

vSDNBox: Um Hardware Especializado de Baixo Custo Gerenciado via SDN

**Fernando N. N. Farias¹, Marcos F. Schwarz², José F. de Rezende²,
Antônio J. G. Abelém¹**

¹Programa de Pós-Graduação em Ciências da Computação – PPGCC
Universidade Federal do Pará – UFPA – PA – Brasil

²Rede Nacional de Ensino e Pesquisa – RNP – Campinas – SP

{fernfnf, abelem}@ufpa.br, {marcos.schwarz, jose.rezende}@rnp.br

Abstract. *Whitebox switches have been an advantageous alternative for the SDN data plane. They allow not only a reduction in CAPEX costs (e.g. in infrastructure) but also, they increase the levels of heterogeneity of manufacturers in the network infrastructure. However, recent tools allowed expanding the performance of whiteboxes based on software-switches with optimizations made with open software and generic hardware. Thus, vSDNBox is a proposal of a software-switched whitebox for software-defined networks that is able to reduce the final cost, without loss of performance because of its management and optimizations made from user space. The results show that the proposal achieves a performance equal or approximate to a hardware-switched whitebox in terms of two important requirements: throughput and latency.*

Resumo. *Os switches whitebox vêm sendo uma alternativa vantajosa para o plano de dados SDN. Eles permitem tanto a redução nas despesas como também a ampliação dos níveis de heterogeneidade de fabricantes na infraestrutura de rede. Recentes ferramentas possibilitaram que os whiteboxes baseados em software-switches pudessem ampliar seus desempenhos apenas com otimizações feitas via softwares abertos em hardwares genéricos. A partir disso, propõe-se o vSDNBox, uma alternativa de whitebox software-switched para redes definidas por software, capaz de reduzir ainda mais os custos e ampliar o desempenho através de seu gerenciamento e otimizações feitas via software no espaço do usuário. Os resultados obtidos comprovam que a proposta consegue ter um desempenho igual ou aproximado de um whitebox hardware-switched através de avaliações de vazão e latência.*

1. Introdução

Mais de 10 anos se passaram desde o início das redes definidas por software (SDNs – *Software Defined Networks*) e, neste tempo, o paradigma SDN tem possibilitado grandes inovações e revolucionado praticamente todos os setores das redes de computadores. Em seu núcleo, as SDNs promovem uma revisão dos hardwares de rede e os softwares que as controlam, através da separação dos planos de controle e de dados. Além disso, por meio de suas interfaces programáveis, SDN habilita novas formas de controlar e gerenciar o comportamento do encaminhamento de dados na rede, de maneira mais inteligente, ágil e menos suscetível a erros [Kreutz et al. 2015].

A principal vantagem trazida pela arquitetura SDN foi a característica de abertura (*openness*) tanto no nível de software quanto de hardware. Desta forma, o software e hardware são vendidos separadamente e os consumidores possuem a liberdade para escolherem entre uma série de opções de sistemas operacionais ou protocolos de controle e gerenciamento. Dentro deste ecossistema, pode-se observar vários níveis de abertura, que originou o chamado *Open SDN*, tais como: padrões abertos (*Open Standards*), softwares livres (*Open Software*), kits de desenvolvimentos (SDKs), interfaces de programação de aplicações (APIs) e hardwares abertos (*open hardware, bare metal* ou *whitebox*) [Shin; Nam e Kim 2012].

Os *whiteboxes* são um novo modelo ou tendência de comutadores ou switches, que diferentes dos switches proprietários (“*blackbox*”), o seu hardware e software são oferecidos separadamente, tornando-os assim muito popular no mundo SDN. Além disso, os administradores têm toda a liberdade sobre o que instalar e configurar dentro destas “caixas”, desde o sistema operacional até as aplicações, passando por soluções de gerenciamento desses dispositivos.

Essa liberdade no gerenciamento trazida pelos *whiteboxes* flexibiliza e facilita o suporte a múltiplos protocolos de encaminhamento, como MPLS, IPv4 & IPv6 e *Ethernet*. Para suportar tais pilhas de software no passado, eram necessários roteadores extremamente robustos e caros, que muitas das vezes só eram operáveis mediante uma infraestrutura inteira utilizando o mesmo modelo de equipamento. Com o uso de *whiteboxes*, isso se tornou simples, escalável e totalmente interoperável entre fabricantes. Mais recentemente, tornou-se um dos principais objetivos do SDN a realização do processamento do encaminhamento de pacote de forma independente do protocolo [Shahbaz et al. 2016].

Apesar dessas vantagens, a realidade dos *whiteboxes* ainda está longe do ideal, por isso ainda há desafios abertos relacionadas a custos, integração, interoperabilidade e desempenho [Kreutz et al. 2015]. Isso é observável devido ao fato que determinados fabricantes continuam colocando características ou elementos específicos em protocolos SDN, tornando-os ainda dependente de hardware. Além disso, para a aceleração de processamento de pacotes, os fabricantes recorrem a hardwares de processamento específico, por exemplo, os ASICs (*Application-Specific Integrated Circuit*) e memórias TCAM, chamados de *hardware-switched*, tornando o equipamento extremamente caro e pouco atrativo no mercado SDN.

Para facilitar e libertar o plano de controle SDN dessas restrições de hardware, pesquisas em DPDK [DPDK 2018] e P4 [P4 2018] vêm buscando capitalizar oportunidades dentro do plano de dados através de uma nova geração de *whiteboxes* compatíveis com hardwares genéricos e programáveis via software, ou seja, hardwares que podem ser gerenciáveis por softwares no espaço do usuário do sistema operacional, a estes *whiteboxes* que têm seu gerenciamento funcional praticamente todo baseado em software dar-se o nome de *software-switched* [Cox et al. 2017].

Este trabalho apresenta uma proposta de *whitebox* de baixo custo, chamada de *vSDNBox*, baseada em *software-switched* e hardware genérico, com objetivo de criar uma “caixa” gerenciável por qualquer protocolo aberto SDN, com desempenho igual ou aproximado a um *whitebox hardware-switched*, de fácil instalação, configuração e atualização de ferramentas e protocolos. O resultado das comparações dos dados coletados mostra que o *vSDNBox* tem desempenho muito próximo a equipamentos *hardware-switched*, e conclui-se que o mesmo pode ser uma alternativa ainda barata e de desempenho equiparável aos *whiteboxes hardware-switched*.

Além desta seção introdutória este artigo ainda está dividido em mais 5 seções. Na Seção 2, tem-se os trabalhos relacionados a elementos que compõem esta proposta. Já na Seção 3, observa-se o referencial teórico conceituando a forma de gerenciamento baseada no modelo de redes definidas por software. Adicionalmente, na Seção 4 tem-se a descrição da proposta, apresentando principalmente os custos utilizados para seu desenvolvimento e as características das principais funcionalidades. Na Seção 5, tem-se a descrição da coleta e a análise dos resultados obtidos na comparação entre o *vSDNBox* e *whitebox* P-3297. E por fim na Seção 6, tem-se as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

No artigo técnico da empresa PICA8 [PICA8 2018], são apresentados os modelos de *whiteboxes* da empresa Pica8 e as vantagens do uso de redes baseadas em *whitebox* devido a liberdade em comprar um hardware separado do seu software de rede. Também, comenta-se a economia feita ao realizar este tipo de escolha. Além disso, o artigo apresenta a arquitetura dos seus *whiteboxes* baseados em hardware ASICs sobre um sistema operacional proprietário chamado PicOS [PICOS 2018].

No artigo, observa-se uma tendência que vem acontecendo com a maioria dos *whiteboxes*, que é a sua integração com hardware de custos extremamente elevados, que é o caso dos ASICs de empresas como *Broadcom*, *Marvell* ou *Mellanox*. Por outro lado, também se nota que as funcionalidades desses *whiteboxes*, tais como a disponibilidade de pilhas de roteamento, acesso ao encaminhamento de pacotes ou habilitar/desabilitar certas quantidades de bridges ou portas no equipamento, só são acessadas, em sua totalidade, após a obtenção de licenças em seu sistema operacional. Ou seja, apesar de serem *whiteboxes*, seus hardwares não possuem bibliotecas abertas para acessar suas funcionalidades e o seu sistema operacional só libera a utilização de algumas características mediante a compra de licenças.

No trabalho de Manggala [Manggala; Hendrawan e Tanwidjaja 2015], observa-se uma análise comparativa entre um *whitebox* e um switch convencional da empresa Cisco. Nele são medidos a latência e a *vazão* através de pacotes ICMP com tamanhos variados de carga útil (ex. no intervalo de 100 a 51200 Bytes). Além disso, o trabalho apresenta uma avaliação sobre o uso de CPU e memória.

No artigo, os resultados apresentados mostram que o switch convencional tem o melhor desempenho quando comparado ao *whitebox*, mas observando os seus dados, percebe-se que esta vantagem é pequena, já mostrando o potencial dos *whiteboxes* baseados em *software-switched*. Também, observa-se nesse trabalho que o software de encaminhamento não é integrado a bibliotecas que permitam a aceleração do processamento de pacote em hardwares programáveis, que melhoraria ainda mais o desempenho de latência e *vazão* desses switches.

3 Referencial Teórico

3.1 Redes Definidas por Software

As Redes Definidas por Software (SDNs) é um paradigma de redes emergente que possibilita o desacoplamento entre os planos de controle e de dados. Onde o plano de controle é responsável por tomar decisões no encaminhamento dos dados de acordo com comportamento programado pela aplicação, enquanto o plano de dados é por onde os dados ou pacotes são encaminhados, envolvendo desde o processamento dos mesmos até a parte física do equipamento, como porta, cabo elétrico, fibra ou antena.

Além disso, as SDNs são conceitualmente controladas por métodos de software, os quais podem ser providos tanto por fabricantes quanto por seus próprios usuários finais ou administradores da rede. Além disso, a programabilidade inserida pelas redes SDN permite uma maior automatização das operações de gerenciamento, bem como, alcançar altos índices de inovação, diferenciação e adaptação.

A partir do SDN, o plano de controle - antes inacessível - é retirado dos dispositivos de rede e migrado para entidades externas que proporcionam a programação remota do plano de dados. Nesse sentido, a infraestrutura e seus componentes passam a serem abstraídas em uma visão de aplicações e serviços de rede, estes, inclusive, podem fazer uso da rede como entidade lógica ou virtual.

SDN define uma forma de implementar políticas de controle de acesso de forma distribuída, a partir de um mecanismo de supervisão centralizado. Nesta arquitetura, cada elemento de rede deve consultar o elemento controlador ao identificar um novo fluxo. O controlador consulta um grupo de políticas globais ou procedimentos para decidir, com base nas características de cada fluxo, como o elemento de encaminhamento deveria tratá-lo. Essa decisão é comunicada ao comutador na forma da programação de uma entrada em sua tabela de encaminhamento com uma regra adequada para o novo fluxo (que poderia, inclusive, ser seu descarte).

A arquitetura SDN, conforme apresentado na Figura 1, consiste de três camadas distintas que são acessíveis através de APIs (*Application Programming Interface*), e nomeadas da seguinte maneira: camada de aplicação, controle e infraestrutura.

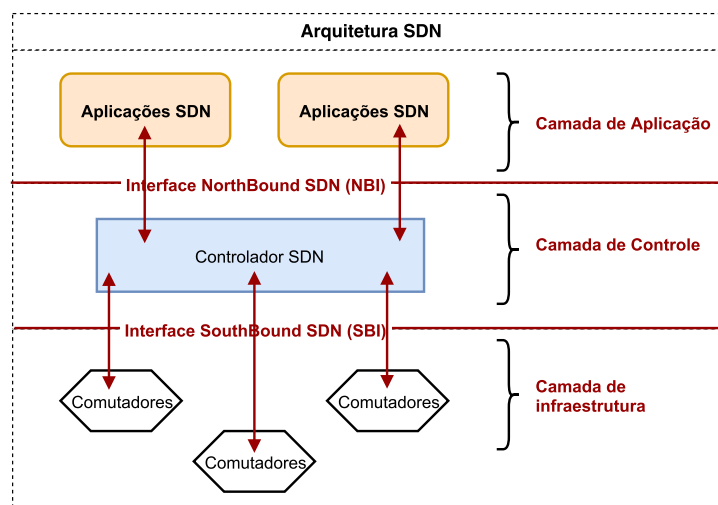


Figura 1. Arquitetura do modelo SDN.

A camada de aplicação consiste nas aplicações dos usuários que consome um ou mais serviços da camada de controle SDN. Ela também é a fronteira entre aplicação e controle usualmente conhecida com *Northbound*. Já a camada de controle provê sólidas funcionalidade de controle, que supervisionam o comportamento do encaminhamento dos fluxos de pacotes através de protocolos de controle como *OpenFlow* [Lara; Kolasani e Ramamurthy 2013]. Por fim, a camada de infraestrutura consiste dos elementos de rede como dispositivos físicos e virtuais que provêm o encaminhamento de pacotes.

3.2 *Whitebox*

O *whitebox* é localizado na camada de infraestrutura do modelo SDN e tem como objetivo encaminhar os fluxos de dados de acordo com comandos oriundos do controlador, para

isso é utilizado um protocolo de *southbound*, por exemplo, o *OpenFlow* que é uma API que padroniza todas as ações enviadas do controlador para *whitebox* e vice-versa.

Diferente de um switch convencional, um *whitebox* é um switch que possui um conjunto mínimo de softwares instalados, ou seja, muitas das vezes o sistema operacional e bibliotecas são de responsabilidade do administrador ou desenvolvedor. A vantagem disso é a diminuição da complexidade e os custos agregados com softwares controladores ou protocolos instalados no equipamento, que com a adoção do SDN deixam de ser uma obrigatoriedade dentro dos mesmos por estarem localizados em um elemento externo.

Além disso, devido a sua flexibilidade e abertura, os *whiteboxes* também são desenvolvidos para suportar uma grande variedade de ferramentas livres (ex. *Openstack*, *Open vSwitch* ou *DPDK*) permitindo ao administrador tomar decisões da compra de equipamentos separados das decisões de softwares, e desta forma poder customizar esses dispositivos de acordo as regras de negócios ou as necessidades específicas de cada rede. Entretanto, um ponto crucial para o sucesso do *whitebox* está ligado ao baixo custo de aquisição e à operação desses equipamentos.

Atualmente, os *whiteboxes* são baseados em hardwares x86 e podem ser divididos em dois tipos: *hardware-switched* e *software-switched*. Os *whiteboxes hardware-switched*, ilustrados na Figura 2 (a), fazem a utilização de um processador de propósito específico dedicado ao processamento de pacotes, totalmente independente do resto do sistema, esse componente é conhecido como dispositivos *ASIC*. Ele provê características ou funcionalidades que não estão integradas ao sistema operacional e são acessadas via uma API proprietária. No caso dos processadores de pacotes, que são os mais utilizados em *whiteboxes*, isso permite que este processamento obtenha excelentes resultados de desempenho para execuções e análises em pacotes de dados comuns ou específicos.

Já nos *softwares-switched*, ilustrado na Figura 2 (b), há um comportamento diferente, pois, o controle da interface é implementado no espaço do usuário do sistema operacional. No entanto, com uso de bibliotecas de aceleração de pacotes que utilizam hardware genéricos é possível programar o comportamento deste hardware, também, do espaço do usuário. Esse processo é conhecido como *Kernel Bypass* [Tan et al. 2017]. Esta técnica vem sendo bastante utilizada para melhorar o desempenho de interfaces de rede permitindo uma aceleração no processamento de pacotes e fazendo uma ponte direta entre a aplicação e o dispositivo, e com isso eliminando o *overhead* da camada de *kernel*. Além disso, através deste processo também é possível utilizar ou alocar, de forma dedicada, núcleos do processador e memória para auxiliarem no processamento desses pacotes.

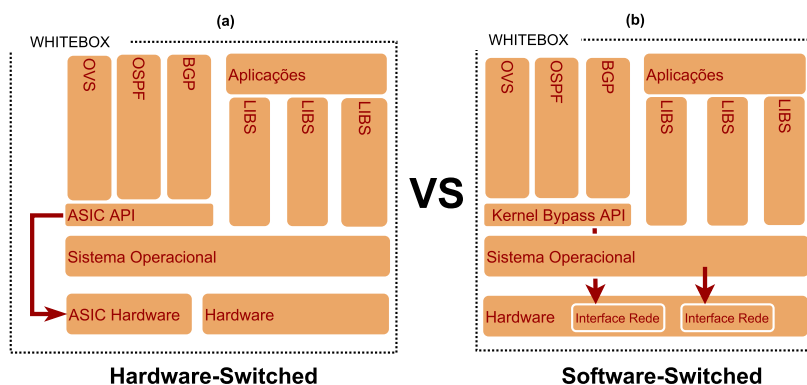


Figura 2. Visão geral de *whiteboxes hardware-switched vs software-switched*.

No entanto, há vantagens e desvantagens em ambos os modelos, por exemplo, em *hardware-switched*, a utilização de hardwares ASIC faz com que o preço do *whitebox* seja bastante elevado, um outro fator é que, dependendo do dispositivo, as funcionalidades são limitadas, bem como, a inclusão de novas funcionalidades é totalmente dependente de acordos com o fabricante, entretanto, seu desempenho é muito maior. Por outro lado, a utilização do modelo *software-switched* causou uma queda nos valores de *whiteboxes*, além de uma abertura para o desenvolvimento de equipamentos mais automatizados e com evolução e criação de novas funcionalidades independente de seus fabricantes.

4. *vSDNBox* Projeto e Implementação

O projeto de construção do *vSDNBox* é fortemente influenciado por aspectos e requisitos que levam em consideração a diminuição de custos, utilização de softwares abertos e otimização do desempenho. Desta forma, a apresentação do *vSDNBox* inicia-se pela definição destes requisitos:

- **Diminuição de custos** – tendo observado que apesar das vantagens trazidas pelo *whitebox hardware-switched* os valores agregados aos mesmos ainda são elevados no ponto de vista de investimento em infraestrutura, com preços bem aproximados aos switches convencionais. No entanto, a partir do avanço dos *softwares-switched* observou-se uma oportunidade para diminuir ainda mais os custos com *whitebox* sem perder desempenho na comutação de dados.
- **Utilização de softwares abertos** – é a base de sustentação da proposta que além de contribuir para a diminuição de custos, ela também, contribui para a atualização e evolução do *vSDNBox*. Essa necessidade permite na implementação do mesmo a possibilidade de agregar ou desagregar funções sem perda da qualidade, flexibilidade ou desempenho.
- **Otimização do desempenho** – o foco principal da proposta é minimizar custos e manter o desempenho, por mais que dispositivos *hardware-switched* sejam mais robustos nesta questão (*i.e.* desempenho), é possível otimizar o *vSDNBox* para que o seu desempenho se aproxime o máximo possível do *hardware-switched*.

Esses requisitos permitiram, de forma criteriosa, direcionar as decisões quanto à definição da arquitetura, modelos de hardwares, e o aprimoramento do desempenho, no *vSDNBox*.

A arquitetura do *vSDNBox* foi desenvolvida baseada no modelo *software-switched* com a utilização de hardware genérico com APIs abertas, facilitando o aprimoramento e, conseqüentemente, a melhora do desempenho alcançado. Sendo assim, na Figura 3 apresenta-se a visão geral da arquitetura definida para *vSDNBox*. Nesta figura, é possível observar 3 níveis de implementações aplicadas à proposta, onde pode-se destacar, a camada de aceleração de pacotes, a camada de controle de encaminhamento e a camada de protocolo de gerenciamento.

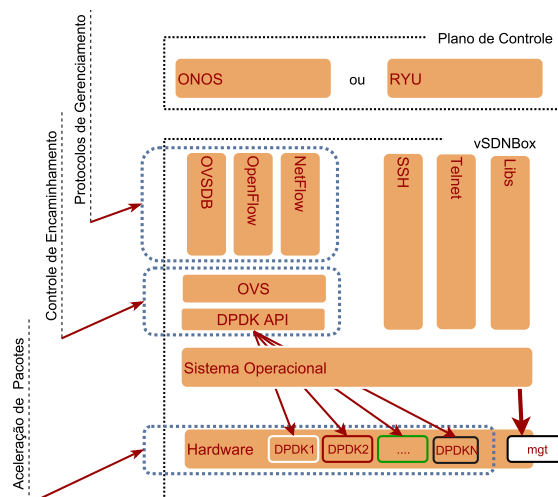


Figura 3. Visão geral da arquitetura do vSDNBox

A camada de aceleração de pacotes é responsável pela gerência dos recursos de cada interface de rede, como: quais núcleos são responsáveis por processar as filas de entrada e saída e/ou memória disponíveis para caches de dados. Nesta iniciativa, a aceleração de pacotes é feita por núcleos do processador, e ignorando algumas camadas de virtualização e de *kernel*. O resultado disso é observado na ampliação da vazão, e na diminuição da latência de processamento de pacotes nas interfaces de redes.

Apesar de existirem outras iniciativas que permitem a aceleração do plano de dados, como: DPDK, *Netmap* [Rizzo 2012], FD.io [FD.io 2018] e XDP-eBPF [XDP 2018]. Tanto a *Netmap* quanto o FD.io são soluções com poucas ou quase nenhuma interoperabilidade com outras ferramentas SDN conhecidas são quase inexistentes. Porém, o XDP-eBPF vem apresentando muitas vantagens, principalmente na questão de desempenho quando comparado ao DPDK, mas a sua aplicabilidade prática ainda é limitada. Logo devido ao fato do DPDK ser uma solução que possui uma maior compatibilidade com as diversas soluções SDN, como também, apresenta o suporte a várias interfaces de redes genéricas. Então, optou-se pelo seu uso na implementação.

Já na camada de controle de encaminhamento, destacado na Figura 3, aplica-se as decisões ou ações nas interfaces aceleradas via software, por exemplo, descarte de pacotes, encaminhamento para outra interface física ou virtual, ou até a retenção da taxa de transmissão nas filas de entrada e saída das portas. Esta camada faz uso do *Open vSwitch* (OVS) [Pfaff et al. 2015] que é uma ferramenta capaz de controlar toda a dinâmica de encaminhamento de pacotes, a partir de comandos oriundos de um controlador SDN, como também, a criação e remoção de características do equipamento, por exemplo, uma bridge ou uma porta virtual para um determinado protocolo (Ethernet ou GRE). Nesse contexto, optou-se pelo OVS por ser compatível com vários protocolos tradicionais (ex. *Ethernet*, *GRE* e *IPv4*), bem como novos protocolos (ex. *VXLAN*, *Geneve*, *NVGRE* e *STT*).

Por fim, na camada de protocolos de gerenciamento tem-se a instanciação dos meios pelos quais é possível controlar e gerenciar remotamente o vSDNBox. Para o controle dos fluxos de pacotes usa-se especificamente o protocolo *OpenFlow*, onde o comportamento dos mesmos é organizado em tabelas do tipo *match-action*, onde o *match* consiste de um conjunto de informações a respeito do cabeçalho do pacote e/ou metadados que são comparados durante a chegada do mesmo. Após a realização do *match* aplica-se um *action* que é a atribuição que trata o comportamento dos fluxos quando a

avaliação do *match* é verdadeira. Porém, para a gerência dos recursos do equipamento utiliza-se o protocolo OVSDB [Pfaff et al. 2015], que basicamente permite alocar e desalocar recursos no *vSDNBox*, por exemplo, criar e deletar interfaces virtuais, obter informações de estatísticas tanto da interface quando da tabela de encaminhamento e configurar parâmetros das interfaces de redes.

O modelo de hardware do *vSDNBox* adotado atende as demandas de infraestruturas de pequenos e médio porte de tráfego de dados. Por isso, resolveu-se criar duas gerações de *vSDNBox*. Na primeira geração, o foco está em obter um hardware mínimo necessário para manter o máximo de transmissão em todas as portas do dispositivo. Portanto, nesta geração, resolve-se trabalhar com interfaces 1 *GbE* (*Gigabit Ethernet*) com no máximo 6 portas, além de um processador com o número de núcleos igual ou maior que o número de portas. A descrição completa do hardware é feita na Tabela I.

Tabela 1. Descrições das configurações dos *vSDNBox* nas gerações 1 e 2

	1º Geração	2º Geração
	Descrição	
Modelo de Placa Mãe	Supermicro 5018-TN7B	Supermicro 5018D-FN8T
Processador	Intel Atom C2758 (8 Núcleos) 2.40 GHz, VT-x	Intel Xeon D-1518 (4 Núcleos e 8 Thread) com suporte a virtualização (VT-x e VT-d), HyperThreads, 1G HugePages
Portas Ethernet	4 x GbE LAN Intel C2000	2 x 10 GbE SFP+ from D-1500 SoC
	2 x GbE LAN Intel i350-AM2	4 x 1 GbE with Intel I350-AM4 1 x 1 GbE with Intel I210
Portas de Gerenciamento	1 x GbE LAN Intel i210	1 x 1 GbE with Intel I210
Memória	8 Gbit DDR3 SDRAM SODIMM	16 Gbit DDR4 SDRAM SODIMM

Na segunda geração, foca-se na ampliação do desempenho no processamento de pacotes, como também, na extensão das características de virtualização tornando-as mais nativas que na geração anterior. Por isso, o processador foi atualizado para suportar atributos específicos de virtualização nativa e processamento de alto desempenho (*VT-d* e *Hyperthread*). Além disso, essa atualização foi necessária para adequação de processamento de pacotes com interfaces de 10 *GbE*. Apesar de serem apresentadas as gerações de 1 e 10 *GbE*, neste trabalho, foca-se apenas na apresentação de resultados da primeira geração, ficando para trabalhos futuros os resultados com 10 *GbE*.

Nos testes iniciais com *vSDNBox*, observou-se que a utilização do sistema operacional, em sua instalação padrão, contribuía para a perda de desempenho no processamento dos pacotes das interfaces DPDK. Isso acontecia devido ao fato dos núcleos disponibilizados ao DPDK ainda estarem vinculados para a alocação de tarefas do sistema operacional, portanto, todo processamento de pacotes concorria com o interrupções e processos locais do sistema operacional, conforme ilustrado na Figura 4(a). Como consequência disso, o desempenho era afetado drasticamente, tanto aumentando a latência quanto diminuindo a vazão de pacotes (*throughput*).

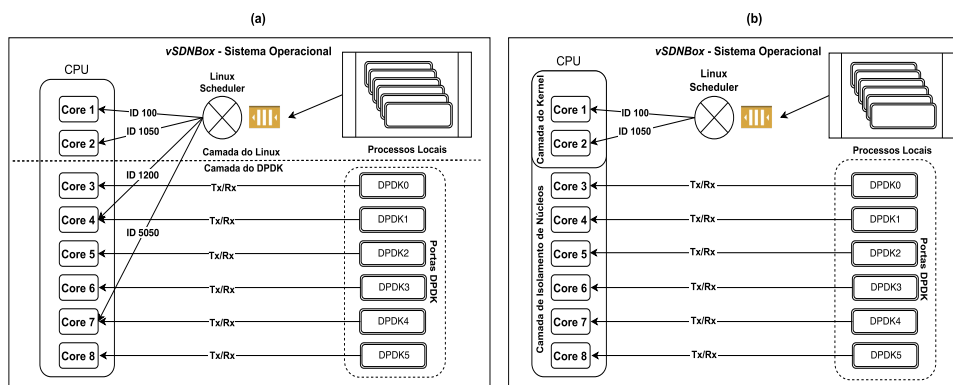


Figura 4. Ajustes no sistema operacional para otimizar a latência e taxa de processamento de pacotes

Para resolver este problema foi necessário efetuar ajustes no sistema operacional com o objetivo de que os núcleos, separados para o DPDK, ficassem isolados exclusivamente para processar as filas de entrada e saída de cada interface de rede DPDK e os demais processos do sistema operacional fossem direcionados para núcleos específicos que não pertenciam ao DPDK, conforme ilustrado na Figura 4(b). Desta forma, conseguiu-se isolar os núcleos alocados ao DPDK dos processos locais, e consequentemente ampliou-se o desempenho em termos de latência e taxa de processamento de pacotes.

Na análise do custo de implementação da proposta, fez-se a comparação entre as duas gerações do *vSDNBox* e o *whitebox hardware-switched* da empresa *Pica8* e modelo P-3297 de 48 portas. Ele foi escolhido por ter características muito similares ao *vSDNBox* como: sistema operacional baseado em Linux e a utilização do *Open vSwitch* na camada de encaminhamento.

Para comparação foram analisados custo com hardware e softwares para que o dispositivo funcionasse com todas as suas características habilitadas. A primeira diferença que se observa é que na maioria dos *whiteboxes* baseados no modelo *hardware-switched* há custos adicionais relacionados ao sistema operacional tanto para uso como para habilitar certas funcionalidades. Por exemplo, o *PICOS* pode ser adquirido e instalado gratuitamente, porém, há limitação de uso para 4 portas GbE e somente uma *bridge*. Enquanto no *vSDNBox* não há essa obrigatoriedade de pagamentos adicionais para liberação de funcionalidades, tudo está disponível abertamente.

Quanto aos custos, apresentados na Tabela 2, observa-se uma diferença de quase US\$5.000,00 dólares entre o modelo *Pica8* quando comparado com *vSDNBox* de 1º geração e de US\$4.900,00 dólares comparado ao *vSDNBox* de 2º geração, não foram levados em consideração os valores com frete, impostos e conversão monetária com o objetivo de facilitar o entendimento.

Tabela 2. Custo comparativos entre o vSDNBox e o P-3297

Whitebox	Hardware	Softwares	Total
<i>vSDNBox</i> 1º Geração	Software-Switched, 6x 1 GbE Portas (\$ 551,99) + Suporte a virtualização	<i>Linux Fedora 27</i> + OVS + DPDK (\$ 0) Licença Open Source GNU	\$ 551,99
<i>vSDNBox</i> 2º Geração	Software-Switched, 2x 10 GbE Portas + 6x 1GbE (\$ 761,35) + Suporte a virtualização	<i>Linux Fedora 27</i> + OVS + DPDK (\$ 0) Licença Open Source GNU	\$ 761,35
<i>Pica8 P-3297</i>	<i>Hardware-Switched</i> , 48x GbE portas (\$ 4.400,00) sem suporte a virtualização	PicOS + OVS (\$ 1.259,00) Licença 1 ano	\$ 5.659,00

5. Análise da Proposta

Esta seção tem o objetivo de validar e avaliar o aa latência e a vazão de pacotes no *vSDNBox* de 1º geração utilizando a arquitetura proposta na Seção 4.1 e compará-la com os resultados obtidos com o *whitebox* sem o uso da aceleração DPDK e, também, com o *whitebox* baseado em *hardware-switched*.

5.1 Cenário Avaliado

Para avaliar o desempenho do *vSDNBox* da 1º geração foram desenvolvidos dois cenários de avaliação, conforme ilustrado na Figura 5. No primeiro cenário, são utilizados apenas uma *bridge* e duas portas físicas de cada equipamento com o objetivo de avaliar o desempenho em um cenário simples.

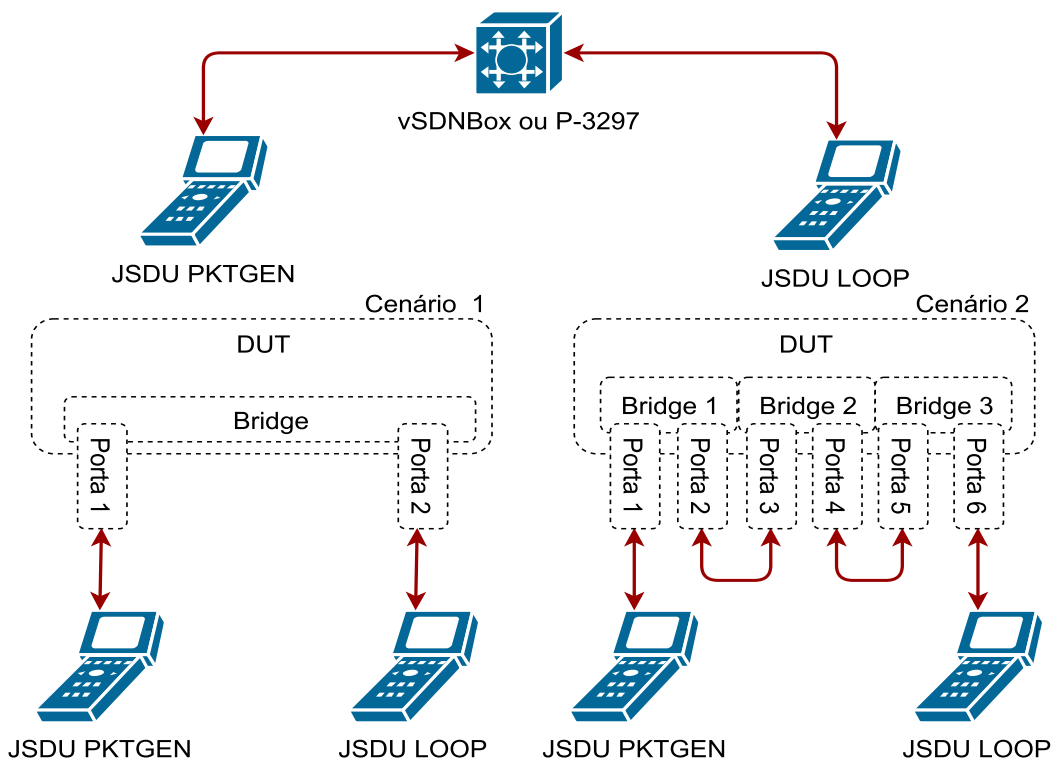


Figura 5. Visão geral dos cenários utilizados para avaliação de desempenho

No segundo cenário, são utilizados três *bridge* e seis portas físicas. Neste caso, o objetivo é avaliar o *vSDNBox* quando todas as portas necessitam de processamento de pacote e com isso avaliar a latência da realização deste processamento, e também verificar a máxima taxa de transmissão sem que haja perda de pacotes.

Para a geração de tráfego foi utilizado o equipamento *SmartClass Ethernet*¹ da empresa VIAVI. Nos testes foram utilizados dois equipamentos, como ilustrado na Figura 5, o primeiro equipamento gera o tráfego (JSDU PKGEN) e o segundo (JSDU LOOP) encaminha todos os pacotes recebidos de volta ao primeiro equipamento. Portanto, todo tráfego utilizado nos testes são bidirecionais e os valores apresentados é a média dos valores obtidos em cada repetição dos testes. Os valores de referência para parâmetros configurados nos equipamentos são baseados na RFC 2544 [Bradner e McQuaid 1999] e estão listados na Tabela 3.

¹ Smart Class Ethernet: <https://www.viavisolutions.com/pt-br/produtos/discontinued/smartclass-ethernet>

Tabela 3. Parâmetros de configurações do gerador de tráfego

Parâmetro	Tipo de Teste	
	Taxa de Processamento	Latência
Tamanhos de Pacotes	64 – 9000 Bytes	64 – 9000 Bytes
Taxa de Transmissão	1000 Mbps	1000 Mbps
Tempo de Duração	60 segundos	60 segundos
Número de Repetições dos Testes	20 Vezes	20 vezes

Por fim, no plano de controle foi utilizado o controlador RYU para fazer o gerenciamento dos redirecionamentos dos tráfegos nas portas físicas, como ilustradas na Figura 5, e também monitorar as estatísticas das portas físicas para compará-las com o equipamento gerador de tráfego.

5.2 Resultados

5.2.1 Resultados do Cenário 1

Conforme descrito na seção anterior, a Figura 6 apresenta os resultados obtidos através das avaliações de desempenhos feitos no *vSDNBox* e no P-3297. Nas Figuras 6 (a) (b) (c), tem-se os resultados de vazão, onde observa-se as vantagens alcançadas pela aceleração via o DPDK. Isto possibilitou que a proposta alcançasse o limite da sua taxa de transmissão (1Gbps - Gigabit por segundo), tanto para pacotes pequenos, no caso de pacotes de 64 Bytes, quanto em pacotes grandes (9000 Bytes). Além disso, também nota-se, nesta avaliação de vazão, que o seu desempenho é equivalente ao equipamento *hardware-switched* (P-3297). Portanto, pode-se concluir que no quesito vazão tanto o *vSDNBox* quanto P-3297 possuem o mesmo desempenho, ainda que a aceleração aplicada no *vSDNBox* seja executada no espaço do usuário do sistema operacional.

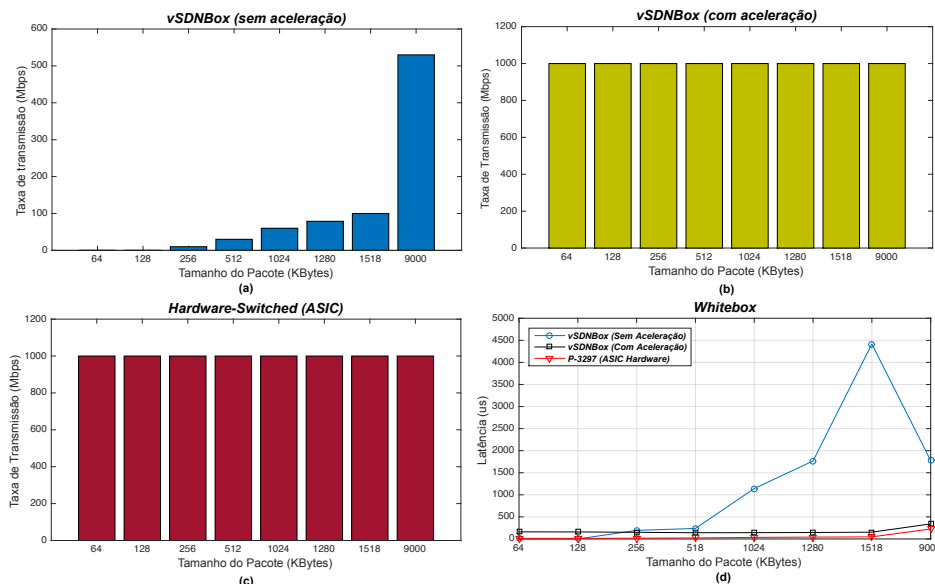


Figura 6. Resultados do Cenário 1

Na Figura 6 (d), observa-se os resultados relacionados às latências. Neste caso, quando comparado o *vSDNBox* com e sem aceleração, observa-se uma diminuição no tempo de processamento dos pacotes, devido a dois fatores. O primeiro é a ausência da sobrecarga de processamento dos pacotes quando há a transição do espaço do usuário

para *kernel*. No segundo fator, o processamento das filas de entrada e saída estão integradas exclusivamente a um conjunto de núcleos isolados do processador, sem interferências de processos ou tarefas locais, e operando de forma dedicada à essas filas. No entanto, houve a necessidade de se fazer um balanceamento da distribuição desse processamento dentro do conjunto de núcleos para manter os valores de latência dentro do patamar de 100 μ s (micro segundos) mediante o aumento no tamanho dos pacotes, ao contrário do que acontece quando não se usa aceleração. Além disso, ao se observar seu desempenho em relação ao dispositivo *hardware-switched* percebe-se um comportamento muito similar, com uma pequena diferença que favorece o P-3297. De uma certa esse resultado já era esperado, dado que o primeiro trabalha a nível de hardware e utiliza um processador superior ao usado no *vSDNBox* de 1º geração. Por isso, com atualização do processador no *vSDNBox* de 2º geração almeja-se obter resultados iguais ou melhores que o P-3297.

5.2.2 Resultados do Cenário 2

Para este cenário pretende-se observar o comportamento do *vSDNBox* utilizando todas as suas portas e com isso perceber como se comporta o processamento de pacotes na utilização de todos os seus recursos, a Figura 7 ilustra os resultados obtidos no cenário 2.

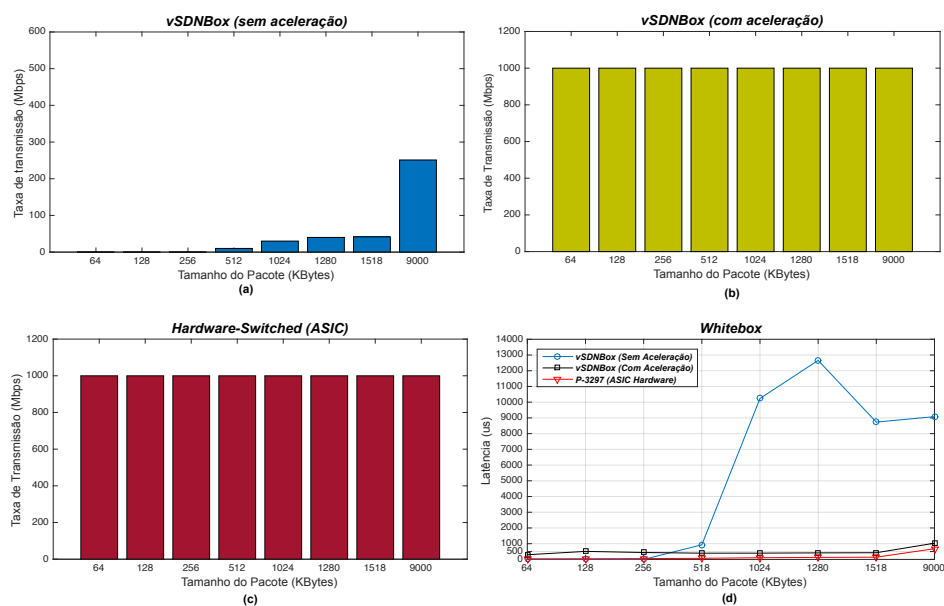


Figura 7. Resultados obtidos no cenário 2.

Observando a vazão na Figura 7 (a) (b) (c), o *vSDNBox* (sem aceleração) não consegue manter a taxa de transmissão sem perdas para pacotes com tamanhos de 64, 128 e 256 Bytes. No entanto, com o uso da solução proposta, a taxa de utilização ou rendimento chega ao máximo permitido pela interface, igualando ao resultado obtido pelo dispositivo *hardware-switched* quando se compara as Figuras 7 (b) e 7 (c). Com isso, valida-se que mesmo em um cenário de carga máxima a proposta consegue ter o mesmo desempenho que um *whitebox hardware-switched*.

Na Figura 7 (d), ilustra-se os resultados de latência que mostram que o valor permanece praticamente constante independente do tamanho dos pacotes que são recebidos ou transmitidos em suas filas. Comportamento similar ao apresentado pelo *hardware-switched*, o qual por ter um hardware específico para este fim, oferece uma latência menor. Contudo, a proposta consegue ter um tempo de resposta curto e constante,

mesmo quando comparado ao P-3297, o que permite concluir que a proposta mesmo em um cenário de congestionamento consegue manter seu desempenho ao máximo em todas interfaces e com latências baixas mostrando a viabilidade de seu uso, seja em um cenário simples ou mais complexo de carga de trabalho.

6. Conclusões e Trabalhos Futuros

Os *whiteboxes* estão evoluindo, acompanhando as crescentes inovações e investimentos em SDN. Além disso, essa categoria de dispositivos vem sendo uma peça fundamental no desenvolvimento do modelo SDN, principalmente por conta da sua abertura, de forma a permitir a evolução dos seus componentes de software sem a necessidade de depender de fabricantes e permitindo o desenvolvimento desse ecossistema de forma aberta.

A proposta *vSDNBox* foi desenvolvida para oferecer principalmente: baixo custo, removendo a dependência de licenças proprietárias e necessidade de hardwares caros, mas ao mesmo tempo buscando-se ampliar o desempenho da proposta. Nesse contexto, os testes realizados com o *vSDNBox* apresentaram que é possível se ter *whiteboxes* por custos ainda mais reduzidos e com desempenhos similares ou até iguais a outros comutadores que possuem hardwares específicos para estas funções.

Um ponto forte da proposta é a arquitetura apresentada, que permite total liberdade na manipulação dos softwares utilizados e também em seu modelo de gerenciamento, podendo ser alterado a medida que o administrador ou desenvolvedor necessite modificá-lo, seja para um modelo mais recente e eficiente ou manter o modelo, já em uso, sempre atualizado.

Os resultados demonstram que a proposta garante elevados valores de desempenho com baixos tempos de processamento, tanto em cenários simples como também em cenários mais complexos, onde todas as portas são utilizadas simultaneamente. Também se observa o forte impacto no desempenho que a proposta apresenta quando comparada ao *software-switched* sem aceleração, tanto na vazão quanto na latência. Além disso, a cada nova geração do *vSDNBox* espera-se que os resultados de desempenho se aproximem cada vez mais dos modelos que utilizam *hardware-switched*, mantendo, porém, a característica de custos bastante reduzidos.

Para trabalhos futuros, pretende-se replicar a metodologia de avaliação de desempenho para o *vSDNBox* de 2º geração e provar que a utilização de modelos mais avançados de processadores e novas tecnologias de virtualização permitirão obter valores ainda melhores que os obtidos pela 1º geração. Além disso, almeja-se avaliar o desempenho da solução em novos cenários com diferentes tipos de pacote (ex. MPLS, VLAN, VXLAN ou GENEVE). Por fim, deseja-se também avaliar o uso de virtualização e seu impacto no desempenho do *vSDNBox*, avaliando assim os limites suportados nesse contexto.

7. Agradecimentos

Os autores agradecem a Rede Nacional de Pesquisa (RNP) pelo apoio financeiro, a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa e pelo apoio financeiro e a Universidade Federal do Pará (UFPA) pelo suporte tecnológico.

Esta pesquisa foi patrocinada pelo H2020 na 4º chamada de colaboratividade EU-BR, no acordo nº 777067 (*NECOS - Novel Enablers for Cloud Slicing*), financiada pela Comissão Europeia de Tecnologia e Ministério da Ciência e Tecnologia.

Referências

- Bradner, S.; McQuaid, J. **Benchmarking Methodology for Network Interconnect Devices**. 1999. Disponível em: <<https://www.ietf.org/rfc/rfc2544.txt>>. Acesso em: 19 mar. 2018.
- Cox, J. H. et al. (2017). Advancing Software-Defined Networks: A Survey. *IEEE Access*, v. 5, p. 1–1.
- DPDK. **Data Plane Development Kit**. 2018. Disponível em: <<https://dpdk.org/>>. Acesso em: 19 mar. 2018.
- FD.io. **The Universal Data Plane**. 2018. Disponível em: <<https://fd.io/>>. Acesso em: 19 mar. 2018.
- Kreutz, D. et al. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, v. 103, n. 1, p. 14–76.
- Lara, A.; Kolasani, A.; Ramamurthy, B. (2013). Network Innovation using OpenFlow: A Survey. *IEEE Communications Surveys & Tutorials*, v. PP, n. 99, p. 1–20.
- Mangala, A. W.; Hendrawan; Tanwidjaja, A. Performance analysis of white box switch on Software Defined Networking using open vSwitch. In: 2015 1ST INTERNATIONAL CONFERENCE ON WIRELESS AND TELEMATICS (ICWT) 2015, **Anais... : IEEE**,(2015)
- P4. **P4 Runtime**. 2018. Disponível em: <<https://p4.org/>>. Acesso em: 19 mar. 2018.
- Pfaff, B. et al. (2015). The Design and Implementation of Open vSwitch. *Nsdi*, p. 117–130.
- PICA8. **Bare Metal Networking- Leveraging “ White Box ” Thinking**. 2018. Disponível em: <<https://fd.io/>>. Acesso em: 19 mar. 2018.
- PICOS. **White Box Networking in Remote and Branch Offices**. 2018. Disponível em: <<http://www.pica8.com/documents/white-box-networking-remote-branch-offices-20170814.pdf>>. Acesso em: 19 mar. 2018.
- Rizzo, L. (2012). NetMap: A novel framework for fast packet I/ O. *Atc '12*, n. 257422, p. 101--112.
- Shahbaz, M. et al. (2016). PISCES: A programmable, protocol-independent software switch. *2016 ACM Conference on Special Interest Group on Data Communication, SIGCOMM 2016*, p. 525–538.
- Shin, M. K.; Nam, K. H.; Kim, H. J. (2012). Software-defined networking (SDN): A reference architecture and open APIs. *International Conference on ICT Convergence*, p. 360–361.
- Tan, J. et al. (2017). VIRTIO-USER: A New Versatile Channel for Kernel-Bypass Networks. p. 13–18.
- XDP. **eXpress Data Path**. 2018. Disponível em: <<https://www.iovisor.org/technology/xdp>>. Acesso em: 19 mar. 2018.