

Uma proposta de arquitetura para o provisionamento de circuitos dinâmicos sobre redes definidas por software

Daniel de Arêa Leão Marques¹, Sidney Cunha de Lucena¹,
Airton Ishimori², Fernando Farias², Antônio Abelém²

¹Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro – RJ

²Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA)
Belém – PA

{daniel.marques, sidney}@uniriotec.br

{airton, fernnf, abelem}@ufpa.br

Abstract. *The OpenFlow technology enables the creation of a programmable layer over the control-plane of a network, in this way dictating the data-plane forwarding behaviour through the use of applications plugged to a network controller. Dynamic Circuit Network is an architecture that permits the scheduling of network resources on virtual circuits, such as bandwidth, over multiple domains with heterogeneous technologies. Recently, researches were conducted in order to integrate both technologies so that dynamic circuits can be dynamically provisioned over OpenFlow domains. The objective of this work is to propose an architecture that enables such provision, maintaining the QoS requisites of a DCN architecture.*

Resumo. *A tecnologia OpenFlow permite a construção de uma camada programável sobre o plano de controle de uma rede, desta forma ditando o encaminhamento no plano de dados através do uso de aplicações associadas a um controlador de rede. Uma rede de circuitos dinâmicos é uma arquitetura que permite agendar recursos de rede, como a largura de banda, através de circuitos virtuais entre múltiplos domínios com tecnologias heterogêneas. Recentemente, pesquisas foram conduzidas visando integrar ambas tecnologias de forma que circuitos possam ser dinamicamente provisionados sobre domínios OpenFlow. O objetivo deste trabalho é propor uma arquitetura que possibilite este provisionamento, mantendo os requisitos de QoS de uma arquitetura DCN.*

1. Introdução

A comunidade científica está cada vez mais realizando projetos colaborativos de grande escala, conforme notado em [Guok et al. 2008]. Estas cooperações costumam utilizar recursos computacionais espalhados ao redor do planeta e são fortemente dependentes de redes de alto desempenho para que seus resultados se integrem e alcancem sucesso. No entanto, os requisitos de rede necessários para oferecer este serviço aos pesquisadores estão além do que a tradicional infraestrutura de encaminhamento por melhor esforço é

capaz de oferecer. Mesmo com a sua evolução ao longo dos anos, essa infraestrutura ainda não oferece, de maneira satisfatória, previsibilidade, flexibilidade e garantias.

Para solucionar este problema de falta de garantia, foi desenvolvida uma arquitetura de rede denominada Redes de Circuitos Dinâmicos (*Dynamic Circuit Network* ou DCN). Esta arquitetura foi proposta por um conjunto de organizações e consiste de uma rede óptica que fornece banda dedicada através de circuitos virtuais para aplicações exigentes [Internet2 2012].

Além da DCN, um segundo paradigma que possibilita novas formas de se controlar uma infraestrutura de rede são as Redes Definidas por *Software* (*Software Defined Network* ou SDN). Neste paradigma, o encaminhamento no plano de dados é realizado pela ação do plano de controle através de uma interface programática, desta forma flexibilizando e otimizando o processo de configuração de uma rede. O maior expoente deste paradigma é o protocolo OpenFlow [McKeown et al. 2008], que padroniza a instalação de tabelas de fluxo nos equipamentos da rede, desta forma ditando o encaminhamento dos pacotes.

Recentemente foi apresentada em [Boote et al. 2011] a integração destas duas tecnologias e, apesar da união ter sido concretizada, a construção de circuitos dinâmicos de forma eficiente e satisfatória sobre uma rede OpenFlow não foi conceitualmente estruturada. Desta forma, não há um consenso sobre como estabelecer uma rede de circuitos dinâmicos sobre um domínio OpenFlow e, ainda assim, garantir o provisionamento dos recursos de rede. No presente trabalho, é proposta uma arquitetura alternativa para a implementação de DCN sobre domínios OpenFlow, assim como uma averiguação de seus mecanismos.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta o ferramental utilizado pelas tecnologias apresentadas, a Seção 3 apresenta a arquitetura vigente utilizada para integrar o DCN com OpenFlow, a Seção 4 apresenta a proposta de uma arquitetura alternativa para o provisionamento de circuitos dinâmicos sobre um domínio OpenFlow, na Seção 5 é feita uma avaliação dos mecanismos empregados nesta nova arquitetura e, na Seção 6, é apresentada a conclusão e os trabalhos futuros.

2. Redes de Circuitos Dinâmicos e OpenFlow

Uma Rede de Circuitos Dinâmicos necessita de um conjunto de aplicações para que a criação de circuitos com banda garantida sobre múltiplos domínios seja viabilizada. Dentre os *softwares* utilizados para esta função, o mais importante é o **OSCARS** [Guok et al. 2006], que é uma ferramenta que atua no interdomínio, ou seja, é responsável pela negociação entre diferentes domínios objetivando a reserva e o provisionamento fim-a-fim de circuitos virtuais com garantia de banda.

A Figura 1 representa a arquitetura DCN. Através de uma interface baseada em *web services*, o usuário agenda uma reserva com o IDC (*Interdomain Controller*, representado pelo *software* OSCARS), que verificará com os outros IDCs ao longo do caminho a disponibilidade de recursos. Caso haja, o OSCARS permitirá o agendamento do pedido, caso contrário, irá indeferir-lo. No momento agendado, as configurações serão encaminhadas para os DCs (*Domain Controllers*), que auxiliam no provisionamento dos circuitos nos respectivos domínios.

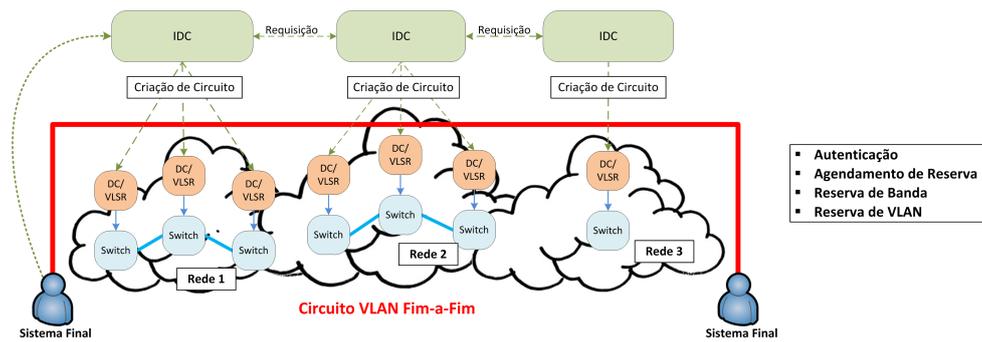


Figura 1. Arquitetura DCN.

A arquitetura de uma rede OpenFlow consiste de um conjunto de *switches* com tecnologia OpenFlow e de um controlador externo, que é o elemento responsável por gerenciar as tabelas de fluxo dos *switches* OpenFlow e determinar o manuseio de pacotes sem uma entrada de fluxo válida nestas tabelas.

Dentre as opções de *softwares* para atuarem como controlador, destaca-se o **NOX-Classic** [Gude et al. 2008]. O **NOX-Classic** é um controlador que fornece uma API que permite o desenvolvimento de *plugins* para ditar o comportamento da rede OpenFlow. Um outro conjunto de ferramentas bastante usado para ambientes experimentais e redes virtualizadas são os chamados *software switches*. Dentre estes, o **Open vSwitch** [Pettit et al. 2010] é um *software switch* multicamadas que suporta as funcionalidades de um *switch* de borda avançado, como NetFlow, sFlow, *port mirroring*, QoS e afins, além de funcionar como *switch* OpenFlow.

Além destas ferramentas, existe o *framework* **QoSFlow** [Ishimori et al. 2012], que estende o protocolo OpenFlow e fornece primitivas que possibilitam o gerenciamento de QoS nestes domínios através de um controlador OpenFlow. Com isto, é possível configurar dinamicamente os mecanismos de QoS nas portas dos *switches* OpenFlow. Para a implementação deste *framework*, são disponibilizados o controlador **NOX QoSFlow** e um *software switch* contendo a API do QoSFlow.

3. Arquitetura básica de DCN sobre Rede OpenFlow

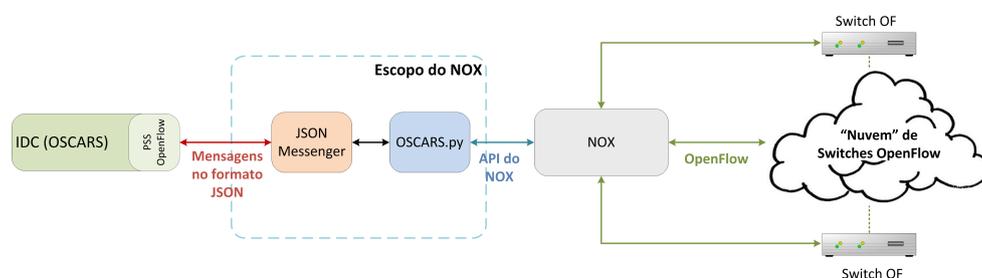


Figura 2. Arquitetura OSCARS com OpenFlow.

A arquitetura apresentada na Figura 2 é o modelo atualmente vigente para permitir uma DCN sobre domínio OpenFlow, consistindo no uso da ferramenta OSCARS em conjunto com o controlador NOX e *switches* do protocolo OpenFlow 1.0.

O funcionamento desta arquitetura começa com o OSCARS, que calcula o caminho de um circuito em uma determinada topologia. Após esta etapa, o PSS OpenFlow, mecanismo que o OSCARS utiliza para interagir com o controlador NOX, se comunica com a aplicação JSON Messenger, aplicação que permite a interação com o controlador a partir de mensagens JSON. O JSON Messenger repassará então as mensagens para a aplicação OSCARS.py associada ao NOX, que serão analisadas e convertidas em comandos de adição ou remoção de fluxos (no caso, VLANs), que deverão ser encaminhados para os *switches* de um domínio OpenFlow.

Apesar desta arquitetura realizar a citada integração, possui como limitação a incapacidade de configurar automaticamente o QoS nos *switches* OpenFlow da rede, o que poderia prejudicar o automatismo da DCN e a flexibilidade inerente a uma SDN, já que o operador teria que manualmente configurar os mecanismos de QoS nos equipamentos. Esta incapacidade acontece porque o protocolo OpenFlow 1.0 não provê a configuração de filas, conforme apontado em [Consortium 2009]. Desta forma, é possível apenas uma abordagem estática de aplicação de QoS, ou seja, usando filas previamente configuradas nos *switches* OpenFlow.

4. Arquitetura NOX-QoSFlow-OSCARS



Figura 3. Integração entre as soluções OSCARS e QoSFlow.

Perante a inabilidade da arquitetura básica provisionar circuitos que estejam plenamente de acordo com a arquitetura DCN, sem a utilização de mecanismos estáticos para prover QoS, foi proposta uma alternativa representada pela Figura 3. Nesta nova arquitetura é usado o OSCARS em conjunto com o QoSFlow, onde as mensagens JSON enviadas pelo OSCARS são tratadas para uso com a API do QoSFlow. Dentre as possíveis formas de se implementar este tratamento, a que se demonstrou mais vantajosa partiu da criação de um novo componente para o QoSFlow. Este novo componente, QoS-OSCARS, está integrado à API do QoSFlow, e é capaz de interpretar as mensagens JSON enviadas pelo OSCARS e delas extrair os dados necessários para executar a reserva de banda solicitada.

Esta arquitetura, por fazer uso do QoSFlow, é capaz de configurar automaticamente o QoS nas portas dos *switches* através do protocolo OpenFlow 1.0, sem a necessidade de ferramentas externas de uso manual. No entanto, é importante notar que caso esta arquitetura faça uso do OpenFlow 1.3, devido às funcionalidades desta nova versão do protocolo, será possível remarcar os pacotes que ultrapassam uma determinada banda sem a necessidade de uma ferramenta externa, assim permitindo a implementação de políticas de QoS mais sofisticadas de maneira automática e nativa.

5. Avaliação do Ferramental Selecionado

Tendo em vista que o *framework* QoSFlow é o principal mecanismo da nova arquitetura, foram realizados dois experimentos para atestar suas funcionalidades. O primeiro expe-

rimento verifica o tempo de resposta das operações de adição, atualização e remoção de configuração dos escalonadores HTB, PFIFO, BFIFO, SFQ e RED no *switch* QoSFlow. O segundo experimento demonstra o impacto do tempo de resposta da API QoSFlow quando o policiamento de banda é constantemente alterado, além de verificar o efeito do policiamento. Para tal, foi utilizada uma topologia contendo dois *hosts* (H1 e H2) ligados a um *switch* QoSFlow baseado no hardware TP-link WR1043-ND. Os enlaces entre os *hosts* e o *switch* utilizaram interfaces físicas de 1Gbps. Em cada *host*, duas máquinas virtuais foram criadas e foi gerado tráfego, via aplicativo *iperf*, entre pares de VMs nos distintos *hosts*. O *switch* QoSFlow é controlado pelo NOX QoSFlow que, através de uma aplicação que faz uso de sua API, utiliza o mecanismo de HTB (*Hierarchical Token-Bucket*) para instanciar duas filas na porta de saída do *switch* para H2. O HTB é apropriado para cenários de DCN.

Pelo HTB transcorreram dois fluxos concomitantemente de H1 para H2: um fluxo usando TCP, que por padrão busca consumir a banda disponível no enlace, e um fluxo usando UDP a 10Mbps. Foi aplicado um policiamento de 10 Mbps durante 30 segundos para o fluxo UDP e, para o TCP, de 6 Mbps por 40 segundos. Após isto, o fluxo UDP é policiado em 4 Mbps enquanto que o policiamento do TCP foi alterado para 10 Mbps, ambos duram 30 segundos. Em seguida foi aplicado um policiamento de 7 Mbps para ambos os fluxos, porém com durações diferentes: no UDP, 40 segundos, e no TCP de 50 segundos. Por fim, o fluxo UDP foi policiado em 4 Mbps até o fim do experimento.

Pode-se observar pelo gráfico apresentado na Figura 4(a) que os tempos de resposta da API QoSFlow para a efetivação de diferentes políticas de escalonamento são reduzidos. Por conseguinte, nota-se que o *switch* QoSFlow possui condições para tornar possíveis as configurações dinâmicas de QoS. A discrepância entre os tempos do HTB e dos demais escalonadores é devido à complexidade deste mecanismo (HTB), que exige maior processamento. No entanto, o mesmo apresentou um desempenho satisfatório, podendo ser aplicado na rede para controle da largura de banda em um cenário com diferentes padrões de tráfego. O bom tempo de resposta da API é refletido pela Figura 4(b). Apesar deste segundo experimento ter utilizado o escalonador HTB, mecanismo com maior tempo de resposta, este não impactou os policiamentos propostos, mostrando a viabilidade do QoSFlow para um cenário de DCN.

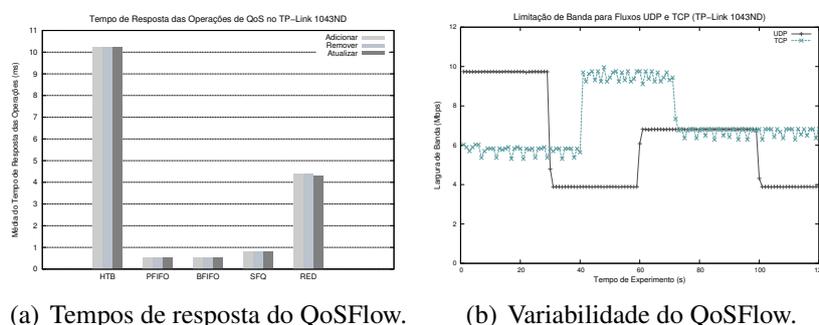


Figura 4. Resultado referente as funcionalidades do QoSFlow.

6. Conclusão e Trabalhos Futuros

Neste trabalho foram observadas duas tecnologias: o OpenFlow, que segrega o plano de controle do plano de dados e permite que este plano se torne uma camada programável; e o

DCN, que é uma arquitetura de redes que permite o estabelecimento de circuitos virtuais sobre múltiplos domínios com garantia de recursos. Apesar destas duas tecnologias já terem sido integradas, a abordagem atual não atende plenamente todos os requisitos de uma DCN.

Neste trabalho foi proposta uma arquitetura alternativa baseada no OpenFlow 1.0 que resolve os problemas encontrados. Caso esta arquitetura faça uso do OpenFlow versão 1.3, suas funcionalidades poderão ser potencializadas, já que esta nova versão do protocolo flexibiliza as formas de aplicação de QoS.

Como trabalho futuro é proposta uma avaliação da arquitetura apresentada na Seção 4, bem como o estudo de viabilidade para se utilizar os escalonadores SFQ e RED. Também será realizada a compatibilização do QoSFlow com o OpenFlow versão 1.3.

Referências

- Boote, J., Brown, A., Lehman, T., and Yang, X. (2011). Dynamic circuit network services and openflow integration. Technical report, Internet2. <http://archstone.east.isi.edu/twiki/pub/ARCHSTONE/OpenFlow/sc-2011-srs-internet2-v2.pdf>.
- Consortium, O. (2009). Openflow switch specification version 1.0.0. Technical report, OpenFlow Consortium. <http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>.
- Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. (2008). Nox: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38(3):105–110.
- Guok, C., Robertson, D., Chaniotakis, E., Thompson, M., Johnston, W., and Tierney, B. (2008). A user driven dynamic circuit network implementation. In *GLOBECOM Workshops, 2008 IEEE*, pages 1–5.
- Guok, C., Robertson, D., Thompson, M., Lee, J., Tierney, B., and Johnston, W. (2006). Intra and interdomain circuit provisioning using the oscars reservation system. In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pages 1–8.
- Internet2 (2012). Dynamic circuit network. <http://www.internet2.edu/network/dc/>. Acessado em 18 Setembro de 2012.
- Ishimori, A., Salvatti, J., Farias, F., Gaspar, L., Granville, L., Cerqueira, E., and Abelém, A. (2012). Qosflow: Gerenciamento automático da qualidade de serviço em infraestruturas de experimentação baseadas em framework openflow. In *III Workshop de Pesquisa Experimental da Internet do Futuro - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2012)*, pages 24–29.
- 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 Comput. Commun. Rev.*, 38(2):69–74.
- Pettit, J., Gross, J., Pfaff, B., Casado, M., and Crosby, S. (2010). Virtual switching in an era of advanced edges. In *2nd Workshop on Data Center - Converged and Virtual Ethernet Switching - 22nd International Teletraffic Congress (ITC 22)*.