

Universidade Federal de São Paulo

Instituto de Ciência e Tecnologia

Redes de Computadores

Trabalho Final de Redes de Computadores

Emulação de Redes com o Netkit

Thiago Hideki Sato (RA: 92341)

Tiago Alexander Leal Ruzzon (RA: 92346)

Prof^a Dr. Valerio Rosset

Dezembro

2017

Sumário:

1	Topologia	1
2	Implementação no Netkit	2
3	Configurações de Upstream e Downstream	3
4	Resultados	4
4.1	Letra A	4
4.2	Letra B	4
4.3	Letra C	6
5	Referências	8

1 Topologia

A Figura 1 apresenta a topologia proposta pelo orientador do trabalho, a fim de ser implementada no Netkit-NG.

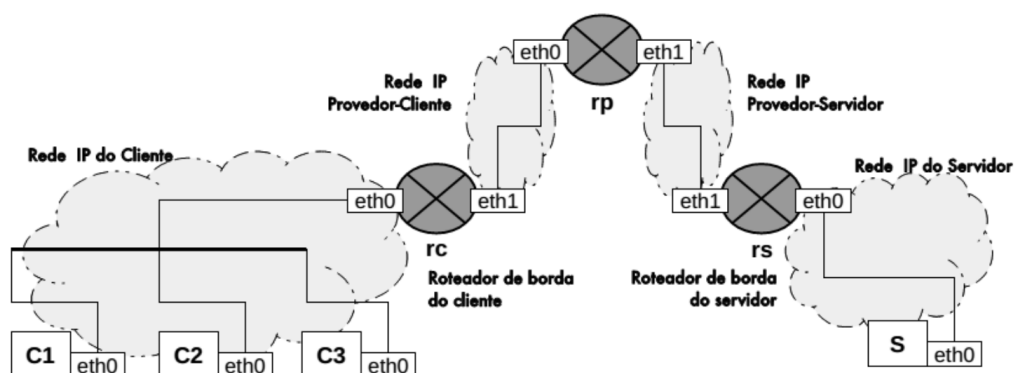


Figura 1: Topologia Proposta

Para implementar esta topologia foi necessário definir endereços IPs para cada enlace, estes endereços são apresentados na tabela 1. Eles foram determinados de maneira arbitrária.

Tabela 1: Definição de Interfaces, IPs e Domínios de Colisão

Dispositivo	Interface	Endereço de IP	Domínio de Colisão
C1	eth0	195.1.1.1	A
C2	eth0	195.1.1.2	A
C3	eth0	195.1.1.3	A
rc	eth0	195.1.1.4	A
rc	eth1	100.1.1.1	B
rp	eth0	100.1.1.2	B
rp	eth1	120.1.1.1	C
rs	eth1	120.1.1.2	C
rs	eth0	210.1.1.2	D
s	eth0	210.1.1.1	D

Em sequência, para melhor entendimento da rede, a figura 2 apresenta um esquemático completo desenvolvido com os endereços IPs em cada enlace.

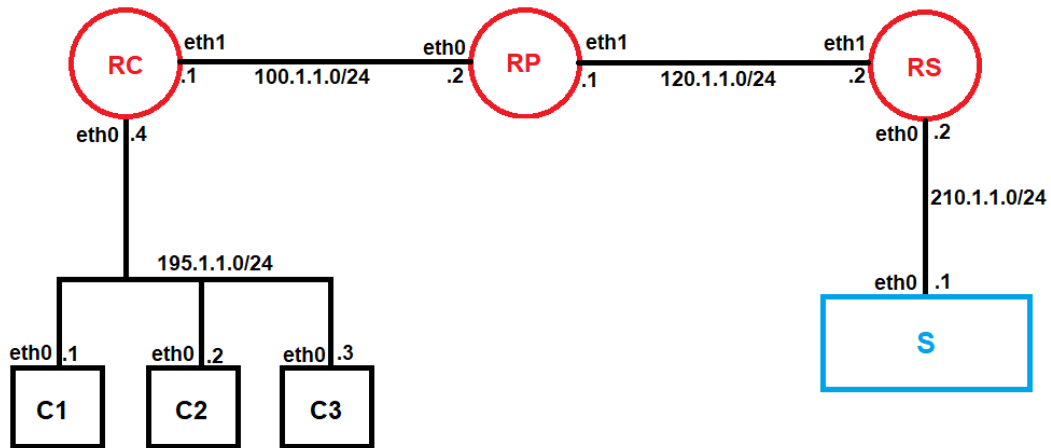


Figura 2: Esquemático Desenvolvido

2 Implementação no Netkit

As figuras 3, 4 e 5 apresentam os arquivos utilizados para emular a topologia da rede definida.

```

1  c1[eth0]="A"
2  c2[eth0]="A"
3  c3[eth0]="A"
4  rc[eth0]="A"
5  rc[eth1]="B"
6  rp[eth0]="B"
7  rp[eth1]="C"
8  rs[eth1]="C"
9  rs[eth0]="D"
10 s[eth0]="D"

```

Figura 3: Arquivo lab.conf

Esse arquivo é responsável para listagem dos nós e suas portas, assim como a definição de em qual domínio de colisão essas portas estão conectadas.

```

1  ifconfig eth0 195.1.1.1 up
2  route add default gw 195.1.1.4
3
4  ifconfig eth0 195.1.1.2 up
5  route add default gw 195.1.1.4
6
7  ifconfig eth0 195.1.1.3 up
8  route add default gw 195.1.1.4
9
10
11 ifconfig eth0 210.1.1.1 up
12 route add default gw 210.1.1.2

```

c1.startup

c2.startup

c3.startup

s.startup

Figura 4: Arquivos .startup de C1, C2, C3 e S

Os arquivos *startup* dos computadores e do servidor definem o IP de cada um na subrede na qual estão inseridos, além da rota padrão (endereço IP de seu roteador).

```

ifconfig eth0 195.1.1.4 up
ifconfig eth1 100.1.1.1 up
route add -net 120.1.1.0 netmask 255.255.255.0 gw 100.1.1.2 dev eth1
route add -net 210.1.1.0 netmask 255.255.255.0 gw 100.1.1.2 dev eth1

```

rc.startup

```

ifconfig eth0 100.1.1.2 up
ifconfig eth1 120.1.1.1 up
route add -net 195.1.1.0 netmask 255.255.255.0 gw 100.1.1.1 dev eth0
route add -net 210.1.1.0 netmask 255.255.255.0 gw 120.1.1.2 dev eth1

```

rp.startup

```

ifconfig eth0 210.1.1.2 up
ifconfig eth1 120.1.1.2 up
route add -net 100.1.1.0 netmask 255.255.255.0 gw 120.1.1.1 dev eth1
route add -net 195.1.1.0 netmask 255.255.255.0 gw 120.1.1.1 dev eth1

```

rs.startup

Figura 5: Arquivos .startup de RC, RP e RS

Os arquivos *startup* dos roteadores inicialmente definem o IP de cada uma de suas portas (assim como os nós acima). Entretanto, é necessária uma configuração adicional de rota, que define um caminho estático o qual os pacotes devem seguir caso se deseje acessar uma das redes que não estão diretamente relacionadas aos roteadores.

3 Configurações de Upstream e Downstream

A figura 6 apresenta a configuração dos roteadores para limitar o tráfego de uma interface de rede. Para isto, foi utilizado a ferramenta *Traffic Control* (TC) do Netkit NG.

Foi proposta a definição da velocidade de *downstream* de 5 Mbps (*megabits* por segundo) e *upstream* de 1 Mbps para os computadores ligados em RC. Enquanto isso, para o servidor, as velocidades tanto de *downstream* quanto de *upstream* são de 50 Mbps.

```
tc qdisc add dev eth1 root tbf rate 1mbit latency 50ms burst 15000      rc

tc qdisc add dev eth0 root tbf rate 5mbit latency 50ms burst 15000      rp
tc qdisc add dev eth1 root tbf rate 50mbit latency 50ms burst 15000

tc qdisc add dev eth1 root tbf rate 50mbit latency 50ms burst 15000      rs
```

Figura 6: Configuração de Traffic Control

Essa configuração foi adicionada aos arquivos *startup* dos roteadores referenciados nas imagens. Os valores de tempo e de tamanho do balde utilizados foram os exemplos apresentados no comando do trabalho.

4 Resultados

4.1 Letra A

Ao ativar o IPERF no servidor S e no cliente C1, a vazão fim-a-fim obtida é apresentada na Figura 2.

Tabela 2: Velocidades de Upload e Download

Quantidade Total de Conexões	Dispositivo	Taxa	Velocidades
1	C1	Upload: Download:	984 Kbits/sec 4,73 Mbits/sec

A vazão fim-a-fim obtida foi um pouco abaixo do definido, apesar de bastante próximas. Isto ocorre pois há certas variações e oscilações na rede, que não é algo exato e perfeito.

4.2 Letra B

Para a execução da ferramenta *traceroute* no cliente C2, foram consideradas duas situações: com carga na rede (enquanto ocorria a execução do IPERF em C1) e sem carga na rede (desativando a execução do IPERF em

C1). O comando utilizado foi *traceroute 210.1.1.1*. A figura 7 apresenta tais resultados.

```
root@c2:~# traceroute 210.1.1.1
traceroute to 210.1.1.1 (210.1.1.1), 30 hops max, 60 byte packets
 1  195.1.1.4 (195.1.1.4)  0.342 ms  0.368 ms  0.174 ms
 2  100.1.1.2 (100.1.1.2)  0.294 ms  0.202 ms  0.152 ms
 3  120.1.1.2 (120.1.1.2)  0.245 ms  0.231 ms  0.205 ms
 4  210.1.1.1 (210.1.1.1)  0.288 ms  0.301 ms  0.273 ms

root@c2:~# traceroute 210.1.1.1
traceroute to 210.1.1.1 (210.1.1.1), 30 hops max, 60 byte packets
 1  195.1.1.4 (195.1.1.4)  0.173 ms  0.103 ms  0.072 ms
 2  100.1.1.2 (100.1.1.2)  152.481 ms  152.320 ms  165.216 ms
 3  120.1.1.2 (120.1.1.2)  165.048 ms  164.910 ms  164.753 ms
 4  210.1.1.1 (210.1.1.1)  164.688 ms  164.538 ms  164.505 ms
```

Figura 7: Latência com e sem carga na rede.

Cada *hop* demonstra um enlace encontrado na rede até o destino. Há quatro *hops* pois, como determinado anteriormente, existem quatro enlaces entre C2 e S. A partir desses valores, o gráfico abaixo foi desenvolvido. A linha em azul demonstra os resultados com carga na rede; já a laranja não possuía carga na rede.

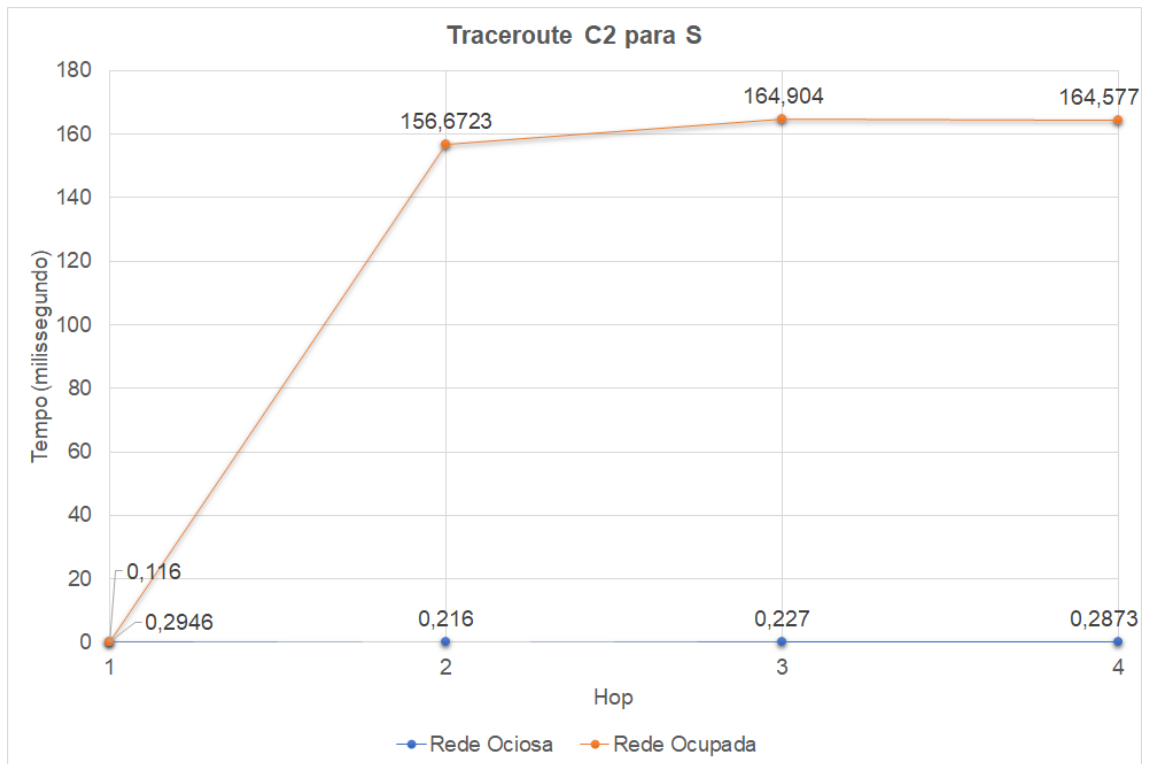


Figura 8: Gráfico de Latência.

Como o *traceroute* apresenta o resultado de tempo de três pacotes (três *round trip times* - RTTs, para se obter um resultado mais consistente), o gráfico gerado apresenta o valor médio dos tempos para cada *hop* apresentado na figura 7.

Observa-se que com carga na rede, os clientes devem concorrer para saber quem enviará os pacotes, podendo ocorrer colisões ou esperando que outro cliente termine de enviar. Por isso, o tempo aumento drasticamente. Entretanto, curiosamente no caso do primeiro enlace (entre C2 e RC), o tempo diminuiu quando a rede estava ocupada. Isso se deve ao fato de que esse enlace é local e não possui limitação de dados com a simulação de *TC* criada anteriormente.

4.3 Letra C

Foi ativado o IPERF no servidor S e em todos os clientes (C1, C2 e C3) ao mesmo tempo e verificou-se a vazão obtida pelos clientes, o resultado é apresentado na tabela 3.

Tabela 3: Velocidades de Upload e Download.

Quantidade Total de Conexões	Dispositivo	Taxa	Velocidades
3	C1	Upload: Download:	307 Kbits/sec 1,72 Mbits/sec
3	C2	Upload: Download:	318 Kbits/sec 1,33 Mbits/sec
3	C3	Upload: Download:	450 Kbits/sec 2,12 Mbits/sec

A partir da tabela com os resultados, foi possível construir os gráficos das figuras 9 e 10.

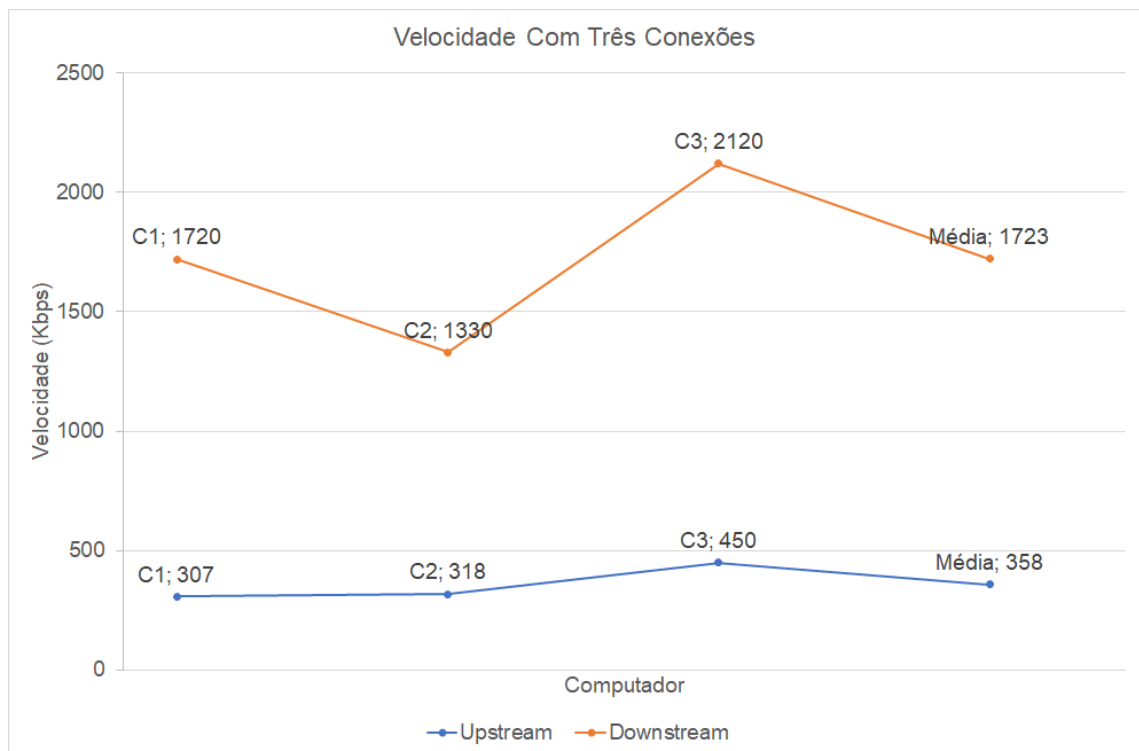


Figura 9: Gráfico de Velocidade Com Três Conexões.

Observa-se através deste gráfico que a velocidade obtida anteriormente para somente uma conexão IPERF foi praticamente dividida por três entre C1, C2 e C3. Isso se deve ao fato de que esses computadores dividem a mesma rede com o mesmo valor de TC. No teste, C3 obteve maior velocidade de que os outros nós, isso pode ter ocorrido por causa da ordem de lançamento

dos comandos IPERF no terminal ou também por oscilações que ocorrem normalmente nas redes.

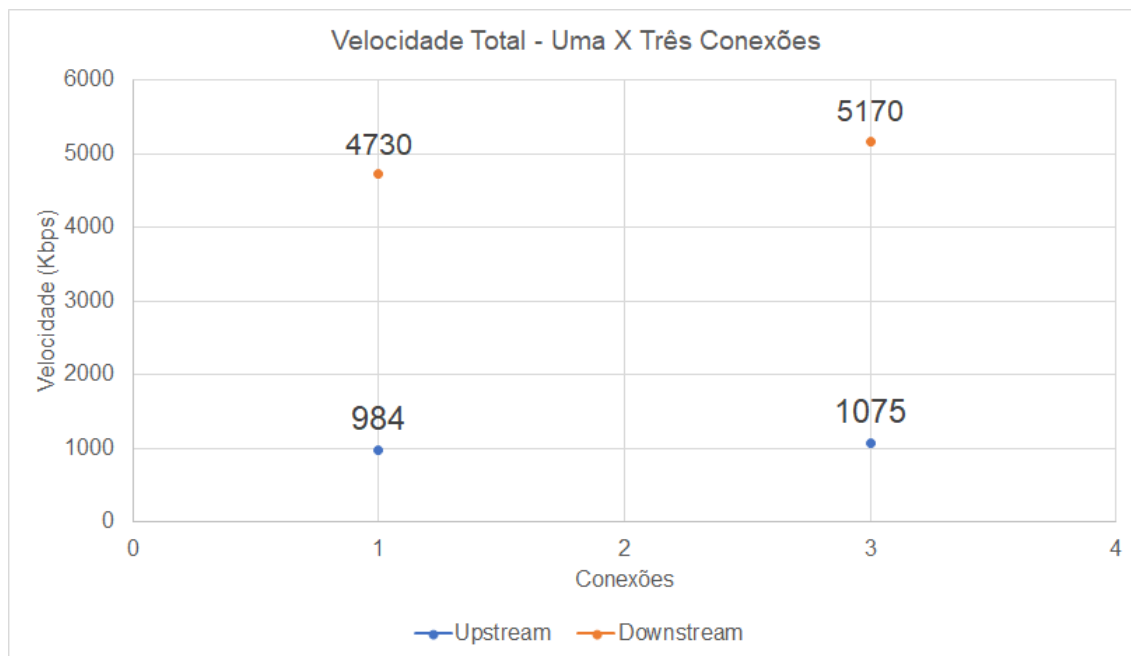


Figura 10: Gráfico de Velocidade Total das Conexões.

Ao comparar a velocidade total de somente uma conexão IPERF (letra A) com três conexões (soma das três), pode-se notar que a velocidade obtida posteriormente foi superior, entretanto, ainda segue próximo do limite de 5 Mbps / 1 Mbps de download / upload.

5 Referências

MITCHELL, Scott. *How to Read a Traceroute*. <<http://www.inmotionhosting.com/support/website/how-to/read-traceroute>>. Acessado em 12 dez. 2017.