

Paulo Henrique Gonçalves Rocha
Orientador: Francisco Airton Pereira da Silva

Avaliação de Desempenho de Placas de Rede 10GbE: Uma Estratégia Baseada em Análise de Sensibilidade

Picos - PI
8 de outubro de 2018

Paulo Henrique Gonçalves Rocha
Orientador: Francisco Airton Pereira da Silva

Avaliação de Desempenho de Placas de Rede 10GbE: Uma Estratégia Baseada em Análise de Sensibilidade

Modelo de Trabalho de Conclusão de Curso em Bacharelado em Sistemas de Informação na Universidade Federal do Piauí. Este modelo está em conformidade com as normas ABNT.

Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
8 de outubro de 2018

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

R672a

Rocha, Paulo Henrique Gonçalves

Avaliação de desempenho de placas de rede 10GbE:
uma estratégia baseada em análise de sensibilidade / Paulo
Henrique Gonçalves Rocha.– 2018.

CD-ROM : il.; 4 ¾ pol. (50 f.)

Trabalho de Conclusão de Curso (Curso Bacharelado em
Sistemas de Informação) – Universidade Federal do Piauí, Picos,
2018.

Orientador(A): Prof. Dr. Francisco Airton Pereira da Silva

1. NIC. DoE. 2. Framework E/S. Pareto. 3. Sensibilidade.
4. Provedor. I. Título.

CDD 004.678

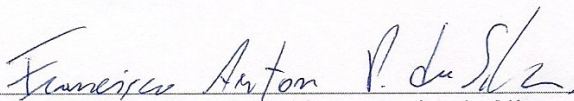
AVALIAÇÃO DE DESEMPENHO DE PLACAS DE REDE 10GBE:
UMA ESTRATÉGIA BASEADA EM ANÁLISE DE SENSIBILIDADE

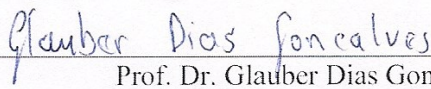
PAULO HENRIQUE GONÇALVES ROCHA

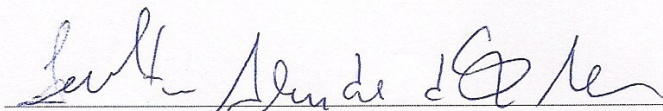
Monografia aprovada como exigência parcial para obtenção do grau de
Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 26 de novembro de 2018


Prof. Dr. Francisco Airton Pereira da Silva
Orientador


Prof. Dr. Glauber Dias Gonçalves
Membro


Profa. Esp. Ivenilton Alexandre de Souza Moura
Membro

Agradecimentos

Agradeço a Deus acima de tudo, por ter me abençoado com saúde, vontade e força para que eu pudesse realizar este trabalho.

Agradeço também a Universidade Federal do Piauí pela oportunidade de realizar este curso.

A Virtex Telecom pelo apoio dado, oferecendo equipamentos e conhecimentos necessários para a realização deste trabalho e também por acreditar na pesquisa científica.

Ao professor Dr. Francisco Airton, pela orientação, paciência, confiança e incentivo.

A todos os meus professores, que foram importantes não só construção do meu conhecimento, como também foram fontes de inspiração e valores que carregarei para sempre.

A toda minha família em especial a meu Pai Rocha, minha Mãe Hilda, minha irmã Patrícia e meu primo Samuel por sempre estarem ao meu lado oferecendo muito amor, confiança e união. Um agradecimento especial a meu primo Jederilson que me auxiliou neste trabalho.

Aos meus amigos André, Davi, João Marcos, Tácio, Wildyson, Renésio, Eduardo, Brena, Jonycássio, Walef, Naara, Douglas, Tomaz, Estevão, Milton, Matheus Lima, Aroldo, Matheus, Rafael Pedro e Tiago que estiveram presentes na minha vida acadêmica e no meu dia a dia durante os últimos anos.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

"Empenhar-se ativamente para alcançar determinado objetivo dá à vida significado e substância. Quem quiser vencer deve aprender a lutar, perseverar e sofrer".

Bruce Lee

Resumo

Com a evolução dos meios de comunicação, cada vez mais dados são transmitidos pelas redes de computadores. Os computadores possuem placas de rede que são componentes eletrônicos obrigatórios para realizar a comunicação entre máquinas. Existe uma variedade de placas com capacidades distintas, desde 100MbE (Megabit Ethernet) até 40GbE, ou mais. A placa de 10GbE é uma das placas mais utilizadas por companhias provedoras de Internet. Frameworks de captura e processamento de pacotes de alta velocidade são utilizados, conjuntamente com as placas de rede. Os frameworks são utilizados a fim de processarem grandes quantidades de pacotes, sem descartá-los. No entanto, é difícil saber qual combinação de placa e framework adotar ao se analisar diversos fatores de desempenho. Gerentes de infraestruturas de redes não possuem ao seu alcance estudos, que os auxiliem na tomada de decisão de quais placas e frameworks adotar de acordo com as suas necessidades. O presente trabalho apresenta uma análise de sensibilidade avaliando o impacto de duas marcas de placas de rede com capacidade 10 GbE e dois frameworks de captura de pacotes. O objetivo é indicar em relação a qual métrica cada combinação tem impacto mais significativo. Diferentes combinações destacaram-se em cenários específicos. A placa Chelsio juntamente com o framework Netmap foi capaz de oferecer uma taxa de vazão maior para os pacotes. A placa Chelsio e o framework PF_Ring exigem menos recursos computacionais para processar pacotes menores. Para processar pacotes maiores a placa Intel e o framework Netmap são menos exigentes. Em relação aos Pacotes Descartados todas as combinações apresentaram resultados semelhantes. Diferentes combinações foram indicadas tornando este trabalho uma ferramenta de auxílio para os profissionais de rede.

Palavras-chaves: NIC. DoE. Framework E/S. Pareto. Sensibilidade. Provedor

Abstract

With the evolution of the media, more and more data are transmitted through computer networks. Computers have network cards that are mandatory electronic components to perform communication between machines. There are a variety of cards with different capacities, from 100MbE (Megabit Ethernet) up to 40GbE, or more. The 10GbE board is one of the boards most commonly used by Internet companies. High-speed packet capture and processing frameworks are used in conjunction with network adapters. Frameworks are used in order to process large quantities of packages without discarding them. However, it is difficult to know which board and framework combination to adopt when analyzing various performance factors. Network infrastructures managers do not have at their disposal studies that help them in deciding which boards and frameworks to adopt according to their needs. The present work presents a sensitivity analysis evaluating the impact of two network card brands with 10 GbE capacity and two packet capture frameworks. The goal is to indicate which metric each combination has the most significant impact. Different combinations have stood out in specific scenarios. The Chelsio board, together with the Netmap framework, was able to offer a higher flow rate for the packages. The Chelsio board and the PF_Ring framework require fewer computational resources to process smaller packets. To process larger packages, the Intel board and the Netmap framework are less demanding. In relation to the Discarded Packages all combinations showed similar results. Different combinations were indicated making this work an aid tool for network professionals.

Lista de ilustrações

Figura 1 – Exemplo de Topologia de Monitoramento de um Provedor ¹	17
Figura 2 – Placa de Rede Intel X520-DA2 ²	18
Figura 3 – Placa de Rede Chelsio T520-SO-CR ³	18
Figura 4 – Estrutura do Framework Netmap	19
Figura 5 – Arquitetura do Framework PF_Ring	20
Figura 6 – Gráfico de Pareto com Efeitos Padronizados	23
Figura 7 – Gráfico de Efeitos Principais da Temperatura na Resistência à Tração	24
Figura 8 – Gráfico de Interações Sinérgica e Antagônica	24
Figura 9 – Cenário Físico dos Testes	31
Figura 10 – Influência dos Fatores na Acurácia (Throughput).	34
Figura 11 – Taxa Média de cada Nível dos Fatores	34
Figura 12 – Interação dos Níveis dos Fatores	35
Figura 13 – Influência dos Fatores na Acurácia (Uso de CPU).	36
Figura 14 – Percentual Médio de cada Nível dos Fatores	37
Figura 15 – Interação dos Níveis dos Fatores	37
Figura 16 – Influência dos Fatores na Acurácia (Perda de Pacote).	38
Figura 17 – Percentual Médio de cada Nível dos Fatores	39
Figura 18 – Interação dos Níveis dos Fatores	40

Lista de tabelas

Tabela 1 – Trabalhos Relacionados	26
Tabela 2 – Fatores e Níveis dos DoE's	31
Tabela 3 – Combinação dos Fatores	32
Tabela 4 – Melhores Combinações Obtidas	40

Lista de abreviaturas e siglas

COTS	Commercial-Off-The-Shelf Software
CPU	Central Process Unit
DMA	Direct Memory Access
DoE	Design of Experiments
DPDK	Data Plane Development Kit
E/S	Entrada e Saída
Gb	Gigabyte
GbE	Gigabit Ethernet
Gb/s	Gigabits por Segundo
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IP	Internet Protocol
NIC	Network Interface Card
PoP	Point of Presence
PPS	Packets per Second
RFC	Request for Comments
QoS	Quality of Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Sumário

1	Introdução	13
1.1	Definição do Problema	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Organização do Trabalho	15
2	Referencial Teórico	16
2.1	Análise de Pacotes	16
2.2	Placas de Rede 10GbE	17
2.3	Frameworks de Captura de Pacotes	19
2.3.1	Netmap	19
2.3.2	PF_Ring	20
2.4	Metodologia <i>Design of Experiments</i>	21
2.4.1	Gráfico de Pareto	22
2.4.2	Gráfico de Efeitos Principais	22
2.4.3	Gráfico de Interação	23
3	Trabalhos Relacionados	26
4	Análise de Sensibilidade	30
4.1	Ambiente de Testes	30
4.2	Resultados Obtidos	33
4.2.1	Throughput	33
4.2.2	CPU	35
4.2.3	Pacotes Descartados	38
4.2.4	Resumo dos Resultados	40
5	Conclusão	42
6	Publicações	43
	Referências	44
	Apêndices	48
	APÊNDICE A Script de Medição da CPU no Framework Netmap	49

APÊNDICE B Script de Medição da CPU no Framework PF_Ring 50

1 Introdução

Um dos produtos do crescimento das redes de computadores foi o surgimento da *Internet*. A Internet é uma rede de computadores que interconecta centenas de milhões de dispositivos de computação ao redor do mundo (KUROSE; ROSS, 2013). Com o surgimento da *Internet* e a sua conseqüente popularização, as pessoas podem comunicar-se e compartilhar dados a todo momento.

À medida que a rede se tornava mais popular, a quantidade de conteúdo só aumentava e os serviços oferecidos ficaram mais complexos. Dentre os serviços, destacaram-se *sites* de pesquisa, comércio eletrônico (*e-commerce*), bancos online (*bankline*), serviços públicos governamentais (*e-government*), entre vários outros. Toda a comodidade proporcionada por esses serviços online, somente repercutiu no crescimento desenfreado da rede com quantidades cada vez mais expressivas de usuários (BRITO, 2018).

Com o aumento gradativo de dados transferidos e compartilhados, os provedores de *Internet* precisam se aperfeiçoar a fim de garantir um serviço de boa qualidade. O tráfego global da *Internet* em 2021 será equivalente a 127 vezes o volume de toda a *Internet* global em 2005. Ao redor do mundo, o tráfego da *Internet* chegará a 30 GB *per capita* até 2021, acima dos 10 GB *per capita* em 2016. O número de dispositivos conectados a redes IP será três vezes maior que a população global em 2021 (CISCO, 2017).

De acordo com uma pesquisa da (STATISTA, 2018b) há uma previsão de crescimento do volume de dados do tráfego IP do consumidor mundial até 2021. Em 2021, o tráfego IP global consumido deverá atingir 232.655 petabytes por mês a uma taxa de crescimento anual composta de 24%. O tráfego IP de Internet consumido totalizou 52.678 PB por mês em 2016 e deverá aumentar para 127.152 PB mensais em 2020. O tráfego IP consumido via dados móveis foi de 5.953 PB por mês em 2016. Na América Latina em 2016, o tráfego atingiu 532 petabytes por mês. Espera-se que esse número aumente para 1,65 exabytes até 2021 (STATISTA, 2018a).

O mercado domiciliar de internet no Brasil é atendido prioritariamente pelas pequenas empresas provedoras de acesso à Internet: 94% daquelas com 1 a 9 pessoas trabalhando, e 92% e 91% daquelas com 10 a 19 e 20 a 49, respectivamente (entre os provedores de grande porte, aqueles com 250 a 499 e mais de 500 pessoas trabalhando, 56% e 52%, respectivamente, atendem esse mercado). Nesse sentido, os pequenos provedores desempenham um papel de fundamental importância para o desenvolvimento da Internet no país. Os pequenos provedores são importantes no que se refere ao atendimento à população das regiões mais afastadas dos grandes centros urbanos (CETIC.BR, 2016).

A camada de transporte da pilha de protocolos TCP/IP, juntamente com o *backbone*, é onde está o principal "gargalo" das operadoras de telecomunicações do Brasil. O investimento nestas camadas está aquém da demanda e, sem uma estrutura sólida neste sentido,

uma provedora não consegue crescer. O desequilíbrio causado por essa falta de estrutura é percebido na degradação dos serviços a longo prazo. Por outro lado, da mesma forma que o investimento é necessário, surgem as dificuldades de capacidade de investimento das empresas (BRITO, 2016).

Os pequenos e médios provedores cada vez mais precisam ir em busca de equipamentos melhores, na qual apresentam altos custos de aquisição. Portanto encontrar soluções mais baratas e que atendam suas demandas de forma mais eficiente é um dos problemas enfrentados por essas empresas.

As interfaces de rede (NIC) com taxas de dados de 1 GbE e 10 GbE (Gigabit Ethernet) são comuns em sistemas conectados a *Internet* (EMMERICH et al., 2015). Toda infraestrutura de rede deve ser avaliada visando uma adequada prestação de serviço, utilizando técnicas de análise. *Sniffing* é uma técnica voltada ao monitoramento e análise de tráfego em rede. Monitorar e observar o tráfego utilizando técnicas como o *sniffing* é um desafio quando se lida com taxas de transmissão e recepção de dados da ordem de 10 Gb/s ou mais.

A tarefa de monitorar um tráfego elevado com ferramentas de *sniffing* é exigente, sendo necessário aplicar poder computacional significativo para tal tarefa (MORENO et al., 2015a). Aquisições de placas de rede com transferências 10GbE ou mais se tornam uma solução adotada pelos provedores. As placas de rede apesar de possuírem capacidade para receber mais pacotes, possuem limitações para processar uma grande quantidade de pacotes. Com a limitação para tratar milhões de pacotes, as placas acabam descartando-os, o que prejudica um estudo da rede. Os frameworks de análise de pacotes em altas velocidades não só capturam pacotes mais rapidamente, como também capturam pacotes com mais eficiência. São exemplos bem conhecidos de frameworks de captura de pacotes: **Netmap** (RIZZO, 2012a), **PF_Ring** (DERI; RING, 2011; ORTEGA, 2015), **DPDK** (INTEL, 2014), entre outros.

Em uma análise de desempenho com placas de rede é importante utilizar um filtro de pacotes de alta velocidade para poder coletar as métricas, com uma perda pequena de informações. A análise do tráfego com perda pequena de pacotes, é importante na compreensão de todos os aspectos e peculiaridades do tráfego. Assim, os administradores de rede podem tomar decisões com mais confiança, pois terão mais informações ao seu dispor.

1.1 Definição do Problema

Apesar de serem tecnologias importantes na medição de qualidade dos serviços de rede, há dificuldades em encontrar estudos que avaliam essas ferramentas. Pequenos provedores precisam dispor de placas de rede 10GbE para suportar os seus respectivos tráfegos e o uso de frameworks é importante no processo de análise e monitoramento. Nenhum trabalho

pesquisado até o momento realizou uma avaliação de quais fatores possuem mais impacto no tráfego de pacotes de uma rede. Os poucos trabalhos que realizam avaliações isoladas de cada tecnologia, não consideraram o efeito de múltiplos fatores conjuntamente. Para pequenos e médios provedores, saber quais Placas de Rede e Framework atendem de forma eficiente as suas necessidades é importante. Já que as companhias provedoras não possuem muitos recursos para serem investidos em poder computacional.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é auxiliar gerentes de infraestruturas computacionais na escolha de placas de rede e framework de processamento de tráfego em situações específicas, oferecendo um estudo de desempenho detalhado sobre tais combinações.

1.2.2 Objetivos Específicos

1. Identificar o impacto dos fatores no Throughput obtido em cada combinação (Placa e Framework).
2. Identificar o impacto dos fatores no Uso da CPU obtido em cada combinação (Placa e Framework).
3. Identificar o impacto dos fatores no Percentual de Pacotes descartados em cada combinação (Placa e Framework).
4. Identificar as interações entre os fatores nas variáveis respostas (Throughput, CPU e Pacotes Descartados).

1.3 Organização do Trabalho

O restante deste trabalho está organizado nos seguintes capítulos: o Capítulo 2 (Referencial Teórico) apresenta uma visão geral sobre os conteúdos abordados nesse trabalho; o Capítulo 3 (Trabalhos Relacionados) apresenta outros trabalhos que realizam avaliação de desempenho em placas ou frameworks de captura de pacotes; o Capítulo 4 (Desenvolvimento) apresenta o ambiente de testes montado e o funcionamento de seus componentes, bem como a análise de sensibilidade realizada sobre ele e, por fim, o Capítulo 5 (Conclusão) apresenta as considerações finais, a contribuição e trabalhos futuros.

2 Referencial Teórico

Neste capítulo será apresentado uma visão geral sobre o que são os frameworks de captura de pacotes em altas velocidades e a sua importância. Abordando também a placas de rede ou NIC, as tecnologias utilizadas no trabalho e as suas principais características. Este capítulo ainda trata da análise de sensibilidade e sua importância em várias áreas.

2.1 Análise de Pacotes

A análise de tráfego em tempo real permite investigar o fluxo de dados compartilhados entre dispositivos comunicantes em um dado momento. Essa investigação permite identificar uma série de problemas que por vezes não são mostrados pelas aplicações, estações de trabalho ou equipamentos. Alguns exemplos de problemas encontrados são: anomalias na rede, pontos de bloqueio, equipamentos e cabeamentos defeituosos, mensagens de sistema não mostradas pelas aplicações, falhas de segurança, tráfego de vírus na rede, pontos de bloqueio na rede, entre outros. (FILHO, 2013)

Este assunto de análise de pacotes está bastante presente na literatura com muitos trabalhos se destacando (PARK; WILLINGER, 2000; BENSON; AKELLA; MALTZ, 2010; MORENO et al., 2015b; LEE; LEVANTI; KIM, 2014). Medições detalhadas de tráfego são necessárias para avaliar os requisitos de capacidade e projetar de forma eficiente a rede (FRALEIGH et al., 2003). Os tópicos de pesquisa que podem se beneficiar do monitoramento em nível de pacote são:

- Desenvolver modelos de tráfego que permitam os operadores de rede determinar a quantidade de provisionamento de recursos necessária em sua rede (BARAKAT et al., 2002).
- Avaliar os trade-offs entre diferentes níveis de granularidade no roteamento e estudar a dinâmica do tráfego entre os POP's (*Point of Presence*) (MEDINA et al., 2002; BHATTACHARYYA; DIOT; JETCHEVA, 2001).
- Desenvolvimento de algoritmos para detectar anomalias de rede, como ataques de negação de serviço e loops de roteamento (HENGARTNER et al., 2002).
- Estudar o desempenho do TCP e identificar onde o congestionamento está ocorrendo na rede (JAISWAL et al., 2007).
- Avaliação da capacidade da rede de oferecer suporte a novos serviços de valor agregado, como telefonia e qualidade de serviço (QoS) (BOUTREMANS; IANNACCONE; DIOT, 2002).

Uma técnica utilizada no processo de análise de tráfego é o *sniffing*. O *sniffing* trata-se de uma técnica de captura de pacotes de rede. Quando os pacotes são transmitidos da origem para o destino, eles percorrem vários dispositivos intermediários. Um nó cuja NIC está definida no modo promíscuo recebe todas as informações percorridas na rede. Cada NIC possui um endereço físico diferente e exclusivo de outra rede. Quando o pacote chega ao NIC, o endereço de hardware do frame corresponde ao endereço físico que o NIC possui, mas se for definido em modo promíscuo, todos os pacotes chegarão a essa NIC. Quando a NIC aceita pacotes, os pacotes são copiados para a memória do driver e depois passam para o kernel e o kernel passa para o aplicativo de análise (KING, 2002). Um exemplo de topologia em que um programa de *sniffing* é utilizada dentro de um provedor é demonstrada na Figura 1.

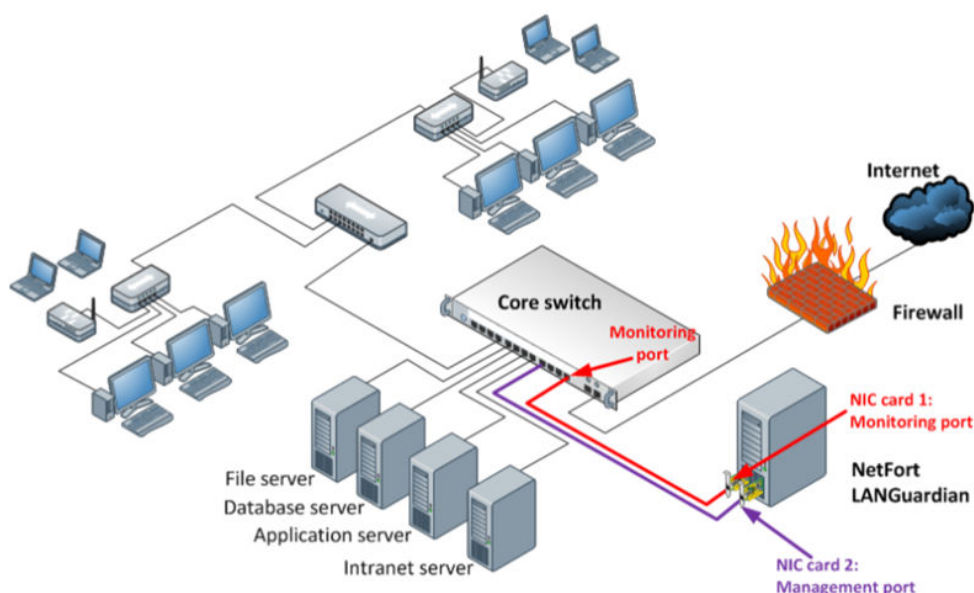


Figura 1: Exemplo de Topologia de Monitoramento de um Provedor ¹.

Como demonstrado no Figura 1, em geral um dispositivo é destinado como um servidor que recebe o tráfego. Esses servidores são equipados com NIC's de 10GbE ou mais, para suportar a quantidade de pacotes. O servidor de monitoramento está conectado diretamente a um *switch* ou roteador, que encaminha o tráfego para o mesmo. Por fim dentro desse servidor um programa de captura de pacotes envia as informações do tráfego ao administrador de rede do provedor.

2.2 Placas de Rede 10GbE

10 Gigabit Ethernet (10GE, 10GbE ou 10 GigE) é um grupo de tecnologias de redes de computadores para transmissão de frames Ethernet com frequência de 10 gigabits por

¹ <http://idijournal.com/network-asset-diagram.html>

segundo. Foi definida inicialmente pelo padrão IEEE 802.3ae-2002. Diferente dos padrões de Ethernet anteriores, no 10 Gigabit Ethernet é definido apenas links ponto-a-ponto full-duplex geralmente conectados por switches de rede (PALMER, 2012).

Para os fins deste trabalho foram utilizadas as placas 10GbE Intel X520-DA2 e Chelsio T520-SO-CR. As placas serão utilizadas nas medições, bem como na avaliação de seu impacto na qualidade do tráfego de pacotes. A placa de rede da Intel é demonstrada na Figura 2 e da fabricante Chelsio na Figura 3.

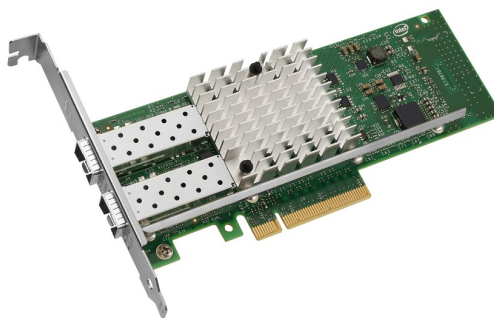


Figura 2: Placa de Rede Intel X520-DA2 ².

O adaptador de rede convergente Intel Ethernet X520 possui conectividade SFP+. A placa X520-DA2 utiliza o Controlador Ethernet 10 Gigabit Intel, denominado Ethernet 82599. O Ethernet 82599 pertence a terceira geração do controlador de rede de 10 GbE da Intel. A placa Intel x520-DA2 foi encontrada com um preço de \$179.99³.



Figura 3: Placa de Rede Chelsio T520-SO-CR ⁴.

² <https://ark.intel.com/pt-br/products/39776/Intel-Ethernet-Converged-Network-Adapter-X520-DA2>

³ https://www.amazon.com/dp/B0038KVYJ0/ref=psdc_13983711_t1_B01IR7SUEK

⁴ <https://www.chelsio.com/wp-content/uploads/2013/10/T520-SO-CR.pdf>

O T520-SO-CR da Chelsio é um adaptador de 10 Gigabit Ethernet Unified Wire, baseado na tecnologia de quinta geração (T5) da Chelsio. Possui porta dupla sem memória com uma interface de barramento de host PCI Express 3.0. A placa T520-SO-CR foi encontrada com um preço de \$219.00⁵

2.3 Frameworks de Captura de Pacotes

Nesta seção são apresentados os frameworks utilizados na pesquisa e na avaliação de desempenho. É apresentado a arquitetura de cada Framework, assim como o funcionamento dos mesmos. Suas principais estruturas internas são explicadas e exibidas.

2.3.1 Netmap

Netmap é um sistema que permite aos aplicativos de leitura e processamento de pacotes realizar um acesso muito rápido aos pacotes na interface de rede, tanto no lado do recebimento quanto do lado da transmissão (RIZZO; LANDI, 2011).

O Netmap foi implementado para vários adaptadores de rede de 1 e 10 Gbit/s. O Netmap utiliza drivers de Placa de Rede e um módulo de kernel modificados em uma tentativa de acelerar a captura de pacotes. O módulo do kernel implementa um buffer de anel. Um buffer de anel é uma memória pré-alocada que, uma vez preenchida, é sobrescrita desde o início de uma forma semelhante a um anel circular. O Netmap tem sua própria API para extrair dados do buffer de anel, na qual inclui uma versão modificada da biblioteca libpcap para suportar aplicativos que utilizam essa biblioteca, como o Wireshark.

A Figura 4 ilustra a arquitetura de funcionamento do Netmap.

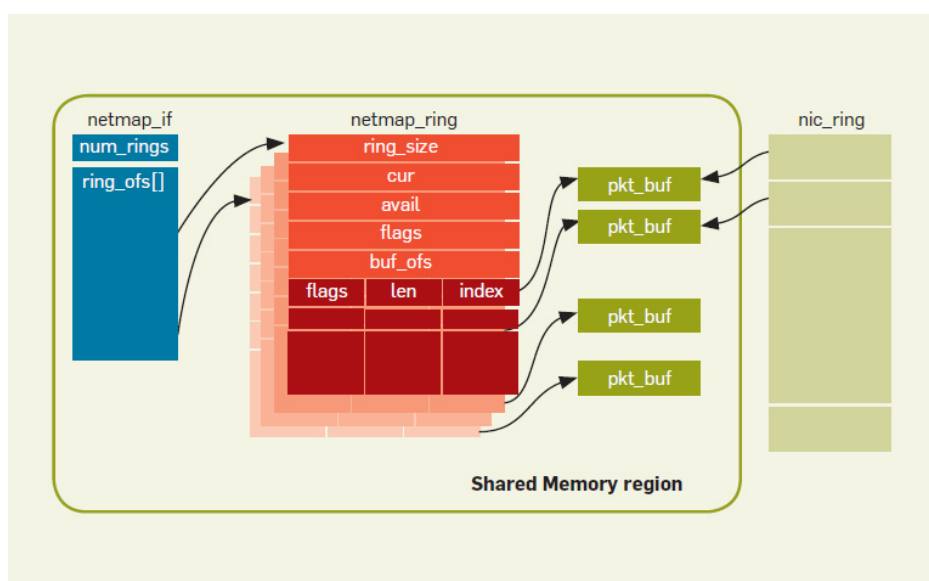


Figura 4: Estrutura do Framework Netmap

⁵ https://www.amazon.com/dp/B00FAU898K/ref=emc_b_5_t

O framework é construído em torno de uma região de memória compartilhada - acessível aos aplicativos kernel e de espaço de usuário - que contém *buffers* e descritores para todos os pacotes gerenciados por uma interface, como demonstrado na Figura 4. Os *buffers* de pacote têm um tamanho fixo, suficiente para armazenar um pacote de tamanho máximo. Isso não implica em fragmentação e em um formato de pacote fixo e simples. Descritores - um por *buffer* - são extremamente compactos (oito *bytes* cada) e armazenados em uma matriz circular que mapeia um-para-um para o anel NIC. Eles fazem parte de uma estrutura de dados chamada *netmap_ring*, que também contém alguns campos adicionais, incluindo o índice (*cur*) do primeiro buffer para enviar ou receber, e o número (*avail*) dos *buffers* disponíveis para transmissão ou recepção. Os *buffers* e descritores do Netmap são alocados apenas uma vez - quando a interface é ativada - e permanecem ligados à interface (RIZZO, 2012b).

2.3.2 PF_Ring

O PF_RING é um framework de distribuição gratuita criando uma pilha circular de cópias de pacotes diretas entre o núcleo do sistema e os aplicativos de leitura de pacotes a nível de usuário (ORTEGA, 2015). Como o Netmap, o PF_RING é uma estrutura que suporta captura de pacotes de até 10 Gbps. O PF_Ring utiliza um módulo de kernel e drivers de NIC modificados. Ele também faz uso de um buffer de anel e requer que os aplicativos usem sua API ou uma versão modificada do libpcap. Com a versão modificada da libpcap, programas de captura de pacotes como o Wireshark ou Tcpcap, podem fazer uso do PF_Ring. O PF_Ring e o Netmap por ignorarem o kernel perdem o processamento de pacotes no Netfilter e não há filtragem de pacotes envolvida. O PF_RING, ao contrário do Netmap, oferece solução em Filtragem de Pacotes de Hardware. A arquitetura do PF_Ring é ilustrada na Figura 5.

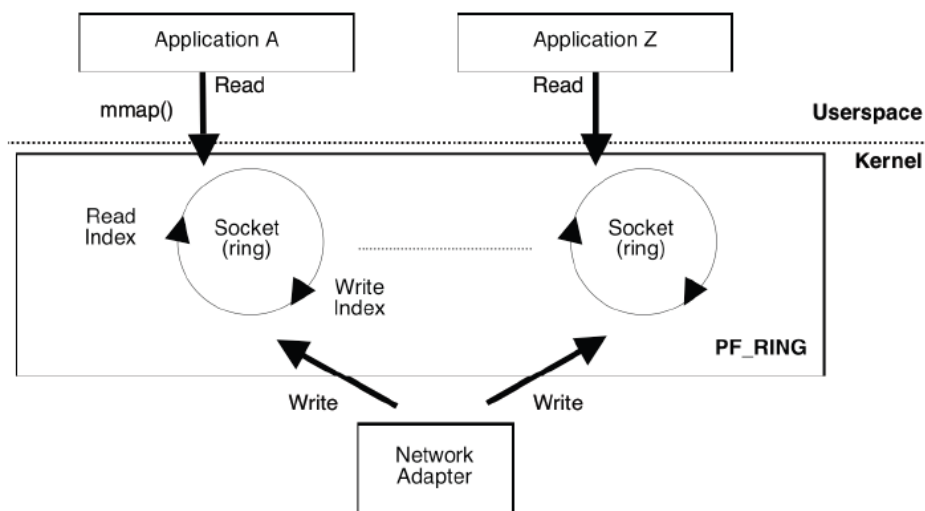


Figura 5: Arquitetura do Framework PF_Ring

O PF_RING utiliza buffers circulares nos adaptadores de rede para copiar e guardar os pacotes recebidos. O buffer é alocado quando um *socket* do PR_RING é criado, e desalocado quando o *socket* é desativado. Cada *socket* têm seu próprio buffer circular. Se um *socket* PF_RING estiver ligado a um *syscall* adaptado, esse adaptador será utilizado no modo só de leitura até o *socket* ser destruído. Sempre que um pacote é recebido do adaptador (geralmente via DMA, direct memory access), o driver passa o pacote para as camadas superiores. No caso do soquete do PF_RING, todos os pacotes de entrada são copiados para o anel do *socket* ou descartados, se necessário. Se o buffer estiver cheio, o pacote é descartado.

Os pacotes recebidos para adaptadores com *socket* PF_RING limitados, por padrão, não são encaminhados para as camadas superiores, mas são descartados depois de serem copiados para os anéis. Essa prática aumenta o desempenho geral, pois os pacotes não precisam ser manipulados pelas camadas superiores, mas apenas pelo anel. O *buffer* de anel de *socket* é exportado para aplicativos de espaço do usuário via função *mmap()*. Os aplicativos do espaço do usuário que desejam acessar o buffer precisam abrir o arquivo e, em seguida, chamar o *mmap()* para obter um ponteiro para o *buffer* circular. O *kernel* copia pacotes no anel e move o ponteiro de gravação para frente. Os aplicativos do espaço do usuário fazem o mesmo com o ponteiro de leitura. Novos pacotes recebidos sobrescrevem pacotes que foram lidos pelos aplicativos do espaço do usuário. A memória não é alocada ou desalocada por pacotes lidos/gravados no *buffer*, mas é simplesmente sobrescrita (DERI et al., 2004).

2.4 Metodologia *Design of Experiments*

O Design of Experiments (DoE -Projeto de Experimentos) está bem estabelecido e é aplicado com sucesso em vários campos da indústria, como produto para desenvolvimento de projetos, (GUNST, 1996), química (LAZIC, 2006) e engenharia de software (KUHN; REILLY, 2002). Existe um quantidade considerável de livros sobre a metodologia (SIEBERTZ; BEBBER; HOCHKIRCHEN, 2017; SELTMAN, 2012; ANTONY, 2006); Portanto, a metodologia geral aqui será descrita brevemente. Em cada parte da aplicação, será exemplificado cada caso metodológico para o estudo de caso.

O conceito geral de DoE foi criado por uma série de experimentos reais simulados em um sistema ou modelo de sistema sob observação. Em cada experimento, um ou vários parâmetros de projeto são alternados e o impacto no comportamento do sistema é avaliado. Quais parâmetro são alterados e como eles são alterados é definido usando um plano de experimento. O objetivo é obter o máximo de informações possíveis usando a menor quantidade de experimentos, e experimentos com o menor custo em termos de esforço computacional ou setup físico. O comportamento do sistema com base nas mudanças de parâmetros é observado usando conjuntos de saídas. No contexto do DoE,

as saídas podem ser referenciadas como “indicadores de performance”, os parâmetros de projeto como “fatores” e os valores das configurações de “Níveis”.

Para analisar o impacto de cada fator no sistema e a interação com os outros fatores, várias combinações de fatores precisam ser testadas e, portanto, exige um experimento. Devido à grande quantidade de combinações possíveis, este é um esforço muitas vezes inviável em termos de tempo e custos. O DoE oferece uma coleção de métodos - referidos como planos experimentais ou tabelas de design - para reduzir a quantidade de experimentos necessários para encontrar informações precisas com o menor número de experimentos possível. O tipo de plano usado depende do objetivo do experimento. Encontrar a tabela de projeto ideal, por exemplo, a menor quantidade de experimentos necessários para descrever o comportamento dos sistemas corretamente, tem sido objeto de intensa pesquisa. Usando um conjunto de métodos estatísticos de avaliação nos dados resultantes do experimento, o impacto, efeitos e interações dos fatores em relação aos indicadores de performance escolhidos é avaliado. Os resultados do experimento podem ser usados para formular um modelo substituto matemático, também chamado de metamodelo em disciplinas de engenharia (MILLER et al., 2014; THOMAS et al., 2014). Neste trabalho foi aplicado a metodologia DoE para avaliação da arquitetura proposta. Três tipos de gráficos são adotados usualmente em estudos com DoE: Gráfico de Pareto, Gráfico de Efeitos Principais e Gráfico de Interação (ESPEJO, 2006).

2.4.1 Gráfico de Pareto

O gráfico de Pareto permite detectar qual o efeito da interação de fatores é mais importante para o processo ou estudo de otimização de projeto com o qual se deve lidar. Exibe os valores absolutos dos efeitos e desenha uma linha de referência no gráfico. Qualquer efeito que ultrapasse essa linha de referência é potencialmente importante. Um gráfico de Pareto é construído como na figura 6, por exemplo. O gráfico mostra que os fatores B (tool geometry) e C (cutting angle) e a interação AC são mais importantes.

2.4.2 Gráfico de Efeitos Principais

O “Gráfico de Efeitos Principais” é o gráfico dos valores médios de resposta em cada nível de um parâmetro de projeto ou variável de processo. Pode-se usar esse gráfico para comparar a força relativa dos efeitos de vários fatores. O sinal e a magnitude de um efeitos principal nos diriam o seguinte:

- O sinal de um efeito principal nos diz a direção do efeito, ou seja, se o valor médio da resposta aumenta ou diminui.
- A magnitude nos diz a força do efeito.

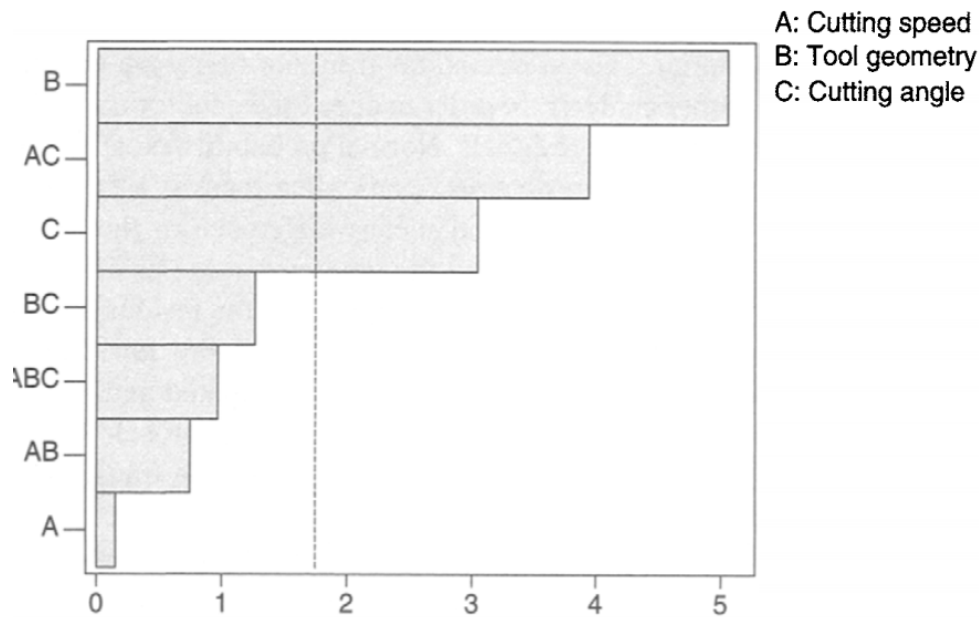


Figura 6: Gráfico de Pareto com Efeitos Padronizados

Se o efeito do parâmetro de projeto ou processo é positivo, isso implica que a resposta média é maior em nível alto do que em nível baixo da configuração do parâmetro. Em contraste, se o efeito for negativo, isso significa que a resposta média em nível baixo de configuração do parâmetro é maior que em nível alto. A figura 7 ilustra o efeito principal da temperatura na resistência à tração de uma amostra de aço. Como pode ser visto na figura, a resistência aumenta quando as configurações da temperatura variam para o nível baixo (ou seja, 1 para 1). O efeito de um parâmetro de projeto ou processo (ou fator) pode ser calculado matematicamente usando a simples equação 2.1:

$$E_f = \bar{F}_{(+1)} - \bar{F}_{(-1)} \quad (2.1)$$

Quando $\bar{F}_{(+1)}$ significa a resposta média na configuração de alto nível de um fator, e $\bar{F}_{(-1)}$ significa a resposta média na configuração de baixo nível de um fator.

2.4.3 Gráfico de Interação

O processo de "Interações", identifica efeitos importantes e determina sua magnitude, logo, as interações entre os efeitos são cruciais. As interações ocorrem quando o efeito de um fator depende do nível de outro fator. Uma medida de design sempre aborda vários fatores. Entender como esses fatores interagem em que magnitude permite para escolher a melhor combinação de medidas, revelando combinações de fatores com efeito cumulativo ou degradante. A interação $I_{A,B}$ entre dois fatores A e B pode ser calculada usando a equação 2.2.

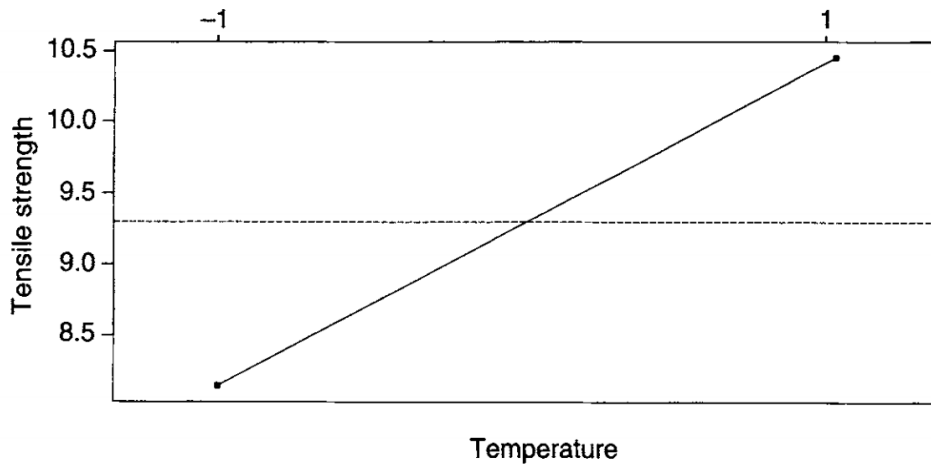


Figura 7: Gráfico de Efeitos Principais da Temperatura na Resistência à Tração

$$I_{A,B} = \frac{1}{2}(E_{A,B(+1)} - E_{A,B(-1)}) \quad (2.2)$$

O $E_{A,B(+1)}$ é o efeito do fator 'A' no nível alto do fator 'B' e $E_{A,B(-1)}$ é o efeito do fator 'A' no nível baixo do fator 'B'.

Para determinar se dois parâmetros de processo estão interagindo ou não, pode-se usar uma ferramenta gráfica simples, porém poderosa, chamada de gráficos de interação. Se as linhas no gráfico de interação forem paralelas, não haverá interação entre os parâmetros do processo. Isso implica que a mudança na resposta média do fator 'A' não depende dos níveis do fator 'B'. Por outro lado, se as linhas não são paralelas, existe uma interação entre os fatores. Quanto maior o grau de afastamento de ser paralelo, mais forte o efeito de interação. Para interação sinérgica, as linhas no gráfico não se cruzam. Por exemplo, a Figura 8(a) é um exemplo de interação sinérgica

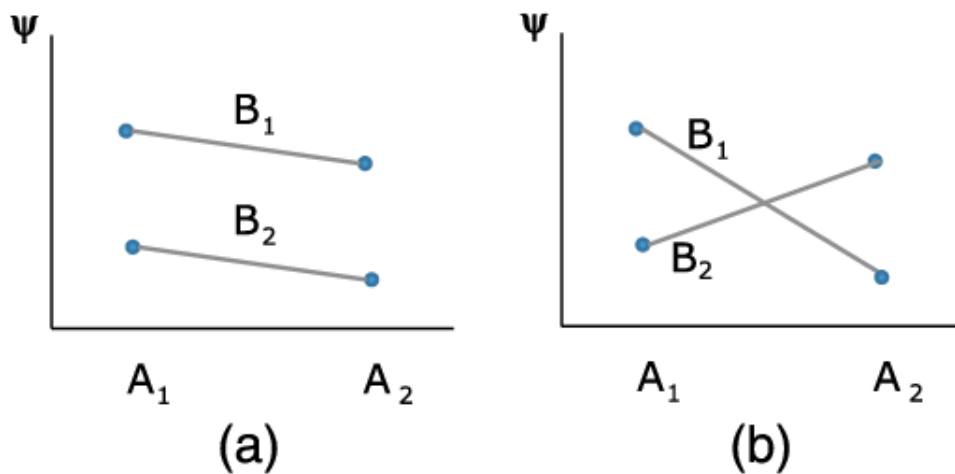


Figura 8: Gráfico de Interações Sinérgica e Antagônica

Na interação antagônica, as linhas no gráfico se cruzam. Isso pode ser ilustrado na Figura 8(b). Neste caso, a mudança na resposta média para o fator A no nível 'A₁' é alta

em comparação ao nível 'A₂'. As mudanças nos níveis do fator 'A' para a resposta média, indica uma dependência do fator 'A' em relação aos níveis do fator 'B'.

3 Trabalhos Relacionados

Este capítulo apresenta alguns trabalhos produzidos acerca de análise de desempenho de NIC's 10GbE ou frameworks de captura de pacote. Na apresentação dos trabalhos foi destacado as correlações com o trabalho proposto bem como as diferenças encontradas. Na Tabela 1 são exibidos os critérios de comparação dos trabalhos relacionados com o trabalho atual. Na Tabela 1 é também destacado em quais quesitos os trabalhos pesquisados, foram diferentes do presente trabalho.

Tabela 1: Trabalhos Relacionados

Trabalho	Análise de Sensibilidade	Métricas Avaliadas	Considerou NIC como fator?	Considerou Framework como fator?
Gallenmüller (2014)	Não	Throughput e CPU	Não	Sim
Larsen e Lee (2014)	Não	Throughput e Latência	Sim	Não
Ammendola et al. (2015)	Não	Latência e Bandwidth	Sim	Não
Barbette, Soldani e Mathy (2015)	Não	Throughput	Não	Sim
Barker (2015)	Não	Perda de Pacote	Não	Sim
Gallenmüller et al. (2015)	Não	Throughput, CPU e Latência	Não	Sim
Nurika et al. (2016)	Não	Throughput	Sim	Não
Kim e Na (2017)	Não	Throughput e Perda de Pacotes	Não	Sim
Marcuzzo e Santos (2017)	Não	Throughput e Latência	Não	Sim
Trabalho Proposto	Sim	Throughput, CPU e Perda de Pacote	Sim	Sim

O primeiro critério de avaliação dos trabalhos foi quais trabalhos fizeram um DoE visando uma análise de sensibilidade dos fatores em relação as métricas utilizadas. Este trabalho foi o único trabalho que realizou a análise de sensibilidade entre os demais trabalhos. O segundo critério foi as métricas utilizadas na avaliação de desempenho. O presente trabalho apresentou mais métricas analisadas em relação a maioria dos demais trabalhos. Por fim os demais critérios foram se os demais trabalhos utilizaram NIC's e Frameworks como fatores. Este trabalho foi o único que utilizou diferentes combinações de NIC e Framework.

No Trabalho de [Gallenmüller \(2014\)](#) um modelo é desenvolvido, demonstrando a relação fundamental entre custos e desempenho destes frameworks de processamento de pacotes. Esta aplicação é gradualmente modificada durante esta tese para demonstrar cenários de processamento de pacotes mais complexos. Assim, os fatores que influenciam a taxa de transferência, como a carga da CPU, a conexão da memória ou o número de pacotes processados simultaneamente, podem ser examinados mais de perto.

A semelhança deste do trabalho proposto com o de [Gallenmüller \(2014\)](#) é que o objetivo a ser alcançado em ambos é que os insights obtidos podem ser aplicados para realizar aplicações de processamento de pacotes. É escolhido a estrutura que melhor se adapte aos requisitos da aplicação. A diferença principal é que [Gallenmüller \(2014\)](#) não leva em conta diferentes placas para realizar a análise, se limitando apenas aos frameworks. O presente trabalho busca realizar uma análise de sensibilidade a fim de identificar com exatidão os fatores mais impactantes na avaliação de desempenho.

Em [Larsen e Lee \(2014\)](#) os autores analisam o estado atual dos avanços da arquitetura de E/S (Entrada e Saída) de alto desempenho e explora os benefícios e as limitações. Com base nas medições e análises, é demonstrado como a E/S é afetada por restrições de latência e throughput. Por fim, os autores sugerem opções para melhorar o desempenho de E/S, com base nessas medidas.

Levando em consideração a parte de análise de [Larsen e Lee \(2014\)](#) percebe-se que as diferenças entre o seu trabalho e este são as métricas analisadas e a ausência de uma análise de sensibilidade. Em [Larsen e Lee \(2014\)](#) os mesmos buscam comparar diferentes fatores em relação a placa de rede, sem a utilização de frameworks de captura de pacotes.

No trabalho de [Ammendola et al. \(2015\)](#) é apresentado o NaNet-10, uma placa de interface de rede de baixa latência. A NaNet-10 é dedicada a sistemas em tempo real, oferecendo quatro canais de E/S de 10 GbE. Os autores buscam avaliar o NaNet-10 e o NaNet-1, o NaNet-1 trata-se de um modelo semelhante ao NaNet-10, sendo o NaNet-10 uma atualização do NaNet-1.

O trabalho proposto nesta monografia mostra-se bem diferente do trabalho de [Ammendola et al. \(2015\)](#) na questão das métricas avaliadas e também do tipo de avaliação. [Ammendola et al. \(2015\)](#) realiza uma análise comparativa dessas placas, levando em consideração as métricas de Latência de *Bandwidth*. Nenhum framework de captura de pacotes foi analisado, e não foi realizada uma análise referente aos impactos nos fatores.

Em [Barbette, Soldani e Mathy \(2015\)](#) foi analisado e comparado quantitativamente vários frameworks de E/S de pacotes, mostrando sua superioridade ao encaminhamento baseado em kernel. Através de uma combinação de técnicas existentes e melhorias próprias, os autores derivaram princípios gerais modernos para o design dos frameworks processadores de pacotes.

O presente trabalho diferencia-se do trabalho de [Barbette, Soldani e Mathy \(2015\)](#) na questão da realização dos experimentos. Em [Barbette, Soldani e Mathy \(2015\)](#) eles avaliam a métrica de Throughput coletado, sendo que como fator é utilizado os frameworks de captura de pacote com diferentes configurações. Cada configuração é um fator diferente utilizado na medição. O presente trabalho com a adição da placa como fator apresenta mais combinações. Por fim este presente trabalho realiza uma análise de sensibilidade ao contrário de [Barbette, Soldani e Mathy \(2015\)](#), que se restringe a uma análise comparativa.

No trabalho de [Barker \(2015\)](#) é realizada uma análise de desempenho entre diferentes

frameworks. A análise realizada pelos autores, busca identificar qual framework apresenta um melhor desempenho na questão do descarte de pacotes. Por fim, no final, é apontado em que situação cada framework se destaca, e qual o melhor cenário que cada um deve ser utilizado.

O presente trabalho difere-se do trabalho de [Barker \(2015\)](#) no tipo de análise realizada. Em [Barker \(2015\)](#) é realizada uma comparação entre os frameworks com base na métrica "pacotes descartados". O trabalho proposto busca uma análise de quais fatores impactam mais as variáveis respostas, sendo realizado um DoE para tal tarefa. Este trabalho utiliza o fator NIC e Framework, ao contrário do trabalho [Barker \(2015\)](#) que utiliza apenas os Framework como fator.

O trabalho de [Gallenmüller et al. \(2015\)](#) são analisados vários frameworks para o pacote de alto desempenho E/S. É introduzido um modelo para estimar e avaliar o desempenho dessas estruturas de processamento de pacotes. Além disso, os autores analisaram o desempenho dos quadros mais proeminentes baseados em medições representativas em cenários de encaminhamento de pacotes. Por fim, foi analisado o *trade-off* entre o Throughput e a latência.

A diferença principal entre este trabalho e o trabalho de [Gallenmüller et al. \(2015\)](#) é a análise de sensibilidade dos fatores. Ambos os trabalhos apresentam métricas semelhantes, bem como o uso de frameworks como fatores. Em [Gallenmüller et al. \(2015\)](#) os autores não realizam medições com diferentes placas de rede como fatores.

O trabalho de [Nurika et al. \(2016\)](#) trata-se uma expansão de uma publicação, dos mesmos autores, sobre otimização simultânea de várias placas de rede em uma simulação de data center, usando algoritmo genético. Os autores realizaram diferentes configurações na placa de rede avaliada. Cada configuração da placa de rede é destinada a diferentes características de transmissão de dados. Nas medições realizadas foi registrada a taxa de transferência, comparando-a com a taxa de transferência da configuração padrão da placa de rede (não otimizada).

O trabalho de [Nurika et al. \(2016\)](#) busca uma análise comparativa do da placa de rede otimizada com a placa em seu estado não otimizado. O presente trabalho mostra-se diferente do trabalho de [Nurika et al. \(2016\)](#), pois utiliza diferentes frameworks de captura de pacotes como fator. O trabalho proposto realiza uma análise de sensibilidade com um maior número de métricas a serem analisadas.

O trabalho de [Kim e Na \(2017\)](#) fornece um método de implementação de comunicação unidirecional baseado em COTS (*Commercial-Off-The-Shelf Software*). Em seguida, foi verificado a vantagem do método de comunicação unidirecional baseado em COTS. Os autores implementaram uma amostra usando a Intel 82580 NIC e PF_RING. Por fim, foi apresentado uma possibilidade que pode contribuir para a realização da comunicação on-line por meio de experimentos sobre desempenho e confiabilidade.

A principal diferença entre o trabalho proposto e o trabalho de [Kim e Na \(2017\)](#)

é o uso da métrica de CPU e também o uso de diferentes placas de redes na análise deste trabalho. Em [Kim e Na \(2017\)](#) os autores utilizam uma placa de rede 10GbE com diferentes analisadores de pacotes, porém a placa não é usada como fator na pesquisa. Neste trabalho proposto uma análise de sensibilidade dos fatores é realizada e [Kim e Na \(2017\)](#) limita-se a uma análise comparativa.

O trabalho de [Marcuzzo e Santos \(2017\)](#) teve como objetivo apresentar uma visão geral sobre algumas das técnicas de aceleração de pacotes mais utilizadas. Dois frameworks onde essas técnicas são implementadas, tiveram o seu desempenho avaliado. As métricas medidas foram a vazão, latência e utilização de recursos e comparadas com a pilha de rede tradicional do Linux. A avaliação teve como objetivo demonstrar sua eficiência em um cenário com alto consumo de recursos de rede.

A avaliação de desempenho da utilizada em [Marcuzzo e Santos \(2017\)](#) foi semelhante em relação a este trabalho, no que se refere as métricas utilizadas. O trabalho de [Marcuzzo e Santos \(2017\)](#) realizou uma avaliação de desempenho comparativa, utilizando diferentes frameworks como fatores. O presente trabalho se difere, portanto, na utilização de frameworks e placas de rede como fatores na avaliação. Neste trabalho outra diferença notada em relação ao trabalho de [Marcuzzo e Santos \(2017\)](#) foi o DoE realizado e a análise de sensibilidade dos fatores.

4 Análise de Sensibilidade

Esta seção descreve os experimentos realizados para analisar os fatores que tiveram mais influência na acurácia das métricas analisadas através da realização de uma análise de sensibilidade utilizando as métricas: Throughput, Uso de CPU e Perda de Pacotes.

4.1 Ambiente de Testes

O ambiente de experimentos foi configurado como ilustra a Figura 9. Foram adotados dois computadores e um aparelho *Mikrotik CCR1036-8G-2S+*¹. Nos experimentos, o *Mikrotik* — também conhecido como *RouterBoard* — foi responsável pela geração do tráfego. A medida de envio do tráfego é PPS (pacotes por segundo), na qual o dispositivo gerava uma rajada de pacotes a cada segundo. A quantidade de pacotes por segundo de cada rajada individual depende do tamanho do pacote utilizado. Pacotes menores são gerados mais rapidamente em relação aos pacotes de tamanho maior, logo mais pacotes por segundo são gerados e enviados mais rapidamente pela *RouterBoard*.

O uso do *CCR1036-8G-2S+* é justificado pelos vários parâmetros de rede disponíveis em sua ferramenta de geração de tráfego, denominada *Traffic-generator*. Esses parâmetros de rede podem ser utilizadas e modificadas para testes, dando uma maior flexibilidade na hora de montar o tráfego a ser gerado. A *RouterBoard* também possui 2 portas que suportam tráfegos de 10Gb/s e que são compatíveis com o *Plugue sfp+*². O *Plugue sfp+* faz parte cabo óptico utilizado, que foi o meio físico empregado para transmitir as informações.

Em cada um dos computadores foi instalada uma placa de rede diferente. As máquinas de igual configuração possuem Processador *Intel i3* com 8GB de Memória RAM e Disco Rígido de 500GB, rodando o Sistema Operacional *Linux Ubuntu 16.04 LTS*. As placas de rede avaliadas foram especificamente a *Intel X520-DA2* e a *Chelsio T520-SO-CR*. Como mencionado anteriormente, os frameworks de leitura e captura de pacotes PF_Ring e Netmap foram adotados, dado a grande quantidade de trabalhos que os utilizam.

Para realizar a avaliação de desempenho, 3 métricas foram mensuradas, são elas: *Throughput*, Perda de Pacotes e Uso da CPU. Essas métricas são definidas na *RFC 2544*, que trata da padronização de testes em dispositivos de rede (BRADNER; MCQUAID, 1999), com exceção da métrica de Uso da CPU. A métrica de Uso da CPU está sendo utilizada, pois foi uma métrica de uso comum nos trabalhos pesquisados.

Na execução de um DoE, o primeiro passo é definir os fatores e níveis do experimento. Para poder realizar a medição das métricas foram definidos três fatores: Placa de Rede,

¹ <https://mikrotik.com/product/CCR1036-8G-2Splus>

² https://produto.mercadolivre.com.br/MLB-1086368142-sfpcisco-para-sfpmikrotik-cabo-twinax-1-metro-_JM

Figura 9: Cenário Físico dos Testes



Framework de Captura de Pacotes e Tamanho de Pacote. As métricas de Placa e Framework possuem dois níveis, que são as placas e os frameworks utilizados. A métrica Tamanho de Pacote possui 5 níveis diferentes, a saber: 64 *bytes*, 128 *bytes*, 256 *bytes*, 512 *bytes* e 768 *bytes*. As variações de tamanho do pacotes também são definidas na RFC 2544, bem como a utilização do protocolo UDP nos pacotes gerados.

A Tabela 2 apresenta os fatores e níveis escolhidos com o objetivo de identificar o fator com maior impacto em cada métrica coletada. Esses fatores e níveis foram utilizados para cada métrica, sendo que cada métrica analisada teve um DoE realizado.

Tabela 2: Fatores e Níveis dos DoE's

Fator	Nível I	Nível II	Nível III	Nível IV	Nível V
Placa	Chelsio	Intel	-	-	-
Framework	Netmap	Pf_Ring	-	-	-
Tamanho de Pacote (Bytes)	64	128	256	512	768

O próximo passo é combinar os fatores e níveis para definir as especificações do experimento. A Tabela 3 apresenta as possíveis combinações dos fatores e níveis das métricas analisadas. O desenho do experimento resultou em 20 combinações para cada DoE.

Para a realização dos experimentos foi utilizado o tempo padrão de 60 segundos, na qual a cada segundo uma rajada de pacotes era enviada e lida. Para cada combinação de níveis de fatores um teste individual de 60 segundos foi executado, coletando 60 medições para a respectiva combinação. Ao fim do teste de 60 segundos, as métricas eram mensuradas em cada combinação. As métricas Throughput e Uso de CPU, tiveram suas médias calculadas em cada combinação ao fim do teste, conforme a equação 4.1.

$$\bar{X} = \frac{1}{60} \sum_{i=1}^{60} x_i \quad (4.1)$$

Tabela 3: Combinação dos Fatores

Nº	Placa	Framework	Tamanho
1	Chelsio	Netmap	64
2	Chelsio	Netmap	128
3	Chelsio	Netmap	256
4	Chelsio	Netmap	512
5	Chelsio	Netmap	768
6	Chelsio	PF_Ring	64
7	Chelsio	PF_Ring	128
8	Chelsio	PF_Ring	256
9	Chelsio	PF_Ring	512
10	Chelsio	PF_Ring	768
11	Intel	Netmap	64
12	Intel	Netmap	128
13	Intel	Netmap	256
14	Intel	Netmap	512
15	Intel	Netmap	768
16	Intel	PF_Ring	64
17	Intel	PF_Ring	128
18	Intel	PF_Ring	256
19	Intel	PF_Ring	512
20	Intel	PF_Ring	768

Onde \bar{X} é a média final dos valores em cada combinação e x_i a quantidade de pacotes lidos em cada rajada. A métrica Pacotes Descartados foi calculada com base no percentual de perda dos pacotes, que foi calculado conforme as equações 4.2 e 4.3.

$$Dx = \sum_{i=1}^{60} tx_i - \sum_{i=1}^{60} rx_i \quad (4.2)$$

$$P = \frac{Dx \times 100}{\sum_{i=1}^{60} tx_i} \quad (4.3)$$

A expressão 'Dx' é a quantidade de pacotes descartados que é obtido através da diferença total entre tx_i que é a quantidade de pacotes enviados em uma única rajada e rx_i que é a quantidade de pacotes recebidos da rajada enviada. Por fim é calculado o percentual de pacotes descartados, representado pela expressão 'P', entre a quantidade total de pacotes enviados e a perda de pacotes obtida.

As métricas foram coletadas utilizando aplicativos provenientes dos próprios frameworks. No caso, quando o framework Netmap era utilizado nos experimentos, o programa, chamado *pkt-gen*³ era utilizado. Através do *pkt-gen*, o Throughput e a quantidade de pacotes eram coletados, da rajada de pacote recebido naquele segundo. O framework

³ <https://github.com/luigirizzo/netmap/tree/master/apps/pkt-gen>

PF_Ring possui um programa de leitura de pacotes que foi utilizado para medir as métricas Throughput e Perda de Pacotes, chamado *pfcount*⁴.

Na RouterBoard foi coletado a quantidade de pacotes enviados, enquanto o número de pacotes recebidos foi coletado pelos programas *pkt-gen* e *pfcount*. Com a quantidade pacotes enviados e recebidos, foi possível mensurar a quantidade de pacotes descartados. Por fim a métrica de Uso de CPU foi mensurado através de dois *scripts* desenvolvidos para tal finalidade. Foi desenvolvido um *script* específico, para realizar a coleta de acordo com o programa do framework utilizado. Os *scripts* são demonstrados no Apêndice A e Apêndice B.

4.2 Resultados Obtidos

Esta seção descreve os resultados obtidos de acordo com os dados coletados no ambiente de testes. As métricas: Throughput, Uso de CPU e Perda de Pacotes serão analisadas com base nos gráficos gerados. Nesta sessão serão interpretados os resultados e explicar os eventos ocorridos, extraíndo as informações pertinentes e descobertas durante os testes.

4.2.1 Throughput

A Figura 10 apresenta o gráfico de Pareto para os fatores relacionados ao Throughput. Neste estudo foi observado que os fatores com impactos mais elevados sobre a medição do Throughput foram: Tamanho de Pacote e Framework. Quando o impacto de um fator é dito alto, quer dizer que alterando o nível daquele fator são obtidos valores bem distintos.

O Tamanho do Pacote possui maior impacto dentre todos os fatores, portanto o Tamanho dos Pacotes de um tráfego de dados é determinante na confiabilidade da medição. Dado que se sabe os fatores com maiores impactos, o próximo passo é saber quais níveis tiveram melhores resultados.

A Figura 11 mostra o gráfico de Efeitos Principais para o fator Throughput. O gráfico representa a resposta média para cada nível de fator ligado por uma linha. No gráfico de efeitos principais se a linha está na horizontal, paralelo ao eixo x, indica a ausência de efeito. Quanto maior a inclinação da linha, maior é a magnitude do efeito principal.

Todos os fatores apresentaram efeitos em seus níveis, sendo o fator Placa o que obteve um menor grau de efeito entre seus respectivos níveis. A Placa de Rede da fabricante Chelsio apresentou uma taxa média de Throughput maior. Para atender a um tráfego com as mesmas especificações, a placa da Chelsio consegue atingir uma Taxa de Transferência maior que a placa da Intel.

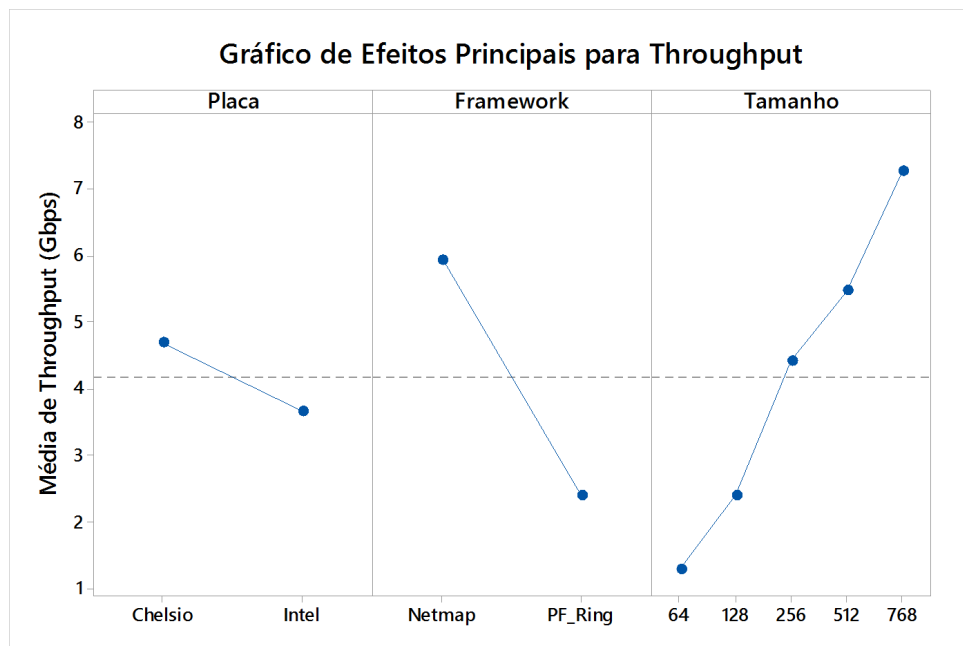
O Framework Netmap apresenta um Throughput médio maior que o Framework PF_Ring, isso significa que o Netmap consegue obter uma maior taxa de transferên-

⁴ https://github.com/avis/PF_RING/blob/master/userland/examples/pfcount.c

Figura 10: Influência dos Fatores na Acurácia (Throughput).



Figura 11: Taxa Média de cada Nível dos Fatores



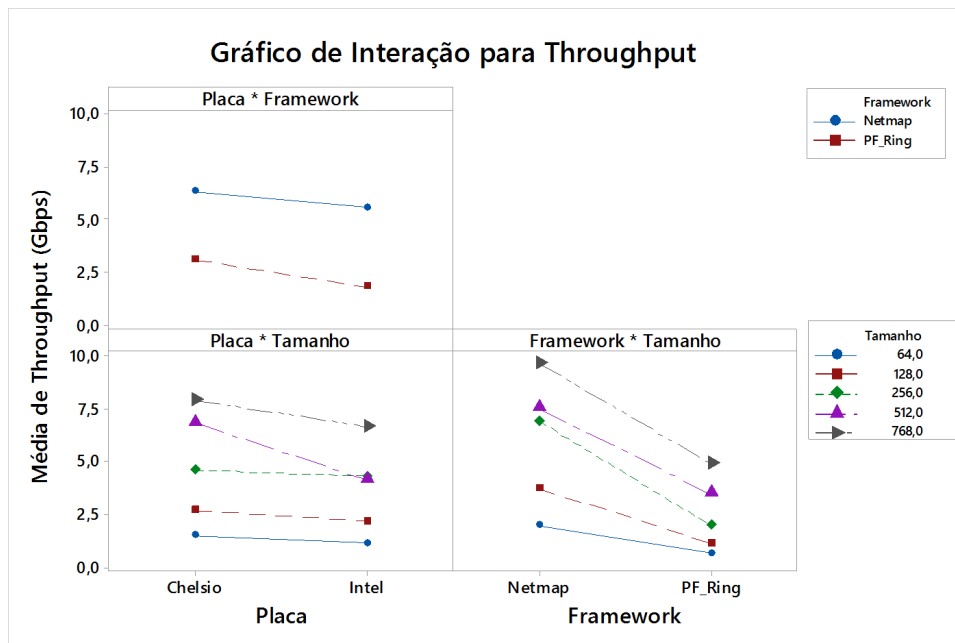
cia média de pacotes em relação ao PF_Ring. Com uma Taxa de Transferência maior, significa que mais pacotes podem ser lidos, o que significa menos pacotes descartados e por consequência um serviço de rede melhor.

Analisando os níveis de Tamanho de Pacote nota-se que quanto maior o tamanho deles, maior o Throughput utilizado. Quanto maior o pacote, mais dados são transportados e portanto maior a vazão ou Throughput necessário para transferi-los.

A fim de saber se a relação entre um fator com a variável resposta medida depende de outro fator categórico, utiliza-se o gráfico de Interações. A Figura 12 mostra os gráfico de

interações, para cada combinação de fatores e suas respectivas interações com a métrica Throughput.

Figura 12: Interação dos Níveis dos Fatores



Ao observar os gráficos nota-se que a Placa da Chelsio com o Framework Netmap obtêm a maior taxa de Throughput médio. Utilizando também a Placa Intel com o Framework Netmap é obtido um desempenho semelhante, porém inferior.

A placa Chelsio utilizando todos os níveis dos pacotes obteve um Throughput médio maior que a Placa Intel. Nota-se que os níveis de pacote 512 bytes e 768 bytes apresentam um grau de interação maior, logo as placas são sensíveis à variação desses níveis.

Ao analisar a relação entre o Throughput médio e o fator Framework observa-se que há interação de todos os níveis do fator Tamanho de Pacote. Com a interação ocorrendo o Throughput médio dos frameworks é dependente das variações dos níveis do fator Tamanho de Pacote. O Framework Netmap demonstrou uma taxa de Throughput médio superior ao Framework PF_Ring com diferentes combinações de Tamanho de Pacotes.

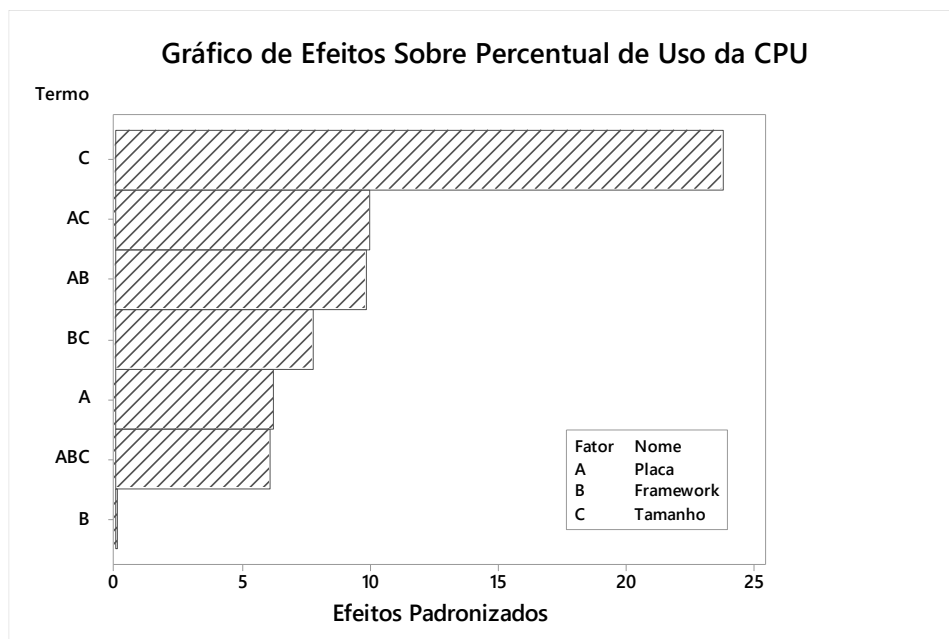
Portanto, um dispositivo equipado com a placa da Chelsio juntamente com o Framework Netmap tem mais capacidade de leitura de dados e por conseguinte menos pacotes são descartados. Essa combinação se mostra a mais adequada para redes que exigem altas taxas de vazão de dados para manter a qualidade de seus serviços.

4.2.2 CPU

A Figura 13 apresenta o gráfico de Pareto para os fatores relacionados ao Uso de CPU. O Tamanho do Pacote possui maior impacto dentre todos os fatores, logo o mesmo é determinante no impacto de consumo de dados da CPU. Os fatores Placa de Rede e Framework não apresentaram significância, no que diz respeito ao impacto ocasionado

na acurácia da métrica de Uso da CPU. Assim, como na métrica Throughput, o fator Tamanho de Pacote, continua sendo o fator mais impactante também na medição do percentual de Uso da CPU. Da mesma forma que da análise anterior o próximo passo é saber quais os resultados dos níveis de cada fator.

Figura 13: Influência dos Fatores na Acurácia (Uso de CPU).

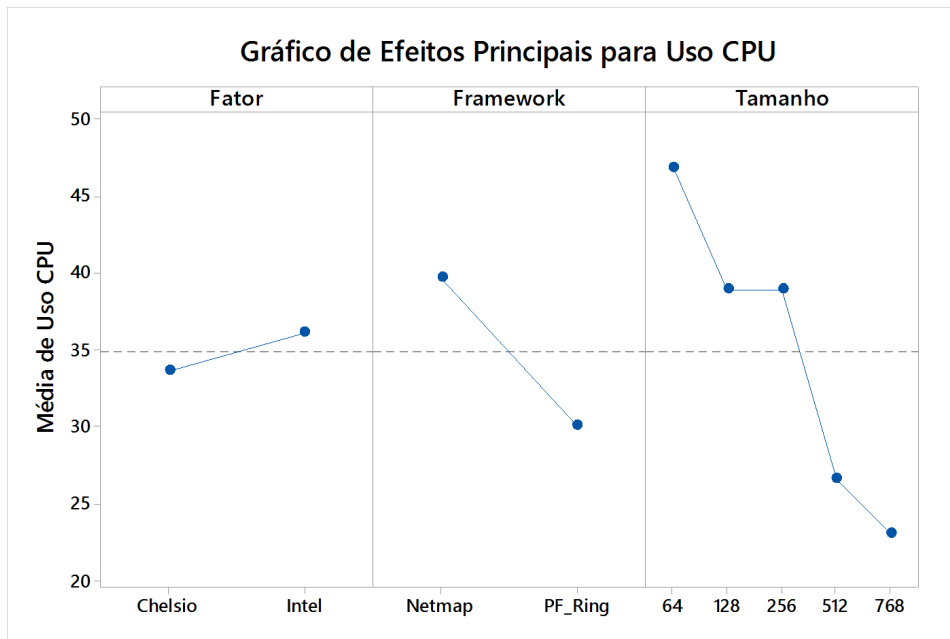


A Figura 14 mostra o gráfico de Efeitos Principais para o fator Uso de CPU. Como todos os fatores não apresentam linhas de ligação entre os níveis na horizontal, conclui-se que todos os níveis dos fatores afetam a resposta, ou seja, possuem efeito. Os fatores Framework e Tamanho de Pacote, apresentam uma escala de magnitude maior do efeito principal.

Analisando o gráfico nota-se que as Placas de Rede não apresentaram uma diferença grande de consumo. A Placa de Rede da Chelsio foi ligeiramente melhor, porém não há diferença significativa na variável resposta, em relação a Placa de Rede da Intel. O framework Netmap apresentou um consumo médio de CPU mais alto em relação ao framework PF_Ring. Os pacotes de tamanho menor apresentaram um consumo de CPU elevado, os pacotes de 64 bytes apresentaram uma média maior. Os pacotes de tamanho maior apresentaram baixa exigência de poder computacional. Cada pacote ao ser recebido pela Placa de Rede e lido pelo Framework tem o seu cabeçalho processado. Quanto mais pacotes chegam a interface de Rede mais CPU é exigida para processar o cabeçalho de cada pacote. Pacotes menores por chegarem em quantidades maiores exigem mais poder computacional, consequentemente mais processamento deve ser alocado pela CPU.

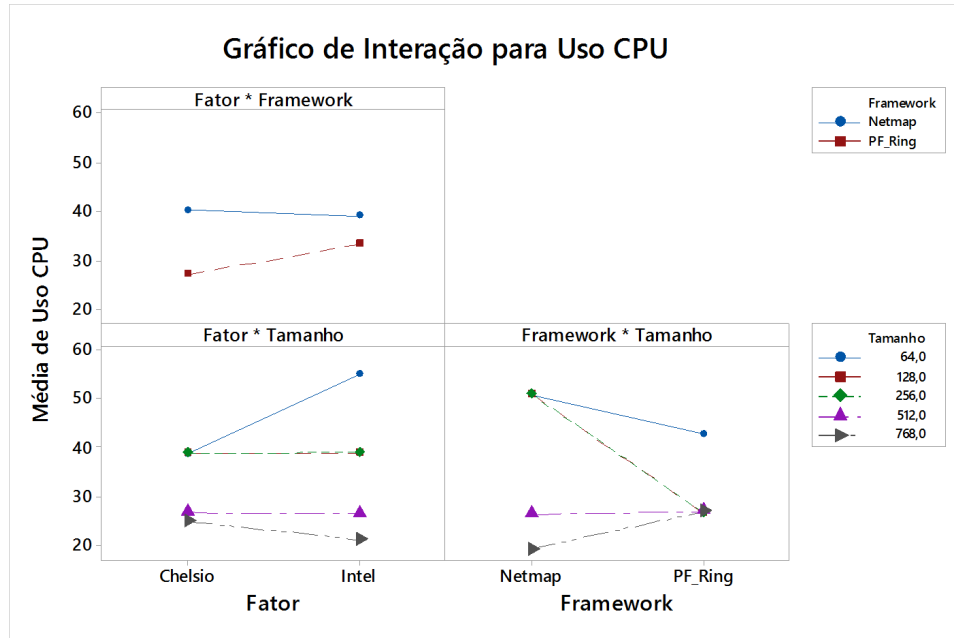
A Figura 15 mostra os gráficos de Interação dos níveis de cada fator, permitindo uma análise mais detalhada em relação ao gráfico de efeitos principais. O gráfico de Interação analisa as combinações específicas dos níveis, a fim de chegar a conclusão qual afeta menos

Figura 14: Percentual Médio de cada Nível dos Fatores



a variável resposta.

Figura 15: Interação dos Níveis dos Fatores



Como analisado no gráfico de efeitos principais, a placa Chelsio apresentou um consumo menor de CPU, juntamente com o framework PF_Ring. A Combinação da Placa da Chelsio com o Framework Netmap, foi a que mais exigiu recursos da CPU. A placa da Chelsio apresentou um consumo de CPU menor em todos os níveis de Tamanho de Pacote, com a excessão do tamanho 768 bytes. A placa Intel, por outro lado exigiu muito processamento com o Tamanho de pacote 64 bytes. Portanto nota-se que a placa da Chel-

sio apresenta um consumo de CPU mais eficiente com pacotes menores e intermediários, já para pacotes grandes a escolha pela placa da Intel é a mais acertada.

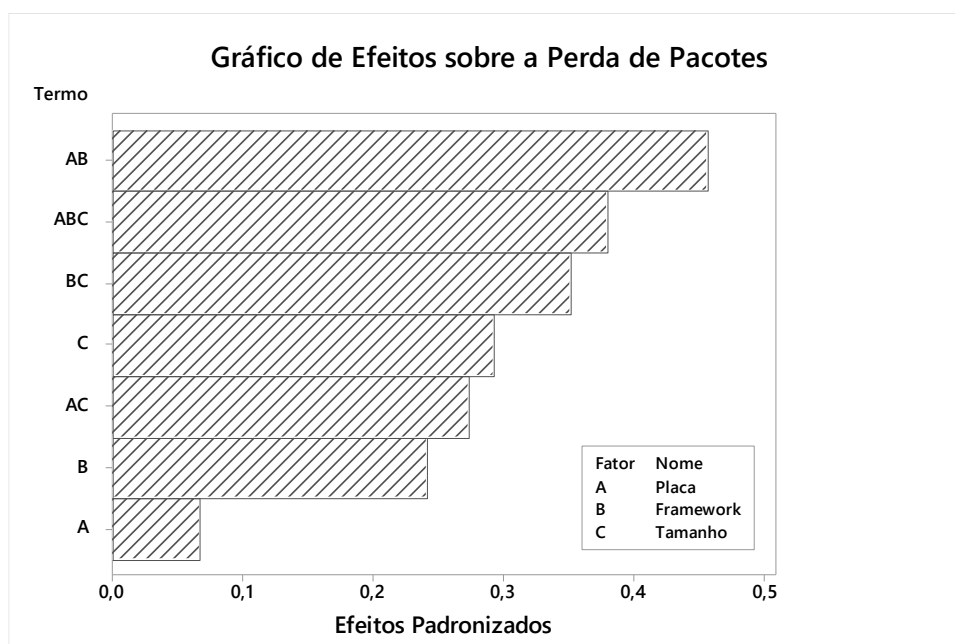
Em relação aos frameworks, o PF_Ring é o que apresenta os melhores resultados para níveis de tamanho inferior: 64 *bytes*, 128 *bytes* e 256 *bytes*. O Framework Netmap, porém, apresentou um percentual médio de uso de CPU menor com o pacote maior: 768 *bytes*. Portanto o Framework PF_Ring consome menos CPU com pacotes menores, já para pacotes maiores a tendência é o Netmap ser melhor.

Por fim conclui-se para a variável resposta Uso De CPU, as melhores combinações são Placa Chelsio e Framework PF_Ring. Para pacotes grandes a Combinação da Placa Intel e Netmap, apresentou bons resultados e é indicada nesta situação.

4.2.3 Pacotes Descartados

Nesta seção é demonstrada os principais resultados obtidos da métrica Pacotes Descartados. Na métrica Pacotes Descartadas, o percentual médio de pacotes descartados foi mensurado a partir dos pacotes descartados durante o experimento. A Figura 16 apresenta o gráfico de Pareto para os fatores relacionados a variável resposta Pacotes Descartados.

Figura 16: Influência dos Fatores na Acurácia (Perda de Pacote).

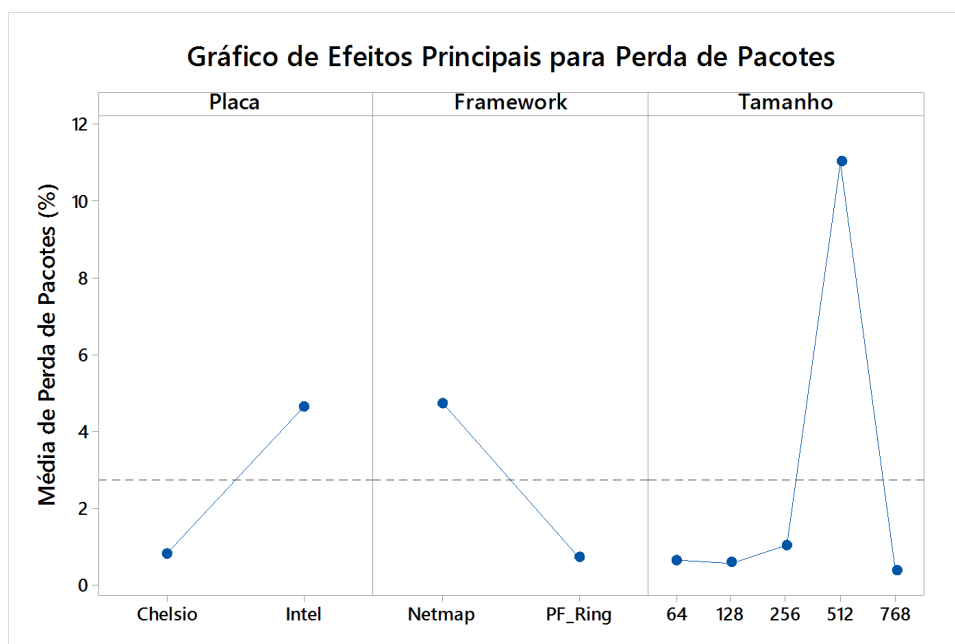


Diferentemente do observado nas outras métricas, os fatores não apresentaram impactos significativos sobre a variável Pacotes Descartados. As combinações desses fatores acabaram sendo mais impactantes. O fator Placa de Rede foi o que menos significativo, indicando que esse fator não acarreta em mudanças no experimento.

A Figura 17 mostra o gráfico de efeitos principais da métrica Pacotes Descartados. Observando o Gráfico nota-se que a variável resposta é afetada pelos níveis dos fatores: Placa

de Rede e Framework. A placa de Rede da Chelsio e o framework PF_Ring apresentaram um percentual menor de pacotes descartados. Em relação ao Tamanho do Pacote os níveis não afetam de forma significativa a variável resposta, exceto o tamanho 512 *bytes*. A suspeita de que o tamanho de pacote 512 *bytes* tenha atingido o pico ilustrado na Figura 17 é algum conflito de configurações entre o Netmap e a Placa Intel. Esse conflito teria causado alguma interferência na alocação dos anéis de memória do Netmap, ocasionando essa perda de pacotes.

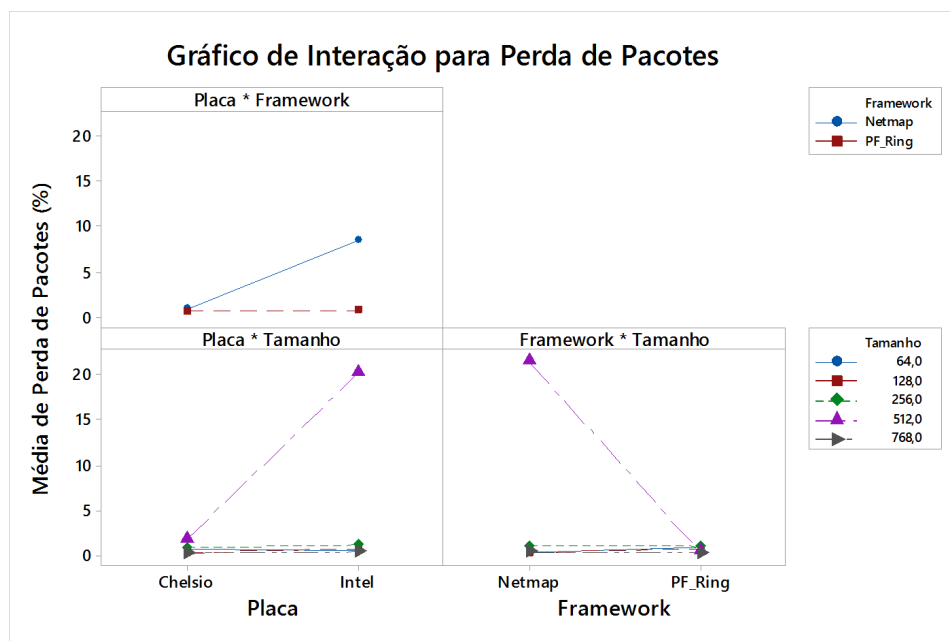
Figura 17: Percentual Médio de cada Nível dos Fatores



O Figura 18 mostra os gráficos de interação da métrica Pacotes Descartados. Observando os gráficos, chega-se a mesma conclusão do gráfico de efeitos principais, na qual os níveis do fator (Tamanho de Pacote) não apresentam interações, logo não afetam a variável resposta. O nível 512 *bytes*, porém apresentou uma elevada perda de pacotes diante da combinação da placa Intel e o framework Netmap. O primeiro gráfico indica que a relação entre a Placa de Rede e o Throughput apresenta pouca dependência do framework PF_Ring já que o mesmo não apresentou interação.

A perda de pacotes foi baixa diante de todas as combinações de níveis. O uso de frameworks é vantajoso para as grandes redes na questão da baixa perda de pacotes. Como demonstrado, no gráfico de Pareto o fator Placa de Rede não apresenta dependência significativa em relação aos níveis do fator Tamanho de Pacote na métrica avaliada. Portanto a escolha de uma placa utilizando como critério a métrica Pacotes Descartados, fica a cargo dos administradores de rede. A combinação da Placa Chelsio e do Framework PF_Ring se mostrou a mais eficiente em relação ao tamanho de pacote 512 *bytes*.

Figura 18: Interação dos Níveis dos Fatores



4.2.4 Resumo dos Resultados

Nesta seção é apresentada as principais conclusões dos resultados obtidos. Na Tabela 4 estão as combinações que obtiveram o melhor desempenho para cada métrica analisada. Na Tabela são exibidas as métricas utilizadas neste trabalho: Throughput, Uso de CPU e Pacotes Descartados. Ao lado de cada métrica há os fatores utilizados na análise de sensibilidade. A combinação trata-se de quais os níveis dos fatores foram melhores para cada métrica.

Tabela 4: Melhores Combinações Obtidas

Métrica	Combinação Ótima		
	Placa	Framework	Tamanho de Pacote (Bytes)
Throughput	Chelsio	Netmap	64,128,256,512,768
Uso de CPU	Chelsio	PF_Ring	64,128,256
	Intel	Netmap	768
Pacotes Descartados	Chelsio	PF_Ring	64,128,256,512,768

Com o conhecimento das melhores combinações, diferentes cenários foram hipoteticamente definidos, à fim de saber qual combinação obtém melhor desempenho em um cenário específico. Os cenários são uma forma de orientar os administradores de rede e demais profissionais a escolher a melhor combinação em situações específicas de tráfego.

- **Cenário 1:** Neste cenário, há uma rede que está enfrentando problemas de gargalo. A rede não consegue "absorver" o tráfego de pacotes. Para essa rede é necessário obter uma boa taxa de Throughput. Observando a Tabela 4 nota-se que a combinação Chelsio e Netmap é a melhor, inclusive para diversos tamanhos de pacotes.

- **Cenário 2:** Neste cenário, um gerente de rede deseja adquirir uma solução que exija menos recursos para tratar o tráfego de dados. Observando a Tabela 4, há duas possibilidades a serem utilizadas a fim de economizar recursos. Caso o tráfego possua muitos pacotes grandes sendo transmitidas em curto espaço de tempo, o ideal é adotar a Placa Intel e o framework Netmap. Caso o tráfego possua pacotes menores, a Placa Chelsio e o framework PF_Ring são as melhores escolhas.
- **Cenário 3:** A rede hipoteticamente está enfrentando perdas elevadas de pacotes. A perda de pacotes ocasiona degradação dos serviços de internet. Deseja-se adotar uma combinação que consiga ler os pacotes com uma perda mínima. Nesse caso a Placa Chelsio e o framework PF_Ring são boas escolhas a serem utilizadas.

5 Conclusão

Este trabalho desenvolveu análises de artefatos de rede (Placas de Rede e Framework de captura e processamento de pacotes). A análise visa auxiliar os administradores de redes, gerentes de infraestruturas, dentre outros profissionais da área a escolher tais artefatos. Busca-se compreender as situações que os mesmos se destacam, indicando qual combinação atende melhor os contextos específicos de tráfego, que esses profissionais precisam. Em específico as placas de redes Chelsio *T520-SO-CR* e Intel *X520-DA2* foram submetidas a experimentos juntamente com os frameworks Netmap e PF_Ring. Foi analisado o comportamento dos mesmos em situações comuns de tráfego de redes, como tráfegos com pacotes de tamanho diferentes. A aplicabilidade do estudo em um ambiente real foi verificada através de análises de sensibilidade. Por meio dessas análises foi possível montar um tráfego que pudesse ser utilizado em um ambiente real. Com os resultados coletados e gerados, notou-se que cada métrica analisada apresentou diferentes situações de uso dos artefatos de rede. Cada situação de uso, levou em conta os impactos dos fatores nas métricas, bem como as interações desses fatores na variável resposta. Para cada variável resposta uma diferente combinação de Placa de Rede e Framework se mostrou mais viável ou não. Portanto, diferentes combinações foram indicadas, conforme os resultados, permitindo assim uma ferramenta de auxílio na tomada de decisão para os profissionais de rede.

Este trabalho teve algumas limitações, uma delas está na quantidade de placas de rede. A utilização de um número maior de Placas de Rede traria mais consistência na validação deste trabalho. No entanto, isso não foi possível devido ao elevado custo financeiro de tais aquisições. Outra limitação foi o uso das configurações padrões de cada Framework. Configurações mais avançadas como *Zero Copy* no Netmap e PF_Ring, poderiam ter sido realizadas, porém não foi possível devido a limitações de hardware das máquinas. Este trabalho poderia ter adotado o uso de gráficos de boxplot para facilitar a visualização dos dados de forma mais clara. Muitos gráficos de boxplot foram gerados devido a quantidade de combinações de níveis dos fatores, sendo optado a não inclusão desses gráficos por questões visuais.

Para trabalhos futuros, será realizada uma análise de sensibilidade com mais variáveis respostas, como uso de memória e velocidade de transmissão média. Nos experimentos seria feita a inclusão de fatores como *Threads* e diferentes serviços de rede, utilizando um mesmo protocolo. Outro ponto a ser abordado é a questão energética das placas de redes, visualizando qual a mais eficiente, levando em conta os fatores utilizados e que poderão ser inclusos. A questão energética é hoje de grande importância em estudos e trabalhos científicos. Mensurar os gastos energéticos, também é importante para empresas levando em conta a necessidade de redução de custos.

6 Publicações

Siqueira, D., Rocha, P. H., Bisneto, T., Rego, P. A., & Silva, F. A. Uma Análise de Desempenho sobre uma MicroFog. **Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance_CSBC)**, [S.l.], v. 17, n. 1/2018, july 2018.

Referências

- AMMENDOLA, R. et al. A multi-port 10gbe pcie nic featuring udp offload and gpudirect capabilities. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2015. v. 664, n. 9, p. 092002. Citado 2 vezes nas páginas 26 e 27.
- ANTONY, J. Taguchi or classical design of experiments: a perspective from a practitioner. *Sensor Review*, Emerald Group Publishing Limited, v. 26, n. 3, p. 227–230, 2006. Citado na página 21.
- BARAKAT, C. et al. A flow-based model for internet backbone traffic. In: ACM. *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*. [S.l.], 2002. p. 35–47. Citado na página 16.
- BARBETTE, T.; SOLDANI, C.; MATHY, L. Fast userspace packet processing. In: IEEE. *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*. [S.l.], 2015. p. 5–16. Citado 2 vezes nas páginas 26 e 27.
- BARKER, S. A. *Comparison of Ring-Buffer-Based Packet Capture Solutions*. [S.l.], 2015. Citado 3 vezes nas páginas 26, 27 e 28.
- BENSON, T.; AKELLA, A.; MALTZ, D. A. Network traffic characteristics of data centers in the wild. In: ACM. *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. [S.l.], 2010. p. 267–280. Citado na página 16.
- BHATTACHARYYA, S.; DIOT, C.; JETCHEVA, J. Pop-level and access-link-level traffic dynamics in a tier-1 pop. In: ACM. *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. [S.l.], 2001. p. 39–53. Citado na página 16.
- BOUTREMANS, C.; IANNACCONE, G.; DIOT, C. Impact of link failures on voip performance. In: ACM. *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*. [S.l.], 2002. p. 63–71. Citado na página 16.
- BRADNER, S.; MCQUAID, J. Rfc 2544. *Benchmarking methodology for network interconnect devices*, 1999. Citado na página 30.
- BRITO, C. Internet no brasil: o problema está na infraestrutura. In: . Abranet - Associação Brasileira de Internet, 2016. Disponível em: <<http://www.abranet.org.br/Artigos/Internet-no-Brasil:-o-problema-esta-na-infraestrutura-1107.html>>. Acesso em: 31.10.2018. Citado na página 14.
- BRITO, S. H. B. *IPv6-O novo protocolo da Internet*. [S.l.]: Novatec Editora, 2018. Citado na página 13.
- CETIC.BR. Os provedores de acesso à internet no brasil: Como atuam e qual sua importância para o desenvolvimento da internet brasileira. In: . CGI.BR/NIC.BR, 2016. v. 8, n. 2. Disponível em: <https://nic.br/media/docs/publicacoes/6-Panorama_Setorial_11_2.pdf>. Acesso em: 31.10.2018. Citado na página 13.

- CISCO, V. Cisco visual networking index: Forecast and methodology 2016–2021.(2017). In: . [s.n.], 2017. Disponível em: <<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>>. Acesso em: 31.10.2018. Citado na página 13.
- DERI, L. et al. Improving passive packet capture: Beyond device polling. In: AMSTERDAM, NETHERLANDS. *Proceedings of SANE*. [S.l.], 2004. v. 2004, p. 85–93. Citado na página 21.
- DERI, L.; RING, M. P. Url <http://www.ntop.org>. *PF_RING.html*, 2011. Citado na página 14.
- EMMERICH, P. et al. Optimizing latency and cpu load in packet processing systems. In: IEEE. *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2015 International Symposium on*. [S.l.], 2015. p. 1–8. Citado na página 14.
- ESPEJO, M. R. *Design of experiments for engineers and scientists*. [S.l.]: Taylor & Francis, 2006. Citado na página 22.
- FILHO, J. E. M. *Análise de Tráfego em Redes TCP/IP: Utilize tcpdump na análise de tráfegos em qualquer sistema operacional*. [S.l.]: Novatec Editora, 2013. Citado na página 16.
- FRALEIGH, C. et al. Packet-level traffic measurements from the sprint ip backbone. *IEEE network*, IEEE, v. 17, n. 6, p. 6–16, 2003. Citado na página 16.
- GALLENMÜLLER, S. Comparison of memory mapping techniques for high-speed packet processing. *Technical University of Munich*, 2014. Citado 2 vezes nas páginas 26 e 27.
- GALLENMÜLLER, S. et al. Comparison of frameworks for high-performance packet io. In: IEEE COMPUTER SOCIETY. *Proceedings of the Eleventh ACM/IEEE Symposium on Architectures for networking and communications systems*. [S.l.], 2015. p. 29–38. Citado 2 vezes nas páginas 26 e 28.
- GUNST, R. F. *Response surface methodology: process and product optimization using designed experiments*. [S.l.]: Taylor & Francis, 1996. Citado na página 21.
- HENGARTNER, U. et al. Detection and analysis of routing loops in packet traces. In: ACM. *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*. [S.l.], 2002. p. 107–112. Citado na página 16.
- INTEL, D. *Data Plane development kit*. 2014. Citado na página 14.
- JAISWAL, S. et al. Measurement and classification of out-of-sequence packets in a tier-1 ip backbone. *IEEE/ACM Transactions on Networking (ToN)*, IEEE Press, v. 15, n. 1, p. 54–66, 2007. Citado na página 16.
- KIM, J.-H.; NA, J.-C. A study on one-way communication using pf_ring zc. In: IEEE. *Advanced Communication Technology (ICACT), 2017 19th International Conference on*. [S.l.], 2017. p. 301–304. Citado 3 vezes nas páginas 26, 28 e 29.
- KING, T. Packet sniffing in a switched environment. *SANS Institute, GESC practical*, v. 1, 2002. Citado na página 17.

- KUHN, D. R.; REILLY, M. J. An investigation of the applicability of design of experiments to software testing. In: IEEE. *Software Engineering Workshop, 2002. Proceedings. 27th Annual NASA Goddard/IEEE*. [S.l.], 2002. p. 91–95. Citado na página 21.
- KUROSE, J. F.; ROSS, K. W. *Computer networking: a top-down approach: international edition*. [S.l.]: Pearson Higher Ed, 2013. Citado na página 13.
- LARSEN, S.; LEE, B. Survey on system i/o hardware transactions and impact on latency, throughput, and other factors. In: *Advances in Computers*. [S.l.]: Elsevier, 2014. v. 92, p. 67–104. Citado 2 vezes nas páginas 26 e 27.
- LAZIC, Z. R. *Design of experiments in chemical engineering: a practical guide*. [S.l.]: John Wiley & Sons, 2006. Citado na página 21.
- LEE, S.; LEVANTI, K.; KIM, H. S. Network monitoring: Present and future. *Computer Networks*, Elsevier, v. 65, p. 84–98, 2014. Citado na página 16.
- MARCUZZO, L. d. C.; SANTOS, C. R. dos. Análise e comparação de técnicas e aceleradores de processamento de pacotes. *ERRC 2017*, p. 2, 2017. Citado 2 vezes nas páginas 26 e 29.
- MEDINA, A. et al. Traffic matrix estimation: Existing techniques and new directions. In: ACM. *ACM SIGCOMM Computer Communication Review*. [S.l.], 2002. v. 32, n. 4, p. 161–174. Citado na página 16.
- MILLER, C. et al. Bim-extracted energyplus model calibration for retrofit analysis of a historically listed building in switzerland. *Proceedings of SimBuild*, v. 2014, 2014. Citado na página 22.
- MORENO, V. et al. Testing the capacity of off-the-shelf systems to store 10gbe traffic. *IEEE Communications Magazine*, IEEE, v. 53, n. 9, p. 118–125, 2015. Citado na página 14.
- MORENO, V. et al. Commodity packet capture engines: Tutorial, cookbook and applicability. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 3, p. 1364–1390, 2015. Citado na página 16.
- NURIKA, O. et al. Real throughput measurements comparison between unoptimized and optimized network cards. *Science International*, v. 29, p. 87–91, 2016. Citado 2 vezes nas páginas 26 e 28.
- ORTEGA, D. D. Pf_ring: Solución de software libre y código abierto para la captura de paquetes en redes de alta velocidad. *Revista Telem@tica*, v. 14, n. 2, p. 13–22, 2015. Citado 2 vezes nas páginas 14 e 20.
- PALMER, M. *Hands-on networking fundamentals*. [S.l.]: Cengage learning, 2012. Citado na página 18.
- PARK, K.; WILLINGER, W. *Self-similar network traffic and performance evaluation*. [S.l.]: John Wiley & Sons, Inc., 2000. Citado na página 16.
- RIZZO, L. Netmap: a novel framework for fast packet i/o. In: *21st USENIX Security Symposium (USENIX Security 12)*. [S.l.: s.n.], 2012. p. 101–112. Citado na página 14.

- RIZZO, L. Revisiting network i/o apis: the netmap framework. *Communications of the ACM*, ACM, v. 55, n. 3, p. 45–51, 2012. Citado na página 20.
- RIZZO, L.; LANDI, M. Netmap: Memory mapped access to network devices. *ACM SIGCOMM Computer Communication Review*, ACM, v. 41, n. 4, p. 422–423, 2011. Citado na página 19.
- SELTMAN, H. J. Experimental design and analysis. *Online at: <http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>*, 2012. Citado na página 21.
- SIEBERTZ, K.; BEBBER, D. V.; HOCHKIRCHEN, T. *Statistische Versuchsplanung: design of experiments (DoE)*. [S.l.]: Springer-Verlag, 2017. Citado na página 21.
- STATISTA. Data volume of consumer-managed internet protocol (ip) traffic in latin america from 2016 to 2021 (in petabytes per month). In: . [s.n.], 2018. Disponível em: <<https://www.statista.com/statistics/728329/consumer-managed-ip-traffic-latin-america/>>. Acesso em: 31.10.2018. Citado na página 13.
- STATISTA. Data volume of global consumer ip traffic from 2015 to 2021 (in petabytes per month). In: . [s.n.], 2018. Disponível em: <<https://www.statista.com/statistics-267202/global-data-volume-of-consumer-ip-traffic/>>. Acesso em: 31.10.2018. Citado na página 13.
- THOMAS, D. et al. Multiscale co-simulation of energyplus and citysim models derived from a building information model. In: *Bausim 2014: Fifth German-Austrian IBPSA Conference*. [S.l.: s.n.], 2014. Citado na página 22.

Apêndices

APÊNDICE A – Script de Medição da CPU no Framework Netmap

Listing A.1: Script de Coleta do Uso da CPU no Netmap

```
#!/bin/bash
while $(pgrep pkt-gen > 0);
do
    # Get the total CPU statistics.
    CPU=(`cat /proc/stat | grep '^cpu '` )
    # Discard the "cpu" prefix.
    unset CPU[0]
    # Get the idle CPU time.
    IDLE=${CPU[4]}

    # Calculate the total CPU time.
    TOTAL=0
    for VALUE in "${CPU[@]}"; do
        let "TOTAL=$TOTAL+$VALUE"
    done

    # Calculate the CPU usage since we last checked.
    let "DIFF_IDLE=$IDLE-$PREV_IDLE"
    let "DIFF_TOTAL=$TOTAL-$PREV_TOTAL"
    let "DIFF_USAGE=(1000*($DIFF_TOTAL-$DIFF_IDLE)
/$DIFF_TOTAL+5)/10"
    echo -en "\rCPU: $DIFF_USAGE% \b\b"
    echo $DIFF_USAGE >> cpu_usage_$1_$2.txt
    # Remember the total and idle CPU times for the
    # next check.
    PREV_TOTAL="$TOTAL"
    PREV_IDLE="$IDLE"
    # Wait before checking again.
    sleep 1
done
```

APÊNDICE B – Script de Medição da CPU no Framework PF_Ring

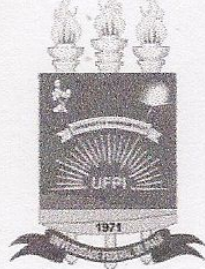
Listing B.1: Script de Coleta do Uso da CPU no PF_Ring

```
#!/bin/bash
while $(pgrep pfcourt > 0);
do
    # Get the total CPU statistics.
    CPU=(`cat /proc/stat | grep '^cpu `)
        # Discard the "cpu" prefix.
unset CPU[0]
    # Get the idle CPU time.
IDLE=${CPU[4]}

    # Calculate the total CPU time.
TOTAL=0
for VALUE in "${CPU[@]}"; do
    let "TOTAL=$TOTAL+$VALUE"
done

    # Calculate the CPU usage since we last checked.
let "DIFF_IDLE=$IDLE-$PREV_IDLE"
let "DIFF_TOTAL=$TOTAL-$PREV_TOTAL"
let "DIFF_USAGE=(1000*($DIFF_TOTAL-$DIFF_IDLE)
/$DIFF_TOTAL+5)/10"
echo -en "\rCPU: $DIFF_USAGE% \b\b"
    echo $DIFF_USAGE >> cpu_usage_$1_$2.txt
    # Remember the total and idle CPU times for the
#next check.
PREV_TOTAL="$TOTAL"
PREV_IDLE="$IDLE"
    # Wait before checking again.
sleep 1

done
```



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
() Dissertação
(X) Monografia
() Artigo

Eu, Paulo Henrique Gonçalves Rocha,

autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação

Análise do Desempenho de Placas de Rede 10GbE: Uma Estratégia Baseada em Análise de Sensibilidade

de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 04 de dezembro de 2018.

Paulo Henrique G. Rocha

Assinatura

Paulo Henrique G. Rocha

Assinatura