

Pedro Spoljaric Gomes – 112344

Tamires Beatriz da Silva Lucena – 111866

Redes de Computadores – Trabalho 1 (Gargalos)

1) Implementação da Rede

As topologias física e lógica da rede foram esquematizada de acordo com a Figura 1.

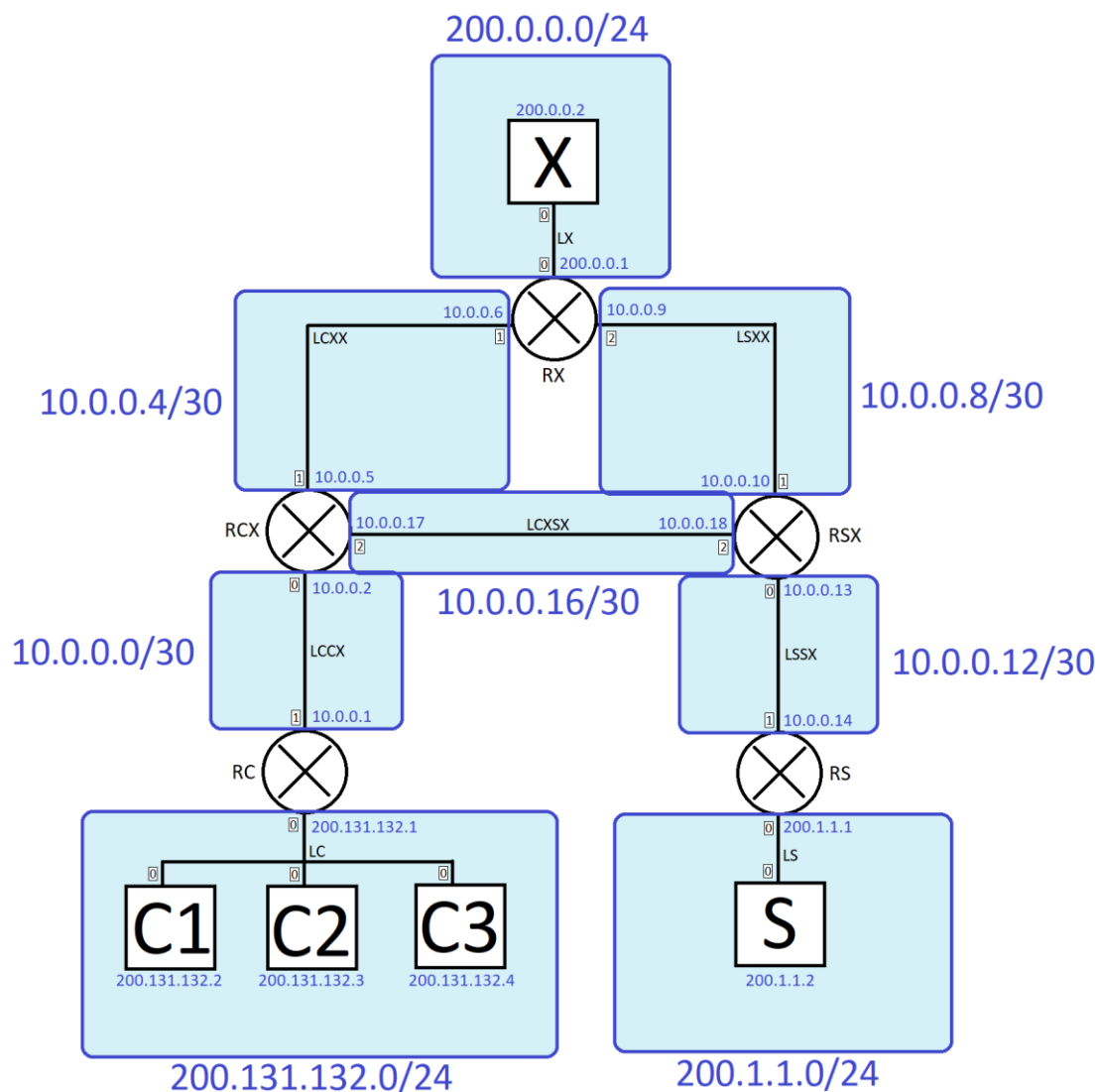


Figura 1: Topologia da rede

As configurações das topologias da rede e do roteamento estático dos roteadores foram feitas nos arquivos de inicialização do projeto e foram enviados juntamente com este relatório.

2) Configuração da Rede

As configurações das capacidades dos enlaces provedor-cliente e provedor-servidor foram definidas, respectivamente, nos arquivos de inicialização dos roteadores RCX e RSX, na interface de rede eth0 de cada um.

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.

A vazão entre os clientes (C1, C2 ou C3) e a máquina X foi definida como 5mbps na direção C -> X e 10mbps na direção X -> C. Essa configuração foi feita na interface de rede eth0 do roteador RCX e pode ser verificada no arquivo de inicialização Rcx.startup. O resultado da configuração foi analisado usando o comando iperf (servidor) na máquina X e iperf (cliente) na máquina C1, como mostra a Figura 2.

```
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.4 sec   6.00 MBytes 4.84 Mbits/sec
[ 5] local 200.131.132.2 port 5001 connected with 200.0.0.2 port 42792
[ 5] 0.0-10.2 sec  10.5 MBytes 8.65 Mbits/sec
```

Figura 2: Teste de link de gargalo entre C1 e X.

Os valores foram bem próximos aos configurados no arquivo de inicialização.

O mesmo foi feito entre servidor e máquina X, a configuração foi feita na interface eth0 do roteador Rsx e a velocidade do link em ambos os sentidos foi configurada em 50mbps, como mostra a Figura 3.

```
[ 5] local 200.1.1.2 port 35433 connected with 200.0.0.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.0-10.0 sec   54.0 MBytes 45.1 Mbits/sec
[ 4] local 200.1.1.2 port 5001 connected with 200.0.0.2 port 55139
[ 4] 0.0-10.1 sec   51.4 MBytes 42.9 Mbits/sec
```

Figura 3: Teste de link de gargalo entre S e X.

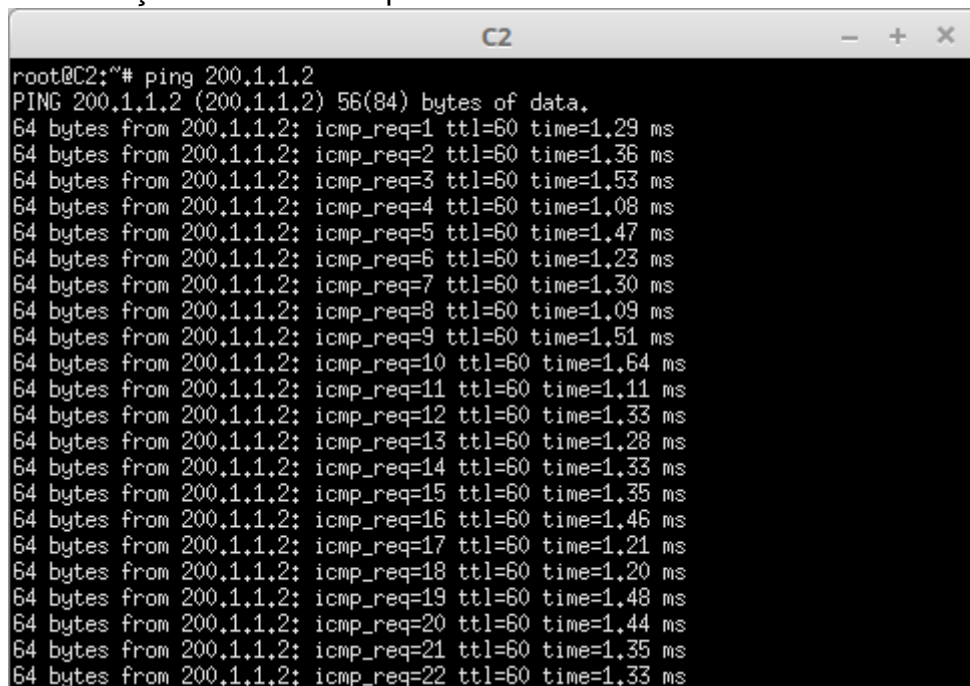
Os valores foram, novamente, bem próximos dos configurados.

Os limites de largura de banda das conexões que passam pelos roteadores Rcx e Rsx foram ajustados através da configuração da rede lógica (camada de enlace).

- (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.

- (i) A Figura 4 mostra o comportamento da latência entre C2 e X sem a execução do comando iperf.

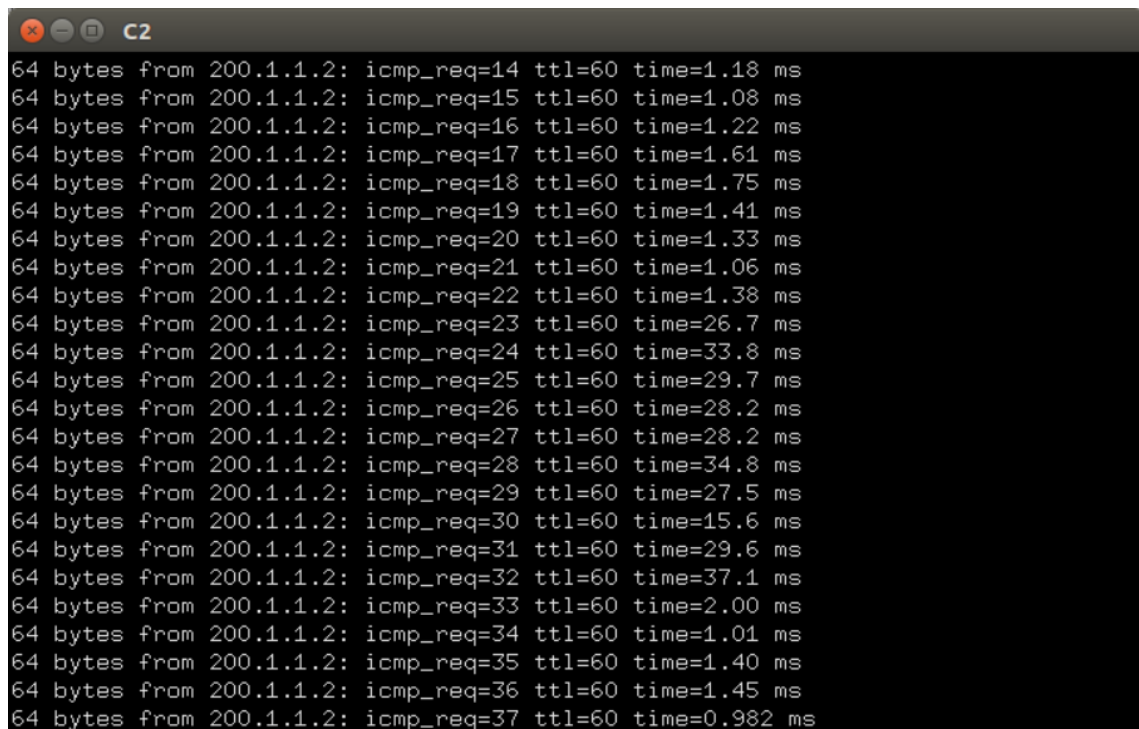


```
root@C2:~# ping 200.1.1.2
PING 200.1.1.2 (200.1.1.2) 56(84) bytes of data.
64 bytes from 200.1.1.2: icmp_req=1 ttl=60 time=1.29 ms
64 bytes from 200.1.1.2: icmp_req=2 ttl=60 time=1.36 ms
64 bytes from 200.1.1.2: icmp_req=3 ttl=60 time=1.53 ms
64 bytes from 200.1.1.2: icmp_req=4 ttl=60 time=1.08 ms
64 bytes from 200.1.1.2: icmp_req=5 ttl=60 time=1.47 ms
64 bytes from 200.1.1.2: icmp_req=6 ttl=60 time=1.23 ms
64 bytes from 200.1.1.2: icmp_req=7 ttl=60 time=1.30 ms
64 bytes from 200.1.1.2: icmp_req=8 ttl=60 time=1.09 ms
64 bytes from 200.1.1.2: icmp_req=9 ttl=60 time=1.51 ms
64 bytes from 200.1.1.2: icmp_req=10 ttl=60 time=1.64 ms
64 bytes from 200.1.1.2: icmp_req=11 ttl=60 time=1.11 ms
64 bytes from 200.1.1.2: icmp_req=12 ttl=60 time=1.33 ms
64 bytes from 200.1.1.2: icmp_req=13 ttl=60 time=1.28 ms
64 bytes from 200.1.1.2: icmp_req=14 ttl=60 time=1.33 ms
64 bytes from 200.1.1.2: icmp_req=15 ttl=60 time=1.35 ms
64 bytes from 200.1.1.2: icmp_req=16 ttl=60 time=1.46 ms
64 bytes from 200.1.1.2: icmp_req=17 ttl=60 time=1.21 ms
64 bytes from 200.1.1.2: icmp_req=18 ttl=60 time=1.20 ms
64 bytes from 200.1.1.2: icmp_req=19 ttl=60 time=1.48 ms
64 bytes from 200.1.1.2: icmp_req=20 ttl=60 time=1.44 ms
64 bytes from 200.1.1.2: icmp_req=21 ttl=60 time=1.35 ms
64 bytes from 200.1.1.2: icmp_req=22 ttl=60 time=1.33 ms
```

Figura 4: Comando ping de C2 para X sem iperf.

Foi observado um comportamento homogêneo da latência entre as máquinas, como um valor próximo de 1,3 ms.

- (ii) Em seguida, o comando ping de C2 para X foi executado ao mesmo tempo que o comando iperf (cliente) na máquina C1 era executado sobre a máquina X, que por sua vez, executava o comando iperf (servidor), e o resultado foi o mostrado na Figura 5.



```
C2
64 bytes from 200.1.1.2: icmp_req=14 ttl=60 time=1.18 ms
64 bytes from 200.1.1.2: icmp_req=15 ttl=60 time=1.08 ms
64 bytes from 200.1.1.2: icmp_req=16 ttl=60 time=1.22 ms
64 bytes from 200.1.1.2: icmp_req=17 ttl=60 time=1.61 ms
64 bytes from 200.1.1.2: icmp_req=18 ttl=60 time=1.75 ms
64 bytes from 200.1.1.2: icmp_req=19 ttl=60 time=1.41 ms
64 bytes from 200.1.1.2: icmp_req=20 ttl=60 time=1.33 ms
64 bytes from 200.1.1.2: icmp_req=21 ttl=60 time=1.06 ms
64 bytes from 200.1.1.2: icmp_req=22 ttl=60 time=1.38 ms
64 bytes from 200.1.1.2: icmp_req=23 ttl=60 time=26.7 ms
64 bytes from 200.1.1.2: icmp_req=24 ttl=60 time=33.8 ms
64 bytes from 200.1.1.2: icmp_req=25 ttl=60 time=29.7 ms
64 bytes from 200.1.1.2: icmp_req=26 ttl=60 time=28.2 ms
64 bytes from 200.1.1.2: icmp_req=27 ttl=60 time=28.2 ms
64 bytes from 200.1.1.2: icmp_req=28 ttl=60 time=34.8 ms
64 bytes from 200.1.1.2: icmp_req=29 ttl=60 time=27.5 ms
64 bytes from 200.1.1.2: icmp_req=30 ttl=60 time=15.6 ms
64 bytes from 200.1.1.2: icmp_req=31 ttl=60 time=29.6 ms
64 bytes from 200.1.1.2: icmp_req=32 ttl=60 time=37.1 ms
64 bytes from 200.1.1.2: icmp_req=33 ttl=60 time=2.00 ms
64 bytes from 200.1.1.2: icmp_req=34 ttl=60 time=1.01 ms
64 bytes from 200.1.1.2: icmp_req=35 ttl=60 time=1.40 ms
64 bytes from 200.1.1.2: icmp_req=36 ttl=60 time=1.45 ms
64 bytes from 200.1.1.2: icmp_req=37 ttl=60 time=0.982 ms
```

Figura 5: Comando ping de C2 para X com iperf ativado.

Foi observado um aumento na latência entre as máquinas durante a execução do comando iperf (cliente), o que pode ser explicado pelo fato de que o comando iperf, na tentativa de medir o limite de banda de uma conexão, tenta ocupar todo o gargalo da conexão com o envio de pacotes, o que dificulta qualquer comunicação que passe pelo mesmo caminho físico (no caso, as conexões entre C1 e X, e entre C2 e X são feitas passando pelo mesmo roteador Rcx, que é o gargalo entre os clientes e a máquina X).

Pode-se notar que a implementação do protocolo IP e de roteamento da camada de rede nos roteadores está dada de maneira correta, e portanto foi possível estabelecer a conexão entre as máquinas.

(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.

Com o comando iperf (servidor) sendo executado na máquina S, o comando iperf (cliente) foi executado em todos os clientes ao mesmo tempo, e o resultado está na Figura 6.

```

S
[ 4] local 200.1.1.2 port 5001 connected with 200.131.132.3 port 43818
[ 6] local 200.1.1.2 port 5001 connected with 200.131.132.4 port 41248
[ 5] 0.0-11.1 sec 1.88 MBytes 1.42 Mbits/sec
-----
Client connecting to 200.131.132.2, TCP port 5001
TCP window size: 40.1 KByte (default)
-----
[ 5] local 200.1.1.2 port 58538 connected with 200.131.132.2 port 5001
[ 6] 0.0-10.4 sec 2.00 MBytes 1.61 Mbits/sec
-----
Client connecting to 200.131.132.4, TCP port 5001
TCP window size: 20.0 KByte (default)
-----
[ 6] local 200.1.1.2 port 44662 connected with 200.131.132.4 port 5001
[ 4] 0.0-11.3 sec 2.88 MBytes 2.13 Mbits/sec
-----
Client connecting to 200.131.132.3, TCP port 5001
TCP window size: 20.0 KByte (default)
-----
[ 4] local 200.1.1.2 port 58452 connected with 200.131.132.3 port 5001
[ 6] 0.0-10.3 sec 5.12 MBytes 4.18 Mbits/sec
[ 5] 0.0-10.7 sec 2.50 MBytes 1.95 Mbits/sec
[ 4] 0.0-10.3 sec 3.50 MBytes 2.86 Mbits/sec
-----
C
64 bytes from 200.1.1.2: icmp_req=23 ttl=60 time=1.23 ms
--- 200.1.1.2 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22147ms
rtt min/avg/max/mdev = 0.502/2.192/13.058/3.292 ms
root@C2:~#
root@C2:~#
root@C2:~#
root@C2:~# iperf -c 200.1.1.2 -r
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
Client connecting to 200.1.1.2, TCP port 5001
TCP window size: 44.1 KByte (default)
-----
[ 3] local 200.131.132.3 port 43818 connected with 200.1.1.2 port 5001
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-11.3 sec 2.88 MBytes 2.14 Mbits/sec
[ 5] local 200.131.132.3 port 5001 connected with 200.1.1.2 port 58452
[ 5] 0.0-10.3 sec 3.50 MBytes 2.85 Mbits/sec
root@C2:~#

C1
-----
Client connecting to 200.1.1.2, TCP port 5001
TCP window size: 60.1 KByte (default)
-----
[ 5] local 200.131.132.2 port 57693 connected with 200.1.1.2 port 5001
[ ID] Interval Transfer Bandwidth
[ 5] 0.0-10.4 sec 6.00 MBytes 4.86 Mbits/sec
[ 4] local 200.131.132.2 port 5001 connected with 200.1.1.2 port 58537
[ 4] 0.0-10.3 sec 10.9 MBytes 8.89 Mbits/sec
root@C1:~# iperf -c 200.1.1.2 -r
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
Client connecting to 200.1.1.2, TCP port 5001
TCP window size: 56.1 KByte (default)
-----
[ 5] local 200.131.132.2 port 57694 connected with 200.1.1.2 port 5001
[ ID] Interval Transfer Bandwidth
[ 5] 0.0-10.4 sec 1.88 MBytes 1.51 Mbits/sec
[ 4] local 200.131.132.2 port 5001 connected with 200.1.1.2 port 58538
[ 4] 0.0-10.8 sec 2.50 MBytes 1.94 Mbits/sec
root@C1:~#

C2
64 bytes from 200.1.1.2: icmp_req=23 ttl=60 time=1.23 ms
--- 200.1.1.2 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22147ms
rtt min/avg/max/mdev = 0.502/2.192/13.058/3.292 ms
root@C2:~#
root@C2:~#
root@C2:~#
root@C2:~# iperf -c 200.1.1.2 -r
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
Client connecting to 200.1.1.2, TCP port 5001
TCP window size: 44.1 KByte (default)
-----
[ 3] local 200.131.132.3 port 43818 connected with 200.1.1.2 port 5001
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-11.3 sec 2.88 MBytes 2.14 Mbits/sec
[ 5] local 200.131.132.3 port 5001 connected with 200.1.1.2 port 58452
[ 5] 0.0-10.3 sec 3.50 MBytes 2.85 Mbits/sec
root@C2:~#

C3
root@C3:~# ping 200.1.1.2
PING 200.1.1.2 (200.1.1.2) 56(84) bytes of data:
64 bytes from 200.1.1.2: icmp_req=1 ttl=60 time=1.79 ms
64 bytes from 200.1.1.2: icmp_req=2 ttl=60 time=1.36 ms
^C
--- 200.1.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 1.365/1.582/1.798/0.216 ms
root@C3:~# iperf -c 200.1.1.2 -r
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
Client connecting to 200.1.1.2, TCP port 5001
TCP window size: 64.0 KByte (default)
-----
[ 3] local 200.131.132.4 port 41248 connected with 200.1.1.2 port 5001
[ ID] Interval Transfer Bandwidth
[ 3] 0.0-10.4 sec 2.00 MBytes 1.62 Mbits/sec
[ 5] local 200.131.132.4 port 5001 connected with 200.1.1.2 port 44662
[ 5] 0.0-10.3 sec 5.12 MBytes 4.17 Mbits/sec
root@C3:~#
```

Figura 6: Comportamento do comando iperf sendo executado nos três clientes simultaneamente.

Foi observado que a largura de banda (limitada pelo gargalo da conexão entre as máquinas presente no roteador Rcx de 5 mbps na direção C -> S, e 10 mbps na direção S -> C) foi dividida pelo protocolo TCP de forma aproximadamente igual entre os clientes que disputavam a ocupação do gargalo (as somas de velocidade de downstream e upstream são próximas ao total limitado).

Caso a transmissão fosse via UDP, um primeiro cliente ocuparia quase toda a capacidade do enlace de gargalo e realizaria a conexão, enquanto os demais teriam que dividir o pouco que sobrou.