

UNIVERSIDADE FEDERAL DE SÃO PAULO
Departamento de Ciência e Tecnologia
Bacharelado em Ciência da Computação

Trabalho 1 (Gargalos)

Anderson Silvério Mendrot Filho 92183

Douglas Diniz Landim 76681

1 Implementação da Rede

1.1 Configuração da topologia física entre nós e da rede lógica

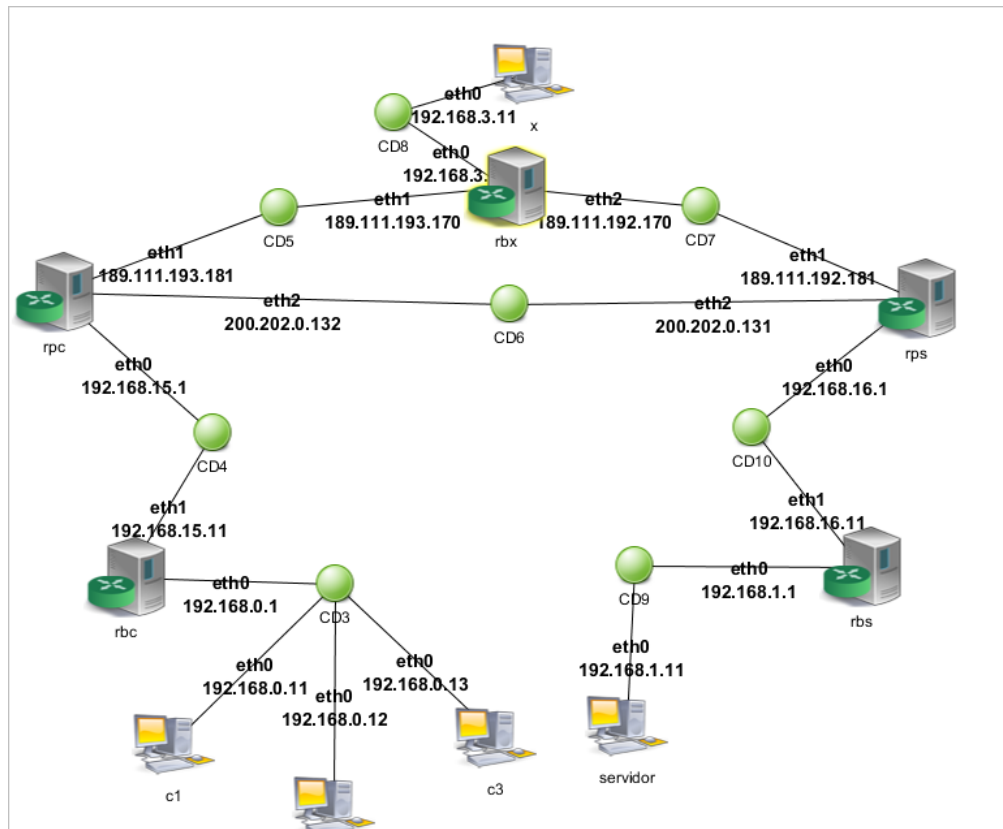


Figure 1: Topologia utilizada no projeto

1.2 Tabelas de roteamento estático implementadas nos roteadores

1.2.1 Roteadores de borda das redes internas de cliente:

Os roteadores de borda de rede de clientes, que possuem apenas 2 interfaces, sendo nesta topologia o rbc e rbs, por padrão encaminham todas as solicitações ao gateway do provedor, como no exemplo abaixo das configurações do roteador da rede de cliente rbc e rbs.

```
route add default gw 192.168.15.1
```

1.2.2 Roteadores do provedor

Os roteadores de provedor já possuem rotas de roteamento mais complexos por conectarem a rede do cliente com no mínimo 2 rotas, para isso precisam possuir uma rota de saída padrão e uma rota de saídas específicas. Neste caso a interface padrão de cliente sempre será a eth0, a interface padrão de saídas específicas será a eth1 e a saída padrão será a eth2, conforme no exemplo abaixo do rpc:

```
route add -net 192.168.0.0/24 gw 192.168.15.11 dev eth0
route add -net 192.168.3.0/24 gw 189.111.193.170 dev eth1
route add -net 189.111.192.170 gw 189.111.193.170 dev eth1
route add default gw 200.202.0.131 dev eth2
```

2 Configurações de 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 Netikit-NG, a qual permite limitar o tráfego de uma interface de rede com uma vazão máxima. Portanto, TC permite realizar traffic shaping.

2.1 Configurações do startup dos roteadores dos provedores do cliente

As configurações de limitação de links nesta topologia ficam no roteador do provedor de modo que o cliente não tenha acesso, e somente o provedor possa gerenciá-las de acordo com cada plano contratado. Isso permite uma troca e upgrade de equipamentos de borda na rede interna do cliente sem impacto nas configurações de seu plano contratado.

Abaixo está um exemplo das configurações do RPC (roteador provedor – cliente), limitando pacotes enviados para eth0 que tem destino a rede do cliente no caso de download, e limitando pacotes enviados para rotas específicas e padrões no caso de upload, na eth1/2.

```
# limitacao para down
tc qdisc add dev eth0 root tbf rate 10mbit latency 50 burst 15000
# limitacao para up
tc qdisc add dev eth1 root tbf rate 5mbit latency 50 burst 15000
tc qdisc add dev eth2 root tbf rate 5mbit latency 50 burst 15000
```

2.2 Configurações do startup dos roteadores dos provedores do servidor

As configurações seguem a mesma lógica para o roteador do provedor do servidor, com a diferença que o upstream e downstream têm a mesma velocidade, logo as configurações são iguais para todas as interfaces.

```
# limitacao para down
tc qdisc add dev eth0 root tbf rate 50mbit latency 50 burst 15000
# limitacao para up
tc qdisc add dev eth1 root tbf rate 50mbit latency 50 burst 15000
tc qdisc add dev eth2 root tbf rate 50mbit latency 50 burst 15000
```

3 Verificação do Desempenho sob os Enlaces de Gargalo

- (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.

Como pode-se ver, a figura 2 apresenta os valores obtidos pelo uso do comando `iperf -s` no X, que possui ip local 192.168.1.11, e o Servidor, que no caso se comporta como cliente, com o comando `iperf -c 192.168.3.11 -r`. Assim, são retornadas informações a respeito do *upload* e *download* do Servidor com X. A informação da largura da banda (*bandwidth*) da rede mostra que a taxa de recepção é de 40.2 Mb/s, ou seja, um arquivo de 402mb levaria 10 segundos para ser transmitido nessa taxa, já calculada no atraso fim a fim. //Situação semelhante ocorre com a taxa de download. Com a largura de banda estipulada no exemplo de 40.9 Mb/s, um arquivo de 409mb poderia ser transmitido em 10s.

```
root@x:~#
root@x:~# iperf -s
Server listening on TCP port 5001
TCP window size: 65,3 KByte (default)
[ 4] local 192.168.3.11 port 5001 connected with 192.168.1.11 port 54642
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.0 sec  48.1 MBytes  40.2 Mbits/sec
Client connecting to 192.168.1.11, TCP port 5001
TCP window size: 40.1 KByte (default)
[ 4] local 192.168.3.11 port 52400 connected with 192.168.1.11 port 5001
[ 4] 0.0-10.0 sec  49.0 MBytes  40.9 Mbits/sec
root@x:~#

[ 5] 0.0-10.0 sec  50.2 MBytes  42.0 Mbits/sec
root@servidor:~# iperf -c 192.168.3.11 -r
Server listening on TCP port 5001
TCP window size: 65,3 KByte (default)
Client connecting to 192.168.3.11, TCP port 5001
TCP window size: 64.0 KByte (default)
[ 5] local 192.168.1.11 port 54642 connected with 192.168.3.11 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.0-10.0 sec  48.1 MBytes  40.2 Mbits/sec
[ 4] local 192.168.1.11 port 5001 connected with 192.168.3.11 port 52400
[ 4] 0.0-10.1 sec  49.0 MBytes  40.9 Mbits/sec
root@servidor:~#
```

Figure 2: Console de X e Servidor, com comandos e resultados

Da mesma forma como ocorre com X e Servidor, a comunicação entre X e máquinas cliente, exibida na figura 3, também possui limitação de largura de banda, ou uma taxa de transmissão do enlace de gargalo especificada. No caso No caso, a largura de banda no teste está com valor 4.77 Mb/s, e após 10s a quantidade de dados enviadas por C1 é de 47.7 Mb, ou uma taxa de transferência

de 5,9625 MB. Porém, por valores de atraso na rede, a transferência real foi de 5.88 MB. Em relação ao *downstream*, de forma correlata, com uma largura de banda de 9.41 Mb/s foram enviados teoricamente 94.1Mb depois de 10s, o que resultaria em 11.7625, mas o valor real transmitido foi de 11.4MB,

```

root@x1:~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 65.3 KByte (default)
-----
[ 4] local 192.168.3.11 port 5001 connected with 192.168.0.11 port 47399
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.3 sec  5.88 MBytes  4.77 Mbits/sec
-----
Client connecting to 192.168.0.11, TCP port 5001
TCP window size: 40.1 KByte (default)
-----
[ 4] local 192.168.3.11 port 52471 connected with 192.168.0.11 port 5001
[ 4] 0.0-10.1 sec  11.4 MBytes  9.41 Mbits/sec

root@c1:~# iperf -c 192.168.3.11 -r
-----
[ 5] local 192.168.0.11 port 5001 connected with 192.168.3.11 port 52469
[ 5] 0.0-10.3 sec  9.88 MBytes  8.03 Mbits/sec
root@c1:~# iperf -c 192.168.3.11 -r
-----
Server listening on TCP port 5001
TCP window size: 65.3 KByte (default)
-----
Client connecting to 192.168.3.11, TCP port 5001
TCP window size: 40.1 KByte (default)
-----
[ 5] local 192.168.0.11 port 47399 connected with 192.168.3.11 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.0-10.3 sec  5.88 MBytes  4.78 Mbits/sec
[ 4] local 192.168.0.11 port 5001 connected with 192.168.3.11 port 52471
[ 4] 0.0-10.2 sec  11.4 MBytes  9.39 Mbits/sec
root@c1:~#

```

Figure 3: Console de X e C1

- (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.

Na imagem, os dados acima da linha vermelha representam o ping antes do comando iperf, e abaixo da linha vermelha, o ping após o comando iperf, apesar do pico de 1.80ms não houveram latencias significativas. Isso significa que mesmo a máquina c2 enviado pacotes à X, e c1 enviando pacotes ao servidor, compartilhando o mesmo processamento nodal e enlace nos roteadores rbc, rpc, os pacotes de ping não colidem ou incrementam atrasos aos pacotes de c1 ao servidor, pois de maneira síncrona os pacotes de ping que saem de C2 e passam pelo processamento nodal, fila, transmissão e propagação nos roteadores rbc e rpc até X fazem etapas sincronizadas e diferentes dos pacotes c1-servidor, e no pior caso onde um pacote de c1 precise ficar em fila para aguardar os pacotes de c2 serem propagados, os pacotes de c1 não serão perdidos pois, c2-x ocupam tempo de transmissão fim a fim, menor do que o tempo permitido de espera dos pacotes de c1-servidor que podem esperar até 50ms, de acordo com o comando tc.

```

X
Server listening on TCP port 5001
root@x:~#

c2
64 bytes from 192.168.3.11: icmp_req=66 ttl=61 time=0.763 ms
64 bytes from 192.168.3.11: icmp_req=67 ttl=61 time=0.433 ms
64 bytes from 192.168.3.11: icmp_req=68 ttl=61 time=1.71 ms
64 bytes from 192.168.3.11: icmp_req=69 ttl=61 time=1.43 ms
64 bytes from 192.168.3.11: icmp_req=70 ttl=61 time=1.76 ms
64 bytes from 192.168.3.11: icmp_req=71 ttl=61 time=1.05 ms
64 bytes from 192.168.3.11: icmp_req=72 ttl=61 time=1.66 ms
64 bytes from 192.168.3.11: icmp_req=73 ttl=61 time=1.49 ms
64 bytes from 192.168.3.11: icmp_req=74 ttl=61 time=1.49 ms
64 bytes from 192.168.3.11: icmp_req=75 ttl=61 time=0.545 ms
64 bytes from 192.168.3.11: icmp_req=76 ttl=61 time=1.72 ms
64 bytes from 192.168.3.11: icmp_req=77 ttl=61 time=1.78 ms
64 bytes from 192.168.3.11: icmp_req=78 ttl=61 time=1.51 ms
64 bytes from 192.168.3.11: icmp_req=79 ttl=61 time=1.75 ms
64 bytes from 192.168.3.11: icmp_req=80 ttl=61 time=1.64 ms
64 bytes from 192.168.3.11: icmp_req=81 ttl=61 time=1.32 ms
64 bytes from 192.168.3.11: icmp_req=82 ttl=61 time=1.71 ms
64 bytes from 192.168.3.11: icmp_req=83 ttl=61 time=1.11 ms
64 bytes from 192.168.3.11: icmp_req=84 ttl=61 time=1.60 ms
64 bytes from 192.168.3.11: icmp_req=85 ttl=61 time=1.81 ms
64 bytes from 192.168.3.11: icmp_req=86 ttl=61 time=0.515 ms
64 bytes from 192.168.3.11: icmp_req=87 ttl=61 time=0.899 ms
64 bytes from 192.168.3.11: icmp_req=88 ttl=61 time=1.51 ms
64 bytes from 192.168.3.11: icmp_req=89 ttl=61 time=1.54 ms
root@c2:~#

servidor
root@servidor:~# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 192.168.1.11 port 5001 connected with 192.168.0.11 port 57933
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.4 sec  5.38 MBytes  4.34 Mbits/sec
root@servidor:~#

c1
root@c1:~# iperf -c 192.168.1.11
Client connecting to 192.168.1.11, TCP port 5001
TCP window size: 44.1 KByte (default)
-----
[ 3] local 192.168.0.11 port 57933 connected with 192.168.1.11 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.3 sec  5.38 MBytes  4.36 Mbits/sec
root@c1:~#

```

Figure 4: Console das máquinas Servidor, C1, C2, X

- (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

No caso dessa situação, a quantidade de tempo de 10s de tráfego de pacotes pela rede para o teste aumentou, justamente porque o número de nós que enviam e recebem pacotes ser maior do que a situação em que um nó individual realiza essas ações.

No caso dessa situação, os clientes c1, c2 e c3 competem para enviar pacotes para uma fila dos roteadores, o que significa que a quantidade de dados transmitidos e recebidos pelos três clientes é menor do que se cada um estivesse operando individualmente. Assim, a taxa de download para o c1 é de 2.96 Mb/s, para o c2, de 2.93 Mb/s e para o c3, de 3.74 Mb/s, o que totaliza 9.43 Mb/s, um valor um pouco abaixo do valor de downstream para os clientes, de 10 Mb/s. Dessa forma, os clientes c1 a c3 enviam, respectivamente, 3.75 MB, 3.75 MB e 4.62 MB após 10s.

No caso do upstream, a soma dos valores de largura de banda para os três clientes totaliza 4.97Mb/s, o que é aproximadamente os 5Mb/s estipulados quando foi feita a limitação dos valores de upload. Assim, após pouco mais de 10s para cada cliente c1 a c3 foram enviados 2.25 MB, 1.88 MB e 2.38 MB



```
servidor
Documents
-----
Netkit phase 2 initialization terminated
[ ok ] Starting enhanced syslogd: rsyslogd.
servidor login: root (automatic login)
root@servidor:~# iperf -s
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
[ 4] local 192.168.1.11 port 5001 connected with 192.168.0.13 port 43818
[ 5] local 192.168.1.11 port 5001 connected with 192.168.0.12 port 37349
[ 6] local 192.168.1.11 port 5001 connected with 192.168.0.11 port 57934
[ ID] Interval      Transfer      Bandwidth
[ 5] 0.0-11.0 sec  2.25 MBytes  1.71 Mbits/sec
Client connecting to 192.168.0.12, TCP port 5001
TCP window size: 20.0 KByte (default)
[ 5] local 192.168.1.11 port 52959 connected with 192.168.0.12 port 5001
[ 6] 0.0-10.7 sec  1.88 MBytes  1.47 Mbits/sec
Client connecting to 192.168.0.11, TCP port 5001
TCP window size: 20.0 KByte (default)
[ 6] local 192.168.1.11 port 40260 connected with 192.168.0.11 port 5001
[ 4] 0.0-11.7 sec  2.38 MBytes  1.70 Mbits/sec
Client connecting to 192.168.0.13, TCP port 5001
TCP window size: 20.0 KByte (default)
[ 4] local 192.168.1.11 port 46356 connected with 192.168.0.13 port 5001
[ 4] 0.0-10.4 sec  4.62 MBytes  3.75 Mbits/sec
[ 5] 0.0-10.6 sec  3.75 MBytes  2.97 Mbits/sec
[ 6] 0.0-10.7 sec  3.75 MBytes  2.93 Mbits/sec

c2
root@c2:~# iperf -c 192.168.1.11 -t 10
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
Client connecting to 192.168.1.11, TCP port 5001
TCP window size: 20.0 KByte (default)
[ 3] local 192.168.0.12 port 37349 connected with 192.168.1.11 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-11.0 sec  2.25 MBytes  1.72 Mbits/sec
[ 5] local 192.168.0.12 port 5001 connected with 192.168.1.11 port 52959
[ 5] 0.0-10.6 sec  3.75 MBytes  2.96 Mbits/sec

c1
root@c1:~# iperf -c 192.168.1.11 -t 10
Client connecting to 192.168.1.11, TCP port 5001
TCP window size: 20.0 KByte (default)
[ 5] local 192.168.0.11 port 57934 connected with 192.168.1.11 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 5] 0.0-10.7 sec  1.88 MBytes  1.47 Mbits/sec
[ 4] local 192.168.0.11 port 5001 connected with 192.168.1.11 port 40260
[ 4] 0.0-10.7 sec  3.75 MBytes  2.93 Mbits/sec

c3
root@c3:~# iperf -c 192.168.1.11 -t 10
Client connecting to 192.168.1.11, TCP port 5001
TCP window size: 84.0 KByte (default)
[ 3] local 192.168.0.13 port 43818 connected with 192.168.1.11 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-11.6 sec  2.38 MBytes  1.72 Mbits/sec
[ 5] local 192.168.0.13 port 5001 connected with 192.168.1.11 port 46356
[ 5] 0.0-10.4 sec  4.62 MBytes  3.74 Mbits/sec
```

Figure 5: Console de Servidor aos clientes C1, C2, C3