

Analisando o Comportamento de Ambientes de Experimentação baseados em OpenFlow

Rodrigo A. de Oliveira¹, Yona Lopes², Débora C. M. Saade², Natalia C. Fernandes¹

¹Departamento de Engenharia de Telecomunicações – PPGEET

² Departamento de Ciência da Computação – Instituto de Computação
Laboratório MídiaCom

Universidade Federal Fluminense (UFF) – Niterói – RJ – Brasil

{rao, yona, debora, natalia}@midia.com.uff.br

Abstract. *The choice of a Software Defined Networking (SDN) experimentation environment proper to the kind of experiment is a frequent challenge for researchers because of the range of existing possibilities, peculiarities of each environment, and restricted budget for purchasing devices. The objective of this study is to analyze the different experimentation environments from the standpoint of the existing OpenFlow switch solutions, considering issues such as performance and costs. Several performance parameters were evaluated in environments with similar network configurations, but different OpenFlow switch, controller and host implementations. The proposal of the article is not to define the best solution, but to provide inputs for choosing the environment that best suits the research requirements, considering performance, controllability, and costs.*

Resumo. *A escolha de um ambiente de experimentação para Redes Definidas por Software (Software Defined Networking - SDN) adequado ao tipo de experimento é um desafio frequente para pesquisadores devido à gama de possibilidades existentes, peculiaridades de cada ambiente e orçamento restrito para compra de equipamentos. O objetivo deste estudo é analisar os diferentes ambientes de experimentação sob a ótica das soluções existente de switch OpenFlow, considerando questões como desempenho e custos. Vários parâmetros de desempenho foram avaliados em ambientes com configurações semelhantes de rede, mas implementações do switch OpenFlow, controlador e clientes diversas. A proposta do artigo não é afirmar qual a melhor solução, mas sim prover insumos para a escolha do ambiente que melhor se adapte às necessidades da pesquisa, considerando desempenho, controlabilidade e custos.*

1. Introdução

As Redes Definidas por Software (*Software Defined Networks - SDN*) constituem um novo paradigma, mudando a forma como as redes de computadores são criadas, modificadas e gerenciadas [Feamster et al. 2014]. Esse novo paradigma abre uma nova perspectiva em ambientes de controle lógico da rede e novas aplicações de rede, que passam a poder ser desenvolvidas de forma simples e livre dos limites da arquitetura atual da Internet. O protocolo OpenFlow [McKeown et al. 2008] surge neste ambiente como uma forma de padronização da configuração por meio de um controle unificado de comutadores, roteadores e pontos de acesso sem fio. Usando o OpenFlow, os dispositivos de rede

passam a conter uma interface de programação simples, que permite o acesso ao controle de fluxo, ou à tabela de comutação do *hardware*.

Dada a sua importância disruptiva, as SDNs e as aplicações de rede desenvolvidas baseadas no protocolo OpenFlow são hoje alvo de grandes investimentos em pesquisas. Estas pesquisas avançam para o estado da arte com soluções inovadoras que trazem diversos benefícios para o setor acadêmico e privado. No entanto, para que estas vantagens sejam de fato reais, a avaliação realizada precisa ser o mais consistente possível. De fato, a importância de testes conclusivos feitos em bons ambientes e de evolução no plano de dados SDN é tamanha que novas propostas, como a de Katta et al. [Katta et al. 2014], surgem na tentativa de aprimorar este plano de dados. Apesar da importância, é comum se deparar na literatura com análises limitadas ou resultados incorretos devido ao mal uso de ferramentas de análise, simulação e emulação.

Este trabalho visa analisar e verificar os pontos fortes e fracos de ambientes de experimentação OpenFlow, provendo insumos para que pesquisadores possam escolher adequadamente a ferramenta que será usada de acordo com as suas necessidades. Os testes realizados visam verificar o desempenho dos ambientes de experimentação, com foco na infraestrutura adotada e através dos resultados obtidos ponderar as características destes ambientes para contribuir na escolha de ambientes a serem usados por pesquisadores.

As próximas seções estão organizadas da seguinte forma: a Seção 2 trata dos trabalhos relacionados; a Seção 3 descreve os tipos de ambientes de experimentação e a dificuldade de escolha; a Seção 4 apresenta a avaliação de desempenho realizada, descrevendo os objetivos e a configuração dos ambientes de teste detalhadamente; a Seção 5 analisa os resultados obtidos; a Seção 6 apresenta uma estimativa de custos aproximada para cada ambiente de teste, assim como uma análise qualitativa dos ambientes de experimentação utilizados e por fim, a Seção 7 apresenta as conclusões e propostas de trabalhos futuros.

2. Trabalhos Relacionados

As SDNs apresentam um modelo de arquitetura inovador capaz de entregar provisionamento automatizado, virtualização e programação de rede para *data centers* e redes corporativas, ganhando continuamente terreno no mercado corporativo e em provedores de serviços em nuvem para *data center*. De fato, diversos artigos têm sido publicados apresentando meios de utilização do OpenFlow [Heller et al. 2010] [Das et al. 2010], o que demonstra a variedade de campos onde a separação entre o plano de dados do plano de controle tem grande utilidade.

No meio acadêmico, diversas pesquisas para aprimorar o conceito e entendimento da utilidade da tecnologia SDN são realizadas, abrangendo diferentes enfoques, como análises dos tipos de controladores, do protocolo OpenFlow [Macedo et al. 2015] e dos equipamentos de diferentes fornecedores que suportam este protocolo [Feamster et al. 2014]. Alguns trabalhos focam suas análises de desempenho em comparações sobre o comportamento dos *switches* quando utilizados com o protocolo OpenFlow e no modo *legacy*, avaliando se as implementações do protocolo OpenFlow, são, de fato, realizadas em *hardware* ou em instâncias de um *software switch* adaptadas ao mecanismo interno do dispositivo [Costa et al. 2016].

Outras avaliações de desempenho de equipamentos OpenFlow focam em uma

camada de abstração de hardware convertendo dispositivos não habilitados para OpenFlow em dispositivos habilitados [Zal and Kleban 2014], com uso de plataforma de teste *OFLOPS*. Algumas avaliações apresentam somente a análise pontual do dispositivo OpenFlow instalado num desktop com sistema operacional *Linux* [Bianco et al. 2010], mensurando como é o comportamento do plano de dados neste cenário, sem comparação com outros meios possíveis de instalação de um dispositivo OpenFlow.

De maneira muito similar outros trabalhos envolvem o emulador Mininet, analisando a sua usabilidade [de Oliveira et al. 2014] ou verificando as suas limitações em diferentes ambientes [Keti and Askar 2015].

Os *testbeds* públicos têm crescido e estimulado inúmeros pesquisadores de rede para executar seus protótipos e experimentos. Com isso, pesquisas abrangentes sobre os desafios encontrados nestes *testbeds* são apresentados [Huang et al. 2016], porém, comparações que envolvam os diversos ambientes possíveis de simulações em OpenFlow com foco em desempenho, capacidade, custos e facilidades de uso, ainda são pouco exploradas.

Neste artigo avaliam-se diferentes meios de soluções de implementações de *switch* OpenFlow para definir quais são mais próprias para a realização de experimentos de novas ferramentas, ressaltando as suas limitações de desempenho e de custos. São feitas medidas que avaliam ambientes baseados em *switches* OpenFlow comerciais, implementados em computadores pessoais, emulados e ainda construídos em *testbeds* públicos. Diferentemente dos outros artigos, a proposta não é meramente avaliar o plano de dados OpenFlow, mas fornecer informações que sejam úteis na definição da plataforma de teste adequada a um experimento específico.

3. Tipos de Ambientes de Experimentação

O paradigma SDN tem ganhado força nos últimos anos e atraído cada vez mais a atenção da comunidade acadêmica e da indústria [Costa et al. 2016]. Com isso, muitos equipamentos de rede passaram a suportar o protocolo OpenFlow. Da mesma forma, o interesse em projetos de pesquisa relacionados ao uso do OpenFlow como *framework* para novas soluções também aumentou [Masoudi and Ghaffari]. Assim, o desenvolvimento de redes definidas por *software* chamou a atenção da comunidade científica nos últimos anos, principalmente devido à flexibilidade que o modelo de controle centralizado apresenta, o que facilita o desenvolvimento de soluções de acordo com a demanda da rede e/ou dos usuários [Ortiz et al. 2016].

Importante salientar que os resultados obtidos nos experimentos realizados são diretamente relacionados ao ambiente utilizado na experimentação. Por exemplo, o ambiente de experimentação tem um efeito notável sobre o tempo necessário para a construção de uma topologia [Keti and Askar 2015], para configuração de um fluxo, para comutação para a porta backup em caso de falha, etc. Além disso, existem diferenças significativas de desempenho e funcionalidade nas implementações em *hardware* do OpenFlow [Keti and Askar 2015]. Resultados inteiros podem ser invalidados caso experimentados em ambientes que não sejam próprios para a avaliação requerida. Caso o ambiente tenha uma limitação que afete diretamente os objetivos dos testes, falsos negativos podem ser gerados e uma boa solução pode ser descartada. Além disso, falsos positivos podem ser encontrados caso a escolha do ambiente de experimentação não seja pensada

previamente.

3.1. Ambiente OpenFlow Simulado e Emulado

Um simulador de redes SDN reproduz o comportamento dos equipamentos de uma rede OpenFlow, através de um modelo, para verificar o comportamento destes. Com isso, um sistema real pode ser avaliado numericamente a partir de um modelo de simulação. Este modelo precisa ser criteriosamente construído para representar o ambiente a ser simulado de forma fiel. Testes cuja a preparação em ambiente reais seja cara, custosa ou demorada têm na simulação uma forma viável de execução. Apesar dos vários simuladores de redes disponíveis, são poucos que oferecem suporte ao OpenFlow. O *Network Simulator 3 (NS-3)*¹ é um exemplo de simulador que é compatível com o OpenFlow, mas que usualmente não é tão utilizado quanto as outras ferramentas de análise devido à sua complexidade para programação do ambiente OpenFlow.

Um emulador de redes SDN é um sistema computacional que imita o funcionamento de *switches* OpenFlow, controladores e/ou *hosts*, tendo resposta em tempo real. O Mininet é um dos emuladores mais utilizados atualmente para a avaliação de redes SDN já que permite a emulação de uma rede OpenFlow com a utilização de apenas uma máquina [Ortiz et al. 2016]. O Mininet utiliza primitivas de virtualização do sistema operacional onde o usuário pode criar, interagir, personalizar e compartilhar um protótipo de rede OpenFlow com *switches*, *hosts* e controladores [Conterato et al. 2013]. Além disso, esse emulador utiliza controladores reais e pode ser integrado com equipamentos externos à emulação.

No entanto, como qualquer outro *software* de emulação, o Mininet tem limitações devido às restrições da plataforma de *hardware/software* onde é executado, o que impede que ele escale para grandes redes [Ortiz et al. 2016]. Essa limitação é devido ao uso de um processo *shell* para emulação e da necessidade de execução de um processo para cada *switch* OpenFlow virtual e/ou *host* em espaço de usuário. O Mininet, até o momento ainda não fornece desempenho e qualidade fiéis a uma rede real [Keti and Askar 2015]. De fato, essa fidelidade depende da disponibilidade de recursos de processamento e memória para emular todos os *switches*, *hosts* e enlces da rede em tempo real [Conterato et al. 2013].

3.2. Ambiente OpenFlow Real

Atualmente, diversos fabricantes já disponibilizam *switches* com suporte ao protocolo OpenFlow. Fabricantes de equipamentos, operadores de rede e de empresas como *Facebook*, *Microsoft* e *Google* já em 2011, formaram a *Open Networking Foundation (ONF)*², com o objetivo de promover o OpenFlow e as redes SDN. O desempenho de um *switch* OpenFlow real pode ser influenciado pela característica do hardware utilizado, já que pode-se implementar o *switch* OpenFlow como *Software Switch (SS)*, ou *Hardware Switch (HS)*. O SS implementa as tabelas de fluxo, campos, entradas, etc, como uma estrutura de dados através de software, geralmente utilizando sistemas UNIX/Linux e memória comum (SRAM - *Static Random Access Memory* - ou DRAM - *Dynamic RAM*). Nesse modelo, os dados são processados através da CPU, e por consequência, o desempenho deste é limitado pelo poder de processamento da(s) CPU(s) utilizada(s). Nos HS geralmente são utilizados uma memória especializada (*Ternary Content Addressable Memory*

¹<https://www.nsnam.org/>

²<https://www.opennetworking.org/>

- TCAM) e sistemas operacionais proprietários para implementar o protocolo OpenFlow e as tabelas de fluxo. Assim, os pacotes são processados através de componentes especializados de *hardware*, orientados para implementação de tarefas específicas em domínios bem definidos, denominados *Application Specific Integrated Circuits* (ASIC). Com isso, a queda de desempenho é mínima.

Uma comparação imediata é que os HSs são considerados mais rápidos do que SSs pois utilizam memórias especializadas para realizar a checagem de regras em paralelo, apesar de possuírem menor capacidade de armazenamento. Com isso, os HSs suportam um número limitado de regras enquanto SS suportam uma quantidade de regras muito maior. Com isso, geralmente, os SSs utilizam altas taxas de processamento para alcançar um menor desempenho que os HSs. No entanto, SSs possuem uma maior flexibilidade para implementar ações mais complexas [Goransson and Black 2014]. Dentro deste cenário, o *Open vSwitch* (OvS) se encaixa como um SS. O OvS é um *switch* virtual multicamada projetado para permitir a automatização maciça da rede através da extensão programática, enquanto ainda suporta interfaces e protocolos de gerenciamento padrão. Ressalta-se que alguns *switches* de mercado, como o WRT54GL da *LinkSys*, utilizam o OvS para implementação. Devido a essas características bem diferentes, trabalhos como o de Katta et al. [Katta et al. 2014] propoem modelos híbridos de *switch* no intuito de combinar as vantagens dos HS e SS em um modelo híbrido que consiga lidar com o *cache* de regras de forma mais eficiente.

Ressalta-se que podem ser montados *testbeds*, privados ou públicos, para conduzir os experimentos tanto com *hardware switches* como com *software switches*. Os *switches* de mercado podem usar ambos os modelos, SS e HS, e ainda podem ser utilizadas máquinas com o OvS emulando os switches. Exemplos de *testbeds* públicos são o FIBRE (*Future Internet Brazilian Environment for Experimentation*)³, o OFELIA (*OpenFlow in Europe: Linking Infrastructure and Applications*)⁴ e o PlanetLab⁵. Estes *testbeds* incluem tanto *hardware switches* quanto *software switches* em suas arquiteturas.

O *testbed* FIBRE utilizado nas simulações, é uma infraestrutura de pesquisa focada em experimentação em redes e é aberta para a utilização por estudantes e pesquisadores brasileiros. Dentre suas funcionalidades plataforma permite a execução de experimentos distribuídos geograficamente em larga escala por meio de alocação de nós sem fio e instanciação de máquinas virtuais nas ilhas de experimentação, seleção de topologias virtuais e alocação de espaço de fluxos OpenFlow tanto nas ilhas de experimentação quanto no backbone e uso de conectividade de 1 Gb/s entre as ilhas de experimentação.

3.3. Escolha do ambiente

A experimentação com redes OpenFlow apresenta muitas possibilidades que diferem muito em custo, desempenho, flexibilidade, facilidade de implementação, etc. Cada uma das características abordadas impacta de forma diferente nos testes realizados. O uso de uma solução mais veloz no processamento pode fazer diferença em testes que precisem medir o tempo de processamento entre a descoberta de uma falha em uma porta e comutação para a backup, por exemplo. Nesse caso, o uso de um emulador, como o

³<http://fibres.org.br/>

⁴<http://www.fp7-ofelia.eu/>

⁵<https://www.planet-lab.org/>

Mininet, pode alterar de forma significativa os resultados.

A escolha do ambiente de testes precisa ser feita com base nos impactos que o ambiente terá sobre o resultado do experimento. Muitos trabalhos não passam por essa escolha e podem ter resultados invalidados por conta do uso de um ambiente não adequado. Por esse motivo, a avaliação dos ambientes OpenFlow para experimentação é essencial, pois resulta em uma maior assertividade dos testes realizados quando escolhido de forma adequada para cada solução. Por tal motivo as simulações foram realizadas em ambientes de simulações distintos como emuladores, *testbeds* privados e públicos. Uma vez que a mesma simulação é realizada nos diferentes ambientes torna-se possível avaliar o impacto de tal escolha no resultado da experimentação.

4. Ambiente de Avaliação

O intuito dos testes realizados por este trabalho é avaliar o comportamento de diversas soluções existentes para construção de ambientes OpenFlow para testes de novas soluções. Os critérios de comparações utilizados se baseiam em simulações que têm enfoque no desempenho do dispositivo OpenFlow. Para isso, são utilizados testes que medem a capacidade do dispositivo tratar requisições simultâneas e distintas, elevado tráfego de dados, protocolos de transporte e requisições solicitadas pelo controlador. Para simulações que necessitam de limitação da largura de banda são utilizados arquivos *bash* configurados e armazenados dentro dos *hosts* clientes com as premissas necessárias para limitação da banda de tráfego.

Essas escolhas são insumos que apontam o caminho para distinguir a eficiência de cada plataforma de experimentação estudada, uma vez que tais análises focam em como será o tratamento do dispositivo OpenFlow para cada solicitação a ele realizada.

4.1. Topologia dos Testes

Para realização dos testes foi projetada uma topologia para atender testes em emuladores, *testbeds* e ambientes físicos igualmente, sendo, esse último, definido por limitações financeiras para obtenção do *hardware*, o que limitou a quantidade de *hosts* e *switches* em todos os ambientes. A topologia consiste de 2 *hosts*, 1 controlador Ryu e uma solução com OvS conforme ilustra a Figura 1. A topologia concebida atende aos testes planejados para análise das soluções e ao mesmo tempo é de fácil implementação. Usando duas máquinas como cliente é possível realizar diferentes testes de comunicação entre os pares de clientes, focando tanto no plano de dados como em testes que estimulem sobrecarregar o dispositivo OpenFlow em utilizar o plano de controle para se comunicar com o controlador.

Os testes duraram 20 segundos, tempo suficiente para realizar as principais simulações de criação de pacotes, geração de tráfego e monitoração de dispositivos, e os resultados são apresentados com um intervalo de confiança de 95%. Foram realizados os testes em cinco ambientes usualmente utilizados para testes com *Openflow*, sendo eles: emulação com o Mininet; uso de *testbeds* públicos, especificamente com FIBRE; montagem de *testbed* privado usando um *switch* comercial, especificamente o Pronto (Pica8 - P3295)⁶; e montagem de *testbed* privado usando um OvS instalado diretamente num

⁶<http://www.pica8.com/documents/pica8-datasheet-48x1gbe-p3290-p3295.pdf>

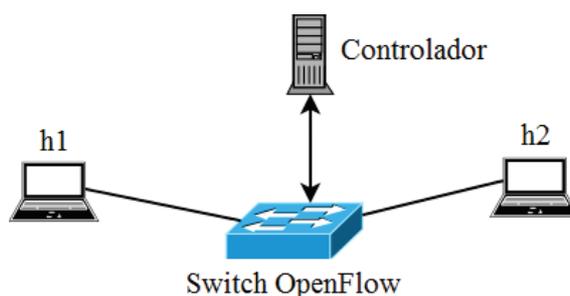


Figura 1. Topologia básica de avaliação.

desktop com 4 GB RAM e outro em máquina com 8 GB RAM, de forma a mensurar a influência na capacidade de desempenho com a variação de memória utilizada. O FIBRE e o *switch* Pronto disponível só trabalham com o protocolo OpenFlow na versão 1.0, com isso, a fim de manter a premissa nos outros ambientes analisados, todos os testes foram realizados com o OpenFlow na versão 1.0.

4.2. Configuração dos Testes

A estrutura física que compõem o *hardware* utilizado para as simulações tem papel fundamental no que tange a influência nos resultados obtidos, máquinas compostas de melhores processadores e memória tendem a desempenhar resultados superiores nas simulações. Por tal motivo, buscou-se usar a mesma configuração de *hardware* no ambiente de instalação da máquina virtual Mininet, no *switch* OvS utilizado no *testbed* privado e no *hardware* utilizado para o controlador Ryu. Qualquer mudança neste *hardware* dificulta a avaliação justa dos experimentos.

Os testes baseados em emulação foram feitos utilizando o Mininet 2.2.1⁷, instalado na plataforma *VirtualBox* usando 2G RAM e 4vcpu, localizada em desktop com 4GB RAM, 55GB HD e sistema operacional XUBUNTU 14.04.1, 32bits i686, Intel(R) Core(TM) i7-2600 3.40GHZ (1 placa com 4 núcleos e 8 threads). Já a construção dos *testbeds* privados foi feita usando, no primeiro caso, um *switch* comercial e, no segundo caso, o OvS 2.0.2 instalado em desktop com as mesmas configurações da máquina citada anteriormente. Alterou-se a memória dessa máquina entre 4 e 8 GB de RAM para avaliar o impacto causado pela memória. Para o ambiente de emulação com o Mininet, foi utilizada a mesma versão OvS 2.0.2 de *switch* OpenFlow.

Para os testes com *testbeds* privados, os *hosts* foram criados com as seguintes configurações: placa Mini-TX dual-core Intel® Atom Processor N2800 (2 núcleos e 4 threads), 2 GB RAM, XUBUNTU 14.04.01, 32 bits, i686. O uso dessas máquinas desonerou os custos financeiros para realização das simulações em *testbeds* privados. No caso do Mininet, os *hosts* foram simulados no próprio ambiente.

Os testes utilizando o FIBRE contaram com equipamentos OpenFlow baseados em placas NetFPGA e *switches* Pronto. Os *hosts* criados possuem sistema operacional GNU/Linux Debian 6.0 e são executados em máquinas virtuais criadas pela própria plataforma, com configuração Linux 3.2.0-4-amd64 SMP Debian 3.2.65-1+deb7u2 x86 64, com 120 MB de memória e 8 GB de HD.

⁷Máquina virtual obtida no site mininet.org

Em todos os testes, foi utilizado o controlador Ryu 4.2.2 em desktop com 4GB RAM, 55GB HD, SO: XUBUNTU 14.04.1, 32bits i686, Intel(R) Core(TM) i7-2600 3.40GHZ (1 placa com 4 núcleos e 8 threads). No caso do ambiente Mininet foi utilizada a mesma versão de controlador, instalada na máquina virtual correspondente ao Mininet. Devido ao uso da linguagem de programação Python e suporte às versões 1.0, 1.2, 1.3 e 1.4 do protocolo OpenFlow o que amplia sua utilização para simulações futuras; optou-se pelo controlador Ryu. A escolha de controladores distintos influencia no tempo de processamento do plano de controle, porém o estudo realizado não foca na comparação de controladores OpenFlow.

Para realizar os testes, foi utilizado o gerador de tráfego *Iperf* (V2.0.5)⁸, o gerador de pacotes *Scapy* (V2.2.0)⁹, o *wireshark*¹⁰ e o *tcpdump*¹¹, instalados nos *hosts* pertinentes e no servidor do controlador conforme necessário.

5. Resultados Obtidos

O primeiro teste realizado avalia o impacto de processamento dos dispositivos sobre os resultados do teste de rede. Para tanto, gerou-se um tráfego TCP com o *Iperf* entre os *hosts* h1 e h2. Para avaliar o impacto do processamento tanto nos *hosts* como nos *switches*, variou-se o tamanho do segmento entre 536 e 1460 B, pois segmentos menores levam a geração de mais pacotes. A banda do enlace também foi variada entre 10 Mbps e 1 Gbps, para variar o número de pacotes gerados/encaminhados. Outra análise realizada foi o impacto do uso do *tcpdump* como ferramenta de monitoração. Novamente, esse é mais um fator que gera sobrecarga na máquina e que tem influência no resultado. O *tcpdump* foi colocado em ambos os *hosts*. A aplicação usada no controlador simula um *switch* padrão, definindo entradas de fluxos com base apenas no MAC de destino.

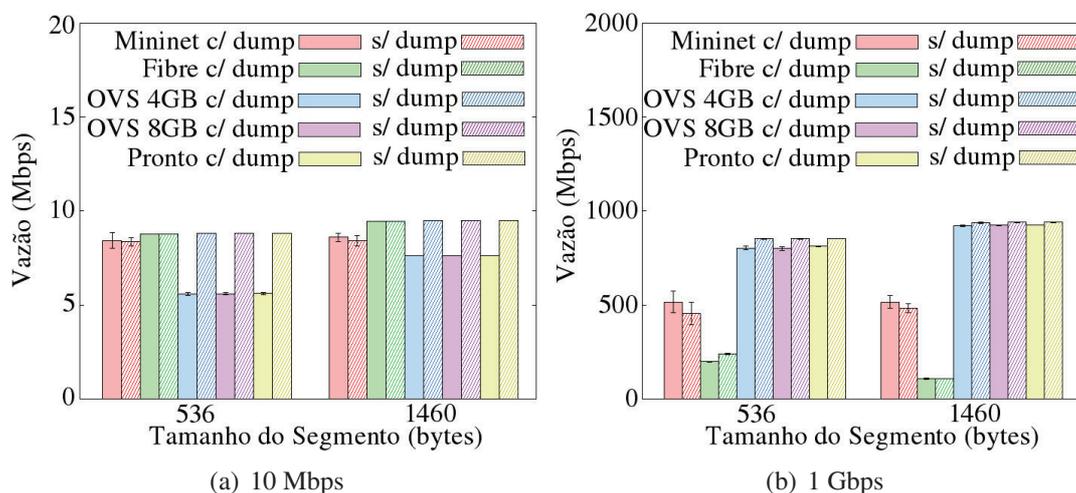


Figura 2. Observação da influência dos valores de tamanho máximo de segmento e uso de monitoração via tcpdump.

No caso das simulações realizadas em ambiente de laboratório real, representado pelo *switch* Pronto e Ovs instalado em desktop físico, a utilização do *tcpdump* influencia

⁸<https://iperf.fr/>

⁹<http://www.secdev.org/projects/scapy/>

¹⁰<https://www.wireshark.org/>

¹¹<http://www.tcpdump.org/>

consideravelmente na vazão obtida, apresentando maior impacto proporcional em enlaces de menor capacidade. Essa característica detectada está relacionada ao *hardware* utilizado para os *hosts*. Pois, o equipamento utilizado como *host* era de capacidade menor, compatível com o custo financeiro mais baixo, porém proporciona um processamento aquém do desejável. Em contrapartida, nos ambientes Mininet e FIBRE, o comportamento foi inconsistente com o esperado, sendo importante salientar que são ambientes onde não é possível ter controle sobre sua composição, uma vez que a plataforma FIBRE é compartilhada com outros pesquisadores em quantidade não verificável, que podem usar o ambiente no mesmo momento, impossibilitando controlar a banda disponível e o ambiente Mininet pode sofrer influências de outras tarefas do sistema operacional.

Na Figura 2(a), o uso de *tcpdump* em nada influenciou o ambiente FIBRE e, na Figura 2(b) pacotes com tamanho de segmento máximo menor, inesperadamente, apresentaram maior vazão. O ambiente FIBRE que se assemelha com um ambiente de uso real, no qual existe pouca controlabilidade dos clientes que o utilizam, o fato de executar os testes de maneira sequencial, ou seja, iniciando a simulação com o tamanho máximo de segmento menor, e variando largura de banda, e uso de *tcpdump* e somente posteriormente a mesma sequência de testes porém com uso de tamanho máximo de segmento maior, pode ter ocasionado o resultado encontrado. Assim, ao utilizar este tipo de ambiente de experimentação é importante executar os testes de forma mesclada, para aumentar a probabilidade que os diferentes cenários sejam expostos às mesmas condições externas. No Mininet, o valor de tamanho de segmento máximo e o uso de *tcpdump*, considerando a margem de erro, em nada influenciaram na vazão dos testes de Iperf. Contudo, ao se estressar o ambiente sem limitar a sua largura de banda, observou-se a mesma limitação, sendo que obteve-se vazão de 9 Gbps sem uso de *tcpdump* e de 4.5 Gbps com o uso de *tcpdump*. Isso é um indicativo importante, pois, no Mininet, a capacidade de processamento é dividida pelos elementos que se colocam na emulação. Se fossem usados mais *hosts* ativos ou mais *switches; links* com 1 Gbps, ou menos, provavelmente seriam afetados, causando uma perda considerável na capacidade do enlace.

A próxima análise realizada foi relativa aos impactos sobre o plano de controle dependendo da plataforma de experimentação que está em uso. Especificamente, buscou-se aumentar o número de eventos de *packet_in*, que são gerados todas as vezes que o *switch* recebe um pacote para o qual não existe entrada de fluxo. A quantidade de eventos gerados é impactada pela capacidade do *host* de gerar pacotes com endereços MAC diferentes, pela capacidade de processamento do *switch* e pela capacidade de resposta do controlador. Os pacotes são gerados no *host 1* utilizando a biblioteca *SCAPY* em todos os ambientes, gerando pacotes ARP para o *host 2*, variando o MAC de origem. Para medir a quantidade de *packet_in* gerados foi habilitado o uso de *wireshark* em h1, h2 e controlador. Uma máquina para o *host 1* com configurações mais robustas foi utilizada nos testes de OvS em desktop e switch Pronto: Intel ® Core i5-M480 2.67 GHz (1 placa com 4 Cores e 8 threads), 4 GB RAM, 500GB of HD, XUBUNTU 14.04.01, 32 bit, i686.

O *host 1*, montado para os testes de OvS em desktop e *switch* Pronto foi o que gerou maior quantidade de pacotes ARP, em média 39000 pacotes ARPs, indicando que um desktop com configurações mais robustas favorece simulações que dependem do desempenho de máquinas que geram pacotes.

Nos testes de OvS em desktop, os pacotes ARP ocasionaram a mesma quantidade

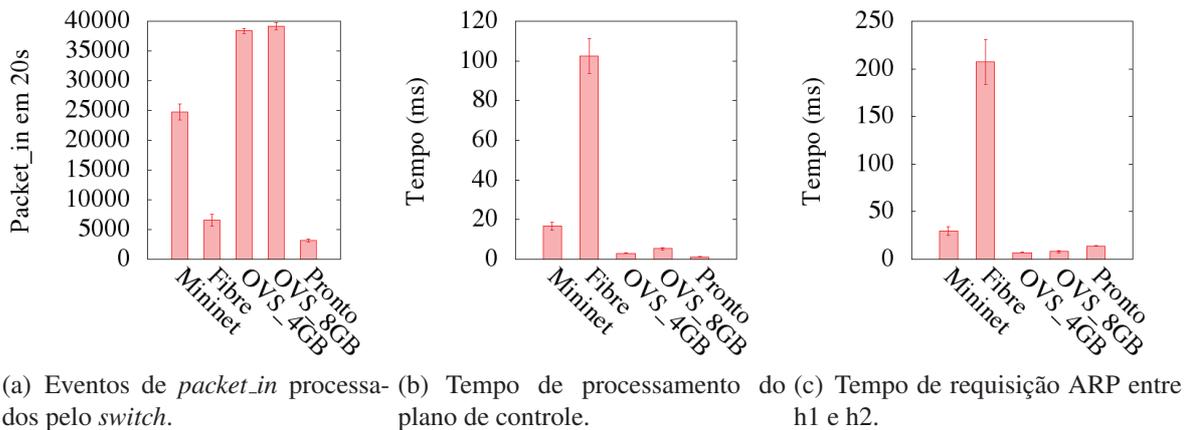


Figura 3. Análise do processamento de eventos de *packet_in* nos diversos ambientes de teste.

de *packet_in* para o controlador Ryu. Para esse teste, o ambiente OvS em desktop obteve o melhor resultado. Porém no caso dos testes em *switch* Pronto, os pacotes ARP gerados em h1, que deveriam criar a mesma quantidade de *packet_in* pelo *switch*, não ocorreu, pois houve perda de pacotes ARP, que não foram convertidos em eventos de *packet_in*. A máquina virtual do Mininet criou em média 25000 *packet_in* provenientes de 25000 chamadas ARP, não havendo perda de pacotes ao longo do processo. O host h1, montado via plataforma FIBRE, criou em média 34000 pacotes ARP, porém devido a perda de pacotes ARP a quantidade de *packet_in* proveniente do *switch* Pronto para o controlador Ryu foi inferior, indicando que este ambiente impactou negativamente o processamento no plano de controle. Tais testes foram mensurados através da análise do arquivo wireshark gerado no *host* h1, verificando a quantidade de pacotes ARP gerados (com MAC de origem distintos) e no arquivo wireshark gerado no controlador, verificando a quantidade de *packet_in* que chegaram no mesmo.

A Figura 3(b) mostra a análise do tempo de processamento entre chegada dos eventos de *packet_in* no controlador e a geração do evento *flow_mod* pelo controlador, que seta uma nova entrada de fluxo no *switch*. Nesse teste, observa-se um desempenho inferior do ambiente FIBRE com o plano de controle. Embora o Mininet tenha processado aproximadamente 4 vezes o número de *packet_in* que o ambiente FIBRE, ele apresenta um tempo de processamento no plano de controle 5 vezes menor que o FIBRE. Os ambientes de *testbed* privados apresentaram um tempo menor que o Mininet e o FIBRE, mesmo no caso do OvS em desktop no qual transitou um número elevado de pacotes no plano de controle. Devido a perda de pacotes ARP, no caso do teste com *switch* Pronto, a quantidade de pacotes no plano de controle foi reduzida, o que proporcionou um processamento rápido.

Mediu-se, como mostrado na Figura 3(c), o tempo entre a requisição ARP feita por h1, com MACs distintos, até a mensagem chegar em h2 e seu retorno até h1. Nesse caso, o ambiente FIBRE foi o que apresentou maiores tempos. Uma vez que o ambiente FIBRE pode estar sendo compartilhado por outros pesquisadores, tal situação pode ter impactado o tempo de processamento do *switch* para criar um número de entradas de fluxo elevada.

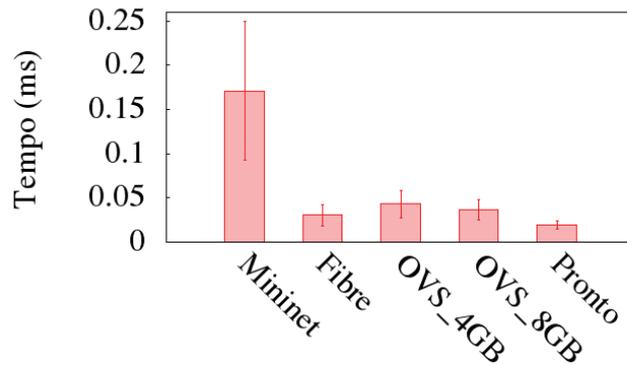


Figura 4. Observação de *JITTER*.

Por fim, a Figura 4 analisa as conexões do tipo UDP, porém o foco está sobre a variação do atraso entre os pacotes sucessivos de dados (*jitter*). Os testes utilizaram tamanho máximo de segmento de 1460 bytes, enlaces de 1 Gbps e taxa de envio de 100 Mbps. Nesta análise, mais uma vez, fica claro como a plataforma Mininet possui desempenho inferior comparado com os outros cenários.

6. Custo Financeiro das Soluções e Análise Qualitativa

Além de observar fatores relacionados ao desempenho, é importante também relacionar os custos financeiros envolvidos para implementação dos ambientes de teste, uma vez que sempre existem limitações financeiras nos projetos. Foi realizado um levantamento, aproximado, nos preços dos elementos utilizados para montagem dos ambientes de experimentação analisados, entre eles o *switch* Pronto Pica8 P-3295¹², os computadores baseados em placas Mini-ITX DN2800MT¹³ e as placas HP Ethernet 1Gb 4-port 331T Adapter¹⁴, que foram usadas na emulação de *switches* em computadores com processador Intel Core i7 2600¹⁵.

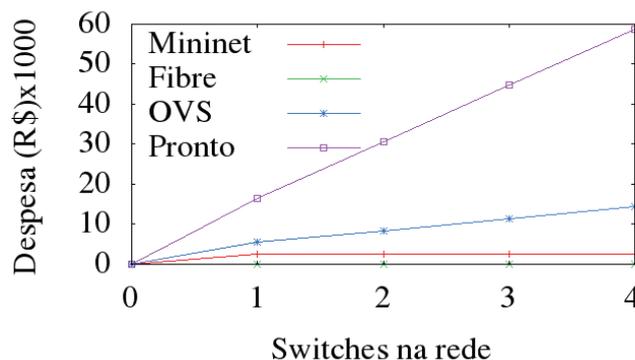


Figura 5. Demonstrativo de gastos por quantidade de *switches* com um *host* conectado por *switch* no ambiente de experimentação.

¹²<http://store.netgate.com/Pica8/P-3295.aspx>

¹³<http://www.atera.com.br/produto/DN2800MT/Placa+m%C3%A3e+Intel+DN2800MT+c-+Atom+Dual+Core+N2800+e+NM10>

¹⁴<http://eutec.com.br/placa-de-rede-hp-ethernet-1gb-4-port-331t-adapter-647594-b21>

¹⁵<http://ark.intel.com/PT-BR/products/52213/Intel-Core-i7-2600-Processor-8M-Cache-up-to-3.80-GHz>

Os valores apresentados na Figura 5 foram obtidos a partir das seguintes estimativas. Com o uso do FIBRE estima-se custo zero de montagem, somente sendo necessário algum dispositivo com acesso à internet. Com o Mininet o custo é determinado pela aquisição de um computador com configurações de memória entre 4GB ou 8GB de RAM e Intel Core i7 2600 (1 placa com 4 núcleos e 8 threads), monitor, teclado e mouse. Nos testes com OVS em desktop o custo é determinado pela aquisição de um desktop para o controlador e um por *switch*; 1 placa Mini-ITX por host implementado; e 1 placa HP Ethernet 1Gb 4-port 331T Adapter por *switch* implementado em computador. Nos testes com *switch* Pronto estima-se o custo referente ao uso de um desktop para funcionamento do controlador Ryu, 1 placa Mini-ITX por *host* e dos *switches* Pronto.

Na Figura 5 apresenta-se a evolução de custo para cada solução em função da quantidade de *switches* a serem utilizados, considerando o uso de um *host* por *switch* e um controlador por ambiente. Para o ambiente FIBRE e Mininet, o aumento de *switches* nas simulações não afeta os custos, uma vez que se trata de uma plataforma digital de uso compartilhado e um emulador de redes, respectivamente. Naturalmente, o aumento do número de *switches* é limitado em ambos casos, seja pelo número máximo de *switches* disponíveis, no caso do FIBRE, ou pela capacidade de processamento e memória da máquina, no caso do Mininet.

Tabela 1. Comparação de parâmetros analisados nos seus respectivos ambientes de simulação.

Parâmetros	Mininet	FIBRE	Mini-itx+ Desktop 4GB	Mini-Itx+ Desktop 8GB	Switch Pronto
Desempenho do <i>switch</i>	Médio	Alto	Alto	Alto	Alto
Desempenho do plano de controle	Baixo	Alto	Alto	Alto	Alto
Desempenho do cliente (host)	Baixo	Alto	Baixo	Baixo	Baixo
Controle de vazão	Baixo	Baixo	Alto	Alto	Alto
Celeridade para implantação	Alto	Baixo	Médio	Médio	Médio
Complexidade para as medições	Baixo	Alto	Médio	Médio	Médio
Custos de implantação	Baixo	N/A	Médio	Médio	Alto
Usa outras versões do OpenFlow	Sim	Não	Sim	Sim	Não
Permite controle in band	Não	Não	Sim	Sim	Não
Permite controle out band	Sim	Sim	Sim	Sim	Sim

A Tabela 1 visa comparar os diversos parâmetros analisados nos ambientes de experimentação averiguados. Outro foco é apresentar o desafio tanto no que tange a implantação quanto a obtenção dos elementos que possibilitam a análise dos ambientes. Os parâmetros foram classificados entre alto, médio e baixo de forma a mensurar a aptidão de cada parâmetro da tabela ao seu respectivo ambiente de experimentação. Em conjunto, foram apresentadas algumas linhas informando a disponibilidade ou não de itens possíveis de existir em ambientes de experimentação.

7. Conclusões e Trabalhos Futuros

Neste artigo, o comportamento de diferentes plataformas de experimentação baseadas em OpenFlow foi avaliado. A escolha do ambiente é uma das premissas para se

alcançar um resultado mais fiel. Uma vez que o desempenho do ambiente afeta diretamente o resultado da validação de novas propostas, testes que avaliem o tempo de processamento do ambiente, vazão, atraso, etc, são essenciais para uma avaliação mais assertiva.

A utilização do ambiente Mininet pode ser uma excelente escolha para o início de aprendizagem sobre o funcionamento do OpenFlow e do conceito de SDN, tendo a vantagem de ser uma máquina virtual obtida diretamente em sites especializados e facilmente colocada em qualquer computador com um *software* de virtualização. Porém, esta pode ser uma solução muito limitada se o usuário visa realizar testes de desempenho. Como observado, o Mininet apresentou, na maioria dos casos, um desempenho muito inferior quando comparado aos demais ambientes de teste.

Soluções que fazem uso de desktops físicos para implementação de *software* de *switch* OpenFlow apresentam bom desempenho e tornando-se uma boa opção para testes e estudos de novas soluções em SDN. O custo de utilização desse mecanismo de implementação apresenta valores financeiros abaixo do que os conseguidos com a utilização de um *switch* físico habilitado para OpenFlow e, como verificado nas simulações, a variação de memória no desktop utilizado não afeta de forma consistente os resultados obtidos. Contudo, dependendo do porte do teste, esta pode ser uma solução inviável em termos de investimento inicial para montagem da rede de teste.

O ambiente FIBRE apresentou aspectos positivos e negativos. O ambiente utiliza tanto *switches* OpenFlow comerciais quanto *switches* emulados em computadores, mas tem alta variabilidade no desempenho observado, pois o ambiente é compartilhado entre pesquisadores. Não existe controle de banda mínima ou de disponibilidade de recursos nos servidores de máquinas virtuais. Por outro lado, o FIBRE tem nós espalhados por diversos locais do Brasil e do mundo, tornando o cenário mais real para experimentação. Além disso, apresentou bons resultados com relação à capacidade dos planos de dado e de controle e custo de implementação zero, tornando-se uma boa ferramenta, dependendo dos requisitos do teste.

Portanto, existem várias possibilidades de ambientes de experimentação para aplicações baseadas em OpenFlow, mas nem todos os ambientes são propícios para qualquer tipo de teste. Além disso, ficam claras as restrições de cada um dos ambientes e quais são os impactos dele sobre os resultados observados. Assim, as conclusões obtidas em cada tipo de ambiente devem sempre ser balizadas pelas restrições específicas do cenário.

Como trabalhos futuros, pretende-se realizar o teste com hosts mais eficientes, permitindo uma comparação mais justa na geração de novos fluxos. Utilizar geradores de pacotes com maior capacidade e menor custo computacional, realizar testes do Mininet usando um controlador externo, desenvolver mais de um *switch* por *desktop* para avaliação do comportamento e avaliar outros testebeds públicos.

8. Agradecimentos

Agradecemos a todas as pessoas que, direta ou indiretamente contribuíram com carinho e atenção durante a construção desse trabalho, especialmente ao CNPq, CAPES e FAPERJ pelo apoio financeiro e a todos os envolvidos da Rede Nacional de Ensino e Pesquisa (RNP) pelo apoio com os testes realizados na plataforma FIBRE.

Referências

- Bianco, A., Birke, R., Giraudo, L., and Palacin, M. (2010). OpenFlow switching: Data plane performance. In *ICC, 2010 IEEE*, pages 1–5. IEEE.
- Conterato, M., de Oliveira, I., Ferreto, T., and De Rose, C. A. (2013). Avaliação do suporte à simulação de redes OpenFlow no NS-3. *Anais da 11a. Escola Regional de Redes de Computadores*.
- Costa, L. C., Vieira, A. B., de Britto, E., Silva, D. F., Gomes, G., Correia, L. H., and Vieira, L. F. (2016). Avaliação de Desempenho de Planos de Dados OpenFlow.
- Das, S., Parulkar, G., McKeown, N., Singh, P., Getachew, D., and Ong, L. (2010). Packet and circuit network convergence with OpenFlow. In *Optical Fiber Communication Conference*, page OTuG1. Optical Society of America.
- de Oliveira, R. L. S., Schweitzer, C. M., Shinoda, A. A., and Prete, L. R. (2014). Using Mininet for emulation and prototyping software-defined networks. In *COLCOM, 2014 IEEE on*, pages 1–6. IEEE.
- Feamster, N., Rexford, J., and Zegura, E. (2014). The road to SDN: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98.
- Goransson, P. and Black, C. (2014). *Software Defined Networks: A Comprehensive Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.
- Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., and McKeown, N. (2010). ElasticTree: Saving Energy in Data Center Networks. In *NSDI*, volume 10, pages 249–264.
- Huang, T., Yu, F. R., Zhang, C., Liu, J., Zhang, J., and Liu, J. (2016). A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. *IEEE Surveys & Tutorials*.
- Katta, N., Alipourfard, O., Rexford, J., and Walker, D. (2014). Infinite CacheFlow in software-defined networks. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 175–180. ACM.
- Keti, F. and Askar, S. (2015). Emulation of Software Defined Networks Using Mininet in Different Simulation Environments. In *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*, pages 205–210. IEEE.
- Macedo, D. F., Guedes, D., Vieira, L. F., Vieira, M. A., and Nogueira, M. (2015). Programmable networks-from software-defined radio to software-defined networking. *IEEE Surveys & Tutorials*, 17(2):1102–1125.
- Masoudi, R. and Ghaffari, A. Software defined networks: A survey.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM*, pages 69–74.
- Ortiz, J., J.Londoño, and Novillo, F. (2016). Evaluation of performance and scalability of Mininet in scenarios with large data centers.
- Zal, M. and Kleban, J. (2014). Performance Evaluation of OpenFlow Devices.