

Validação da Restrição de Banda:

O objetivo deste teste é verificar e validar o funcionamento da restrição de banda e latências referentes à filas, utilizando a ferramenta *iperf* e uma aplicação *UDP (User Datagram Protocol)* para cálculo da média dos *RTT's (Round Trip Time)*.

O cenário utilizado foi de um computador comunicando-se com outro, com o tráfego passando por switches de baixa latência. Ambas as placas de rede foram configuradas para atuar em 100 Mbps *Full Duplex*. Os computadores nas duas pontas estão rodando o sistema operacional *Linux*, distribuição *Ubuntu*, sendo que um deles está com a versão 10.04 e o outro 14.04.

Para possibilitar a utilização do *iperf* para esta tarefa, foi necessário alterar seu código fonte (versão 2.0.5), a fim de possibilitar envios acima da capacidade do enlace (soquete não bloqueante). As modificações constam abaixo:

No arquivo `/src/Client.cpp`:

- Linha 400: Alteração para: `"mSettings->mSock = socket(domain, type | SOCK_NONBLOCK, 0);"`
- Linha 305: Adicionar linha: `"if (currLen < 0 && errno == EAGAIN) { currLen = (mSettings->mBufLen); }`

Inicialmente, pretende-se confirmar o comportamento de envios em *CBR (Constant Bit Rate)* em diversas taxas de envio diferentes. Para isto, mede-se o atraso médio dos pacotes e a porcentagem de perdas de pacotes.

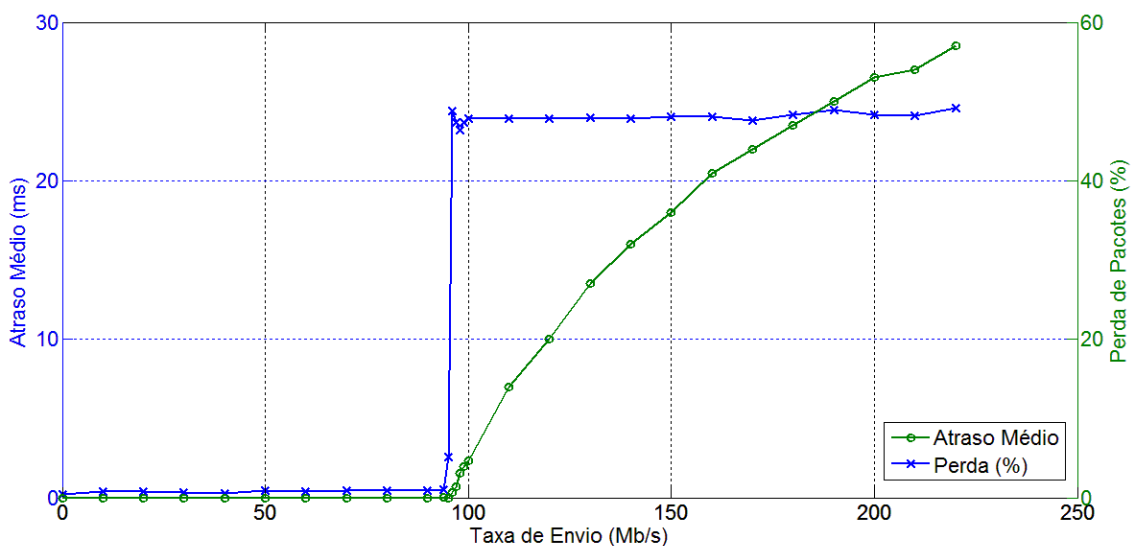


Gráfico 1: Atraso Médio e Perda de Pacotes em função da Taxa de Envio (100Mbps)

Analisando o Gráfico 1, é possível ver que a perda é de 0% até próximo da saturação teórica da rede (100 Mbps), e que depois disso ela começa a aumentar. Esse comportamento é correto, uma vez que esta taxa considera apenas os dados úteis enviados, desconsiderados os bytes de cabeçalhos enviados pelas camadas inferiores do modelo *OSI (Open Systems Interconnection)*. Verificando com o *iperf*, é possível checar que a máxima velocidade de transmissão se restringiu aos 95.7 Mbps.

Para efeitos de cálculo de cabeçalho:

- *UDP: 8 Bytes*
- *IPv4: 20 Bytes*
- *Ethernet: 18 Bytes*
- *Preamble + Interpacket Gap: 20 Bytes*
- *Total: 66 Bytes*

Realizando o cálculo total da velocidade total, considerando que o *iperf* envia dados de 1470 bytes:

$$\frac{95.7}{1470} = \frac{\text{Velocidade Total}}{1470 + 66}$$

$$\text{Velocidade Total} = 99.9967 \text{ Mb/s}$$

Como a medida de velocidade do *iperf* é dada descontando os cabeçalhos, o cálculo da porcentagem de perdas com o enlace saturado é dado pela Equação 1, considerando o limite máximo da placa de rede como sendo 95.7 Mb/s:

$$\% \text{ Perdas} = 1 - \frac{\text{Limite Real do Enlace}}{\text{Taxa Enviada}}$$

Equação 1 – Cálculo da Porcentagem de Perdas

Os valores obtidos para a porcentagem de perda de pacotes se mantiveram muito próximos aos calculados, levando em conta a taxa real de envio do enlace. A Tabela 1 compara os valores medidos e calculados.

Taxa de Envio	Calculado	Medido
90 Mbps	0.00%	0.0%
100 Mbps	4.30%	4.7%
110 Mbps	13.00%	14.0%
120 Mbps	20.25%	20.0%
130 Mbps	26.38%	27.0%
140 Mbps	31.64%	32.0%
150 Mbps	36.20%	36.0%
160 Mbps	40.19%	41.0%
170 Mbps	43.71%	44.0%
180 Mbps	46.83%	47.0%

190 Mbps	49.63%	50.0%
200 Mbps	52.10%	53.0%
210 Mbps	54.40%	54.0%
220 Mbps	56.50%	57.0%

Tabela 1 – Diferença entre Perda de Pacotes (%) medido e calculado

Como esperado, tivemos o atraso médio sempre abaixo dos 1ms para quando o enlace não estava saturado, pelo fato de o envio ser *CBR*. Ao saturar o enlace, este atraso sobe imediatamente para perto dos 24ms, e as perdas de pacotes começam a aparecer.

Estes 24ms são, na verdade, o tempo que o pacote espera na fila de transmissão, que está cheia pelo envio estar acima dos 100Mb/s. Esta fila refere-se ao valor de *Tx* do *Ring Buffer*, que pode ser alterado utilizando a ferramenta *ethtool*, com os devidos privilégios de administrador. Vale notar também que a alteração deste parâmetro não é possível de ser realizada em certas placas de rede.

Observado este detalhe, o próximo teste foi feito a fim de comprovar este comportamento, alterando o tamanho da fila de transmissão da *NIC (Network Interface Card)*. O tamanho ajustado como padrão é de 256 pacotes, porém é ajustável entre 64 e 4096.

Para calcular o atraso teórico em função do tamanho da fila (utilizando a fila cheia e uma *NIC* de 100 Mbps):

$$Taxa\ de\ Saturação\ da\ NIC = \frac{Tamanho\ da\ Fila}{Atraso\ Médio}$$

$$Atraso\ Médio = \frac{8 * Tamanho\ do\ Pacote\ (Bytes) * Número\ de\ Pacotes\ da\ Fila}{100000000}$$

Vale notar estas equações desprezam o tempo de transmissão dos pacotes e de processamento do sistema.

O Gráfico 2 foi gerado utilizando-se uma banda saturada (> 100 Mbps), observando-se então o atraso médio (desconsiderando-se os pacotes perdidos) em função do tamanho da fila.

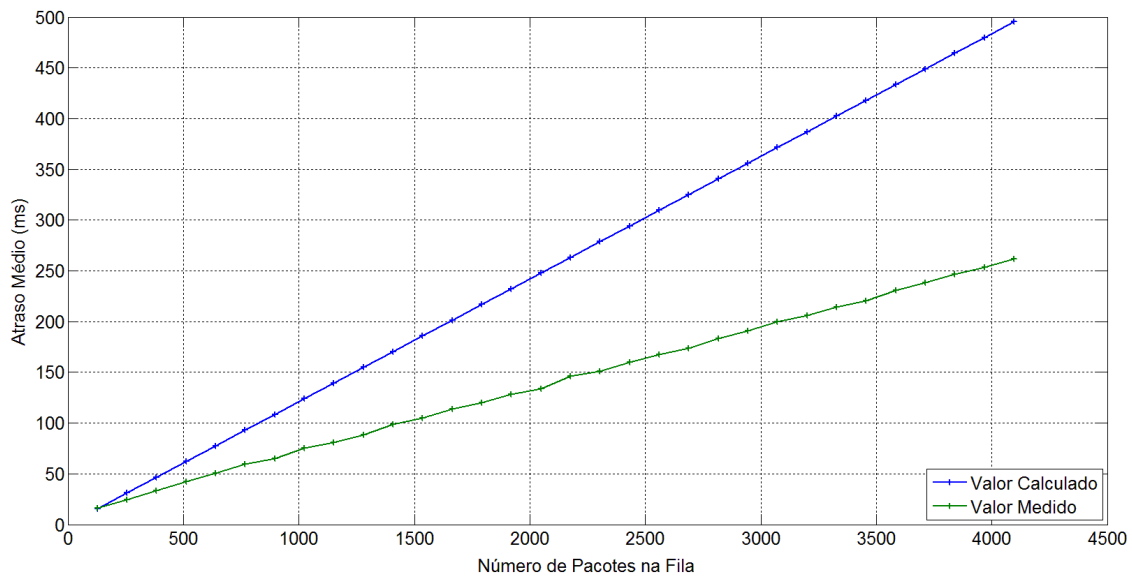


Gráfico 2: Atraso Médio em função do Tamanho da Fila no *Ring Buffer*

O aumento da latência foi de fato linear, entretanto os valores medidos e calculados se divergem muito para filas muito grandes. Isto provavelmente é proporcionado pelo novo algoritmo de *BQL (Byte Queue Limits)* presente nos *kernels* mais recentes do *Linux*, que regula o tamanho da fila automaticamente ¹.

A seguir tomou-se como objetivo verificar a regularidade deste comportamento com a banda limitada. Para exemplos de teste, foram tomados os valores de *100 kb/s*, *1 Mb/s* e *10 Mb/s*. A ferramenta para realizar essas restrições foi o *tbfb (token bucket filter)*, disponível no *tc*.

O *tbfb* foi configurado com os seguintes parâmetros:

- *rate: 100kbit, 1mbit e 10mbit.*
- *limit: 16kB*
- *burst: 1600*

Os valores de *limit* e *burst* foram escolhidos baseados no tamanho do pacote enviado pelo *iperf*, de *1512 bytes*. Desta a forma, a fila sempre terá 10 pacotes, e não permitirá nenhum *burst* adicional.

A latência média é calculada pela mesma forma da fila do *ring buffer*, dividindo-se o tamanho da fila pela taxa limitada. Desta forma, conforme aumentamos o *rate* e mantemos o *limit* constante, a latência média abaixa. Os gráficos a seguir mostram esse comportamento e a validação da limitação de banda pelo *token bucket filter*.

¹ <http://www.coverfire.com/articles/queueing-in-the-linux-network-stack/>

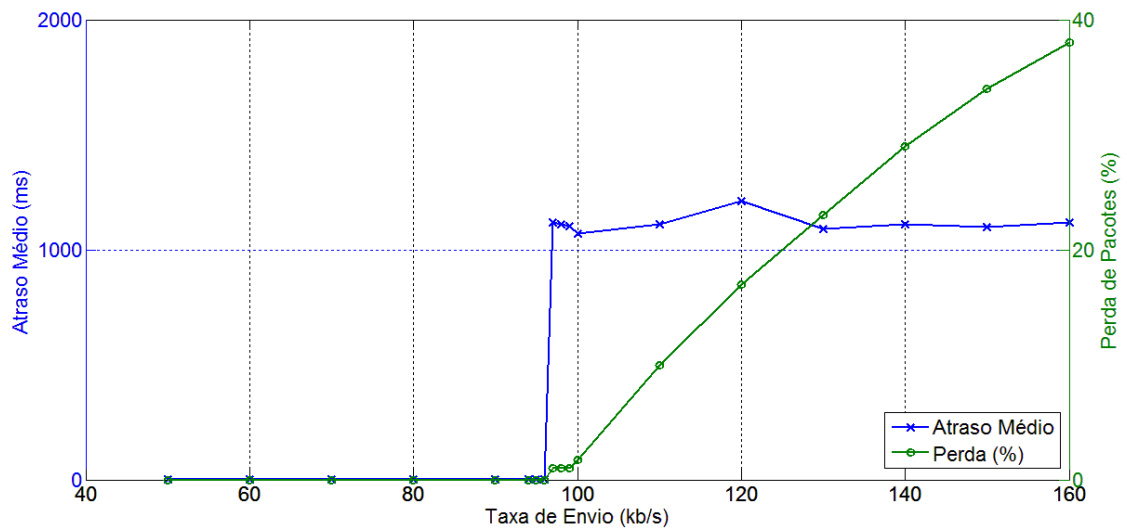


Gráfico 3: Atraso Médio e Perda de Pacotes em função da Taxa de Envio (100kbps)

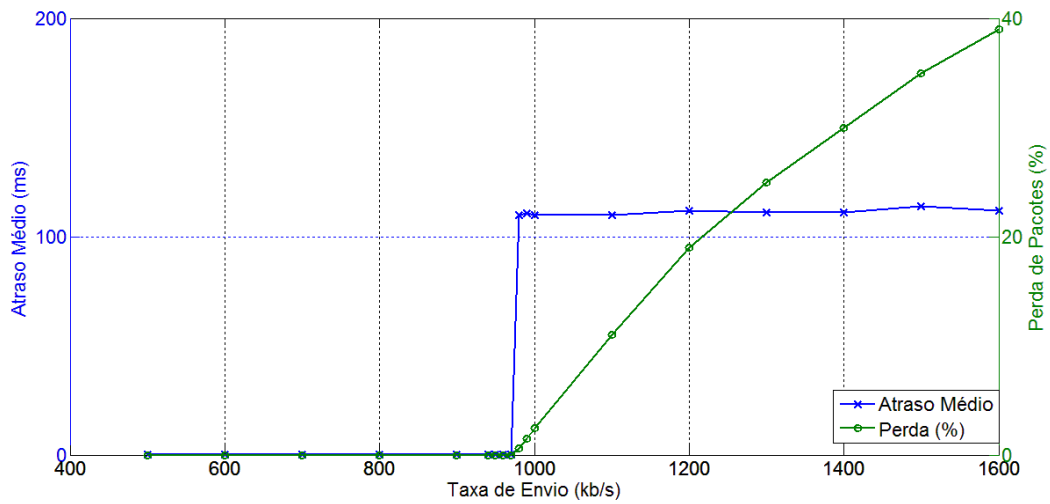


Gráfico 4: Atraso Médio e Perda de Pacotes em função da Taxa de Envio (1Mbps)

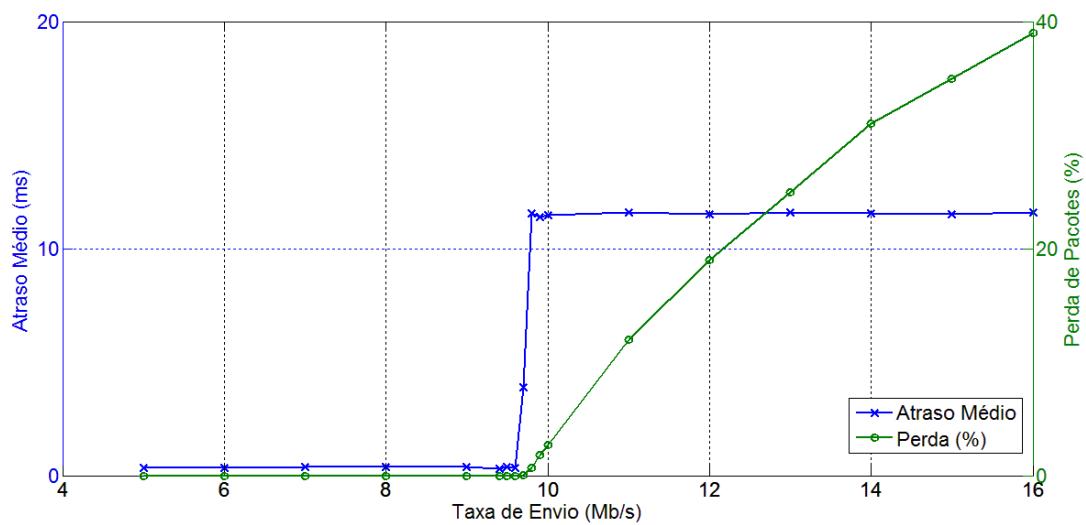


Gráfico 5: Atraso Médio e Perda de Pacotes em função da Taxa de Envio (10Mbps)

É possível perceber que independentemente da taxa de limitação, a perda de pacotes se mantém sempre a mesma conforme aumentamos a saturação da banda. Juntamente com a variância da latência, é possível confirmar que a restrição de banda pelo *tbif* ocorre de forma adequada. Foi observado também que os valores do atraso médio se mantem os mesmos, independente do tamanho da fila do *ring buffer* ou do *txqueuelen*.

Finalmente, foram obtidos valores de atrasos médios para diversos valores de *limit* diferentes, para comprovar a linearidade do algoritmo do *tbif*.

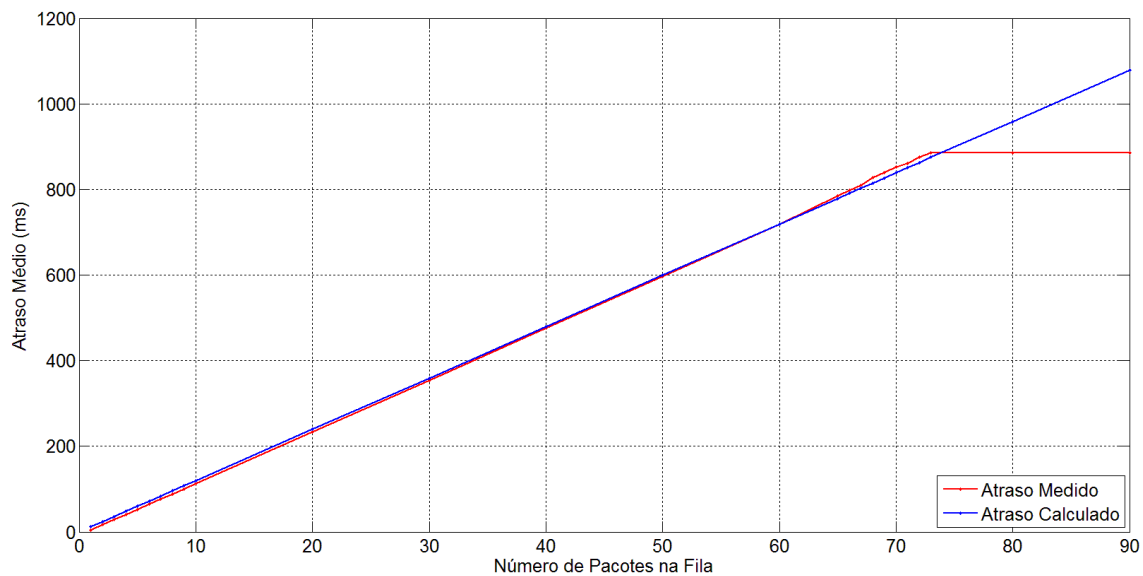


Gráfico 6: Atraso Médio em Função do Número de Pacotes de 1470 Bytes na Fila

Percebe-se que para uma fila de aproximadamente 110kB, o atraso medido para de aumentar, o que significa que este é o tamanho máximo de fila suportado pelo *tbif*. Tendo em vista esses limites da fila, é possível utilizar o *tbif* para restringir a banda em todos os testes necessários, e ainda elimina-se o erro entre o valor calculado e medido imposto pelo *BQL*.

Referências:

<http://www.coverfire.com/articles/queueing-in-the-linux-network-stack/>

<http://sbrc2014.ufsc.br/anais/files/wgrs/ST3-2.pdf>

<http://www.cyberciti.biz/faq/gentoo-centos-rhel-debian-fedora-increasing-txqueueulen/>

<http://www.scottalanmiller.com/linux/2011/06/20/working-with-nic-ring-buffers/>

<http://www.cyberciti.biz/faq/linux-traffic-shaping-using-tc-to-control-http-traffic/>