

OFSwitch13: Viabilizando o uso do OpenFlow 1.3 no ns-3

Luciano Jerez Chaves^{1,2}, Islene Calciolari Garcia², Edmundo R. Mauro Madeira²

¹ Depto. de Ciência da Computação – Universidade Federal de Juiz de Fora (UFJF)
Rua José Lourenço Kelmer, s/n. São Pedro, Juiz de Fora – MG.

² Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Av. Albert Einstein, 1251. Cidade Universitária, Campinas – SP.

{lchaves, islene, edmundo}@ic.unicamp.br

Abstract. *The OpenFlow is a fast evolving technology that enables the implementation of Software Defined Networks. Despite the fact that the Network Simulator 3 (ns-3) already has a module for OpenFlow simulations, it is possible to note that the available implementation is very outdated. As many new major features were introduced up to the latest versions, it is desired to have them available for use. Thus, this paper presents the OFSwitch13: a module to enhance the ns-3 with OpenFlow version 1.3 support. The paper describes the module design and implementation, and a case study scenario to demonstrate some of the available features.*

Resumo. *O OpenFlow é uma tecnologia em rápida evolução que permite a implementação de Redes Definidas por Software. Apesar do simulador de redes ns-3 já possuir um módulo para simulações com o OpenFlow, é possível observar que a versão existente está bastante ultrapassada. Como diversas novas funcionalidades foram introduzidas nas versões mais recentes, é desejável que elas estejam disponíveis para uso. Assim, este artigo apresenta o OFSwitch13: um módulo que habilita o ns-3 para simulações com o OpenFlow versão 1.3. O artigo descreve o projeto e a implementação do módulo, além de um estudo de caso para demonstrar algumas das funcionalidades presentes.*

1. Introdução

Como indicado pela [Open Networking Foundation 2012], os desafios na gerência de redes de estão se intensificando à medida que o número de dispositivos aumenta, sendo que atender às demandas do mercado é praticamente impossível com as arquiteturas tradicionais. Neste contexto, as Redes Definidas por Software (do Inglês *Software Defined Networking* – SDN) surgiram como um paradigma capaz de permitir a implantação de redes de maneira rápida e com baixo custo. Na arquitetura SDN, os planos de controle e de dados são desacoplados, a inteligência é logicamente centralizada em *software* enquanto a infraestrutura da rede é construída com comutadores (*switches*) programáveis de baixo custo. Com a inteligência centralizada é possível simplificar a tomada de decisões com base na visão global (ou parcial) da rede. O protocolo OpenFlow [McKeown et al. 2008] foi a primeira tecnologia desenvolvida especificamente para a realização do SDN, e tem sido adotado por fabricantes de equipamentos e também pela comunidade acadêmica.

Como se sabe, é custoso construir um ambiente de testes contendo vários equipamentos de rede para validar um determinado protocolo ou algoritmo. Nestes casos, ferramentas de *software* economizam muito dinheiro e tempo para realizar esta tarefa. Uma

escolha razoável seria utilizar um ambiente simulado para este fim, como o oferecido pelo simulador de redes *ns-3* [ns-3 2016]. O *ns-3* é um simulador de eventos discretos, desenvolvido para fins educacionais e de pesquisas, e distribuído como *software* livre. Simulações com o *ns-3* podem modelar comutadores OpenFlow através de um módulo disponível na distribuição padrão [OpenFlow for ns-3 2010]. Entretanto, este módulo implementa uma versão ultrapassada do protocolo [OpenFlow v0.8.9 2008]. Como diversas funcionalidades foram introduzidas nas versões mais recentes, é interessante que elas também estejam disponíveis para uso.

Neste cenário, este artigo apresenta o módulo `OFSwitch13` [OFSwitch13 2016] para o simulador *ns-3*. Este módulo permite a realização de simulações com o protocolo [OpenFlow v1.3.5 2015], equipando o simulador com o comutador de rede e com a interface para controlador. Com este módulo é possível interconectar nós da simulação através dos comutadores para enviar e receber tráfego de rede. Para gerenciar os comutadores, a interface para controlador pode ser estendida de maneira a implementar a lógica de controle desejada. O módulo depende de uma biblioteca externa chamada `ofsoftswitch13` [Fernandes e Rothenberg 2014], que de fato implementa e gerencia o caminho de dados do comutador OpenFlow. A biblioteca também possui a ferramenta `dpctl`, usada na criação de mensagens OpenFlow a partir de linhas de comando, e o código necessário para converter as mensagens OpenFlow para o formato de rede.

O restante deste artigo está organizado como segue: a Seção 2 descreve em linhas gerais o protocolo OpenFlow; a Seção 3 detalha a arquitetura do módulo `OFSwitch13` e seus componentes internos; a Seção 4 apresenta o cenário usado para demonstrar a ferramenta; e a Seção 5 conclui o artigo, apontando alguns trabalhos futuros.

2. O protocolo OpenFlow

O protocolo OpenFlow permite a construção de redes SDN através do controle refinado de tráfego em múltiplos equipamentos de redes, inclusive de diferentes fabricantes. O OpenFlow utiliza o conceito de fluxos para identificar um tráfego, com base em regras que podem ser configuradas pelo controlador dinamicamente. A especificação do OpenFlow cobre os componentes internos e as funções básicas do comutador, além do protocolo para comunicação entre comutador e controlador. A Figura 1 ilustra esta arquitetura.

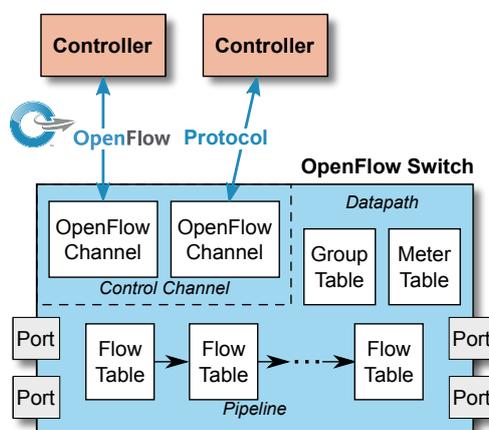


Figura 1. Arquitetura do comutador OpenFlow [OpenFlow v1.5.1 2015].

O caminho de dados do comutador OpenFlow consiste de um *pipeline* com uma ou mais *tabelas de fluxo*, que possuem entradas configuradas pelo controlador. Cada entrada de fluxo consiste de campos extensíveis para identificar o fluxo – *OpenFlow eX-tensible Match* (OXM), contadores, e um conjunto de instruções para aplicar aos pacotes que coincidem com esta entrada. A busca começa na primeira tabela, em ordem de prioridade, e pode continuar pelas tabelas seguintes. Se um determinado pacote atende aos campos de uma entrada, então as instruções associadas à entrada são aplicadas ao pacote. As instruções podem modificar o processamento no *pipeline*, enviando o pacote para alguma das tabelas seguintes. Também podem conter ações que descrevem o encaminhamento do pacote, modificações em campos internos, e processamento por grupos e bandas de medição. A ação mais comum é a de saída, que encaminha o pacote para uma porta de saída. Outras ações disponíveis incluem a ação de grupo e a ação de medição, que direcionam o pacote para a *tabela de grupo* ou *tabela de medição*, respectivamente. Grupos representam conjunto de ações usados em semânticas de encaminhamento complexas, enquanto as entradas de medição permitem que o comutador controle a taxa de *bits* dos fluxos, limitando a banda passante em um valor pré-definido.

Para a comunicação entre comutador e controlador é utilizado o *canal OpenFlow*, protegido por *Transport Layer Security* (TLS), mas que também pode ser construído sobre TCP aberto. É através deste canal que o controlador configura as tabelas no comutador. O comutador deve suportar um único canal OpenFlow ou múltiplos canais, permitindo que diversos controladores compartilhem a tarefa de gerenciamento. O leitor pode consultar a especificação [OpenFlow v1.5.1 2015] para detalhes do protocolo.

3. O módulo OFSwitch13

A Figura 2 mostra a arquitetura do módulo `OFSwitch13`, destacando seus dois principais elementos: o *comutador* e a *interface para controlador* OpenFlow 1.3. Estes elementos interagem com a biblioteca externa `ofsoftswitch13`, e comunicam entre si pelo canal OpenFlow de controle, construído sobre protocolos e interfaces de rede disponíveis no *ns-3*. As subseções a seguir descrevem detalhadamente esta arquitetura.

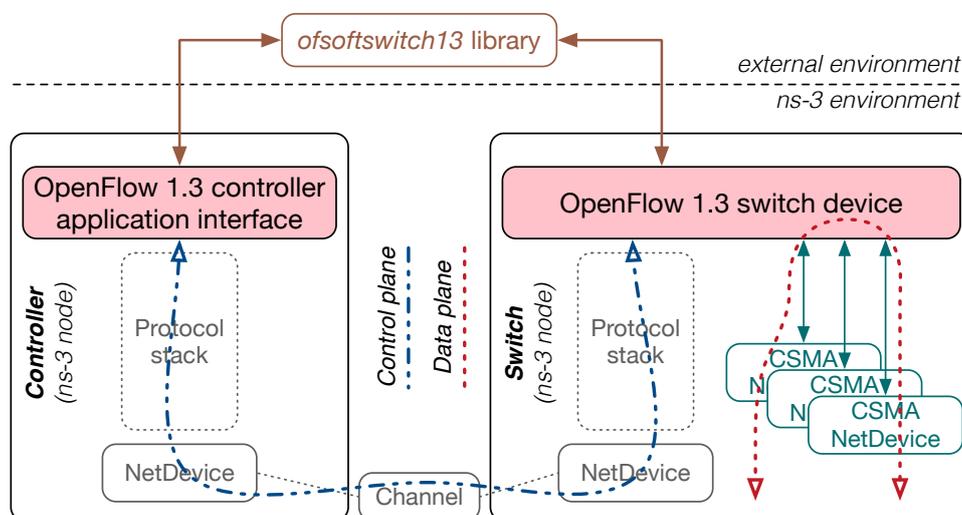


Figura 2. A arquitetura do módulo `OFSwitch13`.

3.1. Comutador OpenFlow 1.3

O comutador OpenFlow 1.3, ilustrado na Figura 3a, é usado para interconectar nós do *ns-3* através de interfaces do tipo *Carrier Sense Multiple Access* (CSMA), agindo como intermediário no encaminhamento dos pacotes. O comutador possui uma coleção de portas, cada uma associada à uma interface CSMA subjacente. Os pacotes entram nas portas através do *OpenFlow trace source* na interface CSMA. Esta nova fonte de rastreamento é configurada em modo promíscuo, e fornece os pacotes recebidos pela interface subjacente com seus cabeçalhos Ethernet. A inclusão desta fonte de rastreamento é a única modificação necessária ao código-fonte do *ns-3* para a integração com o `OFSwitch13`.

A implementação do caminho de dados do comutador (tabelas de fluxo, de grupo e de medição) são fornecidas pela biblioteca `ofsoftswitch13`. Por esta razão, os pacotes que entram no comutador são processados pelas funções da biblioteca. Para modelar o funcionamento do *hardware*, o módulo estima o tempo médio de busca nas tabelas de fluxo, tempo este usado para atrasar o início do processamento do pacote pela biblioteca. Para encontrar um atraso realista, o módulo considera que implementações reais do protocolo OpenFlow se baseiam em algoritmos de busca sofisticados. Como a maioria desses algoritmos realizam busca binária em árvores, a equação $k * \log_2(n)$ é usada para estimar o atraso, onde k é uma constante que representa o tempo necessário para realizar uma única operação do tipo *Ternary Content-Addressable Memory* (TCAM) em *hardware*, e n é o número de entradas nas tabelas de fluxo do comutador.

Pacotes já processados pela biblioteca e marcados com uma ação de saída são enviados para a *fila OpenFlow* na porta de saída. O protocolo OpenFlow provê suporte limitado de Qualidade de Serviço (QoS) através de um mecanismo de filas com configurações distintas, onde uma ou mais filas podem estar associadas à uma porta, e as entradas de fluxos podem fazer o mapeamento dos pacotes nas filas. A fila OpenFlow possui uma coleção de filas internas, que são usadas como filas de prioridade para este fim.

3.2. Interface para controlador OpenFlow 1.3

A *interface para controlador* traz as funcionalidades básicas para a implementação de um controlador OpenFlow 1.3. Ela gerencia uma coleção de comutadores OpenFlow, conforme representado na Figura 3b. Para construir as mensagens de configuração no formato OpenFlow, a interface para controlador se baseia na ferramenta `dpctl`, disponível na biblioteca `ofsoftswitch13`. Com uma linha de comando de sintaxe simples, esta

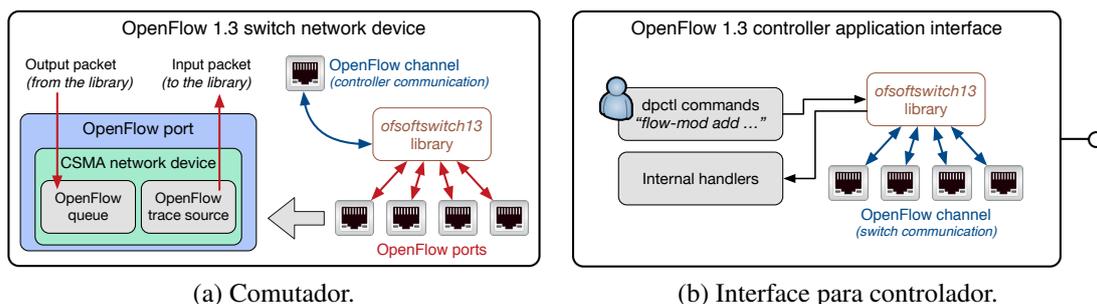


Figura 3. Elementos da arquitetura do módulo `OFSwitch13`.

ferramenta pode ser utilizada para adicionar entradas nas tabelas, solicitar ao comutador por informações de estado, e mesmo modificar outras configurações do equipamento. Para as mensagens OpenFlow recebidas dos comutadores, a interface para controlador utiliza a coleção interna de manipuladores para tratar as mensagens de acordo com seu tipo. Alguns manipuladores podem ser sobrescritos pelos controladores que implementam a interface, de maneira a programar a lógica de controle desejada.

O módulo `OFSwitch13` também possui uma implementação desta interface com a lógica de um *controlador de auto-aprendizagem*. Este controlador funciona como um comutador tradicional, aprendendo em quais portas estão os nós da rede, e simplificando o processo de encaminhamento dos pacotes. Observe que o algoritmo de árvore geradora mínima não está implementado neste controlador, o que o impede de funcionar adequadamente em topologias de rede com ciclos.

3.3. Canal OpenFlow 1.3

O *canal OpenFlow* permite a comunicação entre o controlador e o(s) comutador(es). É através desta interface que o controlador configura e gerencia o caminho de dados. No módulo `OFSwitch13`, o canal OpenFlow é implementado por uma rede dedicada (fora de banda), usando as interfaces e canais disponíveis do *ns-3*. Pode-se optar por um canal OpenFlow compartilhado ou por conexões individuais entre cada comutador e o controlador. Este modelo provê conexões no plano de controle realista, que incluem atraso e, opcionalmente, modelos de erros. Ele também simplifica a análise do protocolo OpenFlow através do mecanismo de rastreamento do *ns-3*, que pode ser usado para gerar arquivos PCAP a serem lidos posteriormente por *softwares* convencionais.

Considerando que as mensagens no canal OpenFlow seguem o formato indicado na especificação, também é possível usar o módulo `TapBridge` do *ns-3* para integrar um *controlador externo*, rodando na máquina local, ao ambiente simulado. Entretanto, este caso de uso ainda não foi validado pelos autores.

3.4. Biblioteca `ofsoftswitch13`

O módulo `OFSwitch13` foi projetado para trabalhar em conjunto com a biblioteca `ofsoftswitch13`. É a biblioteca que, de fato, implementa o caminho de dados do comutador OpenFlow. Isto inclui as portas de entrada e saída, as tabelas de fluxo, de grupo e de medição. A ferramenta `dpctl`, disponível também na biblioteca, é usada para converter linhas de comando em mensagens no formato do protocolo OpenFlow 1.3. Para realizar a decodificação e a análise dos pacotes nas tabelas, a biblioteca utiliza o `NetBee` [NetBee 2016], que permite a descrição dos cabeçalhos de pacotes de redes através da linguagem *NetPDL*.

O código original do `ofsoftswitch13` foi ligeiramente modificado para a integração com o módulo `OFSwitch13` [OFSoftSwitch13 for ns-3 2016]. Entretanto, não houveram modificações na implementação do caminho de dados, que é mantida no repositório original e regularmente sincronizada com o repositório modificado. Para a integração, as portas de entrada e saída do `ofsoftswitch13` foram deixadas de lado, e a biblioteca foi modificada para enviar e receber pacotes diretamente para/do ambiente do *ns-3*. Para isso, todas as funções relacionadas com o envio e recebimento de pacotes foram anotadas como *símbolos fracos*, permitindo que o módulo sobrescreva essas funções

em tempo de ligação. Esta mesma estratégia foi usada para sobrescrever as funções relacionadas ao tempo, garantindo a consistência entre os tempos do simulador e da biblioteca. A integração também se baseia em chamadas de retorno usadas pela biblioteca para notificar eventos como descarte de pacotes por medições de banda, modificação no conteúdo do pacote por instruções nas tabelas de fluxo, pacotes duplicados por ações de grupo, e pacotes armazenados no comutador e enviados ao controlador.

Na tentativa de otimizar o processo de conversão de pacotes entre o formato otimizado do *ns-3* e a representação serial dos mesmos usada pela biblioteca, quando um pacote é enviado para processamento nas tabelas do OpenFlow, o módulo armazena temporariamente o pacote original do *ns-3*. Para os pacotes processados sem modificações no conteúdo, o comutador encaminha o pacote original do *ns-3* pela porta de saída específica. No caso de modificações no conteúdo, o comutador cria um novo pacote com o conteúdo modificado (copiando também os metadados do pacote). Esta abordagem se mostrou mais simples do que identificar quais foram as mudanças realizadas no pacote pela biblioteca.

3.5. Limitações atuais

Uma das limitações do módulo na implementação atual está na compatibilidade apenas com a plataforma GNU/Linux e o GNU *Compiler Collection* (GCC), por conta das modificações na biblioteca para integração com o simulador. Em relação ao protocolo OpenFlow 1.3, são duas as limitações principais: ausência de *conexões auxiliares*, permitindo apenas uma única conexão entre o comutador e o controlador; e ausência de *suporte para múltiplos controladores*, forçando cada comutador a ser gerenciado por apenas um controlador. De acordo com a especificação do OpenFlow, conexões auxiliares poderiam ser criadas pelo comutador para aumentar o desempenho e explorar o paralelismo, enquanto múltiplos controladores aumentariam a confiabilidade já que o comutador poderia continuar operando mesmo que um controlador ou uma conexão apresentassem falhas.

4. Código, documentação e demonstração do módulo

Na página do projeto em <http://www.lrc.ic.unicamp.br/ofswitch13>, os visitantes irão encontrar uma breve descrição da ferramenta, além das informações necessárias para começar a usá-la. Existem ligações para os repositórios onde estão armazenados os códigos (Bitbucket para o código do módulo `OFSwitch13` e GitHub para a biblioteca `ofsoftswitch13` modificada), além da documentação que inclui o tutorial de instalação. Há também uma máquina virtual pré-configurada para facilitar o uso da ferramenta por novos usuários.

O módulo é licenciado sob a GNU GPL (mesma licença do simulador *ns-3*), enquanto a biblioteca modificada acompanha sua licença original BSD. A documentação do código-fonte gerada pelo Doxygen (classes e funções) pode ser consultada no endereço <http://www.lrc.ic.unicamp.br/ofswitch13/doc/html/>. O uso da ferramenta de *issues* no repositório do Bitbucket é encorajada para reportar erros e discutir sobre as funcionalidades do módulo.

Para a demonstração da ferramenta, a Figura 4 mostra a topologia de rede a ser utilizada. Ela representa a rede interna de uma organização, onde os servidores e os clientes estão separados por duas conexões de longa distância de 10 Mbps cada. Do lado servidor, o *comutador de borda* faz o papel do roteador ao responder as requisições dos

clientes e direcioná-las para o servidor interno adequado. Do lado cliente, o *comutador de agregação* é usado para gerenciar as conexões de longa-distância enquanto o *comutador cliente* concentra as conexões. O controlador de auto-aprendizado padrão é usado para gerenciar o comutador cliente, enquanto um novo *controlador de QoS* é responsável pelos demais comutadores. Os códigos do *ns-3* para esta demonstração estão disponíveis no diretório `examples/qos-controller/` do código-fonte do módulo `OFSwitch13`.

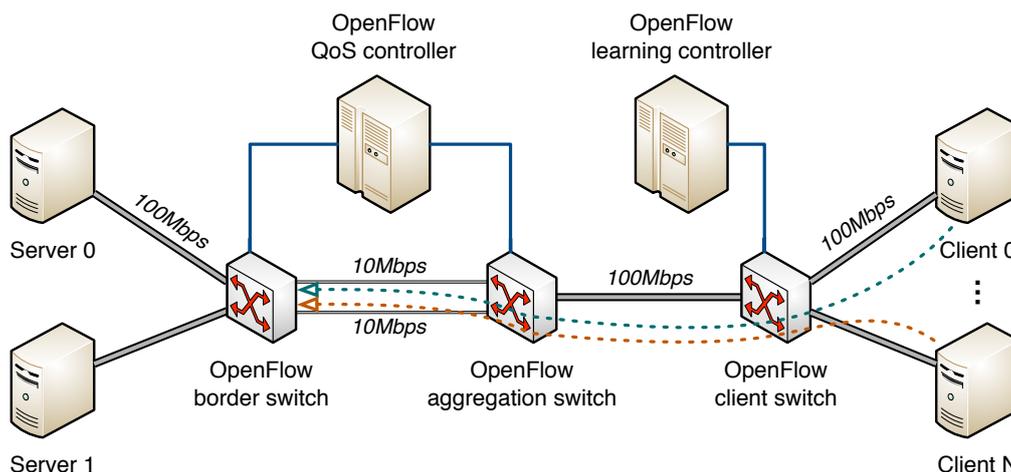


Figura 4. Topologia de rede para a demonstração.

A demonstração consiste em explorar algumas funcionalidades do OpenFlow 1.3 para implementar os seguintes mecanismos no novo controlador de QoS:

- **Agregação de enlaces:** Utilizar a tabela de grupos do OpenFlow (introduzida na versão 1.1), para realizar a agregação dos enlaces de longa distância. A agregação de enlaces pode ser usada para combinar diversas conexões de rede em paralelo, de maneira a aumentar a largura de banda. O objetivo é configurar os comutadores de agregação e de borda para dividir o tráfego pelos enlaces, aumentando a largura de banda entre clientes e servidores para 20 Mbps.
- **Balanceamento de carga:** Utilizar os campos extensíveis do OpenFlow (introduzidos na versão 1.2), para prover um serviço a partir de vários servidores, também conhecido como fazenda de servidores. O objetivo é configurar o comutador de borda para filtrar as requisições de ARP procurando pelo endereço IP do serviço e enviá-las ao controlador, para que este possa respondê-las de maneira a distribuir a carga de trabalho entre os servidores. Assim, o controlador instala as regras de fluxo nas tabelas do comutador de borda para que os próximos pacotes sejam diretamente encaminhados para o servidor escolhido, realizando também as modificações internas nos endereços de origem e destino dos pacotes de maneira que todo o processo seja transparente aos clientes.
- **Controle de fluxo:** Utilizar a tabela de medição do OpenFlow (introduzida na versão 1.3), para limitar a vazão das conexões de cada cliente para um valor pré-definido. O objetivo é configurar o comutador de agregação para controlar a vazão dos fluxos e evitar que eles sobrecarreguem as conexões de longa distância.

5. Conclusões e trabalhos futuros

Este artigo apresentou o módulo `OFSwitch13`, capaz de viabilizar simulações com o OpenFlow 1.3 no simulador de redes *ns-3*. O módulo é distribuído como *software* livre, e requer mudanças mínimas no código-fonte do *ns-3* para integração. Para demonstrar a ferramenta foram exploradas algumas características do OpenFlow 1.3, a saber: uso de tabelas de grupos para agregação de enlaces, o uso de campos extensíveis para balanceamento de carga, e uso de tabelas de medição para controle de vazão. O módulo `OFSwitch13` também foi usado para simulações no *ns-3* integrando o OpenFlow e redes *Long Term Evolution* (LTE), que são descritas em outro trabalho dos mesmos autores [Chaves et al. 2015]. Como trabalhos futuros, pretende-se eliminar as limitações atuais do módulo, além de implementar novas funcionalidades de QoS compatíveis com a recente camada de controle de tráfego do *ns-3*. Como este é um *software* livre, contribuições de desenvolvedores e usuários são sempre bem-vindas.

6. Agradecimentos

Os autores gostariam de agradecer ao Vítor M. Eichemberger e ao Eder L. Fernandes pelas contribuições, à Capes e ao CNPq pelo apoio financeiro (processo 118198/2014-9).

Referências

- Chaves, L. J., Eichemberger, V. M., Garcia, I. C., e Madeira, E. R. M. (2015). Integrating OpenFlow to LTE: some issues toward Software-Defined Mobile Networks. Em *International Conference on New Technologies, Mobility and Security (NTMS)*, páginas 1–5. IEEE.
- Fernandes, E. L. e Rothenberg, C. E. (2014). OpenFlow 1.3 software switch. Em *Simpósio Brasileiro de Redes de Computadores (SBRC)*, páginas 1021–1028. SBC.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., e Turner, J. (2008). OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.
- NetBee (2016). The NetBee library. Disponível em: <http://www.nbee.org/doku.php>.
- ns-3 (2016). Network Simulator 3. Disponível em: <http://www.nsnam.org>.
- OFSoftSwitch13 for ns-3 (2016). OpenFlow 1.3 software switch for ns-3. Disponível em: <https://github.com/ljerezchaves/ofsoftswitch13>.
- OFSwitch13 (2016). The OpenFlow 1.3 module for ns-3. Disponível em: <http://www.lrc.ic.unicamp.br/ofswitch13>.
- Open Networking Foundation (2012). Software-Defined Networking: The new norms for networks. ONF White Paper.
- OpenFlow for ns-3 (2010). GSoC 2010 OpenFlow. Disponível em: <http://www.nsnam.org/wiki/GSOC2010OpenFlow>.
- OpenFlow v0.8.9 (2008). OpenFlow switch spec. version 0.8.9. Open Networking Foundation.
- OpenFlow v1.3.5 (2015). OpenFlow switch spec. version 1.3.5. Open Networking Foundation.
- OpenFlow v1.5.1 (2015). OpenFlow switch spec. version 1.5.1. Open Networking Foundation.