Caracterização do Impacto de Interrupções de Rede e Mitigações de CPU em testes de vazão UDP à 10Gbps

Robson S. Santos[1], João P. M. S. Bitencourt[2], Ronan O. de Andrade[3], Daniel Neto[4], Elenice Pedrosa[4], Janssen Martins[4], Marcos Schwarz[4]

[1]Universidade Federal do Ceará (UFC) – Fortaleza – CE – Brasil

[2]Instituto Federal de Santa Catarina (IFSC) – São José, SC – Brasil

[3]Ponto de Presença da RNP no Piauí (PoP-PI) – Teresina, PI – Brasil

[4]Rede Nacional de Ensino e Pesquisa (RNP)

robson.santos@alu.ufc.br, joao.ms@aluno.ifsc.edu.br, ronan.oliveira@pop-pi.rnp.br, {daniel.neto, elenice.pedrosa, janssen.martins, marcos.schwarz}@rnp.br

Abstract. *The use of systems designed for network performance monitoring and analysis, generating observability, is frequently adopted. The demand for high-performance network analysis, whether in academic environments or national and international collaborations, is constantly growing. Seeking to improve the performance of tests conducted using MonIPÊ for circuit homologation in the RNP, we investigate how configurations such as disabling Linux speculative execution vulnerability mitigations and affinity of network card queue interruptions can influence the performance of UDP transfers in virtualized environments. This study examines how optimizing infrastructure usage can ensure packet loss below 0.1% and guarantee that throughput reaches at least 95% of the available bandwidth during data link testing with support for up to 10 Gbps. The experiments also included performance evaluation of the latest iperf 3.18 version, aiming to ensure greater measurement accuracy and stability. The results highlight the impact of infrastructure on test performance and provide guidelines to enhance the monitoring methodology in high-performance academic networks.*

Resumo. *O uso de sistemas destinados à monitoração e análise de desempenho de redes, gerando observabilidade, é frequentemente adotado. A demanda por análises em redes de alto desempenho, seja em ambientes acadêmicos ou de colaboração nacional e internacional, está em constante crescimento. Buscando melhorar o desempenho de testes realizados utilizando o MonIPÊ em homologações de circuitos na RNP, investigamos como configurações, tais como desativação das mitigações de vulnerabilidades de execução especulativa do Linux e afinidade das interrupções das filas da placa de rede, podem influenciar o desempenho de transferências UDP em ambientes virtualizados. O estudo investiga como a otimização do uso da infraestrutura pode ser utilizada para garantir perdas de pacotes abaixo de 0,1% e garantir que a vazão alcance no mínimo 95% da banda disponibilizada, durante a realização de testes de enlaces de dados com suporte de até 10 Gbps. Os experimentos também incluíram a avaliação de desempenho da versão mais recente do iperf 3.18, visando garantir maior precisão e estabilidade nas medições. Os resultados evidenciam o impacto da*

infraestrutura no desempenho dos testes e oferecem diretrizes para aprimorar a metodologia de monitoramento em redes acadêmicas de alto desempenho.

1. Introdução

O monitoramento do desempenho de redes de computadores é fundamental para garantir a eficiência da comunicação e a estabilidade das aplicações que dependem de conexões de alto desempenho. Em redes acadêmicas e de pesquisa, a avaliação da largura de banda, latência e perdas de pacotes é essencial para identificar gargalos e otimizar a alocação de recursos. Ferramentas como o iperf3[1] e plataformas como o perfSONAR [Hanemann et al. 2005] têm sido amplamente utilizadas para esse fim, permitindo medições padronizadas de throughput e qualidade do serviço (QoS). Com o crescimento da demanda por redes de alta velocidade, especialmente aquelas operando a 10 Gbps, estudos recentes têm explorado metodologias para aprimorar a acuracidade das medições e reduzir variabilidades nos resultados [Al-hamadani and Lencse 2021].

Seguindo essa linha, o MonIPÊ foi desenvolvido como uma plataforma de monitoramento distribuído para redes acadêmicas, utilizando hardware de baixo custo e virtualização para expandir a observabilidade da Rede Nacional de Ensino e Pesquisa (RNP) [Vetter et al. 2014]. Essa solução tem se mostrado eficaz na integração com ferramentas como o perfSONAR para medições fim a fim.

No contexto da RNP, a plataforma MonIPÊ desempenha um papel fundamental na análise de performance de circuitos a serem contratados e no mapeamento de falhas, auxiliando os operadores dos Pontos de Presença (PoPs). O serviço de homologação de circuitos, foco deste estudo, foi incorporado posteriormente ao MonIPÊ com o objetivo de validar a qualidade dos enlaces contratados. Esse serviço é executado em uma máquina virtual (VM) dedicada em cada PoP e utiliza contêineres Docker com iperf3 para realizar testes de desempenho. Esses contêineres, executados na VM do PoP, estabelecem uma rede em loopback entre o PoP e o cliente, permitindo a avaliação da vazão, latência e perda de pacotes. A ***Figura 1*** ilustra o teste de circuito, que consiste no envio de pacotes entre dois contêineres, com o tráfego sendo roteado pelo cliente, garantindo que o enlace seja testado em ambas as direções. Com a crescente adoção de enlaces acima de 1 Gbps, a necessidade de suportar transmissões de até 10 Gbps trouxe desafios técnicos que exigem otimizações no ambiente de teste para garantir medições precisas e confiáveis.

Um dos principais desafios identificados no processo de homologação de circuitos foi a dificuldade em atingir os parâmetros de qualidade exigidos nos testes de tráfego UDP. De acordo com os requisitos técnicos para contratação de circutitos da RNP, a taxa de perda de pacotes deve permanecer abaixo de 0,1%, enquanto a vazão alcançada deve atingir pelo menos 95% do valor contratado com a operadora. No entanto, na solução atual do MonIPE em testes realizados em circuitos acima de 1 Gbps, esses requisitos não são atendidos de forma consistente, evidenciando possíveis gargalos e limitações na infraestrutura, componentes e configuração da solução de homologação que podem estar comprometendo a precisão das medições.

Diante desse cenário, o presente estudo busca investigar como ajustes e otimizações no ambiente virtualizado podem impactar a precisão das medições de vazão UDP

[1] `https://software.es.net/iperf/`

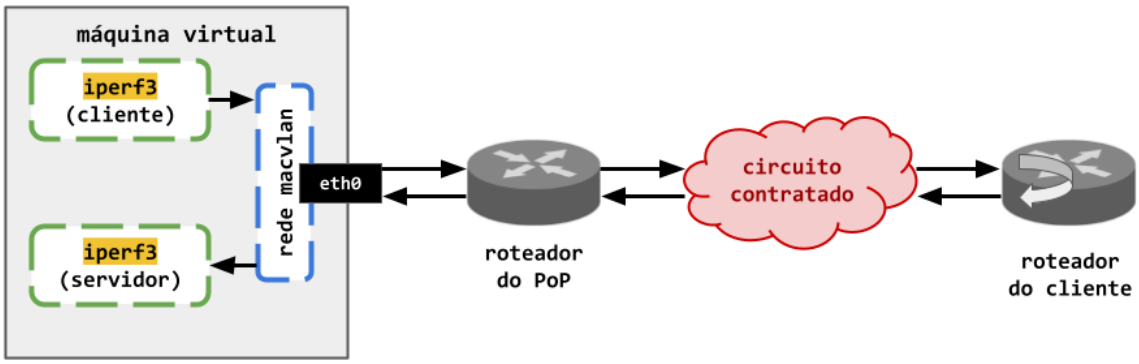

Figura 1. Arquitetura da homologação de circuitos: contêineres iperf3 na VM do PoP testam o circuito em ambos os sentidos.

em infraestruturas de 10 Gbps. Para isso, são exploradas diferentes abordagens para criação de um ambiente de desenvolvimento e testes, com a utilização de configurações, tais como PCI Passthrough, NUMA-aware CPU Pinning, afinidade das interrupções das filas da placa de rede e a opção kernel Linux de desativar as mitigações de vulnerabilidades de execução especulativa (mitigations=off), avaliando como esses ajustes influenciam a estabilidade e a confiabilidade dos testes. Para as medições de rede, foi utilizada a versão 3.18 do iperf3, a mais recente até a data deste estudo, que inclui melhorias significativas nos testes UDP em comparação com versões anteriores. Dessa forma, este trabalho aprimora as otimizações utilizadas pelo MonIPÊ, oferecendo uma metodologia confiável para homologação de enlaces de alto desempenho e contribuindo para o avanço das melhores práticas no monitoramento de redes acadêmicas e de pesquisa.

2. Contribuições

Este trabalho apresenta uma abordagem sistemática para a avaliação de desempenho de redes de 10Gbps em ambientes virtualizados, com foco na comunicação UDP e na garantia de zero perda de pacotes. Para isso, propomos um ambiente de testes automatizado e conteinerizado, permitindo a replicação dos experimentos e a análise de diferentes configurações de *tuning*.

Nossa investigação se concentra em três eixos principais:

- Infraestrutura: Análise do impacto da virtualização e do uso de MTU 9000 na comunicação UDP de alta velocidade.
- Ferramentas de Teste: Implementação de um arcabouço replicável para experimentos, utilizando conteinerização para automação dos testes.[2]
- Otimizações e Resultados: Comparação de diferentes cenários de *tuning* considerando impacto das interrupções das filas da placa de rede, bem como da desativação das mitigações do Linux de vulnerabilidades de execução especulativa de CPU ((mitigations=off) para avaliação do desempenho da versão mais recente do iperf 3.18.

Os experimentos visam identificar as melhores estratégias para alcançar zero perda de pacotes em comunicações UDP de 10Gbps utilizando apenas um fluxo, fornecendo diretrizes práticas para otimização de ambientes virtualizados de alto desempenho.

[2]`https://git.rnp.br/melhorias-monipe/teste-automatizado`

Perguntas de Pesquisa:

- Se existem, quais são as configurações de *tuning* necessárias para garantir zero perda de pacotes em testes UDP sobre redes de 10Gbps com apenas um fluxo?
- De que forma a conteinerização pode facilitar e tornar replicável a avaliação de desempenho em ambientes de rede de alta velocidade?
- Quais são os principais impactos de desempenho entre diferentes configurações de *tuning* para interrupções da placa de rede, bem como das mitigações de execução especulativa de CPU?

3. Trabalhos Relacionados

A análise de redes acadêmicas e de alto desempenho tem sido foco de diversos estudos, principalmente no que se refere à otimização de infraestrutura, ferramentas de medição e configurações avançadas para testes de redes de alta velocidade. Embora existam propostas voltadas para monitoramento e avaliação de tráfego em redes de 10 Gbps, ainda há desafios a serem solucionados, especialmente no que diz respeito à homogeneização de cenários de teste, precisão das medições e impacto das configurações do ambiente virtualizado.

3.1. Infraestrutura para Monitoramento de Redes Acadêmicas

A infraestrutura de redes acadêmicas de alto desempenho requer soluções que garantam medições confiáveis de tráfego TCP e UDP, especialmente em ambientes virtualizados. O perfSONAR tem sido amplamente utilizado como ferramenta para medições fim a fim, mas estudos recentes têm buscado superar suas limitações de granularidade e *overhead*. Em [Mazloum et al.] os autores propõem a integração do perfSONAR com *switches* programáveis P4, permitindo a coleta passiva de dados reais da rede em nível de fluxo, o que amplia a visibilidade, reduz a carga imposta por testes ativos e melhora a detecção de eventos como *microbursts* e perdas causadas por DTNs mal configurados.

Ainda, diversos trabalhos têm discutido o impacto da virtualização no desempenho de testes de rede, especialmente quando há a necessidade de minimizar perdas de pacotes e garantir latências reduzidas. Estudos recentes também apontam que ganhos de desempenho só são obtidos quando melhorias no kernel Linux são acompanhadas de tunings avançados. [Schwarz et al. 2024] analisaram os impactos do MSG ZEROCOPY e do BIG TCP no *throughput* de redes de 100 Gbps, evidenciando que o uso combinado de técnicas como iommu=pt, optmem_max, CPU pinning, pacing de pacotes e tunings específicos no kernel 6.8 pode aumentar em até 38% o desempenho em WANs quando comparado ao kernel 5.15.

3.2. Ferramentas de Teste e Automação

O iperf3 é uma das ferramentas mais utilizadas para medições de vazão, latência e perdas de pacotes em redes TCP e UDP. O seguinte estudo [ANDRADE 2021] avaliou o desempenho do iperf3 utilizando o testbed Mentored, identificando gargalos na comunicação inter-ilhas e variações de throughput. No entanto, o estudo não explora a padronização das medições e os impactos da escolha da versão do iperf3 nos resultados. Nosso trabalho aborda essa lacuna ao comparar diferentes versões do iperf3 (3.11 e 3.18), executadas em contêineres com base em perfSONAR Tools, verificando discrepâncias entre versões e otimizando a execução dos testes.

Outro problema recorrente na literatura é a inconsistência de medições em simulações de rede. O estudo [Hardin et al. 2023] analisou a confiabilidade de medições feitas com o Mininet e iperf3, demonstrando que o tamanho da janela TCP e as configurações do *buffer* podem distorcer significativamente os valores de vazão. Nosso trabalho se diferencia ao utilizar máquinas virtuais completas (KVM) com acesso direto à NIC via PCI Passthrough, eliminando as limitações identificadas nesse estudo e garantindo maior fidelidade nas medições.

Além disso, os estudos anteriores não abordam automação para execução de testes e coleta de resultados, algo essencial para garantir reprodutibilidade e padronização. Nosso estudo propõe um arcabouço automatizado, baseado em *scripts* e análise estatística, permitindo a execução padronizada dos testes e a comparação detalhada entre diferentes configurações.

3.3. Tunings e Impacto na Performance

A otimização da infraestrutura de testes é um dos diferenciais do nosso estudo. O artigo de [Lei et al. 2021] analisou o impacto do Sockperf em redes Ethernet de 100 Gbps, demonstrando que a utilização da CPU em redes *overlay* é significativamente maior. Embora tenha contribuído para a análise de latência e *jitter*, o estudo não explora ajustes no sistema operacional ou impactos da alocação das interrupções das filas da placa de rede em diferentes CPUs no desempenho dos testes. Nosso trabalho complementa essa abordagem ao investigar e avaliar diferentes estratégias de alocação das interrupções, garantindo melhor aproveitamento dos recursos computacionais, aumentando a consistência dos testes e o desempenho, bem como minimizando a perda nos testes UDP.

O estudo de [Kempf et al. 2024] avaliou o impacto do tamanho do buffer e do offloading da NIC em redes de alta taxa de transmissão, demonstrando que buffers de tamanho padrão são insuficientes para QUIC e UDP, resultando em perdas desnecessárias de pacotes. Nosso trabalho expande essa análise ao investigar como ajustes específicos no Linux podem melhorar a estabilidade das medições em cenários 10Gbps.

Já o estudo [Gál et al. 2021] analisou sessões paralelas de TCP e UDP, destacando como diferentes configurações de buffer e algoritmos de congestionamento impactam a taxa de transferência e estabilidade da rede. Embora tenha explorado cenários de comunicação massivamente paralela, o estudo não se concentrou exclusivamente em UDP, nem avaliou otimizações específicas para redes de alto throughput. Nosso trabalho avança nessa área ao realizar testes exclusivamente com UDP, avaliando sistematicamente cenários com diferentes níveis de mitigação de segurança e alocação de CPU, e aplicando ajustes refinados em *buffers* e pilha de rede no Linux, garantindo maior estabilidade e minimização de perdas em redes de 10 Gbps.

Em contraste com os trabalhos anteriores que analisam aspectos isolados da virtualização, ferramentas de medição ou *tunings* específicos, o presente estudo propõe uma abordagem integrada e reprodutível para avaliação de desempenho em enlaces de 10 Gbps, incorporando avanços em automação, isolamento de recursos e análise estatística. A principal inovação está na criação de um ambiente de testes controlado, com máquinas virtuais configuradas para acesso direto à placa de rede via *PCI Passthrough* e mapeamento fixo dos núcleos de CPU. Também foram desativadas algumas proteções do sistema e isolados núcleos específicos, visando reduzir interferências e garantir maior previsibilidade na

execução dos testes.

A metodologia desenvolvida inclui um conjunto de *scripts* que automatiza a orquestração dos testes, coleta e sumariza os dados gerados, facilitando a análise de variáveis como vazão, uso de CPU e taxa de perda de pacotes. Além disso, foram avaliadas múltiplas afinidades de interrupção da placa de rede, permitindo identificar gargalos causados pela alocação dinâmica padrão do Linux. A proposta também difere pela adaptação do perfSONAR Tools com a versão mais recente do iperf3 (v3.18) e pela comparação entre diferentes configurações de *tunings*, incluindo ajustes nos *buffers* UDP e uso de *Jumbo Frames*. Todos esses ajustes, combinados com a execução de testes bem estruturados em dois cenários distintos, com e sem mitigações de segurança, reforçam a relevância deste trabalho. A proposta se destaca por oferecer uma solução prática, atualizada e de fácil replicação para avaliar o desempenho de redes acadêmicas de alta velocidade em ambientes virtualizados.

4. Metodologia

A fim de replicar e avaliar o desempenho da solução do MonIPE ilustada na ***Figura 1***, foi desenvolvida uma infraestrutura de testes baseada nas mesmas características do ambiente de produção, como virtualização e contêineres. Os experimentos foram realizados em máquinas virtuais executadas através do *hipervisor* KVM (*Kernel-based Virtual Machine*). Abaixo na ***Tabela 1***, estão as características do *hardware* sobre o qual as máquinas virtuais foram instanciadas:

Tabela 1. Características do *hardware* utilizado.

Componente	**Descrição**
Servidor	Dell R620
Informações da memória RAM	16 GB, DDR3, 1666 MHz
Modelo do processador	Intel(R) Xeon(R) CPU E5-2609 v2, com 8 núcleos
Tipo de armazenamento	HDD
Placa de rede de 10 Gbps	NetXtreme II BCM57800 1/10 Gigabit Ethernet 168a

Fonte: Autor.

O sistema operacional utilizado no servidor físico é o Ubuntu 20.04, com o *kernel* Linux 5.15.0-131-generic, que foi configurado para que permitisse a passagem direta de dispositivos de rede para a máquina virtual (*PCI Passthrough*), além do uso de diferentes camadas de isolamento dos núcleos de CPU. Além disso, os núcleos 0, 2, 4 e 6 estão isolados do escalonador de processos. Dessa forma, os seguintes argumentos foram adicionados aos parâmetros de inicialização:

- `intel_iommu=on`;
- `iommu=pt`;
- `isolcpus=0,2,4,6`.

As máquinas virtuais foram instanciadas com o auxílio da ferramenta de automação Vagrant, utilizando a extensão *vagrant-libvirt*, e possuem as seguintes características:

- Sistema Operacional openSUSE Tumbleweed;
- *Kernel* versão 6.13.7-1-default;

- 4 núcleos de CPU, sendo que cada núcleo virtual é associado a um núcleo físico isolado diferente;
- CPU em modo *host-passthrough*, garantindo o uso máximo dos recursos do hardware subjacente;
- alocação no mesmo nó NUMA do *host* onde está localizada a placa de rede, isso ajuda no desempenho ao garantir que memória e CPU sejam alocadas no mesmo nó entre a aplicação e o dispositivo de rede;
- Mapeamento entre as CPUs virtuais da máquina virtual e CPUs físicas do *host*, de forma que cada CPU virtual possui um CPU físico dedicado;
- 6 GB de memória RAM;
- Passagem direta da placa de rede de 10 Gbps, NetXtreme II BCM57800 1/10 Gigabit Ethernet 168a.

No sistema virtualizado, o Docker foi instalado para facilitar a implantação e gerenciamento de contêineres. Dois contêineres foram instanciados, sendo utilizada a imagem com o *perfSONAR Tools*, adaptada com a versão mais atualizada do iperf3 (v3.18). O pacote *perfSONAR Tools* consiste em um conjunto de ferramentas projetadas para a execução de medições sob demanda, incluindo iperf, iperf3 e owamp, que são amplamente utilizadas na literatura. Cada contêiner foi configurado com um endereço IP próprio, simulando dois nós distintos dentro da mesma rede, garantindo um ambiente de teste isolado e controlado. O primeiro contêiner foi configurado como servidor iperf3, aguardando conexões de teste, enquanto o segundo contêiner atuou como cliente, executando sucessivas medições utilizando protocolo UDP.

Para a obtenção dos resultados, foram desenvolvidos *scripts*[3] para automação dos testes de vazão de forma padronizada e sumarização dos resultados obtidos por meio de análise estatística e geração de gráficos. A automação executa as medições sob diferentes cenários, garantindo a reprodutibilidade e permitindo a comparação precisa entre diferentes configurações de *tunings*.

Para a realização dos testes, primeiramente, a ferramenta principal utilizada é a rotina `executa-experimento` desenvolvida em *shell-script*, que permite definir parâmetros como os núcleos de CPU utilizados pelo cliente e pelo servidor, o arquivo `docker-compose.yml` com a descrição do cenário, o nome do teste, a duração, a largura de banda máxima e o número de repetições. Caso não sejam especificados, alguns desses parâmetros assumem valores padrão (1 rodada, 10 segundos de duração e 10Gbps de vazão), garantindo flexibilidade na configuração dos experimentos.

Após a execução dos testes, os resultados são armazenados em um diretório especificado pelo usuário. Para a análise e sumarização dos dados obtidos, utiliza-se o script `sumarizar-experimento.py`, que recebe como entrada o diretório de resultados e os apelidos dos testes realizados. A saída gerada apresenta informações como o uso da CPU por núcleo, a vazão de rede em Gbps e a taxa de perda de pacotes, além de gráficos comparativos nos formatos `png` e `svg`. Esses gráficos incluem visualizações do uso da CPU, da vazão e da taxa de perda, tanto para rodadas individuais quanto para comparações gerais entre diferentes experimentos.

Esse procedimento possibilita uma avaliação detalhada do desempenho da rede e da utilização de recursos computacionais nos testes conduzidos. A metodologia adotada,

[3] `https://git.rnp.br/melhorias-monipe/teste-automatizado`

baseada em automação e reprodutibilidade, permite uma análise sistemática dos resultados, facilitando a identificação de padrões e variações no comportamento da infraestrutura testada. Além disso, a geração automática de gráficos e estatísticas proporciona uma interpretação visual dos dados, tornando mais eficiente a compreensão do impacto das diferentes configurações testadas.

Para garantir um ambiente otimizado durante os testes, foram aplicadas diversas configurações no sistema operacional Linux e no adaptador de rede Ethernet. Abaixo está a descrição das principais configurações adotadas.

Sistema Operacional Linux

- **Mitigations=off**: Essa opção desativa mitigações de vulnerabilidades de CPU como Spectre, Meltdown, L1TF e MDS, reduzindo a sobrecarga no processamento. Contudo, a sua aplicação deve ser avaliada com a equipe de segurança.
- **UDP Buffer Configuration**: Ajustados os buffers de memória para soquetes UDP a fim de evitar perda de pacotes devido à insuficiência de espaço para armazenamento temporário.

Configurações do Adaptador de Rede Ethernet

- **Jumbo Frame**: Ativado para melhorar a eficiência na transmissão de grandes volumes de dados contíguos, reduzindo a sobrecarga de cabeçalho de pacotes.
- **Direcionamento de Fluxo TX e RX**: Configurado RSS (*Receive Side Scaling*) para distribuir tráfego de entrada em múltiplos núcleos da CPU e XPS (*Transmit Packet Steering*) para alinhar a transmissão com os núcleos apropriados, melhorando o desempenho do sistema.

Nos experimentos realizados, foram avaliados dois cenários distintos para análise de desempenho em ambientes virtualizados, considerando diferentes parâmetros de *tunings*. Em todos os cenários, no *Host* foi utilizado um ambiente com suporte a I/O Memory Management Unit (IOMMU) em modo "*pass-through*"(pt). No sistema virtualizado, foram configuradas *Jumbo Frame*, que aumentam o tamanho máximo dos pacotes transmitidos, e ajustes na configuração de memória para o protocolo UDP, visando reduzir *overheads* e melhorar a vazão dos pacotes.

No **cenário 1**, a mitigação de vulnerabilidades foi mantida ativada (**mitigations=auto**) tanto no *host* quanto no sistema virtualizado, proporcionando um nível de segurança adequado, mas possivelmente impactando o desempenho devido às proteções contra vulnerabilidades.

Por fim, no **cenário 2**, a mitigação foi completamente desativada (**mitigations=off**) tanto no *host* quanto no sistema virtualizado, eliminando quaisquer penalidades associadas às proteções de segurança, o que resulta em maior desempenho, mas ao custo de maior exposição a vulnerabilidades conhecidas.

Adicionalmente, o *daemon* `irqbalance` foi removido, permitindo a atribuição manual das interrupções a núcleos específicos. Nos dois cenários cada um dos CPUs da máquina virtual foi isolado para uma função específica, de forma a aumentar o determinismo dos testes e facilitar a detecção de gargalos relacionados ao processamento: No CPU 0 são

executados os *scripts* de teste e processos do sistema, enquanto o CPU 1 foi alocado para a execução do servidor *iperf3*. O CPU 2 foi dedicado à execução do cliente *iperf3*, e o CPU 3 permaneceu livre, possibilitando a análise de sua influência na carga de processamento e na alocação de recursos da rede.

A segunda característica avaliada foi o impacto da alocação das interrupções das filas da placa de rede em diferentes CPUs. Por padrão, as filas da placa de rede são alocadas dinamicamente em todos os CPUs. Sendo assim, a cada execução dos testes, tanto as interrupções da fila de envio (irq-tx), quanto as de recebimento (irq-tx), podem ser alocadas em diferentes CPUs, bem como, durante o próprio teste, o escalonador do Linux pode realocar as interrupções para outros CPUs. Esse comportamento compromete a consistência das medições e é uma das fontes de perdas detectadas. Para caracterizar esse impacto, em cada cenário foram executadas 5 (cinco) possibilidades de afinidade das interrupções:

- rx2/tx2 - Ambas as interrupções de recebimento e envio alocadas no CPU 2, onde também é executado o iperf3 cliente;
- rx2/tx1 - A interrupção de recebimento alocada no CPU 2 que executa o iperf3 cliente e a interrupção de envio junto ao iperf3 servidor;
- rx1/tx2 - A interrupção de recebimento alocada no CPU que executa o iperf3 servidor e a interrupção de envio junto ao iperf3 cliente;
- rx1/tx1 - Ambas as interrupções de recebimento e envio alocadas no CPU 1, onde também é executado o iperf3 servidor;
- rx3/tx3 - Ambas as interrupções de recebimento e envio estão alocadas no CPU 3, que está livre.

5. Resultados

Os experimentos realizados nos cenários 1 e 2 permitiram avaliar o desempenho da vazão UDP, uso de CPU e a perda de pacotes, considerando diferentes configurações.

No primeiro cenário, cujos sistemas operacionais do hospedeiro e da máquina virtual estavam configurados com o argumento de *kernel* `mitigations=auto`, pode-se observar que a alocação das interrupções em diferentes *cores* impacta diretamente o desempenho da vazão e a utilização de CPU, sendo a melhor vazão alcançada na configuração ”rx3/tx3”, sendo mais que o dobro da pior configuração ”rx2/tx2”, conforme mostrado na Figura2. Essa diferença pode ser explicada pelo fato da configuração ”rx3/tx3”utilizar as interrupções no *core* ocioso. Nos demais casos, as interrupções estão competindo por tempo de processamento nos núcleos que estão executando a aplicação de transferência, o que impacta diretamente o desempenho dos testes.

Dentre os testes que não usam o *core* ocioso, a configuração ”rx1/tx1”teve o melhor resultado, onde é possível constatar que o processo mais intenso em processamento é o do cliente *iperf*, sendo ele o gerador de pacotes. Nos testes ”rx1/tx2”e ”rx2/tx1”, em que as interrupções das filas de transmissão e recepção ficaram em núcleos alternados, percebe-se a maior degradação na vazão quando as interrupções da fila de recepção compartilham o núcleo do cliente *iperf*.

Na Figura 3, fica demonstrado que, em todos os cinco testes, o núcleo do cliente apresenta constante uso máximo de processamento. Isso explica a degradação massiva quando qualquer uma das interrupções recai sobre o mesmo. Além disso, observando o teste ”rx3/tx3”, é possível comprovar que o núcleo do servidor possui uma carga de

processamento menor que a do cliente. Adicionalmente, no teste com maior degradação no desempenho, ”rx2/tx2”, é possível perceber que, à medida que o cliente sofre gargalo, o servidor recebe menos pacotes para processar e, como consequência, consome menos processamento. Observando o teste ”rx1/tx1”na Figura 3 e o valor de perda para o mesmo na Tabela 2, nota-se que houve sobrecarga do *core* que executa `iperf` servidor.

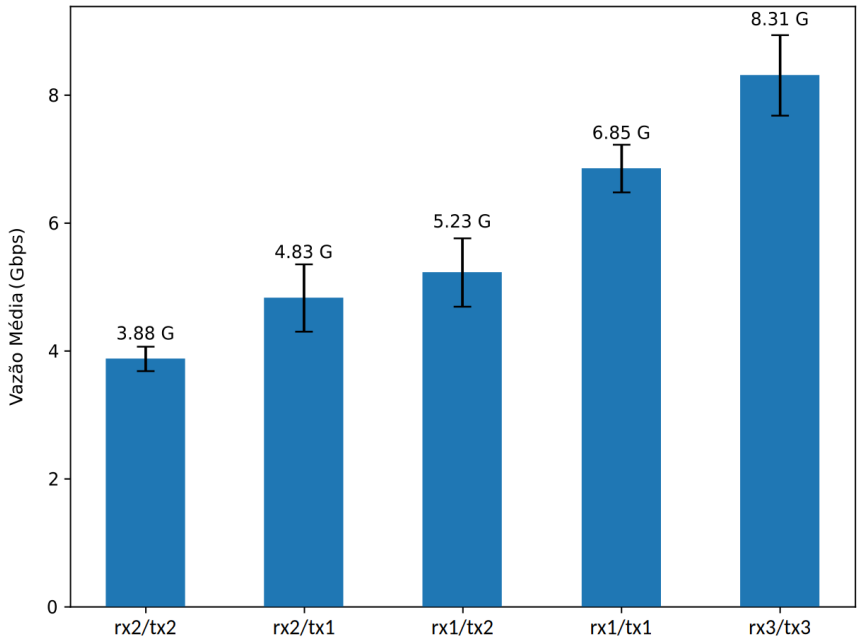

Figura 2. Vazão do servidor IPerf 3.18 no cenário 1.

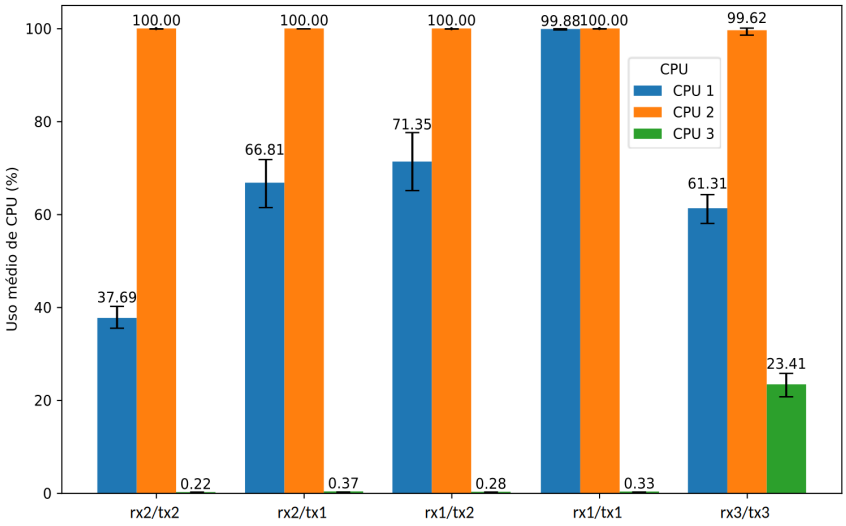

Figura 3. Uso de CPU, mostrando os núcleos 1, 2 e 3 no cenário 1.

No segundo cenário, com o argumento `mitigations=off`, tanto no *host* quanto na máquina virtual, verificou-se uma maior taxa de transferência, indicando que a remoção das mitigações pode ser benéfica para ambientes que buscam o máximo de desempenho ou onde esse tipo de vulnerabilidade não seja um fator crítico, por exemplo, ambientes que apenas executem aplicações conhecidas e que não recebam acessos de terceiros.

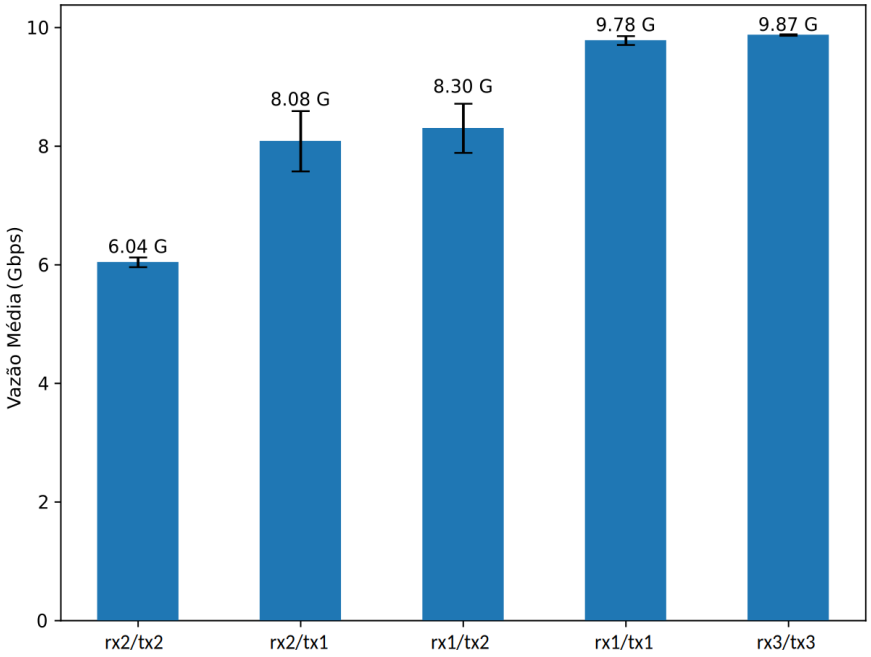

Figura 4. Vazão do servidor IPerf 3.18 no cenário 2.

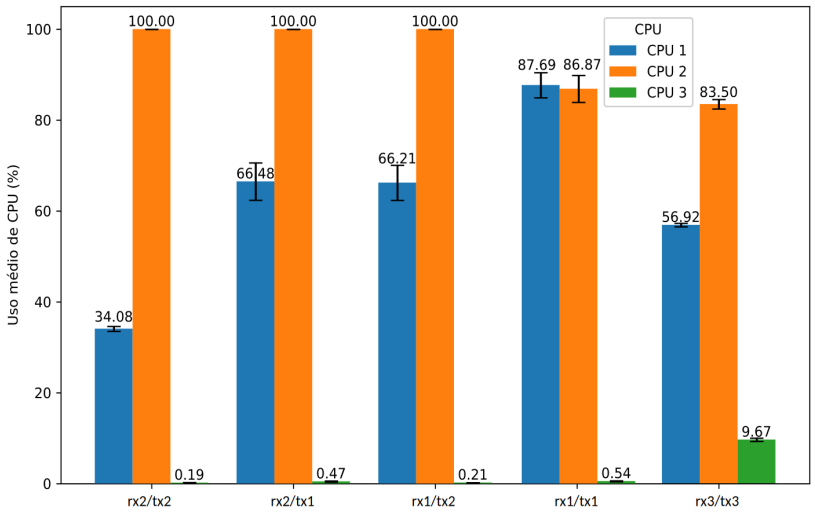

Figura 5. Uso de CPU, mostrando os núcleos 1, 2 e 3 no cenário 2.

Na Figura 4 é possível notar resultados maiores de vazão para todos os cinco testes realizados, se comparados com o Cenário 1, chegando a um ganho de até 67% como pôde ser observado pela diferença entre o teste ”rx2/tx1”entre os dois cenários. O melhor resultado foi novamente observado no teste ”rx3/tx3”, que atingiu a vazão máxima suportada pela placa de rede utilizada. O segundo melhor resultado ficou no teste ”rx1/tx1”, conforme também ocorreu no Cenário 1. O pior teste foi ”rx2/tx2”, cujo gargalo ocorreu no cliente. Nos testes ”rx2/tx1”e ”rx1/tx2”seguiram com níveis próximos de vazão.

Na análise de consumo de processamento, a Figura 5 mostra que o menor consumo para o cliente e servidor foi no teste ”rx3/tx3”. Vale ressaltar que, apesar do núcleo

do servidor ter uso mais elevado do que em ”rx2/tx2”, este só possui valor mais baixo por conta de gargalo no processo do cliente. O segundo melhor resultado ficou com o teste ”rx1/tx1”, que traz o benefício de utilizar somente dois núcleos de processamento para o experimento. Nesse teste, os núcleos do cliente e servidor alcançaram volume de processamento médio próximos, conseguindo vazão superior a 9,5 Gbps. Para os testes ”rx2/tx1”e ”rx1/tx2”, nota-se que, em ambos os casos, o consumo de processamento ficou praticamente o mesmo no cliente e no servidor em cada teste.

Teste	Cenário 1	Cenário 2
rx2/tx2	0,0000	0,0001
rx2/tx1	0,0000	0,0001
rx1/tx2	0,0000	0,0001
rx1/tx1	0,1881	0,0093
rx3/tx3	0,0019	0,0016

Tabela 2. Perdas de cada cenário, nas diferentes configurações de interrupção.

Na Tabela 2, são mostrados os valores de perda para cada teste em cada cenário. É possível observar que a maior perda ocorreu quando todas as interrupções foram alocadas no núcleo de processamento do servidor e as mitigações estavam todas habilitadas. Isso indica que o cliente gerou mais pacotes do que o servidor conseguiu processar.

6. Conclusão

Os experimentos realizados demonstraram a importância da configuração adequada dos parâmetros de rede e virtualização para otimizar as transferências de alto desempenho em sistemas virtualizados. A análise comparativa dos dois cenários revelou que a remoção das mitigações de CPU contra vulnerabilidades de execução especulativa, embora implique níveis menores de segurança, resultou em um aumento significativo de até 67% na vazão de testes UDP com apenas um fluxo, utilizando, inclusive, somente dois núcleos de CPU para a realização do experimento, como foi o caso do teste ”rx1/tx1”do Cenário 2. Além disso, as técnicas de *tunings* utilizadas provaram ser estratégias eficazes para aprimorar a eficiência da transmissão de dados em ambientes virtualizados.

Os resultados obtidos fornecem diretrizes para a otimização do desempenho em redes virtualizadas, especialmente para aplicações que exigem baixa perda de pacotes e alta vazão de dados. Estudos futuros podem explorar o impacto de outras configurações avançadas, como diferentes algoritmos de balanceamento de carga e técnicas de *offloading*, além de investigações sobre a segurança das configurações utilizadas.

Referências

Al-hamadani, A. T. H. and Lencse, G. (2021). A survey on the performance analysis of ipv6 transition technologies. *Acta Technica Jaurinensis*, 14(2):186–211.

ANDRADE, A. M. (2021). Análise da aderência do testbed mentored para a condução de experimentos de ataque distribuído de negação de serviço na iot.

Gál, Z., Kocsis, G., Tajti, T., and Tornai, R. (2021). Performance evaluation of massively parallel and high speed connectionless vs. connection oriented communication sessions. *Advances in Engineering Software*, 157:103010.

Hanemann, A., Boote, J. W., Boyd, E. L., Durand, J., Kudarimoti, L., Łapacz, R., Swany, D. M., Trocha, S., and Zurawski, J. (2005). Perfsonar: A service oriented architecture for multi-domain network monitoring. In *Service-Oriented Computing-ICSOC 2005: Third International Conference, Amsterdam, The Netherlands, December 12-15, 2005. Proceedings 3*, pages 241–254. Springer.

Hardin, B., Comer, D., and Rastegarnia, A. (2023). On the unreliability of network simulation results from mininet and iperf. *International Journal of Future Computer and Communication*, 12(1).

Kempf, M., Jaeger, B., Zirngibl, J., Ploch, K., and Carle, G. (2024). Quic on the fast lane: Extending performance evaluations on high-rate links. *Computer Communications*, 223:90–100.

Lei, J., Munikar, M., Suo, K., Lu, H., and Rao, J. (2021). Parallelizing packet processing in container overlay networks. *EuroSys 2021*.

Mazloum, A., Kfoury, E., AlSabeh, A., Gomez Gaona, J. A., and Crichigno, J. Enhancing visibility on a science dmz with p4-perfsonar. *Available at SSRN 5210525*.

Schwarz, M., Tierney, B., Vasu, K., Dart, E., Rothenberg, C. E., Bezerra, J., and Valcy, I. (2024). Recent linux improvements that impact tcp throughput: Insights from r&e networks. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 775–784. IEEE.

Vetter, M., Vetter, F., Stanton, M., Moura, A., Machado, I., Lopes Melo, E. T., Rhoden, G. E., Pescador, R., Brandtner, P., and Cordeiro, L. (2014). Monipê: um serviço de monitoramento de desempenho de redes usando soluções em hardware de baixo custo e virtualização de infraestrutura.