

Avaliação de Desempenho de Planos de Dados OpenFlow

Leonardo C. Costa¹, Alex B. Vieira¹
Erik de Britto e Silva², Daniel F. Macedo²
Geraldo Gomes³, Luiz H. A. Correia³, Luiz F. M. Vieira²

¹Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF) – Juiz de Fora, MG – Brasil

²Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

³Departamento de Ciência da Computação
Universidade Federal de Lavras (UFLA) – Lavras, MG – Brasil

leonardocosta@ice.ufjf.br, alex.borges@ufjf.edu.br

{erik,damacedo,lfvieira}@dcc.ufmg.br

gccgomes@sistemas.ufla.br, lcorreia@dcc.ufla.br

Abstract. *Software Defined Networks are characterized by decoupling the data plane and the control plane of network switches. The OpenFlow protocol implements this concept, and due to the advantages of lower operation expenditures and greater innovation in the network, it is found today in various equipment sold by many companies. Despite being widely employed in industry and research, in the literature does not exist a systematic evaluation of OpenFlow data plane performance. In this paper we present a comparison of different OpenFlow implementations, both on hardware and software switches, in order to validate the maturity of existing commercial solutions. We also compare some existing equipment operating in OpenFlow mode versus its traditional mode of operation. Results show strong evidences that some vendors implement OpenFlow in software, and also that the choice of implementation significantly impacts the performance of the switch.*

Resumo. *As Redes Definidas por Software (SDN) são marcadas pela separação dos planos de dados e de controle. O protocolo OpenFlow é uma implementação desse conceito que, devido às vantagens de menor custo de operação e maior inovação na rede, é encontrado hoje em diversos equipamentos comercializados por várias empresas. Apesar de ser muito utilizado na indústria e na pesquisa, não existe na literatura uma avaliação sistemática do desempenho do plano de dados OpenFlow. Neste trabalho apresentamos uma comparação de diferentes implementações OpenFlow, tanto em hardware quanto software, tendo em vista validar a maturidade das soluções comerciais existentes no mercado, e se os equipamentos existentes, quando operando em modo OpenFlow, operam com qualidade igual ou superior ao modo tradicional de operação. Os resultados mostram fortes indícios de que alguns fabricantes implementam OpenFlow em software, e também que a escolha da implementação impacta significativamente no desempenho do switch.*

1. Introdução

O paradigma de Redes Definidas por *Software* (SDN) vem mudando a forma de como criar, modificar e gerenciar as redes de computadores [Feamster et al. 2014]. Ao separar o plano de controle do plano de dados, a evolução e a inovação da rede são facilitadas. Assim, dentre outras coisas, novos serviços podem ser implantados mais rapidamente, diminuindo o custo operacional das redes. Além disso, novas pesquisas se tornaram possíveis em ambientes reais, sem que se afetasse o tráfego ou a disponibilidade da rede de produção [Feamster et al. 2014].

Considerando esse contexto, o paradigma SDN tem atraído a atenção tanto da comunidade acadêmica quanto da indústria. Apesar de existirem propostas mais flexíveis de SDN, tais como o POF e o P4 [Song 2013, Bosshart et al. 2014], o OpenFlow, que está intimamente ligado a SDN, tem recebido uma grande atenção da indústria. Parte dessa atenção deve-se a simplicidade de se implementar OpenFlow nos projetos existentes de comutadores e roteadores. Isso despertou o interesse de vendedores de equipamentos, operadores de redes, e de empresas como Google, Microsoft e Facebook, empresas essas que, em 2011, formaram a Open Networking Foundation (ONF), que tem o objetivo de promover a tecnologia OpenFlow e o paradigma SDN no mercado [ONF 2015]. Como consequência do crescente interesse da indústria pelo OpenFlow, cada vez mais equipamentos de rede, como *switches* e roteadores, passaram a suportar o protocolo [Kreutz et al. 2015]. Mais ainda, diversas redes de produção, como a WAN da Google [Jain et al. 2013], bem como redes de proporções continentais como a rede de pesquisa e cooperação na América Latina AmLight [Ibarra et al. 2015] já utilizam SDN.

Entretanto, uma solução inovadora pode não se tornar um padrão da indústria devido a limitações de desempenho, segurança ou outros fatores. Por exemplo, o paradigma de redes ativas, proposto no início do milênio, também permitia a programação de comutadores [Tennenhouse et al. 1997]. Devido à sua escalabilidade limitada e problemas de segurança, não foi adotada em redes comerciais.

Tendo em vista a popularidade do OpenFlow, é importante verificar se as suas implementações existentes, tanto na forma de equipamentos comerciais quanto em implementações via *software open source*, possuem características de desempenho e robustez adequadas para o uso em redes de produção. Apesar de existirem alguns trabalhos na literatura que avaliam o desempenho dos controladores OpenFlow [Khondoker et al. 2014], ainda são raros os trabalhos que enfocam o plano de dados dos *switches* OpenFlow, bem como suas implementações e mecanismos internos [Bianco et al. 2010, Sünner 2011, Appelman and De Boer 2012]. Esses trabalhos são esforços no sentido de avaliar o desempenho do OpenFlow em comparação com os modos normais de operação de um *switch* e também em comparar diferentes implementações de *switches* OpenFlow, mas considerando sua relação com o plano de controle. Existe, portanto, uma lacuna na avaliação de desempenho dos *switches* que implementam o protocolo OpenFlow quando consideramos apenas o plano de dados, descartando qualquer interação entre controlador e *switches*.

Tendo em vista esse contexto, o presente trabalho tem como objetivo avaliar a qualidade de diferentes implementações OpenFlow de *hardware switches* comerciais, além de implementações *open source* de *software switches*. Serão avaliadas métricas tradicionais de desempenho, que levam em conta aspectos como latência e jitter, em de-

terminadas operações do *switch*. Assim, com este trabalho, verificaremos se um administrador de rede poderia comprar um *switch* compatível com OpenFlow com a mesma facilidade que hoje se compra um *switch* Ethernet, ou se a compatibilidade e desempenho das implementações ainda são altamente dependentes do fabricante e modelo do equipamento. Deve-se ressaltar que, nessa comparação, procuramos analisar apenas aspectos referentes ao plano de dados dos *switches*, como encaminhamento de pacotes ou outros mecanismos internos, descartando a influência do controlador nas operações analisadas. Além disso, fizemos comparações entre o desempenho de cada *switch* em seus modos *legacy* (normal) e OpenFlow, para identificar se há vantagem no uso do protocolo OpenFlow. Finalmente, procuramos verificar se é possível identificar, a partir de métricas de desempenho da rede, se as implementações do protocolo são realizadas, de fato, em *hardware*, ou se são apenas instâncias de um *software switch* adaptadas ao mecanismo interno do *switch*. Os resultados mostram que ainda existem diferenças significativas de desempenho e funcionalidade nas implementações em *hardware*.

O restante deste artigo está organizado da seguinte forma. A seção 2 apresenta os trabalhos relacionados. A seção 3 descreve a metodologia de avaliação, enquanto a seção 4 discute os resultados obtidos. A seção 5 conclui o artigo, apresentando as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

O advento do paradigma de Redes Definidas por Software (SDN) vem mudando a forma como projetamos e gerenciamos as redes de computadores. Segundo os trabalhos de [Feamster et al. 2014, Braun and Menth 2014, Sezer et al. 2013, Macedo et al. 2015], o paradigma vem ganhando a atenção nos últimos anos porque, além de abordar a falta de programabilidade em arquiteturas de rede existentes, permitiu a inovação na rede de uma forma mais fácil e rápida, através de sua ideia básica, que é a separação dos planos de controle (que decide como manipular o tráfego) e de dados (que manipula o tráfego de acordo com as decisões estabelecidas pelo plano de controle). Dessa forma, novas pesquisas e desenvolvimentos tecnológicos se tornaram possíveis em redes em produção, sem que se afetasse o tráfego ou a disponibilidade da mesma. Assim, diversos serviços de rede estão sendo repensados, de forma a torná-los mais flexíveis.

O paradigma SDN atraiu a atenção da indústria e da comunidade acadêmica, e parte dessa atenção se voltou para o protocolo OpenFlow. O OpenFlow [McKeown et al. 2008] foi definido como um protocolo aberto para a programação da tabela de fluxos de *switches* e roteadores, o que passou a permitir o particionamento do tráfego da rede de acordo com os interesses de seus operadores. O protocolo ainda pode ser definido como uma API que interconecta os equipamentos de rede (plano de dados) a um controlador (plano de controle) [Feamster et al. 2014, Sezer et al. 2013]. O OpenFlow acabou se tornando a plataforma SDN de referência, utilizada na maioria dos desenvolvimentos e em pesquisas envolvendo o paradigma [Macedo et al. 2015]. E com o crescente interesse da indústria no OpenFlow, cada vez mais equipamentos de rede, como *switches* e roteadores, de empresas como HP, NEC, Pronto, Extreme, Cisco, Brocade, Juniper e Huawei, por exemplo, passaram a suportar o protocolo [Feamster et al. 2014].

Uma série de pesquisas relacionadas à SDN e ao padrão OpenFlow foi realizada

nos últimos anos. Contudo, ainda são raros os trabalhos que enfoquem o plano de dados dos *switches* OpenFlow, bem como suas implementações e mecanismos internos.

Alguns trabalhos abordam, de certa forma, a temática do plano de dados. Entretanto, tais trabalhos possuem abordagem e objetivos diferentes dos nossos. Por exemplo, [Bianco et al. 2010] analisam o desempenho de um único *switch*, implementado virtualmente sobre uma ponte (*bridge*). Os autores comparam o desempenho desse *switch* em modo OpenFlow, com o desempenho do mesmo operando através de roteamento IP e de *switching* Ethernet (modo *legacy*). Ou seja, eles comparam técnicas de encaminhamento distintas, utilizando o mesmo equipamento. Esse trabalho apresenta métricas de desempenho bem definidas, com foco em vazão de encaminhamento e latência de pacotes, sob condições normais e de sobrecarga, com diferentes padrões de tráfego e diferentes tamanhos de pacotes na rede.

O trabalho de [Sünnen 2011] também avalia o desempenho de um único *hardware switch* OpenFlow. Algumas das métricas de avaliação dependem fortemente do desempenho do controlador, que não é foco de estudo do nosso trabalho.

Por sua vez, [Appelman and De Boer 2012] realizam uma comparação entre diferentes *switches* OpenFlow. Foram comparadas três plataformas OpenFlow, sendo dois *hardware switches* e uma versão em *software* do Open vSwitch. Contudo, a avaliação contou fortemente com a participação do controlador, o que influenciou de forma considerável as avaliações conduzidas.

Finalmente, [Rotsos et al. 2012] propõem um framework aberto e genérico que permite o teste de *switches* OpenFlow. Eles desenvolveram uma ferramenta que testa a interação entre o mecanismo de encaminhamento do *switch* e a aplicação de controle remota, tanto para *hardware switches* quanto para *software switches*. O trabalho apresenta a ferramenta, assim como avalia algumas implementações de *switches* OpenFlow. Assim como nos trabalhos supracitados, a participação do controlador não permite que os resultados sejam conclusivos sobre o desempenho único e exclusivo dos *switches*.

Dessa forma, destacamos os seguintes diferenciais do nosso trabalho: (i) Comparamos diferentes implementações OpenFlow de *hardware switches* comerciais e de *software switches open source*. Essa comparação é realizada através de métricas de desempenho tais como latência e *jitter*. (ii) Analisamos apenas aspectos referentes ao plano de dados dos *switches*, descartando a influência do controlador. (iii) Comparamos os desempenhos de cada *switch* em seus modos normal e OpenFlow. (iv) Indicamos quando é vantajoso ou não utilizar o protocolo OpenFlow nos *switches* analisados. Em parte, nossos resultados nos permitem avaliar se as implementações do protocolo são feitas em *hardware*, ou se são instâncias de um *software switch* adaptadas ao mecanismo interno do *switch*.

3. Metodologia de Avaliação

Topologia de avaliação. Foi montada uma topologia típica SDN para as avaliações e testes de comparação das diferentes implementações de *switches* OpenFlow, como mostra a Figura 1. De forma geral, a topologia é composta por três máquinas interligadas através de um *switch* OpenFlow. Uma das máquinas atua como cliente, uma como servidor e a terceira como controlador da rede. Montamos essa topologia de modo que em

cada cenário de teste apenas o *switch* fosse variado. De fato, utilizamos máquinas e rede dedicadas aos experimentos, assim como disponibilizamos a mesma largura de banda em todos os testes, de modo que houvesse o mínimo de impacto possível causado pela rede ou pela configuração das máquinas. Como as aplicações executadas nos computadores são extremamente leves, controlador, cliente e servidor não operaram em saturação total. Também realizamos testes prévios com a vazão de cada *switch* e obtivemos valores máximos de vazão muito semelhantes entre si, de tal modo que poderíamos descartar a influência da vazão nos testes.

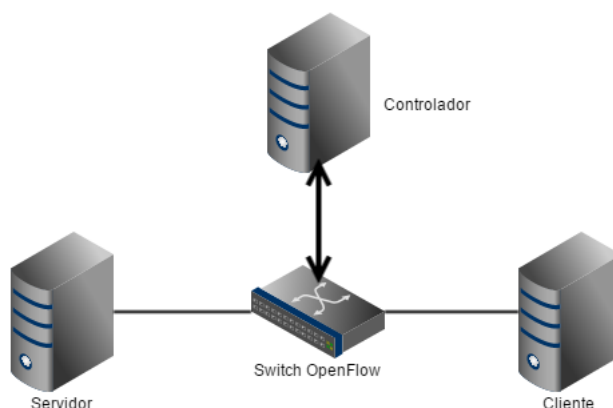


Figura 1. Arquitetura da rede de avaliação.

Para instalar as regras OpenFlow de avaliação, utilizamos o controlador POX [POX 2015]. O controlador instala regras com *hard timeout* ilimitado, no momento da inicialização da rede. As regras ficam em atividade até o momento em que o *switch* é desconectado. Nos testes com *switches* operando em modo *legacy*, sem estar ativo o protocolo OpenFlow, o controlador não realiza nenhuma operação.

Switches avaliados. Com o surgimento do OpenFlow e sua grande aceitação no mercado, várias empresas passaram a produzir equipamentos com suporte a OpenFlow, seja através de prototipação, de atualização de *firmware* ou do desenvolvimento direto dessa tecnologia [Rotsos et al. 2012]. Neste trabalho serão avaliadas e comparadas algumas dessas implementações OpenFlow para *hardware switches* comerciais e *software switches* conhecidos, considerando a versão 1.0 do OpenFlow. A Tabela 1 apresenta os *switches* avaliados e sumariza suas características.

Caracterização da carga. Os testes consistem no envio de pacotes UDP entre a máquina cliente e o servidor através do switch OpenFlow a ser avaliado. Para isso, foi implementado um gerador de tráfego próprio, composto por dois módulos: um módulo instalado na máquina cliente e outro no servidor. O módulo cliente envia os pacotes UDP para o servidor, com o *timestamp* de envio do pacote. O módulo executado no servidor, ao receber os pacotes, os retorna para o cliente. O módulo cliente, ao receber a resposta, calcula o Round Trip Time (RTT) de cada um dos pacotes utilizando os *timestamps* de envio e de chegada. A partir disso, o módulo cliente pode computar as métricas de interesse, como atraso e *jitter* dos pacotes. Vale ressaltar que a precisão dos *timestamps* está na casa dos microssegundos, a mais exata possível, segundo a RFC 3161.

Tabela 1. Switches avaliados no trabalho.

Marca	Modelo	S.O. Firmware	HW / SW Switch	Núm. de portas	CPU
Extreme	Summit x460-24p	ExtremeXOS 15.4.2.8	HW	28	Single Core CPU 500 MHz
Extreme	Summit x440-48p	ExtremeXOS 15.4.2.8	HW	52	Single Core CPU 500 MHz
HP	HP2920-24G	Firmware K 15.5 i	HW	24	Tri Core ARM1176 625 MHz
LinkSys	WRT54GL	OpenWRT Pantou	SW	4	Broadcom BCM5352 200 MHz
Open vSwitch	OvS 2.3.0	Linux Ubuntu 14.04	SW	-	Intel Core i7 CPU 2.80 GHz

Variamos o tamanho dos pacotes UDP entre 64, 128 e 256 bytes, para avaliar o desempenho dos *switches* com diferentes *overheads* de processamento de dados. Esses valores foram escolhidos com base nos trabalhos relacionados e na RFC 2544. Os resultados de cada teste correspondem ao envio de 10.000 pacotes, e as curvas apresentam intervalos de confiança com 99%.

Foram definidos quatro cenários para a avaliação de desempenho das implementações OpenFlow de cada *switch*. Para cada um deles, foi elaborado um tipo de teste diferente, de acordo com as especificidades de cada caso.

4. Resultados

Nesta seção apresentaremos os resultados mais significativos dos experimentos realizados nos diferentes *switches* com suporte a OpenFlow.

4.1. Desempenho em diferentes tipos de match

O primeiro cenário definido avalia o desempenho do *switch* na realização de diferentes tipos de casamento (ou *match*). Definimos cinco tipos genéricos de *match*, compostos por subconjuntos de matches individuais formados dentre os 12 atributos da versão 1.0 do OF (Tabela 2). O objetivo é verificar se o tempo de execução dos *matches* é influenciado pelos diferentes atributos, e, principalmente, analisar a diferença do desempenho dos mesmos tipos de *match* em diferentes *switches*.

Tabela 2. Subconjuntos de atributos utilizados para matches.

Tipo de Match	Atributos de Match OpenFlow
Porta	Porta do <i>switch</i>
MAC	Endereços MAC de origem e destino
IP	Ethertype e endereços IP de origem e destino
Porta UDP	Ethertype, endereços IP de origem e destino, protocolo da camada de transporte e portas TCP/UDP de origem e destino
Exato	Porta do <i>switch</i> , endereços MAC de origem e destino, ethertype, ID da VLAN, prioridade da VLAN, endereços IP de origem e destino, protocolo da camada de transporte, IP ToS e portas TCP/UDP de origem e destino

Na Tabela 3 listamos os intervalos de confiança dos atrasos médios de pacotes de 64 bytes, obtidos na realização de cada um dos diferentes tipos de *match*, em cada

switch. Podemos verificar que em cada *switch* as diferentes operações de *match* possuem intervalos de atraso muito próximos um dos outros. Isso leva a crer que o tipo ou a quantidade de atributos utilizados para realizar um *match* não afetam substancialmente o atraso dos pacotes, quando comparamos os diferentes tipos de *match*. A única exceção ocorreu na realização do *match* exato no *switch* HP. O intervalo de confiança desse *match* no *switch* HP não foi coincidente com o dos demais tipos de *match*. O tempo de atraso médio foi destacadamente menor, quando comparado aos demais *matches*. Ainda nesta seção, apresentaremos uma possível explicação para isso.

Tabela 3. Intervalo de confiança do atraso para diferentes tipos de match (em ms).

Switch	Tipos de Match				
	Porta	MAC	IP	Porta UDP	Exato
Extreme x460-24p	0.125 - 0.127	0.124 - 0.126	0.124 - 0.126	0.124 - 0.126	0.121 - 0.125
Extreme x440-48p	0.174 - 0.175	0.131 - 0.133	0.130 - 0.132	0.119 - 0.121	0.117 - 0.119
HP	1.069 - 1.147	1.049 - 1.113	1.064 - 1.309	1.038 - 1.106	0.743 - 1.001
LinkSys - OpenWRT	1.025 - 1.051	1.017 - 1.045	1.019 - 1.049	1.017 - 1.046	1.018 - 1.047
Open vSwitch	0.514 - 0.519	0.502 - 0.508	0.504 - 0.508	0.513 - 0.517	0.508 - 0.511

Ao compararmos o mesmo tipo de *match* em diferentes *switches*, notamos uma substancial diferença entre os intervalos de confiança. Conseguimos notar três comportamentos distintos, para os tipos de *match* por porta, MAC, IP e porta UDP. Os *switches* Extreme (x460 e x440) apresentam os menores valores de atraso médio de todos os *switches* comparados, com aproximadamente 0.1 ms de atraso médio. Por sua vez, o Open vSwitch apresenta um valor médio de atraso próximo a 0.5 ms. Já o *switch* HP e o OpenWRT Pantou rodando no LinkSys WRT54GL [Pantou 2015] possuem os piores desempenhos, com os intervalos de confiança do atraso beirando 1 ms. Para ilustrar essa situação, o gráfico da Figura 2 apresenta as distribuições cumulativas (CDF) dos atrasos dos pacotes em cada *switch*, para o *match* por IP. Vale ressaltar que esse comportamento é observado também nas curvas dos *matches* por porta, MAC e porta UDP. No gráfico, é possível notar a diferença entre os três grupos formados a partir das distribuições. Os valores médios do atraso do *switch* HP e LinkSys OpenWRT são fortemente influenciados por atrasos de pacotes superiores a 2 ms.

Contudo, para o *match* exato, notamos uma pequena diferença. Como dissemos anteriormente, o *switch* HP possuía um atraso médio menor quando comparado ao dos outros tipos de *match*. Observando o gráfico da Figura 3, com a distribuição cumulativa dos atrasos dos pacotes para o *match* exato, notamos que a curva correspondente ao *switch* HP fica deslocada mais à esquerda, se aproximando da curva do Open vSwitch. Isso significa que os tempos de atraso dos pacotes nesse *match*, para o *switch* HP, são relativamente menores quando comparados aos atrasos nos outros *matches* no mesmo *switch*, em cerca de 21%. Acreditamos que isso se deve a algum mecanismo do *switch* HP que realiza *matches* exatos possivelmente em *hardware*, de forma mais rápida, ao contrário dos *matches* não exatos, que devem ser realizados em *software*, de forma mais lenta. Mesmo assim, para o *match* exato, os *switches* Extreme continuaram como os mais rápidos e a implementação OpenWRT do LinkSys como o mais lento dos *switches* analisados.

Para testarmos a influência do tamanho do pacote nos *matches*, realizamos os mesmos testes variando o tamanho dos pacotes para 128 e 256 Bytes. Quando comparamos

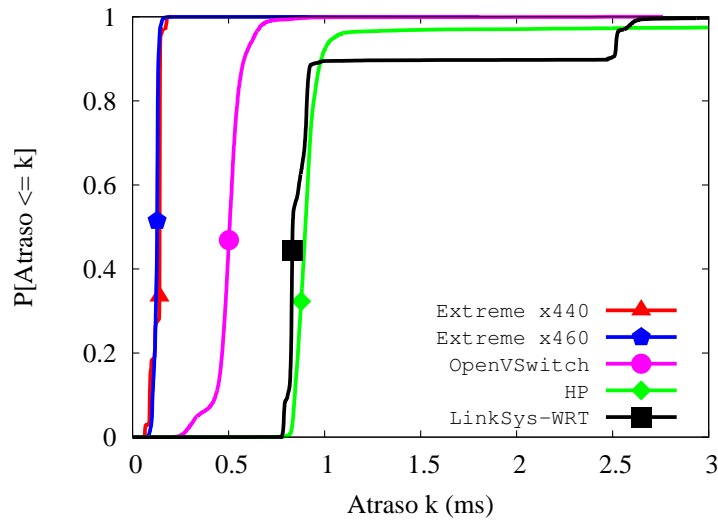


Figura 2. Distribuições cumulativas dos atrasos para o tipo de *match* por IP.

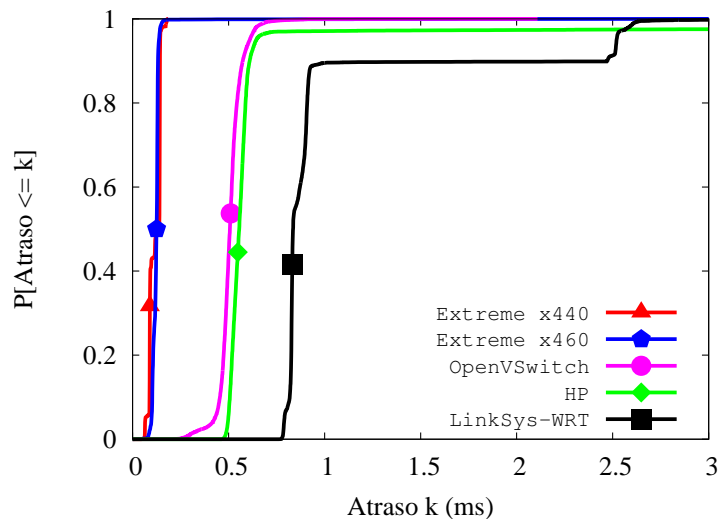


Figura 3. Distribuições cumulativas dos atrasos para o tipo de *match* exato.

os resultados para tamanhos diferentes de pacote, notamos que os *switches* HP e Extreme x460 tiveram um aumento maior nos tempos de atraso. Os outros não tiveram aumentos significativos nos atrasos. Podemos notar essa situação ao comparar o gráfico da Figura 4 (distribuições cumulativas dos atrasos para *matches* exatos com pacotes de 256 bytes) com o gráfico da Figura 3 anterior (com pacotes de 64 bytes). O deslocamento das curvas referentes aos *switches* Extreme x460 e HP é perceptível entre os gráficos.

O gráfico da Figura 5 mostra a CDF dos *jitters* de pacotes em *matches* exatos. Observamos que os *jitters* dos *switches* Extreme foram notoriamente menores quando comparados aos outros, ficando abaixo de 1 ms. Por sua vez, a curva do *switch* HP se assemelha às curvas dos *software switches* estudados. Isso pode corroborar a nossa suspeita de que a implementação OpenFlow do *switch* HP é apenas uma instanciação de um *software switch* rodando em memória, dentro de um *hardware switch*.

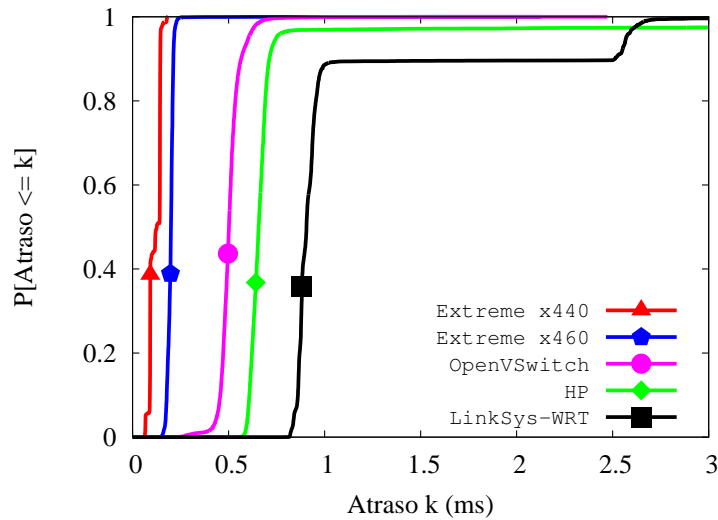


Figura 4. Distribuições cumulativas dos atrasos para o tipo de match exato em pacotes de 256 bytes.

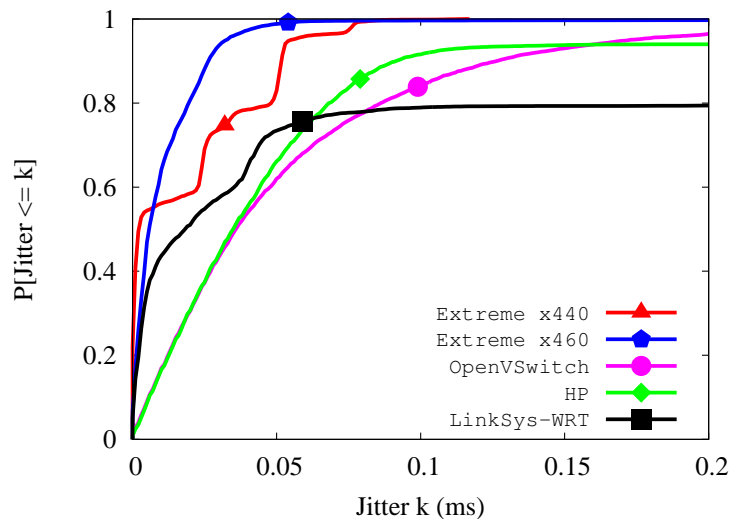


Figura 5. Distribuições cumulativas dos jitters para pacotes de 64 bytes.

4.2. Desempenho dos switches nos modos legacy e OpenFlow

O segundo cenário definido compara os desempenhos de um determinado *switch* quando operando nos modos OpenFlow e normal (*legacy*). Comparamos o encaminhamento dos pacotes entre a porta de chegada e a porta de saída do *switch* nos dois modos de operação, para verificar se existe alguma diferença de desempenho entre os modos.

Para reproduzir o encaminhamento porta a porta de um *switch* normal no modo OpenFlow, instalamos previamente uma regra no controlador que realizava *matches* no atributo de porta de entrada (*in_port*) e repassava os pacotes que sofriam esse *match* para uma porta de saída determinada. Só assim seríamos capazes de comparar os dois modos de operação de forma igualitária.

No gráfico da Figura 6 mostramos o desempenho dos *switches* em modo

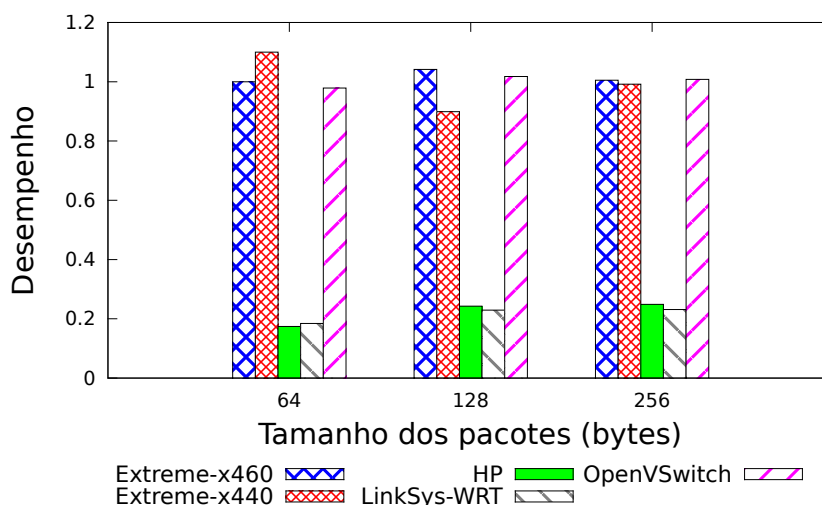


Figura 6. Desempenho dos switches em modo OpenFlow em comparação ao modo legacy.

OpenFlow quando comparados ao desempenho dos mesmos em modo normal (*legacy*), para três tamanhos diferentes de pacotes. A comparação entre os desempenhos foi obtida através da métrica de *Speedup*, que consistiu na divisão da latência no modo normal pela latência no modo OpenFlow. No gráfico, valores próximos a 1.0 correspondem a desempenhos semelhantes dos modos OpenFlow e normal. Valores menores que 1.0 indicam uma piora no desempenho no modo OpenFlow, enquanto valores superiores a 1.0 indicam um desempenho superior utilizando OpenFlow. Os intervalos de confiança são pequenos, por isso não estão reproduzidos na imagem.

Ao analisar o gráfico, nota-se que o desempenho do modo OpenFlow é bem próximo ao desempenho do modo *legacy* nos dois switches Extreme (x460 e x440) e no Open vSwitch. Já os desempenhos do modo OpenFlow nos switches HP e LinkSys-OpenWRT são bem inferiores aos desempenhos em modo *legacy*, chegando a apenas 20% do desempenho normal em ambos, aproximadamente. Essas observações nos levam a crer que já é vantagem, em alguns equipamentos, utilizar a tecnologia OpenFlow. Nesses casos, a utilização do OpenFlow praticamente não acarretou sobrecarga em operações normais do switch. Além disso, é possível pensar que os mecanismos de encaminhamento internos desses switches podem também ser utilizados pela tecnologia OpenFlow, o que explicaria um desempenho semelhante entre os dois modos.

4.3. Desempenho com uma ou muitas regras na tabela de fluxos

O terceiro cenário definido avalia o desempenho do switch em duas situações: quando a tabela de fluxos possui apenas uma regra, e esta retorna um *match*; e quando a tabela possui múltiplas regras, mas apenas uma delas retorna um *match*. Nosso objetivo é verificar se o desempenho pode ser influenciado pela disposição das regras na tabela de fluxos, pelo processamento interno de cada switch ou pelo tipo de regra instalado.

O primeiro passo para realizar esse teste foi a descoberta do limite máximo de regras que poderíamos instalar em cada switch no modo OpenFlow. A Tabela 4 apresenta o máximo de regras instaladas nos experimentos, além do número máximo de regras

suportadas pelos *switches* em modo *legacy*. Ao analisar esses dados, notamos que, na maioria dos casos, o número de regras suportadas no OpenFlow é menor do que o número de regras suportadas nativamente.

Tabela 4. Quantidade de regras suportadas pelos *switches* nos modos *legacy* e OpenFlow.

Switches	Quantidade de Regras Suportadas no Modo <i>Legacy</i>	Quantidade de Regras Observadas no Modo OpenFlow
Extreme x460-24p	2048	1200
Extreme x440-48p	1024	248
HP	2048	16000
LinkSys - OpenWRT	100	100
Open vSwitch	1000000	750000

Ao investigar o motivo dessa diferença, notamos que os *switches* Extreme (x460 e x440) armazenam as regras OpenFlow em formato de *access-lists* (ACLs) em *hardware*, assim como fazem com as regras quando operando em modo normal. Isso pode explicar o desempenho semelhante dos dois modos de operação (OpenFlow e *legacy*) para os *switches* Extreme, mostrados no cenário anterior. Como, no modo OpenFlow, existem mais alguns atributos ou operações a serem armazenadas, e o tamanho da ACL é definido como *double*, menos regras são suportadas no modo OpenFlow. Já o Open vSwitch possui capacidade de armazenamento de até 1 milhão de regras em uma tabela *hash* desenvolvida para armazenar apenas *matches* exatos, mas escolhe o limite de acordo com os tipos de regras e as condições da rede. Para armazenar regras de *matches* não exatos, o Open vSwitch utiliza uma tabela linear de até 100 posições.

Em relação ao *switch* HP, na instalação da tecnologia OpenFlow, ele passa a possuir além de uma tabela em *hardware*, uma outra tabela em *software*, com uma grande capacidade de armazenamento de regras, e que não pode ser desativada. Esse fato pode indicar que, possivelmente, essa implementação do OpenFlow seja uma implementação em *software* do Open vSwitch, sem maior integração ao projeto de *hardware* do *switch* HP. Isso explicaria o baixo desempenho no modo OpenFlow do *switch* HP, apresentado no cenário anterior.

Tabela 5. Intervalos de confiança dos atrasos em tabelas com uma e múltiplas regras (em ms).

Switches	Uma regra	Múltiplas regras
Extreme x460-24p	0.121 - 0.125	0.124 - 0.128
Extreme x440-48p	0.109 - 0.110	0.114 - 0.115
HP	1.049 - 1.120	1.046 - 1.330
LinkSys - OpenWRT	1.028 - 1.055	1.215 - 1.242
Open vSwitch	0.508 - 0.511	0.513 - 0.517

Na Tabela 5 estão listados os intervalos de confiança do atraso médio dos pacotes em duas situações: no caso em que a tabela de fluxos contém apenas uma regra, e no caso em que a tabela de fluxos possui múltiplas regras falsas e apenas uma regra como verdadeira. Em ambos os casos, os *matches* foram exatos. A quantidade de regras instaladas, no último caso, equivale ao número limite de regras encontrado no experimento anterior, listado na Tabela 4 anteriormente. Vale ressaltar que nos *switches* com regras escritas

em ACLs, a regra verdadeira foi inserida por último na tabela de fluxos. Já nos switches baseados em *software*, como as regras são inseridas utilizando índices *hash*, é impossível determinar sua efetiva posição na tabela de fluxos.

Podemos notar que, na maioria dos *switches*, os intervalos se interceptam ou apresentam diferença de poucos microssegundos entre si. Isso pode indicar que a grande quantidade de regras instaladas e a disposição das mesmas na tabela de fluxos não sobrecarregam significativamente os *switches*, nem causam um sensível aumento nos atrasos. A única exceção a essa observação ocorre com o LinkSys-OpenWRT, que possui um aumento médio no atraso dos pacotes em torno de 0.2 ms.

4.4. Desempenho na modificação de campos do cabeçalho dos pacotes

O quarto e último cenário avalia o desempenho de cada *switch* em modificações de campos específicos do cabeçalho de cada pacote. No teste desse caso, foram realizadas operações de modificação em campos do cabeçalho dos pacotes como endereços MAC e IP de destino, prioridade de VLAN, IP *Type of Service* e porta UDP, com o objetivo de comparar essas operações nos diferentes *switches*. Vale ressaltar que as operações de modificação de cabeçalho são opcionais no OpenFlow 1.0.

Tabela 6. Intervalos de confiança dos atrasos médios em modificações dos campos de cabeçalho (em ms).

Tipo de modificação	Switches				
	Extreme x460	Extreme x440	HP	LinkSys-OpenWRT	Open vSwitch
Sem modificação	0.125 - 0.127	0.114 - 0.115	1.121 - 1.212	1.032 - 1.058	0.508 - 0.511
MAC	0.125 - 0.129	0.113 - 0.115	1.128 - 1.203	1.034 - 1.061	0.511 - 0.516
IP	-	-	1.107 - 1.178	1.034 - 1.060	0.510 - 0.515
Prioridade de VLAN	0.125 - 0.128	-	1.107 - 1.183	-	-
IP ToS	-	-	1.122 - 1.194	1.030 - 1.056	0.506 - 0.510
Porta UDP	-	-	1.071 - 1.132	1.033 - 1.060	0.512 - 0.515

Na tabela 6 estão listados os intervalos de confiança dos atrasos dos pacotes quando estes sofrem ações de modificação em cinco campos de seus cabeçalhos, individualmente. Para efeito de comparação, colocamos também os valores do caso em que os pacotes não sofrem nenhuma alteração em seus cabeçalhos. Para todos os casos, não se notou um considerável aumento nos tempos de atraso médio dos pacotes nas modificações, quando comparados ao caso em que não há modificação. Isso indica que a operação de modificação pode ocorrer na ordem de poucos microssegundos. Entretanto, notamos que ainda existe falta de suporte para algumas operações de modificação em alguns *switches*. De fato, essa limitação está prevista na documentação de cada um deles. Essa falta de suporte, aliada ao fato de que só é permitido realizar uma única operação de modificação de cabeçalho por regra, mostra que ainda existem aspectos a se melhorar nas implementações OpenFlow já existentes, sobretudo em *hardware*. Devido a certas configurações da arquitetura, algumas operações de modificação não puderam ser realizadas e não foram inseridas na tabela.

5. Conclusão

Neste artigo avaliamos o desempenho do plano de dados de *hardware* e *software switches* OpenFlow. O objetivo deste trabalho foi avaliar a maturidade das implementações existentes, dado que o OpenFlow já se tornou uma tecnologia com diversos usos em redes de produção, e o seu uso tende a aumentar nos próximos anos.

A avaliação levou em consideração três *hardware switches* comerciais, e dois *software switches open source*. Foram avaliados o atraso de entrega de pacotes, o desempenho dos *switches* com diferentes tipos de regras de *match* de cabeçalhos, e se houve uma diferença de desempenho dos *switches* em modo OpenFlow em relação a um *switch* L2 convencional.

Os resultados mostraram que os *switches* implementam OpenFlow de formas diferentes. Alguns o implementam em *hardware*, outros em *software*. Além disso, a forma como a implementação foi realizada afeta significativamente o desempenho dos *switches*.

Como trabalhos futuros, pretendemos expandir a lista de *switches* avaliados, estudando alguns *switches* muito empregados pela comunidade de pesquisa Brasileira e do mundo. Por exemplo, pretendemos avaliar os *switches* Pica8, Datacom e NetFPGA (utilizados nas ilhas FIBRE¹).

Referências

- Appelman, M. and De Boer, M. (2012). Performance analysis of openflow hardware. Semester thesis project report, University of Amsterdam.
- Bianco, A., Birke, R., Giraudo, L., and Palacin, M. (2010). Openflow switching: Data plane performance. In *IEEE International Conference on Communications*, pages 1–5.
- Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and Walker, D. (2014). P4: Programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95.
- Braun, W. and Menth, M. (2014). Software-defined networking using openflow: Protocols, applications and architectural design choices. *Future Internet*, (6):302–336.
- Feamster, N., Rexford, J., and Zegura, E. (2014). The road to SDN: an intellectual history of programmable networks. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 87–98. ACM New York, NY, USA.
- Ibarra, J., Bezerra, J., Morgan, H., Lopez, L. F., Cox, D. A., I., Stanton, M., Machado, I., and Grizendi, E. (2015). Benefits brought by the use of openflow/sdn on the am-light intercontinental research and education network. In *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, pages 942–947.
- Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., and Vahdat, A. (2013). B4: Experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13*, pages 3–14, New York, NY, USA. ACM.

¹<https://fibre.org.br/>

- Khondoker, R., Zaalouk, A., Marx, R., and Bayarou, K. (2014). Feature-based comparison and selection of software defined networking (sdn) controllers. In *2014 World Congress on Computer Applications and Information Systems, WCCAIS 2014*, pages 1–7.
- Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.
- Macedo, D. F., Guedes, D., Vieira, L. F. M., Vieira, M. A. M., and Nogueira, M. (2015). Programmable networks-from software-defined radio to software-defined networking. *IEEE Communications Surveys and Tutorials*, 17(2):1102–1125.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. In *SIGCOMM Comput. Commun. Rev.*, volume 38, pages 69–74, New York, NY, USA.
- ONF (2015). What is ONF? Disponível em: <https://www.opennetworking.org/images/stories/downloads/about/onf-what-why.pdf>. Acesso: novembro/2015.
- Pantou (2015). Pantou : OpenFlow 1.0 for OpenWRT. Disponível em: http://archive.openflow.org/wk/index.php/Pantou.:_OpenFlow_1.0_for_OpenWRT. Acesso: novembro/2015.
- POX (2015). Pox wiki. Disponível em: <https://openflow.stanford.edu/display/ONL/POX+Wiki>. Acesso: novembro/2015.
- Rotsos, C., Sarrar, N., Uhlig, S., Sherwood, R., and Moore, A. W. (2012). *OFLOPS: An open framework for OpenFlow switch evaluation*, volume 7192 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- Sezer, S., Scott-Hayward, S., Chouhan, P.-K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., and Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. In *IEEE Communications Magazine*, volume 51, pages 36–43.
- Sünnen, D. (2011). Performance evaluation of openflow switch. Semester thesis project report, Swiss Federal Institute of Technology Zurich.
- Song, H. (2013). Protocol-oblivious forwarding: Unleash the power of sdn through a future-proof forwarding plane. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN '13*, pages 127–132, New York, NY, USA. ACM.
- Tennenhouse, D., Smith, J., Sincoskie, W., Wetherall, D., and Minden, G. (1997). A survey of active network research. *Communications Magazine, IEEE*, 35(1):80–86.