

# Parâmetros balizadores para experimentos com comutadores *OpenFlow*: avaliação experimental baseada em medições de alta precisão.

Rafael Emerick Z. de Oliveira, Pedro P. Piccoli Filho,

Moisés R. N. Ribeiro, Magnos Martinello

**Resumo**—A medida de latência, em escala de microssegundo, e a perda de quadros impostas pelas implementações correntes de comutadores *OpenFlow* são objeto de estudo desse trabalho. Ao contrário dos trabalhos anteriores que dependem de equipamentos de medição de alto custo, este trabalho utiliza como ferramenta de avaliação de desempenho o *Oflops* em conjunto com gerador de tráfego em *NetFPGA*. Foram detectadas, entretanto, limitações dessa ferramenta em alguns cenários de teste, além de discontinuidades na resposta de implementações padrão em *software* e em *hardware*. Tais informações são úteis no balizamento de experimentos relacionados à Internet do Futuro baseados em plataformas abertas.

**Palavras-Chave**—*OpenFlow*, *Oflops*, *NetFPGA*.

**Abstract**—The latency measurement, in a microsecond scale, and the frame loss ratio imposed by the current open-source implementations of *OpenFlow* switches are the object of study of this paper. Unlike previous studies that rely on high cost measurement apparatus, this paper uses as a tool for performance evaluation *Oflops* in conjunction with the *NetFPGA* traffic generator. There are, however, limitations of that method in some test scenarios, and discontinuities in response *OpenFlow* standard implementations in software and hardware. Such information are useful in planning the experiments related to the Future Internet based on open platforms.

**Keywords**—*OpenFlow*, *Oflops*, *NetFPGA*, *Open vSwitch*.

## I. INTRODUÇÃO

O conceito de programabilidade em redes é baseado na separação entre o plano de controle, responsável pelos protocolos e pelas tomadas de decisão que resultam nas tabelas de encaminhamento, e o plano de dados, que cuida da comutação e do repasse dos quadros na rede. Nos equipamentos de redes legadas, o plano de controle é executado no próprio equipamento a partir dos protocolos implementados no sistema operacional embarcado, impedindo qualquer tomada de decisão baseada em algoritmos que não tenham sido previstos nestes protocolos previamente instalados. Para que seja possível "quebrar" esta restrição, faz-se necessária uma forma de permitir que o equipamento encaminhe os quadros a partir de protocolos implementados externamente, de maneira que o plano de controle seja independente e que não se limite aos protocolos implementados

pelo fabricante. Este é o principal objetivo das *Software Defined Networks (SDN)*.

A tecnologia *OpenFlow* [1] apoia-se nesse conceito de redes definidas por *software*, como uma interface de programação de aplicativos que definem o comportamento do encaminhamento de quadros na rede. Além disso, a tecnologia define um protocolo entre os comutadores e o controlador, que viabiliza o controle logicamente centralizado. Isto promove uma versatilidade no encaminhamento de quadros no plano de dados, controlados externamente por aplicações em execução nos sistemas operacionais de rede, os quais são executados nos controladores. O *OpenFlow* se apresenta como uma proposta de padrão de interface pragmática, com um protocolo definido entre plano de controle e plano de dados, permitindo a instalação de regras de encaminhamento nas tabelas de fluxos dos equipamentos. Estes fluxos são baseados em diversos parâmetros de protocolos de camadas distintas como Ethernet, IP, TCP e UDP, e em versões mais recentes do *OpenFlow*, inclui MPLS. Atualmente ele é a implementação mais difundida tanto no meio acadêmico, como nas implementações de equipamentos e sistemas comerciais de SDN.

Por ser relativamente nova, e oferecer grande versatilidade para prototipação de novas propostas de comunicação, a tecnologia baseada em *OpenFlow* levanta muitos questionamentos a respeito do desempenho oferecido pela solução. Avaliar o desempenho de novas aplicações neste paradigma tem sido um desafio, devido à reduzida oferta de ferramentas de baixo custo e de alta precisão dedicadas à análise do desempenho quantitativo dos comutadores de código aberto disponíveis. Há estudos que analisam questões de desempenho do protocolo em si, propondo uma abordagem que reduz a frequência de acesso ao controlador, o que efetivamente melhora a eficiência da comunicação [2]. Para questões relativas às implementações do *OpenFlow*, outros estudos avaliam o desempenho do protocolo em comparação com abordagens convencionais de encaminhamento e roteamento [3]. Além destes, há estudos que detalham aspectos funcionais de maneira comparativa entre implementação *OpenFlow* em *software*, no caso o *Open vSwitch (OVS)*, implementação em *NetFPGA* e implementações comerciais em *hardware*, basicamente utilizando a versão 1.0 do protocolo [4]. Uma limitação deste último estudo para o fornecimento de parâmetros balizadores para experimentos com comutadores *OpenFlow* reside na ausência de resultados comparativos entre o *OVS* e

implementações em *hardware*. Além disto, as análises utilizaram equipamentos geradores de tráfego, que são de alto custo e pouca versatilidade para explorar novas características e novas propostas de comunicação em redes definidas por *software*. Portanto, há necessidade de plataformas e de métodos para medição e avaliação de desempenho de comutadores *OpenFlow* que suportem alta precisão e sejam de baixo custo, permitindo avaliações customizáveis e provendo uma interface de medição tanto para o plano de dados quanto para o plano de controle.

Atualmente, uma boa opção para este fim é o *framework* de medição *Oflops* [7], uma ferramenta que viabiliza o desenvolvimento rápido de testes de casos de uso para implementações *OpenFlow* tanto em *software* quanto em *hardware*. O *framework* conta com um gerador de quadros que funciona tanto em nível de *software* quanto de *hardware* que viabiliza medições versáteis e com baixo custo de operação. Um PC básico com Linux e uma placa *NetFPGA* Gigabits são os recursos mínimos que viabilizam medições de alta precisão na escala de microssegundos. Neste presente trabalho, buscou-se verificar aspectos funcionais básicos desse *framework*. A principal motivação é conhecer de maneira prática sua real capacidade e suas eventuais limitações. Isto foi feito de maneira complementar à avaliação comparativa entre a implementação *OpenFlow* padrão em *hardware* e em *software*.

O foco do presente trabalho está em analisar o tempo de resposta do encaminhamento no plano de dados de diferentes implementações de comutadores *OpenFlow* disponíveis, de maneira a verificar o comportamento de operações básicas, como troca de VLANs. Dessa maneira, informações quantitativas para futuros experimentos, relacionados às camadas mais baixas dos modelos de comunicação podem ser obtidas. Parâmetros balizadores são necessários na composição de uma solução de comunicação como, por exemplo, a latência de operações de encaminhamento com alterações de cabeçalho. Com este propósito, foram analisados as implementações do comutador *OpenFlow* em *NetFPGA* de Stanford [6], e duas configurações de máquinas rodando o comutador em modo *kernel* do *OVS* [5] para avaliar a sensibilidade do desempenho em relação à capacidade computacional disponível para o comutador. Para garantir uma precisão nas medições em um nível de granularidade abaixo de milissegundos, nossa abordagem apoia-se em *hardware* especializado na forma de *NetFPGA*. A geração de quadros e as medidas são obtidas com base na plataforma de medições *Oflops* [7]. A latência e a perda de quadros são as métricas de desempenho escolhidas para permitir o balizamento dos resultados de aplicações que rodam em soluções abertas de redes definidas por *software*. Mais ainda, esta contextualização é fundamental para experimentos mais complexos relacionados à Internet do Futuro.

## II. METODOLOGIA E AMBIENTE DE EXPERIMENTAÇÃO

A metodologia adotada consiste em utilizar a plataforma de medições *Oflops*. Esta solução suporta três formas de operação: modo usuário, modo *kernel* e modo baseado em *hardware*. Na terceira opção, todo o processo de medição envolvendo a geração, a marcação e a captura dos quadros é realizada por uma implementação em *NetFPGA*, alcançando um nível de

precisão na ordem de microssegundos. A plataforma é composta por cinco *threads* para tratar diferentes tipos de eventos: (1) Geração de Pacotes de Dados; (2) Captura de Pacotes de Dados; (3) Canal de Controle; (4) Canal SNMP, que realiza coletas SNMP assíncronas; e (5) Gestão de Tempo. Além disso, o *Oflops* oferece um controlador de rede *OpenFlow* que viabiliza medição do desempenho de interações relativas ao plano de controle dos comutadores.

A arquitetura da plataforma de medição está apresentada na Figura 1. Nesta, percebe-se que a medição é feita de maneira ativa com a conexão direta do comutador à plataforma de medição. O plano de dados do comutador é conectado às interfaces geradoras de tráfego, o qual pode ser gerado de duas maneiras: via módulo de *kernel* (*pcap*) ou via gerador de tráfego em *hardware* implementado em *NetFPGA*. Somente no último caso é possível alcançar precisão em nível de microssegundo na medição do tempo de resposta do comutador no plano de dados. Paralelamente, pode-se realizar medições no plano de controle da rede, assim é possível medir tempo de instalação de regras, tempo de efetivação de regras, etc. Na camada intermediária, da Fig. 1, temos a plataforma do *Oflops* em si. Essa camada de *software* é responsável por viabilizar que os módulos criados pelo usuário sejam executados via compartilhamento de bibliotecas. A linguagem C é usada para programação dos módulos. Nestes, é feito o processamento da lógica dos testes, a caracterização do tráfego gerado, além das contabilizações e estatísticas necessárias nos experimentos.

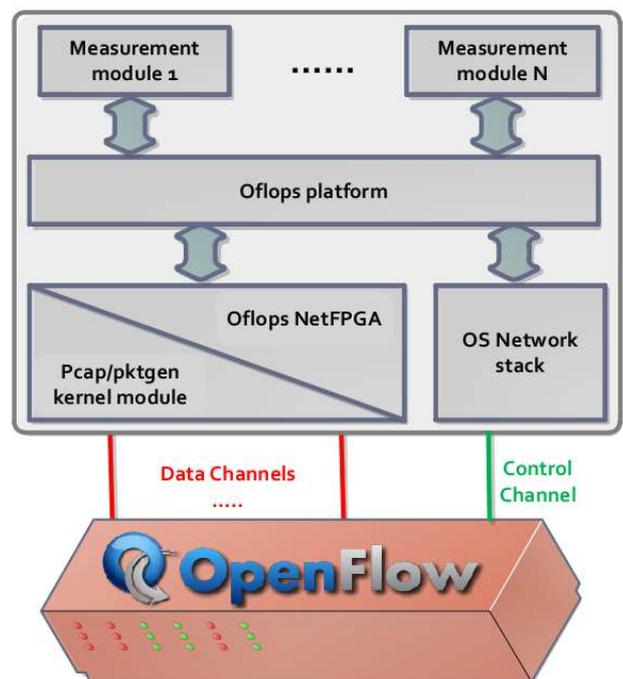


Fig. 1. Arquitetura do *Oflops*, plataforma de medição utilização[7].

Para o *Oflops* foi utilizado um PC de hospedeiro, com processador AMD Athlon IIX4 640 3Ghz, como barramento em 800Mz e palavra de 64bits, com cache L1 e L2, respectivamente de 512 KB e 2MB, 4GB de RAM DIMM

DDR2 com clock em 400MHz e placa *NetFPGA* com gerador de quadros do *Oflops*.

Para os comutadores avaliados, utilizou-se um comutador *OpenFlow* em *hardware*, na *NetFPGA* 1Gbps, com a implementação padrão de Stanford, e a implementação *OpenFlow* em *kernel* Linux do *OpenVswitch* 1.4.0, instalando duas opções de configuração de PC, conforme resume a Tabela I.

Utilizou-se, para o *testbed*, o módulo *Action Delay* disponibilizado pelo *Oflops*. O módulo instala uma regra no comutador (e.g., alteração de VLAN), e então, gera um volume grande de quadros (i.e., mais de 30.000) para executar as medições. A partir daí, as informações de *timestamps* obtidas do gerador de quadros são processadas pelo módulo via *framework Oflops* e obtêm-se, então, a média, o desvio padrão e a taxa de perda de quadros.

TABELA I- DETALHES DA CONFIGURAÇÃO DOS COMUTADORES *OPENFLOW*

Sw Openflow	CPU*	Memória*	Interface Rede (plano de dados)
OVS1	AMD Athlon(tm) 64 3GHz 128KB(L1) 512MB(L2)	2GB DDR2 667MHz	Eth1/Eth2: RTL-8169 Gigabit Ethernet 1Gbps clock 66MHz 32bits
OVS2	Intel(R) Xeon(R) CPU 3075 @ 2.66GHz 32KB(L1) 4MB(L2)	4GB DDR2 800MHz	Eth1:82572EI Gigabit Ethernet 1Gbps clock 33MHz 32 bits Eth2:RTL-8169 Gigabit Ethernet 1Gbps clock 66MHz 32bits
NetFPGA	AMD Athlon(tm) II X4 128KB(L1) 512KB(L2)	4GB DDR2 400MHz	Portas Gigabits da NetFPGA

\*PC Host

### III. RESULTADOS OBTIDOS

Para quantificar os possíveis valores inerentes ao sistema de medição, o primeiro passo foi ligar as interfaces do *Oflops* entre si, de maneira que os quadros não era enviados até os comutadores alvo, gerando um *loopback*. A Fig. 2 apresenta resultados para o sistema de medição no qual o tráfego é gerado a uma taxa constante, variando-se o tamanho dos quadros. Até o tamanho de quadro de 400 bytes a latência na plataforma de medição é zero. Em seguida, aumenta-se gradativamente o tamanho do quadro, tendo como resultado 1µs para quadros até 1400 bytes, e 2µs para valores superiores. Nota-se que o desvio padrão das medições é muito inferior ao valor médio, por isso não é visível nos pontos apresentados na Fig. 2.

No caso da Fig. 3, a carga oferecida é variada para um tamanho fixo de quadro, em 150 bytes. Considerando valores superiores a 580Mbps, houve um aumento significativo, tanto da média como do desvio padrão inerente ao sistema de medição.

As limitações intrínsecas observadas no *loopback do Oflops* operando sobre a *NetFPGA*, nas Fig. 2 e Fig. 3, possivelmente estão associadas à forma de processamento dos cabeçalhos realizados na interação entre a SRAM e a implementação do gerador de pacotes na FPGA. No primeiro caso, da Fig. 2, nota-se um incremento determinístico de

latência causado, possivelmente, durante o preenchimento da *payload* do quadro. O *Oflops* somente escreve o conteúdo variável do cabeçalho dos quadros na SRAM, o restante do preenchimento é feito pela implementação do gerador de tráfego em FPGA. Para quadros grandes, possivelmente a estrutura de preenchimento deve utilizar mais ciclos de *clock*, o que pode justificar o incremento praticamente determinístico no tempo de resposta. Da mesma forma, na Fig. 3, a sobrecarga na SRAM da *NetFPGA* pode ser a responsável pela incremento da latência observada para taxa a partir de 580Mbps (cerca de 480K pacotes por segundo). Entretanto, a julgar pela variabilidade expressa pelo desvio padrão da Fig. 3, deve haver um processo de enfileiramento quando em regime de sobrecarga.

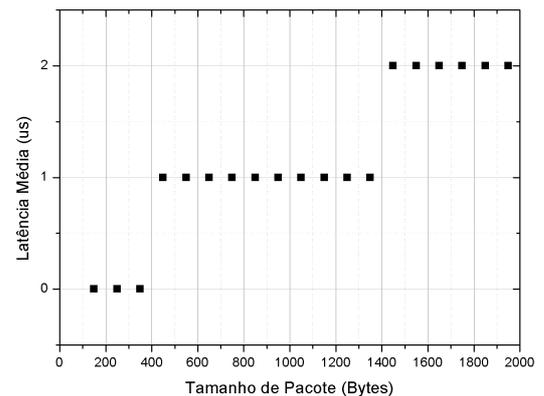


Fig. 2. Avaliação com a carga em 10Mbps e variação do tamanho do quadro.

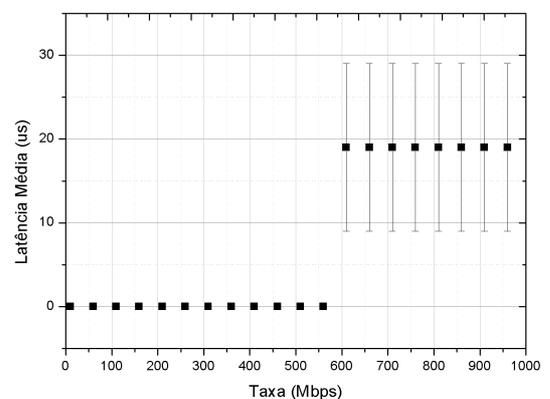


Fig. 3. Avaliação com o tamanho de quadro em 150 bytes e variação na carga de tráfego.

Uma vez caracterizado o desempenho em *loopback*, pode-se agora avaliar o desempenho comparativo dos comutadores *OpenFlow*. A Fig. 4, traz a latência média *versus* o tamanho de pacote para uma carga fixa de 10Mbps. Percebe-se que a diferença dos recursos disponíveis entre os comutadores *OVS1* e *OVS2*, listados na Tabela 1, não tem impacto significativo no tempo de resposta para variação de tamanho de quadros. O

desempenho entre cerca de 30 a 90  $\mu$ s no repasse dos quadros. Entretanto, o *OVS* apresentou alta latência para quadros superiores a 1500 bytes, devido à fragmentação imposta pela configuração da interface de rede da máquina hospedeira, elevando a latência em cerca de 3 ordens de magnitude. Para o comutador *OpenFlow* em *NetFPGA*, tem-se um desempenho uniforme, atingindo latências inferiores a 10 $\mu$ s para quadros até 800 bytes.

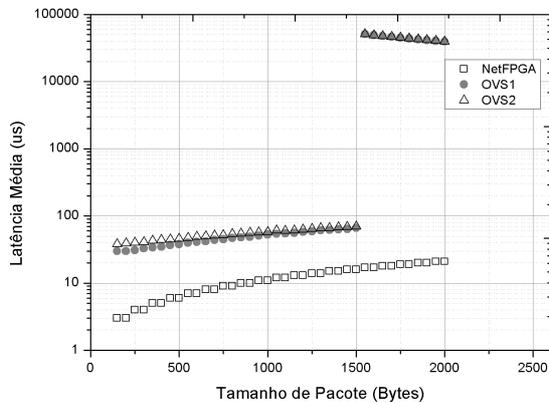


Fig. 4. Variação de tamanho de quadro com carga em 10Mbps.

A taxa de perda de pacotes foi, também, avaliada e o resultado é apresentado na Fig. 5. Para taxa fixa em 10Mbps obteve-se valor nulo para a perda de quadros limitados a 1500 bytes. Acima deste valor, a fragmentação dos quadros impediu o Oflops de obter medidas de perda de quadro. Nota-se, ainda, que em 10Mbps o desvio padrão das medidas é muito menor que a média, e por isso não é visível na Fig. 4 e Fig. 5.

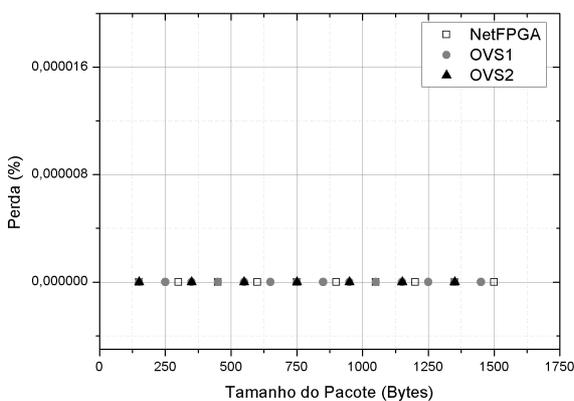


Fig. 5. Variação de tamanho de quadro com carga em 10Mbps

Pode-se notar pela Fig. 6 que o comutador *OpenFlow* em *NetFPGA* é 10 vezes mais rápido no repasse dos quadros do que o *OVS* para taxa até, aproximadamente, 260Mbps. A partir deste valor, a superioridade no tempo de resposta é de quase quatro ordens de grandeza. Na variação de carga, percebeu-se que o comutador em *NetFPGA* manteve a latência constante. A alteração apresentada na Fig. 6, para taxa em torno de

580Mbps, deve-se a incremento inerente do sistema de medição mostrado na Fig. 3. No caso dos comutadores com o *OVS*, a disponibilidade de memória mostrou-se significativa para o redução da latência média. A quantidade de memória e a velocidade de barramento diferenciaram o desempenho das duas implementações do *OVS*, favorecendo o *OVS2*, uma vez que este possui o dobro de memória e apresenta um barramento superior, conforme Tabela 1.

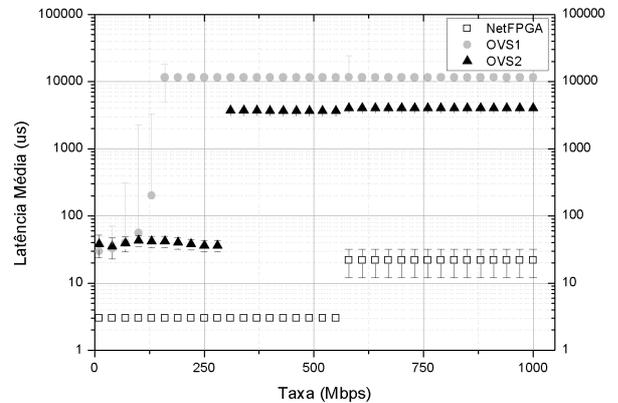


Fig. 6. Variação da vazão com o tamanho de quadro em 150 bytes

Em relação a taxa de perda de quadros, na Fig. 7, verificou-se que o comutador em *NetFPGA* manteve taxa de perda nula. Para o *OVS1*, um pequeno acréscimo é observado a partir de 125Mbps, o qual coincide com o aumento de latência na Fig. 6. Todavia, para o *OVS2* notou-se uma expressiva taxa de perda de quadros, chegando a valores superiores a 10%. Esta incongruência de resultados entre latência e perda de quadros constata a capacidade do *framework* de medição de detectar a influência de pequenas nuances de hardware. Note na Tabela 1, que as interfaces de rede do *OVS1* são idênticas, e para o *OVS2* temos heterogeneidade onde uma das placas é *onboard* de inferior capacidade de barramento. Portanto, no repasse de quadros entre estas interfaces, sob aumento de carga, percebeu-se o pior desempenho do *OVS2* em relação ao *OVS1*.

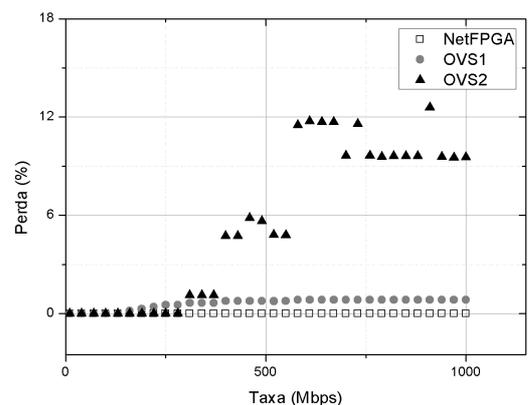


Fig. 7. Variação da carga de tráfego com o tamanho de quadro em 150 bytes

## IV. CONCLUSÃO

O *framework* *Oflops* provê instrumentação em *hardware* *NetFPGA* permitindo testar as capacidades dos comutadores *OpenFlow* com precisão em nível de microssegundos. Entretanto, este *framework* não havia ainda sido testado em relação aos quesitos de latência inerente. Avaliou-se, neste trabalho, o *Oflops* sob dois cenários de carga. Posteriormente, foi analisado o desempenho de três diferentes comutadores *OpenFlow* com implementações abertas em *software* e *hardware*.

Os resultados obtidos mostram que o *Oflops* apresenta significativa latência inerente a partir de uma taxa de 580Mbps para o ambiente utilizado. Isso se deve, provavelmente, a detalhes de implementação de *hardware* do gerador de tráfego envolvendo suas interações com a memória SRAM da *NetFPGA*.

A análise dos comutadores *OpenFlow* mostrou que a latência média de comutação referente à implementação *OpenFlow* em *NetFPGA* é inferior a 10 $\mu$ s independentemente do tamanho do pacote, considerando um tráfego de até 1 Gbps. Se o comutador for implementado em *software*, como é o caso do *OVS*, o desempenho da latência média para operações de alteração de VLAN e de comutação foi subiu para, aproximadamente, 10 milissegundos para taxas superiores a 500Mbps.

Como trabalhos futuros, primeiramente, planeja-se descer em nível de implementação na *NetFPGA* para uma investigação mais detalhadas a respeito das limitações do gerador de tráfego para os cenários utilizados nesse trabalho.

Além disso, planeja-se estender a análise dos comutadores *OpenFlow* incluindo outras implementações comerciais e ampliando o escopo dos experimentos, com cenários de engenharia de tráfego e testes de desempenho sob falhas, tanto no plano de dados como no plano de controle.

## REFERÊNCIAS

- [1] McKeown, N., et al. "OpenFlow: Enabling Innovation in Campus Networks", March, 2008, <http://www.openflowswitch.org/documents/openflow-wp-latest.pdf>
- [2] Mogul, J. C., Tourrilhes, J., Yalagandula, P., Sharma, P., Curtis, A. R., and Banerjee, S. (2010). Devoflow: cost-effective flow management for high performance enterprise networks. In Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets '10, pages 1:1–1:6, New York, NY, USA. ACM.
- [3] Mateo, M. P. "OpenFlow Switching Performance," July 2009. [http://www.openflow.org/downloads/technicalreports/MS\\_Thesis\\_Polito\\_2009\\_Manuel\\_Palacin\\_OpenFlow.pdf](http://www.openflow.org/downloads/technicalreports/MS_Thesis_Polito_2009_Manuel_Palacin_OpenFlow.pdf).
- [4] Appelman, M., De Boer, M., Van Der Pol, R., "Performance Analysis of OpenFlow hardware" February 2012. <http://staff.science.uva.nl/~delaat/rp/2011-2012/p18/report.pdf>.
- [5] Casado, M., K. T. R. R. and Shenker, S. (2010). Virtualizing the network forwarding plane. In Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow., New York, NY, USA. ACM.
- [6] Naous, J., Erickson, D., Covington, G. A., Appenzeller, G., and McKeown, N. (2008). Implementing an openflow switch on the *NetFPGA* platform. In Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS '08, pages 1–9, New York, NY, USA. ACM.
- [7] Rotsos, C., Sarrar, N., Uhlig, S., Sherwood, R., and Moore, A. W. (2012). *Oflops*: An open framework for openflow switch evaluation. In PAM, pages 85–95.