

Avaliação de desempenho de roteadores

Eduardo Schulz¹

¹ Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS – Brasil

`schulzEduardo@edu.unisinos.br`

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1. Introdução

Quando pensamos em roteadores geralmente temos em mente equipamentos com *hardware* especializado, como *Application-Specific Integrated Circuitss* (ASICs). Como o nome implica, são circuitos integrados com propósito único (alterar) ser usados apenas para esta única função. Porém, as funções de um roteador não precisam ficar limitadas à um *hardware* especializado, em muitos casos pode ser usado computadores convencionais, ou seja por *software* para a realização dessas tarefas.

2. Fundamentação Teórica e Trabalhos Relacionados

Os primeiros roteadores, ainda nos tempos da ARPANET, eram baseados em uma arquitetura de *software*. *Interface Message Processor* (IMP) começou a ser instalado e utilizado a partir do outono de 1969 [Heart et al. 1970]. Os primeiros ASICs só começaram a serem desenvolvidos a partir de 1976, sendo que está tecnologia apenas começou a entrar no mercado de roteadores no ano de 1998 com o Juniper M40.(referencias faltando)

Software Routers por muito tempo foram desconsiderados por não serem tão eficientes quanto roteadores ASICs porém nos dias de hoje com o advento de processadores *multi-core* em até mesmo computadores de baixo custo e de *multi-queue NICs* podemos ter um desempenho muito melhor que no passado usando *software*[Beifuß et al. 2016] Além disso para certas aplicações esses roteadores podem escalonar de uma forma melhor que roteadores ASICs. Porém para grande parte dos casos esses roteadores apresentam um custo mais alto comparados com roteadores de *hardware*.

Para grande parte das aplicações *hardware routers* são superiores nos quesitos de desempenho assim como no quesito de custo. Um exemplo seria uma roteador para uso domiciliar, dificilmente seria justificavel se investir em um *software router* para essa aplicação levando em conta pequena elasticidade necessária. Outro exemplo pode ser um roteador que conecta um AS! (AS!), onde o tamanho de fluxo é tão grande que uma

solução de baseada em *software* não seria capaz de suportar. Mas existiriam casos onde o desempenho é semelhante a um roteador ASIC? Neste trabalho será investigado se a performance de dois computadores, um de uso geral e outro de baixo custo, são capazes de competir com um roteador ASIC.

3. Metodologia

Serão apresentados quatro cenários neste trabalho, que visam comparar o desempenho e capacidade de fluxo de dados na rede. Serão analisados roteadores ASIC assim como roteadores rodando linux, com o intuito de comparar o desempenho da rede.

O cenário I representará um ambiente de controle, ou seja, não possuirá roteadores. Ambos computadores se comunicaram utilizando um *switch* TP-Link de 100 Megabit de banda. O cenário II e III será um ambiente com um roteador linux entre os computadores, sendo que no cenário II o roteador terá grande poder computacional comparado com o cenário III. Já no cenário IV será testado um roteador ASIC rodando *firmware* customizado.

Cada cenário será testado com uma série de testes utilizando diversos tamanhos de datagramas UDP. Será levado em conta *throughput*, pacotes por segundo, latência e recursos consumidos. A ferramenta que será utilizada nestes testes será o iPerf3. Serão executados 10 testes para cada tamanho de datagrama por um período de 30 segundos.

Tem-se como hipótese que o cenário I terá o melhor desempenho comparado com o restante e que o cenário II e IV teram resultados aproximadamente parecidos. Se espera que no cenário III o desempenho será afetado pela falta de recursos do roteador.

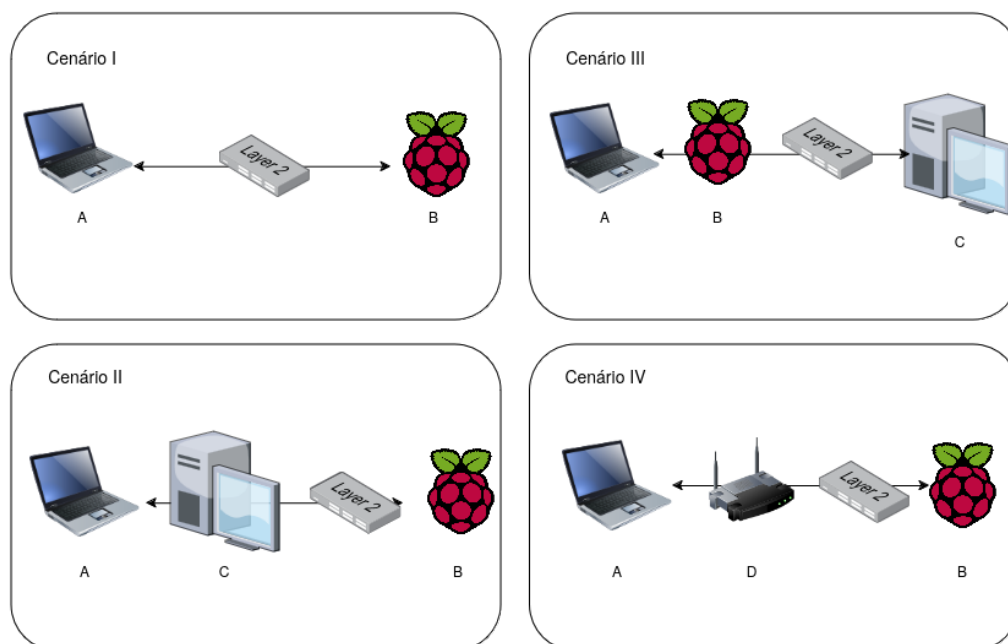


Figura 1. Cenários dos testes a serem realizados.

4. Avaliação de desempenho e funcionalidade

Nessa sessão será apresentado os ambientes dos cenários de testes, principalmente como foram configurados e seus respectivos *hardwares*. Também será exposto os resultados dos experimentos realizados neste trabalho.

4.1. Equipamentos

Como explicado na seção 3, os testes serão realizados seguindo cenários com diversos equipamentos, que devem realizar funções diferentes dependendo do cenário selecionado. A coleta de métricas de consumo de processador, memória, entrada e saída de *bits* pela interface de rede foram realizadas usando o *software* Prometheus em conjunto com o Grafana para a visualização gráfica dos dados.

A - Thinkpad T440s		C - Custom Build	
Processador	Intel® Core™ i5 4200U	Processador	Intel® Xeon® E5-2670v3
RAM	8GB DDR3	RAM	16GB DDR4 ECC
SO	NixOS GNU/Linux	SO	NixOS GNU/Linux
kernel	6.1.79	kernel	6.1.79
B - Raspberry Pi 4		D - TPLink WR741ND	
Processador	Broadcom BCM2711	Banda	FastEthernet 100Mbit/s
RAM	4GB LPDDR4	Firmware	OpenWRT
SO	RPiOS Lite GNU/Linux	Núm. Portas	4
kernel	6.6.20+rpt-rpi-v8	Lançamento	2016

Tabela 1. Especificação dos Equipamentos

4.1.1. Configuração dos Equipamentos

No **cenário I** os equipamentos A e B foram configurados para receberem endereços Internet Protocol (IP) pelo serviço *Dynamic Host Configuration Protocol* (DHCP).

No **cenários II e III**, o roteador um ip estático assim como o segundo computador. O primeiro computador recebeu um endereço IP dinâmico dado pelo roteador acima do roteador.

```
# No roteador
# systemctl net.ipv4.ip_forward
# ip addr add 10.0.1.1/24 dev {iface1}
# iptables -t nat -A POSTROUTING -o {iface2} -j MASQUERADE

# No segundo computador
sudo ip addr add 10.0.1.2/24 dev {iface}
sudo ip route add 192.168.0.0/24 via 10.0.1.1 dev {iface}
```

Já no **cenário IV** o roteador também foi configurado para receber um endereço dinâmico na porta *Wide Area Network* (WAN) e fixo na porta *Local Area Network* (LAN). Toda configuração foi feita pela interface gráfica *web* disponibilizada pelo *OpenWRT*. O segundo computador não precisou ser configurado pois o mesmo ganhou endereço IP dinâmico pelo servidor DHCP do roteador.

4.2. Resultados

4.2.1. Throughput

Como é possível notar na figura 2 o cenário de controle, isto é o I, possuiu a maior taxa de transferência entre todos os cenários. Os cenários II e III tiveram resultados semelhantes porém, o II teve uma taxa ligeiramente superior nos datagramas maiores em comparação

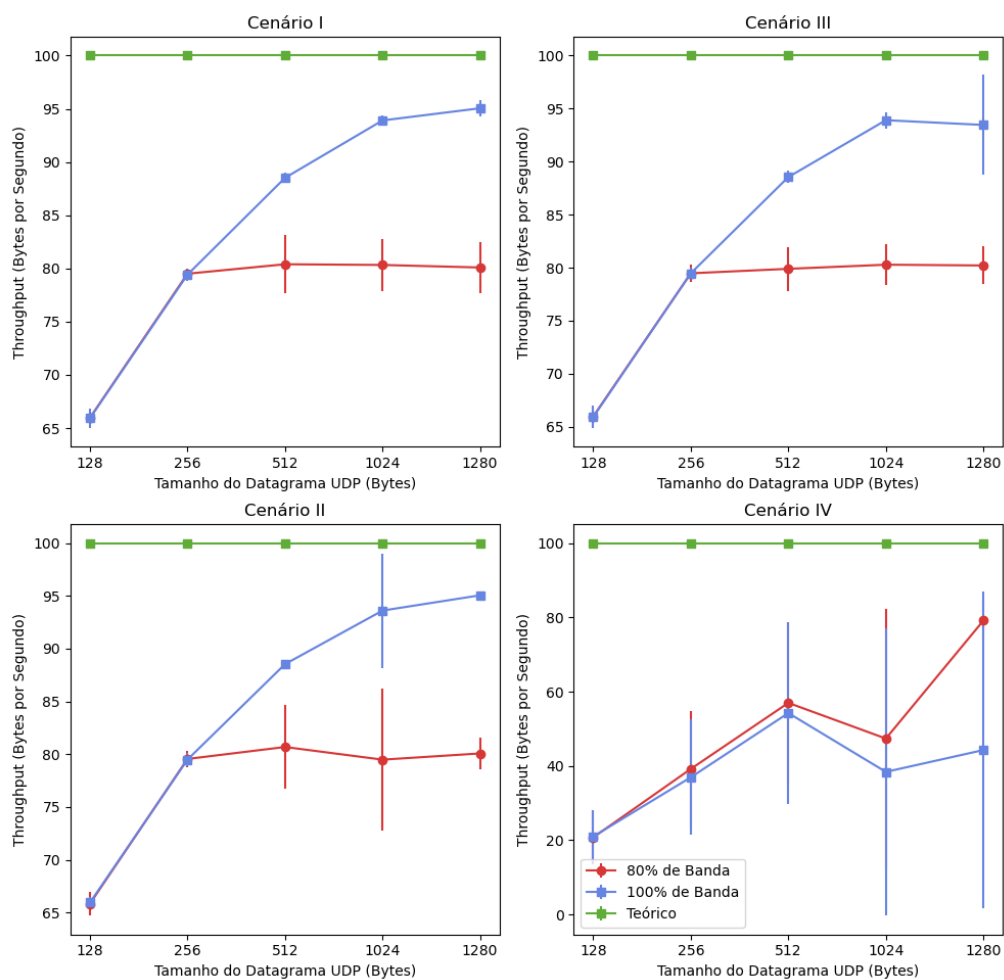


Figura 2. Throughput atingido com datagramas de tamanhos diferentes.

com o cenário III, isto é com uma variância maior também. Já o cenário IV teve o pior dos resultados registrados, com taxas de transferência inferior aos demais e com uma variação elevada.

4.2.2. Número de Pacotes

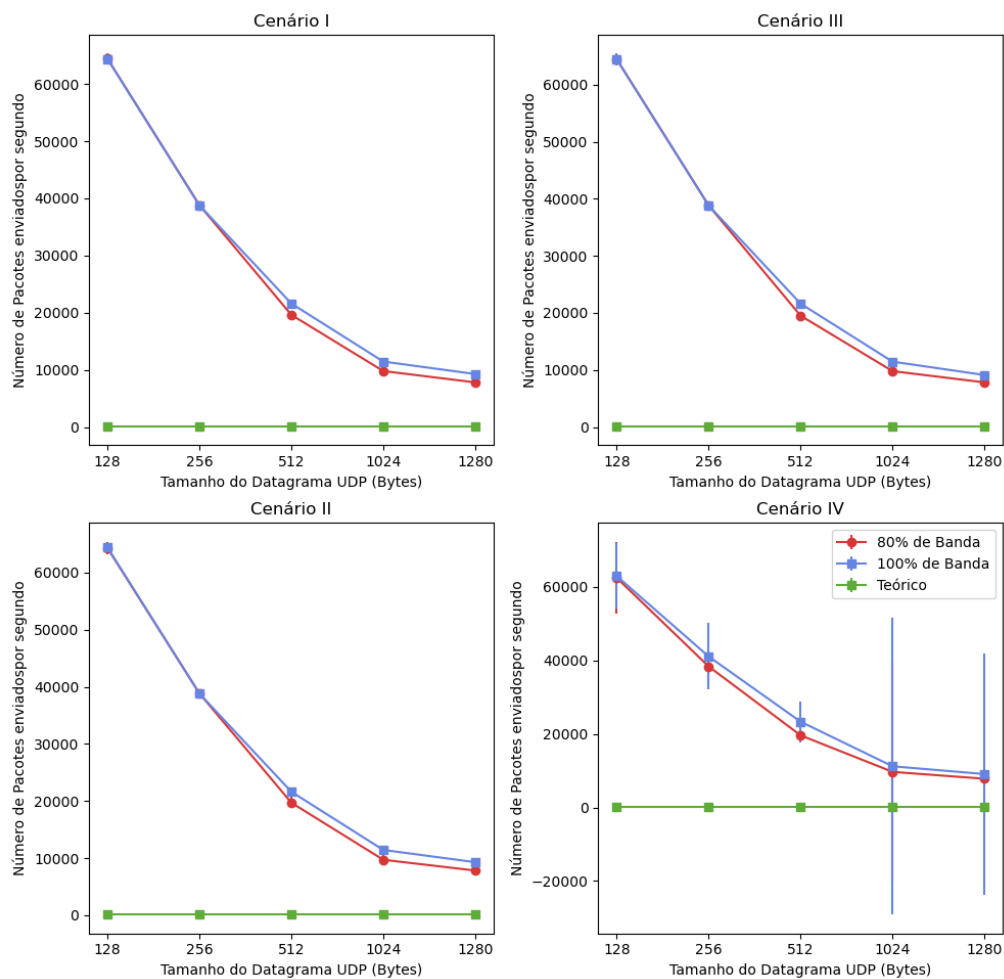


Figura 3. Número de datagramas enviados por segundo em cada teste realizado.

Assim como nos outro gráfico 2, podemos notar um padrão. Nã figura 3 podemos notar pouca diferenca entre os cenários I, II e III, onde todos apresentam um número de pacotes enviados por segundo semelhante. É possível notar que novamente o cenário IV possuiu resultados bem diferentes dos demais, apresentando desvio padrão muito superior que os demais cenários, principalmente nos datagramas com tamanho de 1024 bytes.

4.2.3. Jitter

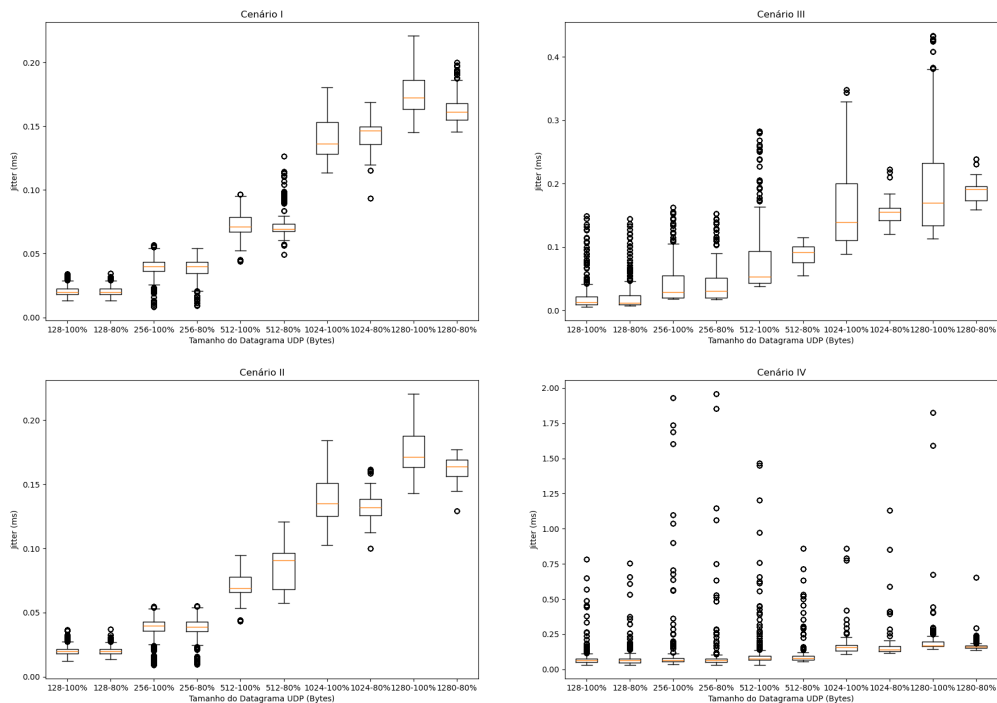


Figura 4. Boxplot dos dados de jitter coletados nos testes seguindo tamanho de datagrama.

Nesta figura 4 podemos analisar as médias assim como os *outliers* coletados pelo iPerf3 quando medindo o jitter. Como podemos notar o cenário I e II tiveram resultados semelhantes. O cenário III por sua vez teve um *jitter* superior aos cenários I e II e com um *outliers* mais notáveis. O cenário IV teve o pior dos resultados comparado com os demais, apresentando latências muito superiores e variações altíssimas.

4.2.4. Perda de Pacotes

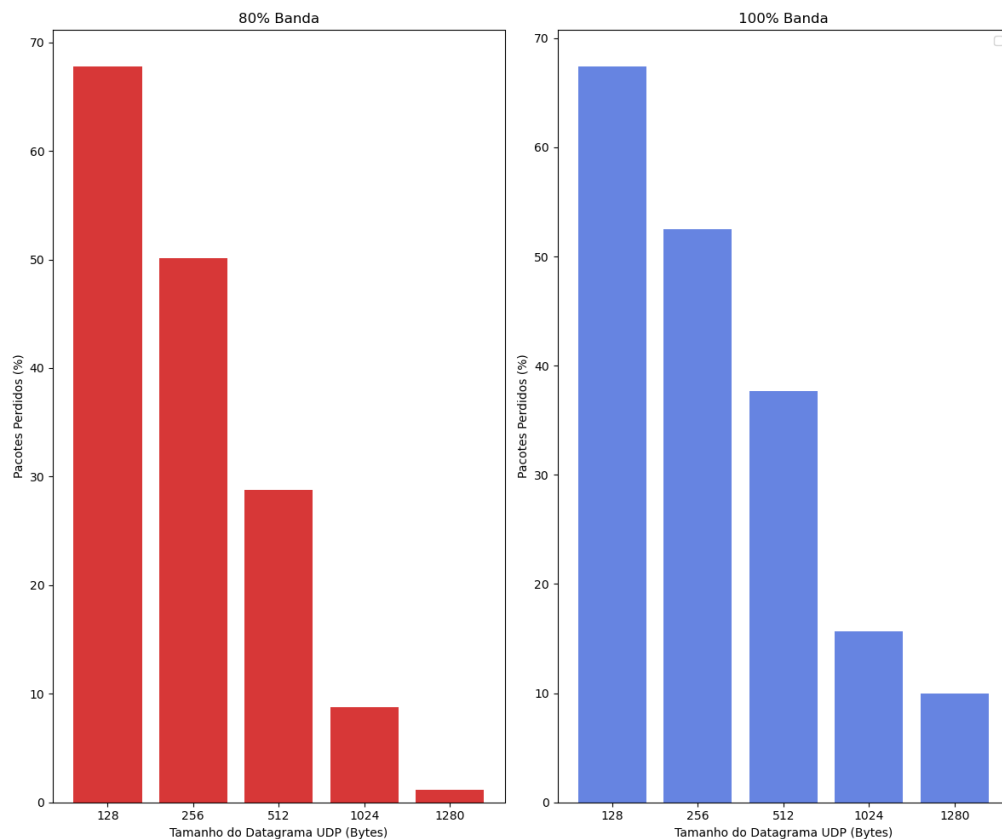


Figura 5. Perda de Pacotes.

Nesta figura 5 é apenas exposto a perda de pacotes ocasionada no cenário IV. Os demais cenários não apresentaram perdas de pacotes significativas, isto é, perdas superiores a 1%. É claramente visível que o roteador deste cenário não foi capaz de suportar tal carga, tendo casos que as perdas ultrapassam 67%.

É possível notar que o roteador teve maior dificuldade lidando com datagramas menores, isso se dá porque o maior custo computacional acontece na quebra do cabeçalho. Então em fluxos com datagramas menores mais pacotes são enviados e por consequência mais pacotes precisam ser desmontados pelo roteador. Já com datagramas maiores o número de pacotes pode ser menor como é visto na figura 3.

5. Conclusão e Discussões

Referências

- Beifuß, A., Runge, T. M., Raumer, D., Emmerich, P., Wolfinger, B. E., and Carle, G. (2016). Building a low latency linux software router. In *2016 28th International Teletraffic Congress (ITC 28)*, volume 01, pages 35–43.
- Heart, F. E., Kahn, R. E., Ornstein, S. M., Crowther, W. R., and Walden, D. C. (1970). The interface message processor for the arpa computer network. In *Proceedings of the May 5-7, 1970, Spring Joint Computer Conference, AFIPS '70* (Spring), page 551–567, New York, NY, USA. Association for Computing Machinery.