

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO – UFES  
CENTRO TECNOLÓGICO – CT  
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO  
DISCIPLINA DE PROJETO ORIENTADO DE GRADUAÇÃO I

## **ANTEPROJETO**

### **Implementação de funções de rede virtualizadas usando Click**

---

Aluno: Felipe Belsholff Pina

---

Orientador: Prof. Dr. Magnos Martinello

---

Coorientadora: Profa. Msc. Cristina Klippel  
Dominicini

Vitória, Espírito Santo, Brasil  
Dezembro/2016

## 1. Introdução

O modelo de Virtualização de Funções de Redes (NFV – Network Function Virtualization) foi proposto em 2012 por grandes empresas de telecomunicações em conjunto com o Instituto Europeu de Padronização de Telecomunicações (ETSI – *European Telecommunications Standards Institute*), tendo a redução de custos de implantação e operação de telecomunicações como principal objetivo. A proposição se baseava em desacoplar os *softwares* dos seus *hardwares* hospedeiros e de propósito específico – o conjunto é normalmente chamado de *middlebox* [3] – para que seus ciclos de vida cada vez mais distintos fossem melhor gerenciados [1]. Este gerenciamento envolvia, por parte das operadoras de telecomunicações, o desenvolvimento de serviços de rede customizados para esses *middleboxes* [1], visando atender precisamente os seus interesses econômicos diante dos desafios técnicos que a atualidade tem proporcionado.

As pesquisas na área vêm apresentando resultados promissores, mas também muitos desafios a serem enfrentados pelas empresas e pelos pesquisadores [1]. Os principais envolvem NFV com Redes definidas por Software (SDN – *Software Defined Networks*), Computação em Nuvem (*Cloud Computing*), Virtualização e Internet das Coisas (IoT – *Internet of Things*), além da expansão do uso da internet por meio da inclusão digital [4]. Versando sobre a tecnologia em si, os desafios passam principalmente por: escalabilidade, gerenciamento, eficiência energética, desempenho, eficiência na alocação de recursos, segurança, privacidade, confiabilidade, dentre outros [1].

## 2. Motivação e Justificativa

Em busca de alta performance, *middleboxes* foram inicialmente implementados em *hardware*, apresentando alto custo e provendo flexibilidade limitada [5]. Diante de um cenário de mercado de telecomunicações atual, com requisitos mais rigorosos, o modelo de soluções de redes proprietário e intimamente embarcado sobre o *hardware* está sofrendo mudanças para absorver as novas demandas.

No que diz respeito aos equipamentos, com o advento da virtualização, diversos destes – *switches*, roteadores, balanceadores de carga, *proxies*, etc – estão sendo substituídos pelos equivalentes virtuais, muitas vezes implementados diretamente nos serviços de fornecedores

das camadas virtuais, e em outros casos como máquinas virtuais. Essa tendência é chamada Função de Rede Virtual (VNF – *Virtual Network Function*) [1]. Já os equipamentos que ainda saem das fábricas estão cada vez mais preparados para se tornarem programáveis por quem desejar, como, por exemplo: os roteadores residenciais que são capazes de executar *firmwares* de terceiros como *DD-WRT* [6], *OpenWRT* [7], *Tomato* [8], dentre outros, além dos proprietários que geralmente os embarcam de fábrica.

No que toca a flexibilidade, pode-se novamente retornar ao exemplo anterior, mencionando que a separação de equipamentos e sistemas de rede favoreceu a criação de soluções livres, colaborativas, mais robustas, mais fáceis de manter, mais confiáveis, tanto com a atuação de empresas quanto a atuação de comunidades de desenvolvedores, contribuindo assim para a qualidade do serviço prestado pela empresa e pelo serviço recebido pelo usuário [9]. A lista de vantagens se repete olhando para a virtualização de funções de redes, adicionando o desempenho, a eficiência e escalabilidade das diversas redes virtuais que trafegam pela estrutura física.

A partir dos investimentos pesados, um novo *boom* na área de pesquisa em softwares de rede surgiu. Uma das propostas que ganharam nova visibilidade, por meio de um sistema operacional personalizado para sua execução [3], foi o *Click* [2, 5]: um *framework* apresentado em 1999, voltado para o desenvolvimento de *middleboxes* em *software* de uso geral, baseando-se na conexão entre uma infinidade de elementos menores e bem simples. Com o *Click*, a construção *desses middleboxes* se torna modular e em alto nível inicialmente, havendo a possibilidade, caso necessário, de se inserir, estender e fabricar elementos ao manusear seus códigos. Como o projeto se baseia em plataformas livres, a integração com os mais diversos projetos é extremamente facilitada.

Neste contexto, o tema deste projeto, que contempla a análise de desafios e a proposta de soluções para implementação de funções de rede virtualizadas usando a plataforma *Click* mostra-se muito relevante e atual.

Os estudos sobre o tema proposto são capazes de fortalecer e trazer novos conhecimentos relacionados à graduação em Ciência da Computação. A rápida evolução da computação, em um processo de ganho de escala com base em virtualização em geral, tem possibilitado o reestudo e desenvolvimento de diversas subáreas, como no caso do tema. É possível citar as

seguintes áreas de conhecimento relacionadas ao tema de estudo: redes de computadores, sistemas operacionais, programação procedural, estruturas de dados, otimização e análise de desempenho em algoritmos e sistemas, metodologia de pesquisa. Somam-se ao tripé de ensino os estágios supervisionados e não-supervisionados, os projetos de pesquisa e extensão, e projetos independentes, trazendo ainda mais conhecimento a ser testado, renovando-o, e contribuindo para o crescimento do estudante e da sociedade em geral ao realizar as aplicações devidas.

### **3. Objetivos**

O objetivo geral desse trabalho passa por analisar desafios propostos para o provisionamento de *middleboxes* usando a plataforma *Click*, além de propor soluções para o problema de virtualização de funções de rede, saindo da tradicional plataforma de *hardware* especializado.

Tratando do objetivo específico, deseja-se avaliar a viabilidade e eficiência dessa plataforma e de seus módulos para a implementação de funções de redes virtualizadas.

### **4. Metodologia**

Primeiramente, será feito um estudo exploratório dos trabalhos relacionados com o modelo de virtualização de funções de rede, além do provisionamento de *middleboxes*. Tal estudo se dará através da leitura de artigos científicos e outras literaturas que exponham o estado da arte relacionado a área, e a partir disso, entender os desafios e as funcionalidades necessárias para viabilizar a implementação de NFV. Depois, serão feitos estudos sobre as diferentes tecnologias habilitadoras do modelo e, em especial, a plataforma *Click*, explorando suas vantagens e desvantagens.

Após essa primeira fase, será realizado o levantamento das funcionalidades exigidas para virtualizar funções de rede e alguns tipos de funções de rede que podem ser implementadas pela plataforma *Click*. Após essa análise, será projetada uma proposta para NFV usando a plataforma *Click*.

Então, essa proposta será implementada por meio de um protótipo para demonstração dos conceitos. Ao fim, os resultados dos testes com o protótipo serão analisados para se tirar

resultados mais conclusivos com relação ao atendimento das funcionalidades necessárias para habilitar NFV. Além disso, o protótipo será avaliado em relação ao desempenho da solução proposta.

## **5. Resultados Esperados**

A partir das conclusões obtidas com esses estudos, serão propostas soluções para problemas conhecidos no problema da implementação de funções de rede virtuais utilizando a plataforma Click. Será analisado, sobretudo, a viabilidade desse modelo, tendo em vista os recursos utilizados e os requisitos mínimos que deverão ser atendidos para viabilizar a utilização de NFV em substituição a soluções que utilizam hardware de propósito específico.

## **6. Cronograma**

Para desenvolvimento desse projeto, pretende-se cumprir o seguinte cronograma de execução de atividades, apresentado na Tabela 1.

1. Realizar revisão bibliográfica de trabalhos relacionados a NFV, com foco na construção de VNFs e associando ao *framework Click*.
2. Levantar as funcionalidades exigidas para NFV; especificar e projetar modelos para a proposta.
3. Projetar uma solução para o problema funções de rede virtualizadas usando *Click*.
4. Implementar protótipo da solução projetada; prover testes para a implementação.
5. Realizar avaliações de desempenho para a implementação.
6. Produzir monografia.

Atividade	Fevereiro	Março	Abril	Maio	Junho
<b>1</b>					
<b>2</b>					
<b>3</b>					
<b>4</b>					
<b>5</b>					
<b>6</b>					

*Tabela 1 - Cronograma mensal das atividades*

## 7. Recursos Necessários

Esse trabalho consistirá na implantação de um protótipo no *datacenter* do Núcleo de Estudos em Redes Definidas por Software (NERDS) da UFES, que tem como objetivo simular um ambiente de rede, habilitando soluções NFV utilizando *Click*.

Para a instalação e configuração da plataforma Click, será preciso inicialmente pelo menos uma máquina com recursos razoáveis de memória, processamento e armazenamento que disponha de um *hypervisor* de código aberto; podendo ser necessário até 6 máquinas físicas ou virtuais para simular uma rede de pequeno porte, de acordo com o avanço do projeto.

O presente trabalho leva em conta módulos já existentes no *Click* que permitem habilitar soluções NFV por meio de VNFs. A maioria destes módulos se encontram em constante desenvolvimento, sujeitos a mudanças significativas ao longo do tempo. É também papel desse trabalho acompanhar todas estas mudanças e adequar o trabalho, tendo em vista os recursos existentes atualmente no laboratório NERDS.

## 8. Fundamentação Teórica e Tecnologias Necessárias

Esta seção apresenta uma breve fundamentação teórica para entendimento dos temas abordados por este trabalho.

### a. Middleboxes

Os *middleboxes* são dispositivos de propósito único que implantam serviços ao longo da rede física, promovendo segurança, otimização e maior gerenciamento. São exemplos os *switches*, roteadores, *firewalls*, balanceadores de carga, priorizadores de pacotes, limitadores de banda, *proxies*, filtros de conteúdo, etc [3]. Durante muito tempo foram exclusivamente dispositivos de hardware com software embarcado, com alto custo e ciclo de vida reduzido, muito em função das empresas produtoras desses equipamentos realizarem obsolescência programada no software do dispositivo, não atualizando-o quando fosse conveniente. Juntando isso ao rápido desenvolvimento das redes de computadores, que também obrigava a atualização do parque de equipamentos, causando alta dependência de seus consumidores.

No entanto, esse modelo vem sendo superado dia após dia com o uso das tecnologias de NFV e da evolução dos periféricos computacionais de uso geral, possibilitando dessa forma a criação de *middleboxes* virtuais ou de *software* sem perda de performance [10], barateando custos a longo prazo, seja pelo aumento do ciclo de vida ou pelo ganho de escalabilidade advindo da virtualização de soluções de uma forma geral, desde servidores físicos a redes de computadores e telecomunicações.

O método mais fácil de construção dessas soluções de *software* passa pela construção de máquinas virtuais com alta vazão, leves, simplificadas quanto a suas configurações, e que permita alta flexibilidade para melhoramentos técnicos e administração [10], podendo até mesmo automatizar a gerência de rede sem precisar mover cabos na rede física [1].

## **b. Virtualização de Funções de Rede – NFV**

Para além da pequena explicação feita na seção de introdução, vamos focar na arquitetura base do modelo NFV, dividida em diversos elementos bem definidos e responsáveis tanto pela parte técnica quanto pela gerencial que o modelo prevê:

- **Função de rede virtualizada (VNF – *Virtual Network Function*):** trata-se da implementação em *software* de uma função de rede – comumente um *middlebox* – desenvolvido com base no modelo NFV. ClickOS [3] é uma das soluções já existentes.

- **Gerente VNF:** responsável pelas operações relacionadas ao ciclo de vida de uma VNF, como por exemplo as quatro básicas (criação, alteração, consulta e exclusão).
- **Camada de Virtualização:** é responsável por abstrair os recursos de *hardware* e ligar as VNFs à infraestrutura virtualizada. Normalmente, esta funcionalidade é fornecida sob a forma de *hypervisors* que permitem a execução de máquinas virtuais em *hardware* de propósito geral. *Xen* [11] e *KVM* [12] são exemplos [3].
- **Gerente da Infraestrutura Virtualizada:** é responsável por virtualizar e gerenciar os recursos configuráveis de computação, rede e armazenamento e controlar sua interação com as VNFs. Ele aloca máquinas virtuais em *hypervisors*, armazenam seus discos virtuais em recursos de armazenamento disponíveis na infraestrutura e gerencia toda sua forma de conectividade na rede, seja através de roteamento interno ou externo. Um exemplo de gerente neste caso é o *Openstack* [13].
- **Sistemas de Suporte Operacional (OSS – *Operational Support Systems*):** são responsáveis por suportar os processos internos da operadora como inventários de rede, provisionamento de serviços, configuração de elementos de rede e gerenciamento de falhas.

### c. Virtualização de Funções de Rede – NFV

*Click* trabalha com elementos simples, assemelhando-se com caixas, que devem ser conectadas para formar um elemento maior e relevante como um firewall ou balanceador de carga, por exemplo. A passagem de pacotes pode ocorrer por vários tipos de contadores, classificadores, diversos manipuladores de pacotes, tratadores de erros, analisadores de protocolos, etc. Estes foram pré-implementados e fazem parte dos elementos padrão do *framework*, cuja entrada é dada por arquivos de texto com sintaxe apropriada. É importante ressaltar que por ser de código aberto, tanto as melhorias nos atuais elementos, estendendo-os, quanto a criação de novos são livres.

O controle de fluxo é feito com base na arquitetura produtor-consumidor, por meio de operações de envio (*push*) e requisição (*pull*), onde cada elemento possui suas portas de entrada e saída e assumir seu comportamento no fluxo. No entanto, elementos de diferentes



comportamentos não podem se acoplar diretamente, dependendo assim de uma fila para que isso ocorra. A intenção é evitar gargalos de fluxo e processamento desnecessário com estados de espera ocupados. Seguem dois diagramas ilustrativos:

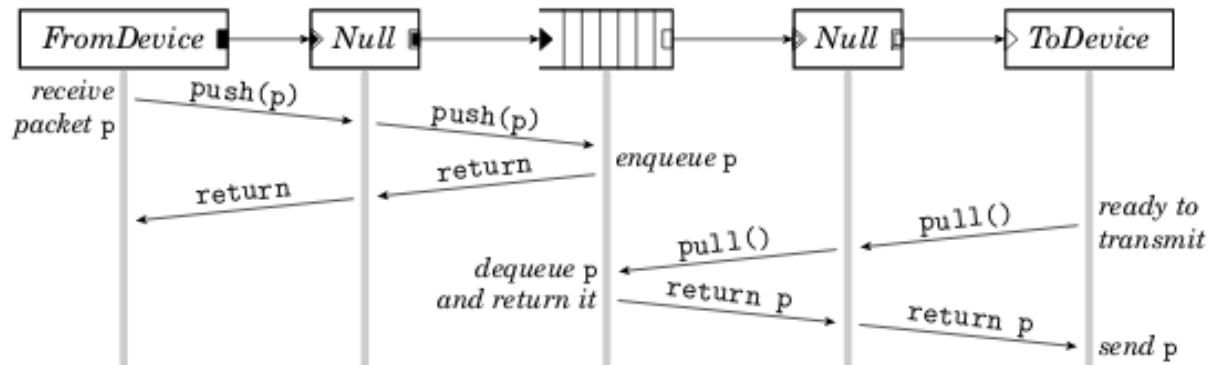


Figura 1 – Evitando a espera ocupada. Perceba como a fila evita que o *Null* à esquerda fique aguardando os elementos a direita ficarem prontos. Com isso ganha-se tempo.

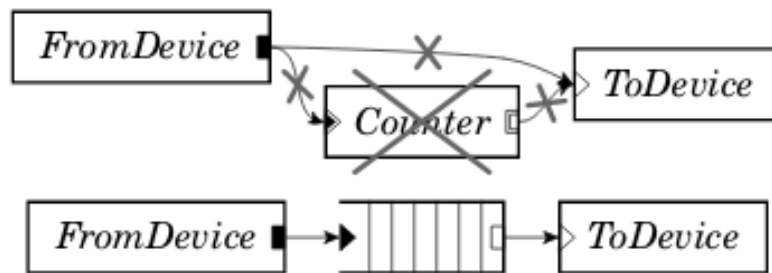


Figura 2 – Diversas formas de conectar elementos. Perceba que elementos não podem fazer parte de operações *push* e *pull* ao mesmo tempo e que ligações diretas entre elementos *push* e *pull* também não são permitidas.

Importante dizer que o *Click* é capaz de operar em nível de usuário e de *kernel*, sendo o primeiro preferencial para a programação e o segundo para utilização, devido aos privilégios adquiridos em *kernel-space*. Nesse modo de operação, em testes, o *Click* foi capaz de ser mais rápido que *middleboxes* de entrada de linha, mesmo a performance não sendo o principal foco do trabalho na época em que foi concebido [2].

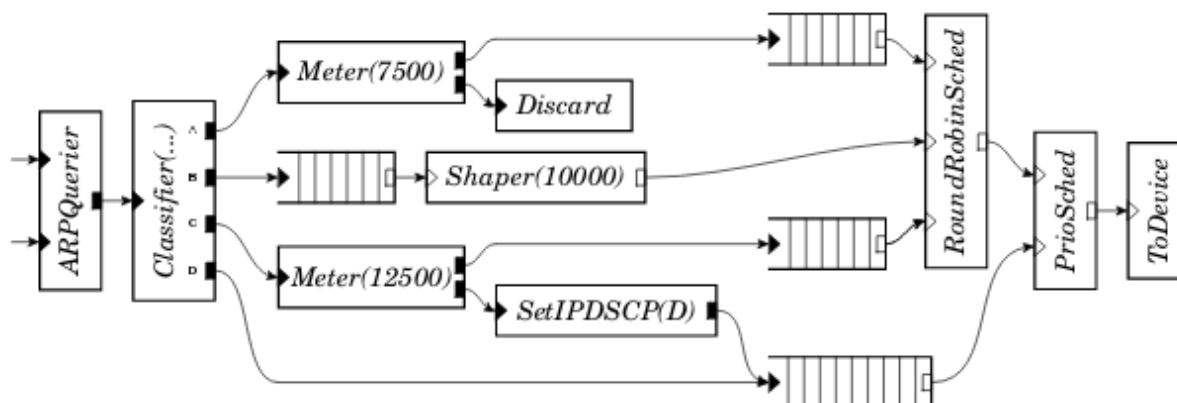


Figura 3 – Condicionador de tráfego implementado em Click.

#### d. ClickOS

Apresentado em 2014, a plataforma *Click* foi a escolhida de um projeto de pesquisa da Universidade politécnica de Bucareste junto com a Nippon Electric Company europeia – NEC Europe – para ser a tecnologia principal de um projeto de virtualização de funções de rede voltado para a criação de funções de rede virtuais que possam substituir *middleboxes* de hardware. Surgiu assim o *ClickOS*.

Ao disponibilizar um sistema operacional baseado no *MiniOS* – que faz parte do Xen – extremamente enxuto, com menos de 12MB de carga, o ClickOS trata dois dos principais problemas relacionados aos *middleboxes* de *software*: escalabilidade e economia de recursos. Outro problema que a equipe responsável foi capaz de lidar é com necessidade de altas taxas de transferência necessárias. Várias otimizações foram realizadas a partir do mapeamento de entrada e saída ser feito pelas máquinas virtuais diretamente em memória física, evitando métodos tradicionais implementados nos *hypervisors* como o tráfego através de *drivers* intermediários (*netback*) e *drivers* virtuais (*netfront*), e evitando também o uso de *switches* virtuais que não foram otimizados para o uso da memória compartilhada entre máquina física e máquina virtual (*memory grants*), como por exemplo o *OpenvSwitch* [14], substituído pelo *VALE Switch* [15], que posteriormente também foi melhorado pela equipe e ganhou um novo nome: *ClickOS Switch*.

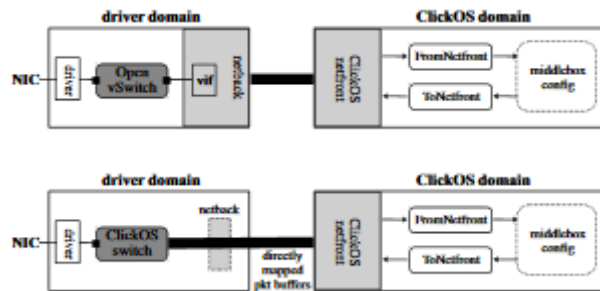


Figura 4 - Acima o design padrão de redes usado pelo hypervisor Xen. Abaixo o design modificado pelos pesquisadores.

Aliás, grande parte dessas otimizações foram reportadas pelos pesquisadores para seus desenvolvedores para que possam incluí-las em suas versões regulares. É o caso do *Xen*, *OpenvSwitch*, *kernel* do *Linux* e o próprio Click, que recebeu novos elementos em meio ao desenvolvimento do *ClickOS*.

## 9. Referências

- [1] MIJUMBI, Rashid et al. Network function virtualization: State-of-the-art and research challenges. **IEEE Communications Surveys & Tutorials**, v. 18, n. 1, p. 236-262, 2015.
- [2] MORRIS, Robert et al. The Click modular router. **ACM SIGOPS Operating Systems Review**, v. 33, n. 5, p. 217-231, 1999.
- [3] MARTINS, João et al. ClickOS and the art of network function virtualization. In: **Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation**. USENIX Association, p. 459-473, 2014.
- [4] THE NEXT billion geeks: Mobile internet in emerging markets. **The Economist**, Dadri, 2 set. 2010. Disponível em: <<http://www.economist.com/node/16944020>>. Acesso em: 1 dez. 2016.
- [5] LAUFER, Rafael et al. CliMB: enabling network function composition with click middleboxes. In: **Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization**. ACM, p. 50-55, 2016.

[6] TAYLOR, Nathan. Supercharge Your Router. *APC (Bauer Media Group)*. v.32, n.7, p. 48-52, 2012. ISSN: 07254415.

[7] KIM, Cheong Ghil; KIM, Kuinam J. Implementation of a cost-effective home lighting control system on embedded Linux with OpenWrt. **Personal and ubiquitous computing**, v. 18, n. 3, p. 535-542, 2014.

[8] NICKEL, Eduardo Maltauro; BESSA, William Kuhl Svoboda Marques. **Sistema Embarcado Com Acesso Sem-Fio**. 2010. 46 f. Tese (Graduação) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal do Paraná, Curitiba, 2010. Disponível em: <<http://www.cricte2004.eletrica.ufpr.br/ufpr2/tccs/158.pdf>>. Acesso em: 01 dez. 2016.

[9] STALLMAN, Richard Matthew; LESSIG, Lawrence; GAY, Joshua. **Free Software, Free Society: Selected Essays of Richard M. Stallman**. Boston: Gnu Press, 2002. 224 p. Disponível em: <<https://www.gnu.org/philosophy/fsfs/rms-essays.pdf>>. Acesso em: 02 dez. 2016.

[10] LEE, Jeongkeun et al. No more middlebox: integrate processing into network. In: **ACM SIGCOMM Computer Communication Review**. ACM, 2010. p. 459-460.

[11] FRASER, Keir; HAND, Steven; PRATT, Ian. **Xen Project**. 2003. Disponível em: <<https://www.xenproject.org>>. Acesso em: 03 dez. 2016.

[12] QUMRANET. **Kernel-based Virtual Machine**. 2007. Disponível em: <<http://www.linux-kvm.org/>>. Acesso em: 03 dez. 2016.

[13] NASA; RACKSPACE. **Openstack**. 2010. Disponível em: <<https://www.openstack.org>>. Acesso em: 03 dez. 2016.

[14] LINUX FOUNDATION. **OpenvSwitch**. 2009. Disponível em: <<http://openvswitch.org>>. Acesso em: 03 dez. 2016.

[15] RIZZO, Luigi; LETTIERI, Giuseppe. Vale, a switched ethernet for virtual machines. In: **Proceedings of the 8th international conference on Emerging networking experiments and technologies**. ACM, 2012. p. 61-72.