.

Configure as capacidades dos enlaces dos provedores para atender os contratos estabelecidos entre as partes. Para tanto, utilize a ferramenta TC (*traffic control*) que está instalada nas VMs do Netkit-NG, a qual permite limitar o tráfego de uma interface de rede com uma vazão máxima. Portanto, TC permite realizar *traffic shaping*. Verifique os comandos no ANEXO II.

**Observação:**

- A resolução deste exercício deverá estar em um arquivo texto, especificando exatamente os comandos dados em cada um dos roteadores envolvidos.

**Referências:**

<http://man7.org/linux/man-pages/man8/tc.8.html>

<https://wiki.debian.org/TrafficControl>

## 3) Verificação do Desempenho sob os Enlaces de Gargalo

Considere que os acessos à “Internet” foram contratados nos respectivos provedores para atender a rede do servidor S e a rede dos clientes C. É importante verificar se os provedores estão fornecendo os serviços conforme contratados. Para isso, utilize a ferramenta IPERF para medir a vazão fim-a-fim com uma máquina na rede externa. Verifique os comandos no ANEXO II.

- (a) Ative o servidor IPERF na máquina X. Em uma das máquinas cliente e no servidor S, verifique a vazão fim-a-fim obtida com X nas duas direções e explique os resultados.

- (b) Agora ative o servidor IPERF na máquina S. Na máquina C1, conecte o cliente IPERF ao servidor S. No cliente C2, utilize a ferramenta `ping` com destino para a máquina X. Verifique a latência de C2 a X e explique os resultados a partir de duas situações: (i) com carga na rede, ou seja, durante a operação do IPERF entre C1 e S; (ii) sem carga na rede, desativando a execução do IPERF.
- (c) Ative o IPERF no servidor S e em todos os clientes, C1, C2 e C3 ao mesmo tempo. Verifique a vazão obtida nos clientes e explique os resultados

**Observação:**

- A resolução deste exercício deverá estar em um arquivo texto com a explicação de como e por qual razão foram obtidos os resultados observados. Nas respostas argumente pontos de vista teórico (Seção 1.4 do livro de J. Kurose) e prático (o impacto da operação dos protocolos da pilha TCP/IP nos resultados obtidos).

**Referências:**

<http://linux.die.net/man/1/iperf>  
<http://linux.die.net/man/8/traceroute>

## ANEXO I: Instalação do Netkit-NG

1) Acesse <https://netkit-ng.github.io/> e baixe os arquivos:

- Netkit-NG core 3.0.4: netkit-ng-core-32-3.0.4.tar.bz2
- Sistema de arquivos 7.0: netkit-ng-filesystem-i386-F7.0-0.1.3.tar.bz2
- Kernel 3.2: netkit-ng-kernel-i386-K3.2-0.1.3.tar.bz2

2) Descompacte os arquivos:

Abra o terminal, em /home/usuario/, e dê os comandos:

```
$ tar -xjSf netkit-ng-core-32-3.0.4.tar.bz2
$ tar -xjSf netkit-ng-filesystem-i386-F7.0-0.1.3.tar.bz2
$ tar -xjSf netkit-ng-kernel-i386-K3.2-0.1.3.tar.bz2
```

3) Configure as variáveis de ambiente:

Abra o arquivo /home/usuario/.profile com um editor de texto (nano, pico, vim, gedit, ...) e na ultima linha insira as configurações das variáveis de ambiente do NetKit:

```
export NETKIT_HOME=/home/usuario/netkit
export MANPATH=$NETKIT_HOME/man
export PATH=$NETKIT_HOME/bin:$PATH
```

4) Exporte as variáveis de ambiente:

Abra o terminal e, na sua /home/usuario/ dê o comando:

```
$ . .profile
```

5) Verifique as configurações do Netkit:

Abra o terminal e, em /home/usuario/netkit/, execute o script:

```
$ cd $NETKIT_HOME
$ ./check_configuration.sh
```

Caso sejam verificados que outros pacotes precisam ser instalados, dê o comando no terminal:

```
$ sudo apt-get install <nome_pacote>
```

## ANEXO II: Comandos de interesse

### 1) Configuração de IP da interface de rede:

```
$ ifconfig <interface> <ip_dispositivo>/< mascara>
```

### 2) Definição de gateway (roteador) padrão

```
$ route add default gw <ip_roteador>
```

### 3) Definição de rotas na tabela de roteamento dos roteadores

```
$ route add -net <ip_rede>/< mascara> gw <ip_roteador> dev <interface>
```

### 4) Alteração do MTU de uma interface de rede

```
$ ifconfig <interface> mtu <valor>
```

### 5) Captura de pacotes:

```
$ tcpdump -i <interface> -w /hostlab/<arquivo.pcap>
```

Parâmetros:

-i: interface de rede

-w: salva pacotes capturados em arquivo

Observação: para interromper a captura, dê Ctrl+C no terminal.

### 6) Controle do tráfego:

Adição de controle:

```
$tc qdisc add dev <interface> root tbf rate <taxa> latency <tempo> burst <bytes>
```

Parâmetros importantes:

qdisc: *queueing discipline*, refere-se aos algoritmos de escalonamento de rede ou algoritmos de enfileiramento.

tbf: *token bucket filter*, é um dos algoritmos bem conhecidos para limitar a transmissão de uma interface de rede. Sua implementação consiste em um balde (*bucket*) que é constantemente preenchido com informações simbólicas, chamadas *tokens*, a uma taxa específica (*token rate*). Para transmitir, o pacote enfileirado precisa ser associado a um *token* que chega ao balde, e então o *token* é removido e o pacote colocado no enlace. Dessa forma, os pacotes são transmitidos conforme a taxa do *token*.

Parâmetros a serem preenchidos:

<taxa>: especifica a taxa máxima de transmissão em bits/s. Exemplo: "4mbit"

<interface>: interface de rede. Exemplo: "eth0"

<tempo>: especifica o tempo máximo que um pacote espera para que um *token* fique disponível. Exemplo: "50ms".

<bytes>: tamanho do balde em bytes. Exemplo: "15000".

Remoção de controle:

```
$tc qdisc del dev <interface> root
```

### 7) Teste de desempenho da rede:

No servidor:

```
$ iperf -s
```

No cliente:

```
$ iperf -c <ip_servidor> -r
```

Parâmetros importantes:

-c: define operação do IPERF como cliente

-r: realiza medição de desempenho nos dois sentidos (*upstream* e *downstream*)