Uma Arquitetura para Isolamento de Redes Virtuais Usando a Abordagem Híbrida Xen e OpenFlow*

Diogo Menezes Ferrazani Mattos , Lyno Henrique Gonçalvez Ferraz e Otto Carlos Muniz Bandeira Duarte

[1]Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

Resumo. *A virtualização de redes é a técnica que provê o compartilhamento de recursos de rede, permitindo a coexistência de múltiplas redes lógicas sobre a mesma infraestrutura física. Contudo, o isolamento entre redes virtuais é um requisito fundamental e também um dos principais desafios da virtualização de redes. Um desafio importante é garantir o isolamento entre duas ou mais redes virtuais que possuam espaços de endereçamento IP com interseção de endereços. A ferramenta de virtualização deve ser capaz de classificar quais pacotes pertencem a cada rede. Este artigo propõe uma arquitetura para o isolamento de redes virtuais em uma ferramenta de virtualização híbrida que combina Xen e OpenFlow. A arquitetura de rede proposta associa uma aplicação de controle de redes OpenFlow com um esquema de marcação de etiquetas para cada rede virtual. Um protótipo foi desenvolvido e sua avaliação revela que a arquitetura proposta isola completamente o tráfego das redes virtuais mesmo que possuam o mesmo endereço IP. Além disso, a solução proposta impede que uma máquina virtual que pertença a uma dada rede virtual escute os pacotes de outras redes virtuais, mesmo que sejam enviados em difusão.*

Abstract. *Network virtualization is the technique that provides resource sharing and enables multiple logical networks to coexist over the same physical infrastructure. Isolating virtual networks, however, is one of the challenges of network virtualization. The virtualization tool must be able to classify packets that belong to each network even if the virtual network share the same IP address space with others. In this paper we propose a system for isolating virtual networks. Our proposal associates a control application of OpenFlow networks with a labeling scheme for each virtual network. Labeled packets are only delivered to virtual networks that share the same label. We developed a prototype and its evaluation shows that the proposed system prevents a virtual machine, which belongs to a given virtual network, to listen packets of other virtual networks, even if they are broadcast packets. The results also show that two virtual networks can share the same address space, and even then, traffic from each virtual network is isolated.*

1. Introdução

A virtualização de rede é uma tecnologia essencial para prover um ambiente de experimentação para propostas para a Internet do Futuro, assim como também é uma efetiva proposta pluralista para a Internet, na qual diversas redes executam sobre um mesmo substrato físico [Mattos et al. 2011, Feamster et al. 2007]. A virtualização separa a função desempenhada por um elemento de rede de sua realização física. Assim, permite experimentar diversos protocolos e serviços inovadores no núcleo da rede, pois

*Este trabalho foi realizado com recursos da FINEP, FUNTTEL, CNPq, CAPES, FAPERJ e UOL.

cada rede virtual, chamada de fatia de rede, é isolada das demais. Entretanto, as iniciativas atuais de virtualização de redes ainda enfrentam desafios para garantir desempenho, requisitos de qualidade de serviço exigidos pelas aplicações que executam nas redes virtuais e isolamento seguro do tráfego entre as redes virtuais [Mattos and Duarte 2012, Barabash et al. 2011].

O isolamento de redes virtuais é divido em duas vertentes importantes: o isolamento de recursos [Fernandes and Duarte 2010, Mattos and Duarte 2012] e o isolamento da comunicação entre nós de uma rede virtual [Barabash et al. 2011]. O isolamento de recursos garante que as redes virtuais operem de forma independente, assim, o uso de recursos de um roteador virtual não interfere no desempenho dos demais. O isolamento de recursos é importante porque evita a negação de serviços, já que uma rede virtual não consegue exaurir os recursos de outra rede virtual. O isolamento da comunicação das redes virtuais, por sua vez, é um outro requisito essencial, pois garante que a comunicação de uma rede virtual só alcance os nós que de fato pertencerem a essa rede virtual. O isolamento da comunicação de redes virtuais é essencial em meios de comunicação em difusão, como o Ethernet [Perlman et al. 2011], pois um pacote em difusão, ou com múltiplas destinações, deve ser acessível somente pelos nós que pertençam a rede virtual correta, enquanto os demais nós não devem receber tais pacotes [Huang 2005, Barabash et al. 2011]. O isolamento da comunicação das redes virtuais é importante também por segurança, uma vez que impede uma rede virtual de bisbilhotar (*eavesdropping*) os pacortes de outras redes virtuais. Outro ponto importante do isolamento da comunicação é a restrição do espaço de endereçamento e do alcance dos endereços de uma rede virtual. A ferramenta de virtualização de redes deve prover a facilidade de uma rede virtual usar o mesmo espaço de endereçamento de outra rede virtual, por exemplo o mesmo endereço IP, que outra [Barabash et al. 2011]. Garantir o isolamento da comunicação e isolamento de recursos, mantendo o desempenho das redes virtuais é um desafio da virtualização de redes [Egi et al. 2007, Fernandes et al. 2010].

Este artigo propõe uma arquitetura de virtualização de redes para isolar recursos e isolar também a comunicação de redes virtuais. A proposta se apoia no paradigma da separação de plano, no qual o encaminhamento e o controle da rede são desacoplados [Pisa et al. 2010, Wang et al. 2008]. A ideia central da proposta é que os pacotes de uma rede virtual permaneçam restritos aos nós dessa rede e não interfiram no funcionamento das demais, assim o tráfego de uma rede virtual não interfere no desempenho de outras redes. O objetivo da arquitetura proposta é a virtualização de redes em que os nós tenham a impressão de que a infraestrutura física é dedicada somente para os pacotes da sua rede virtual. Para tanto, a proposta baseia-se no sistema híbrido de virtualização de redes XenFlow [Mattos et al. 2011, Mattos and Duarte 2012], que combina a ferramenta de virtualização de computadores Xen com a interface de programação de aplicação (API) para redes OpenFlow. Assim, as principais contribuições da proposta são: i) a garantia de isolamento da comunicação entre redes virtuais; ii) o mapeamento de funções de isolamento de redes virtuais para primitivas do plano de dados; e iii) a combinação da solução de isolamento da comunicação com a proposta de isolamento de recursos [Mattos and Duarte 2012], garantindo a virtualização de redes em um ambiente em que haja garantia de recursos para redes virtuais e, ao mesmo tempo, garantia de que os pacotes de uma dada rede sejam somente entregues aos nós destinados a receber.

As principais propostas para prover o isolamento de redes virtuais visam somente o isolamento de recursos [Mattos and Duarte 2012, Fernandes and Duarte 2010, Fernandes and Duarte 2011] ou se baseiam no encapsulamento dos pacotes de redes virtuais para prover o isolamento da comunicação das redes virtuais [Barabash et al. 2011, Perlman et al. 2011]. A arquitetura proposta, no entanto, usa o paradigma da separação de planos para garantir o máximo desempenho do encaminhamento de pacotes e realiza

tanto o isolamento de recursos quanto o isolamento da comunicação. O isolamento da comunicação é alcançado através da marcação dos pacotes de cada rede virtual com uma etiqueta que indica a qual rede o pacote pertence. O padrão 802.1Q, que define o funcionamento de VLANs (*Virtual Local Area Network*), é usado para marcar os pacotes de acordo com as regras definidas pela aplicação OpenFlow que traduz as regras do plano de controle para o plano de dados. O isolamento de recursos de rede consiste no mapeamento das redes virtuais em filas de encaminhamento definidas no plano de dados. A proposta deste artigo estende o sistema XenFlow [Mattos et al. 2011, Mattos and Duarte 2012], que realiza a separação de planos usando o Xen para executar o plano de controle e OpenFlow para o plano de dados. A proposta deste artigo, então, conjuga a separação de planos do XenFlow, adicionando a marcação de pacotes por VLAN como esquema de isolamento de redes virtuais. Um protótipo da proposta foi implementado e a sua avaliação revelou que a abordagem proposta executa tanto o isolamento da comunicação quanto o isolamento do uso de recursos entre redes virtuais.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta as principais propostas de virtualização de redes no Xen e suas limitações. A arquitetura de redes virtuais isoladas proposta é detalhada na Seção 4. A avaliação do sistema e os seus resultados são apresentados na Seção 5. A Seção 6 conclui o artigo.

2. Trabalhos Relacionados

O isolamento da comunicação das redes virtuais associado à separação de planos e à garantia de recursos para cada rede virtual é um desafio para as principais propostas de virtualização de redes [Mattos and Duarte 2012, Pisa et al. 2010, Fernandes and Duarte 2011, Fernandes et al. 2010]. Uma das principais formas de se alcançar a virtualização de redes é usar uma plataforma de virtualização de máquinas e as máquinas virtuais executarem funções elementos de rede. Nesse sentido, o Xen é uma das principais ferramenta usadas para a virtualização de redes [Wang et al. 2008, Egi et al. 2007] em computadores pessoais.

A plataforma de experimentação PlanetLab permite que pesquisadores compartilhem uma infraestrutura física, através da virtualização de rede, para executar experimentos distribuídos [Chun et al. 2003]. A plataforma PlanetLab apresenta uma interface de rede especial que permite aos pesquisadores isolarem o tráfego de suas redes dos demais, a VNET [Huang 2005, Bavier et al. 2006]. A VNET é uma rede sobrecamada que fornece às máquinas virtuais um ambiente isolado na Camada 2 e, portanto, agnóstico ao protocolo da camada de rede. Para tanto, essa proposta se baseia no uso de protocolos de tunelamento para realizar a comunicação isolada entre máquinas virtuais. Outra proposta de isolamento de redes aplicada ao PlanetLab é o VIOLIN [Jiang and Xu 2005]. O VIOLIN também se baseia no conceito de redes sobrecamada, contudo, o isolamento das redes é realizado na camada de aplicação. Para realizar o isolamento, o VIOLIN encapsula os pacotes das redes virtuais e os envia em uma rede sobrecamada implementada por suas aplicações.

Considerando o isolamento de redes em um ambiente de Centro de Dados, outra proposta é a Nuvem Privada Virtual (*Virtual Private Cloud* VPC–) [Wood et al. 2009], que aplica o isolamento de redes na Computação em Nuvem. A proposta VPC aplica uma infraestrutura de VPN (*Virtual Private Network*) entre elementos na nuvem. A ideia central é criar VPCs dinamicamente a partir de repositórios de recursos configuráveis em nuvens e conectá-los aos locais que necessitam de recursos através de VPNs. Já a arquitetura de redes DOVE (*Distributed Overlay Virtual Ethernet*) para Centros de Dados é uma proposta de virtualização de redes que provê isolamento [Barabash et al. 2011]. A arquitetura DOVE permite a criação de redes virtuais com espaços de endereçamentos

isolados e dinâmicas sobre uma infraestrutura física comum. O funcionamento do DOVE baseia-se no encapsulamento de pacotes e na criação de uma rede de sobrecamada para permitir a separação entre rede virtual e a rede física subjacente. O encaminhamento e o roteamento dos pacotes das redes virtuais ocorre de acordo com os pacotes encapsulados [Barabash et al. 2011]. O isolamento de redes virtuais é alcançado nessa proposta pelo uso de um identificador de rede que é adicionado ao cabeçalho extra da rede sobrecamada DOVE. Assim, os pacotes com um dado identificador só é entregue às máquinas virtuais que compartilhem o mesmo identificador de rede virtual.

O isolamento da comunicação de redes virtuais pode ser alçado usando o Open vSwitch [Pfaff et al. 2009], um comutador de software projetado tanto para ser usado em ambientes virtualizados, quanto para transformar um computador comum em um comutador programável. O Open vSwitch é um comutador por *software* que realiza a comutação dos pacotes entre máquinas virtuais e a rede física. O Open vSwitch realiza o isolamento da comunicação de redes virtuais através da marcação de etiquetas de VLAN nos pacotes provindos das máquinas virtuais. Assim, ao ser encaminhado na rede, seja entre máquinas virtuais hospedadas em uma mesma máquina física, ou seja, entre máquinas físicas distintas, os pacotes são marcados com a etiqueta da VLAN a que pertencem. Dessa forma, um pacote só é entregue à máquina virtual se sua interface pertencer à VLAN que está marcada no pacote.

As propostas XNetMon [Fernandes and Duarte 2010] e XNetMan [Fernandes and Duarte 2011] aplicam o Xen como ferramenta de virtualização de redes e executam algoritmos para o controle do uso de recursos por cada rede virtual. Contudo, tais propostas não garantem o completo isolamento de tráfego entre redes virtuais, pois a separação do tráfego apenas se baseia na classificação de pacotes de acordo com o endereço da rede virtual a que se destinam. Assim, isolamento se restringe a redes virtuais com espaços de endereçamento disjuntos. O sistema de virtualização de redes XenFlow [Mattos et al. 2011] combina virtualização de máquinas virtuais Xen com a interface de programação de aplicação (API) OpenFlow. O XenFlow aplica o conceito de separação de planos [Pisa et al. 2010], mas não realiza o isolamento de recursos ou da comunicação de redes virtuais. Já a proposta QFlow [Mattos and Duarte 2012] estende o XenFlow e aplica o isolamento de recursos através do mapeamento de requisitos de Qualidade de Serviço em recursos disponíveis no plano de encaminhamento OpenFlow. No entanto, nem o XenFlow nem o QFlow isolam a comunicação das redes virtuais.

A ideia básica das propostas acima apresentadas para prover o isolamento da comunicação das redes virtuais é realizar o encapsulamento dos pacotes das redes virtuais e, assim, criar uma rede sobrecamada que interconecte os nós de uma dada rede virtual. No entanto, as propostas apresentadas falham ao se considerar o paradigma da separação de planos, em que o encaminhamento é realizado pelo plano de dados, enquanto o controle da rede está restrito ao plano de controle. Nesse sentido, arquitetura de rede proposta neste artigo isola redes virtuais mesmo no cenário de separação de planos, sem a necessidade de criar uma rede sobrecamada, inserindo uma etiqueta de VLAN nos pacotes de cada rede virtual. A proposta estende o sistema XenFlow [Mattos et al. 2011] e aplica o isolamento de recursos do sistema QFlow [Mattos and Duarte 2012]. A ideia central da proposta deste artigo é realizar o isolamento da comunicação de redes virtuais através de primitivas do plano de dados, como marcar a VLAN que um pacote pertence, para que seja possível realizar o roteamento em redes virtuais, isolando uma rede virtual das demais, sem que os pacotes dessa rede sejam tratados pelo plano de controle.

3. O Encaminhamento de Pacotes em Redes Virtuais

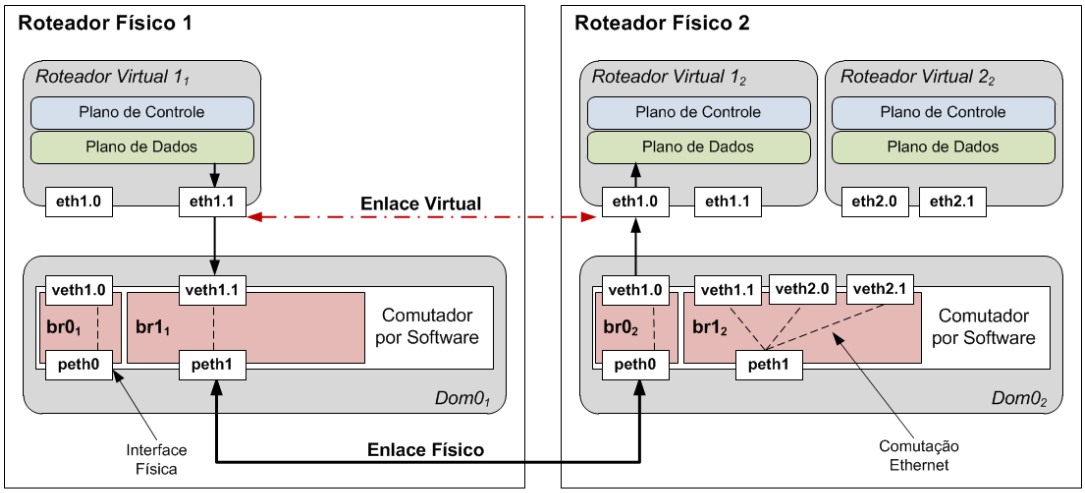
A arquitetura de redes virtuais proposta considera o paradigma de separação de planos, no qual as funções do roteador virtual são divididas em dois planos, o plano de

controle e o plano de dados. O plano de controle, que é executado dentro da máquina virtual criada com a plataforma Xen e é responsável pelas tarefas de controle, como a atualização da tabela de roteamento. Por outro lado, o plano de dados é criado utilizando-se o OpenFlow que é responsável pelo encaminhamento dos pacotes.

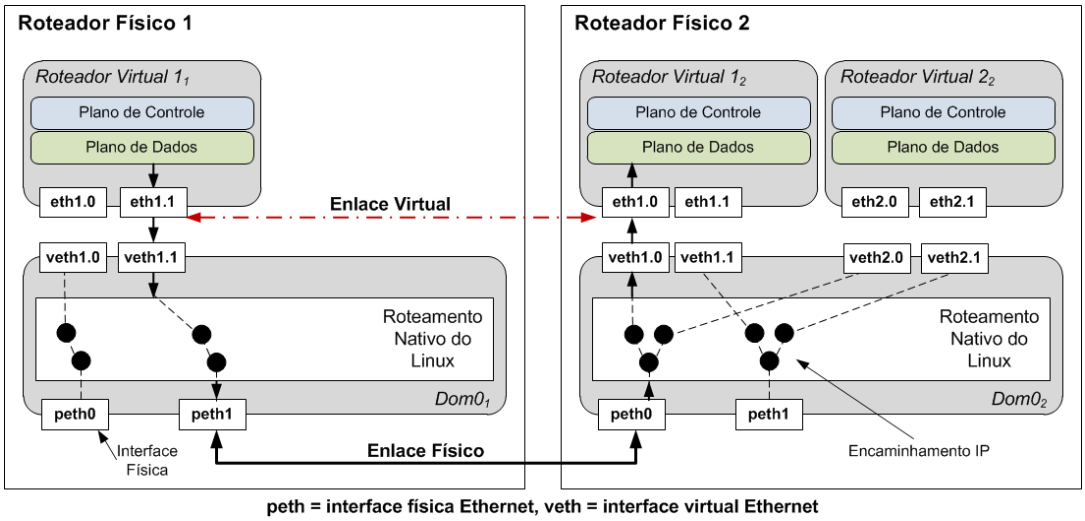
O sistema de virtualização Xen permite, por padrão, três modos principais para realizar o encaminhamento de pacotes entre as interfaces das diferentes máquinas virtuais e as interfaces da máquina física: o modo *bridge*, o modo *router* e o modo NAT (*Network Address Translation*). Esses mecanismos multiplexam (comutam ou roteiam) o tráfego de saída dos pacotes originados pelas diferentes máquinas virtuais e demultiplexam (comutam ou roteiam) o tráfego de chegada de pacotes de uma (ou diversas) interface de entrada E/S com destino às diferentes máquinas virtuais [Egi et al. 2007]. Os dois principais mecanismos discutidos nesse trabalho são o modo *bridge* e o modo *router*.

A Figura 1 compara os modos *router* e *bridge* do Xen. A figura mostra dois roteadores físicos (Roteador Físico 1 e Roteador Físico 2) com duas interfaces físicas Ethernet cada um (`peth0` e `peth1`). Um enlace físico liga a interface física Ethernet `peth1` do Roteador Físico 1 com a interface física Ethernet `peth0` do Roteador Físico 2. O enlace virtual mostrado na figura é estabelecido com a associação do Roteador Virtual 1 do Roteador Físico 1 com o Roteador Virtual 1 do Roteador Físico 2. O modo *bridge* define uma ponte Ethernet, *bridge Ethernet*, que interconecta as interfaces da máquina física às interfaces das máquinas virtuais, como mostrado na Figura 1(a). Nesse caso, as interfaces físicas são representadas por interfaces virtuais, ligadas ponto-a-ponto. A ponte é um comutador por *software* e, portanto, encaminha os pacotes para máquina virtual baseado no encaminhamento da camada de enlace, MAC. Outro modo de encaminhamento de pacotes no Xen é o modo *router*, mostrado na Figura 1(b). O modo *router* encaminha os pacotes para a máquina virtual através do roteamento dos pacotes de acordo com as informações da camada de rede, no caso, baseado no endereço IP. O modo *router* cria um roteador entre a máquina física e as máquinas virtuais. Assim, a máquina física se apresenta como mais um salto na rede IP entre máquina virtual e o restante da rede.

Essas arquiteturas de rede são suficientes e apresentam um bom funcionamento e também um bom desempenho para consolidação de servidores [Egi et al. 2007]. O modo *bridge* permite que as máquinas virtuais se comuniquem como se estivessem em uma mesma LAN (*Local Area Network*). Já o modo *router* permite a agregação de rotas para um determinado conjunto de máquinas virtuais que estão sobre uma mesma máquina física, diminuindo os requisitos de memória sobre os dispositivos de encaminhamento da infraestrutura física [Barabash et al. 2011], já que as rotas divulgadas na rede física endereçam grupos de máquinas virtuais, ao invés da divulgação dos endereços de todas as máquinas virtuais de forma plana e não agregada, como ocorre no modo *bridge*. Contudo, essas arquiteturas de rede não são plenamente suficientes para a virtualização de redes. Os modos *bridge* e *router* falham na arquitetura com separação dos planos de controle e de dados. A separação de planos é essencial para prover desempenho às redes virtuais [Pisa et al. 2010, Mattos et al. 2011]. Na separação e planos, o plano de controle é responsável por calcular as rotas, já o plano de dados é responsável por encaminhar os pacotes nas rotas corretas. A separação de planos relaciona-se com o roteamento de pacotes, ou seja, o encaminhamento de pacotes de acordo com o cálculo de uma rota mais adequada ao endereço de rede do pacote. Assim, quando o modo *bridge* é considerado, a separação de planos falha, pois a essa é uma tarefa que depende das informações da camada de rede. Já no modo *router*, a arquitetura com separação de planos é viável [Pisa et al. 2010], pois são criadas tabelas de roteamento no plano de encaminhamento do Domínio 0 que são cópias das tabelas dos roteadores virtuais. Assim, os pacotes de cada rede virtual são encaminhados de acordo com a tabela de rotas do roteador virtual correspondente através dos mecanismos nativos do *kernel* do Linux. No



(a) Encaminhamento de pacotes através do modo *bridge* do Xen.



(b) Encaminhamento de pacotes através do modo *router* do Xen.

Figura 1. Comparação entre dois modos de encaminhamento de pacotes no Xen, o modo *bridge* e o modo *router*.

entanto, a separação de planos usando o modo *router* apresenta alguns desafios. O principal desafio é o isolamento da comunicação das redes virtuais que se divide em dois problemas principais: identificar a qual rede virtual o pacote pertence e tratar os pacotes com múltiplas destinações, por exemplo, a difusão (*broadcast*) e a destinação múltipla ou difusão seletiva (*multicast*).

A identificação trivial de qual rede um pacote pertence é através de seu endereço IP de destino. No entanto, a identificação trivial falha quando duas redes virtuais distintas usam o mesmo espaço de endereçamento IP. Quando os espaços de endereçamento não são disjuntos, não é possível identificar a qual rede uma pacote pertence e, assim, não há isolamento entre redes virtuais. O segundo desafio de identificação de pacotes em redes virtuais consiste no encaminhamento de pacotes com endereços multidestinatários ou de difusão (*multicast/broadcast*), pois neste caso um único endereço de IP se refere a diversos hospedeiros, ou seja, pacote multidestinatário ou de difusão possui um endereço específico que designa a multidestinação. A multidestinação não é uma lista de endereços

individuais, mas sim um endereço de grupo e isto ocasiona perda na semântica hierárquica de qual rede um IP pertence. Assim, os pacotes são destinados a endereços IP padrões e não é possível diferenciar entre uma rede ou outra apenas pelo seu endereço IP. A solução de usar espaços de endereçamentos isolados para identificar as redes virtuais falha, pois o IP *multicast* não pertence à faixa de IP destinado às redes virtuais e, consequentemente, não pode ser encaminhado pelas tabelas de rota do Domínio 0. O problema de encaminhamento do pacote de *broadcast* é ainda mais complexo, pois há casos em que o pacote de *broadcast* nem possui cabeçalho IP, como no caso do ARP (*Address Resolution Protocol*). Dessa forma, a definição de qual rede o pacote pertence deve ser feita na camada de enlace. Contudo, o modo *router* nativo trata somente a camada de rede. Assim, a virtualização de redes introduz alguns desafios cuja solução depende de ações que considerem o endereçamento da camada de enlace, no caso Ethernet, e permita o encaminhamento pelo endereçamento da camada de rede, o IP.

Uma possível solução para separar o tráfego de uma rede virtual do tráfego das demais é realizar a multiplexação e demultiplexação de pacotes entre interfaces de rede virtuais e físicas através marcação de pacotes com etiquetas, ou *tags*, de VLAN, como provido pelo o Open vSwitch [Pfaff et al. 2009]. O Open vSwitch age como uma ponte Ethernet, porém com algumas opções especiais, como por exemplo o uso de VLAN, que é o acréscimo de uma informação para a multiplexação/demultiplexação no cabeçalho do pacote. Assim, a ideia básica ao se usar o Open vSwitch é marcar as interfaces de rede virtuais com uma etiqueta (*tag*) comum às interfaces que pertençam a um mesmo domínio de *broadcast*. Essa etiqueta, com o identificador da VLAN, é aplicada a todos os pacotes que saem da interface de rede virtual marcada com ela. A etiqueta, nesse caso, é usada como um identificador do enlace virtual ou do domínio de *broadcast* virtual ao qual o pacote pertence. Dessa forma, duas redes virtuais podem possuir a mesma faixa de endereçamento sem que uma interfira na outra, garantido o isolamento do espaço de endereçamento e dos pacotes de *broadcast*. No entanto, como o Open vSwitch age como uma ponte Ethernet, as mesmas limitações do modo *bridge* do Xen se aplicam, não permitindo que seja realizado o paradigma da separação de planos.

O Open vSwitch oferece o modo OpenFlow que é capaz de realizar o processamento de pacotes em diversas camadas seguindo a interface de programação de aplicação (API – *Application Programming Interface*) OpenFlow [McKeown et al. 2008]. O OpenFlow permite o controle de plataformas de encaminhamento através da definição de fluxo que consideram campos das camadas de enlace, de rede e superiores para definir as regras de encaminhamento dos pacotes. Logo, uma solução para fazer a separação de planos com o isolamento de redes virtuais é usar o Open vSwitch se comportando como um comutador programável OpenFlow realizando separação de planos com isolamento. Essa é a ideia básica da arquitetura de redes isoladas proposta nesse artigo.

4. A Arquitetura Proposta

A separação de planos na arquitetura proposta é alcançada da seguinte maneira. Na máquina virtual, executa o `Client` que é um aplicativo que executa em segundo plano e verifica se há atualização na tabela de rotas ou na tabela ARP da máquina virtual e as envia para o `Server`[1], no Domínio 0 da máquina física que a hospeda. O `Server` é basicamente um procurador, *proxy*, que recebe as conexões de todos os clientes, trata as mensagens, as responde e repassa as informações de todos os clientes concentradas e resumidas para a aplicação que executa sobre o POX [2], um controlador do comutador

[1] A comunicação entre o *Client* e o *Server* ocorre através de interfaces de redes dedicadas à comunicação entre máquinas virtuais e Domínio 0. Todas as comunicações de controle são encriptadas e autenticadas através da troca de certificados, usando o protocolo SSL v3.0 (*Secure Socket Layer*).

[2] http://www.noxrepo.org/pox/about-pox/.

OpenFlow que permite o desenvolvimento de aplicações em *Python*. A aplicação desenvolvida sobre o POX é a que mantém a estrutura de dados para armazenar as tabelas de rota de cada máquina virtual e, também, faz a tradução das rotas em fluxos OpenFlow. Contudo, como o Open vSwtich não é capaz de marcar os pacotes com a etiqueta de VLAN, padrão 802.1Q, ao mesmo tempo em que permite o controle pelo OpenFlow, um elemento importante da arquitetura proposta é o marcador de VLAN que se insere entre as interfaces de rede das máquinas virtuais e o comutador Open vSwitch que implementa o plano de dados OpenFlow. Os componentes da arquitetura proposta estão explicitados na Figura 2.

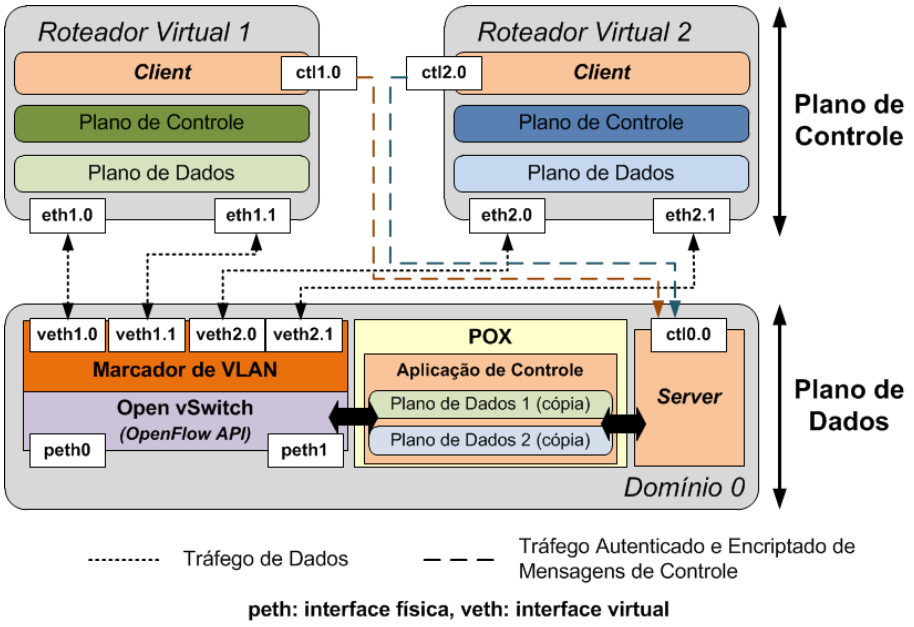


Figura 2. A arquitetura proposta, baseada na virtualização de redes XenFlow, com marcador de VLAN, o comutador Open vSwitch, a aplicação POX e para a construção das tabelas e o Server para a comunicação das tabelas de rotas do roteador virtual para o Domínio 0. Toda comunicação de controle é encriptada e autenticada através do protocolo SSL.

A ideia chave da proposta se baseia no uso do modo OpenFlow, para ter ao mesmo tempo informações da Camada Ethernet, de VLAN e da Camada IP através dos 12 campos da especificação OpenFlow, e também na inserção do marcador de VLAN entre as interfaces virtuais e comutador OpenFlow. Assim, o pacote que entra ou sai de uma máquina virtual obrigatoriamente deve marcado, ou desmarcado, com a etiqueta da VLAN a que pertence. O elemento marcado de VLAN é configurado no momento de criação das máquinas virtuais com o identificado (ID) da rede virtual que a interface de rede virtual pertence. A função do marcador de VLAN é inserir a etiqueta de VLAN em todo o pacote que saia da máquina virtual em direção ao comutador por *software* e de retirar a etiqueta de VLAN de todos os pacotes que saem do comutador em direção à máquina virtual. Esse funcionamento torna a marcação e o encaminhamento por VLAN transparente para as máquinas virtuais. O marcador de VLAN também é o responsável por garantir que os pacotes que não pertençam a uma dada rede virtual cheguem a máquinas virtuais de outras redes, pois o marcador de VLAN descarta os pacotes que chegam até ele, mas não têm a etiqueta de VLAN com o identificador correto.

A tradução de uma rota em fluxo ocorre da seguinte forma. Ao chegar um pacote no comutador por *software* (Open vSwitch) de um Domínio 0, se não existir um fluxo ao

qual o pacote se adeque, o pacote é enviado ao POX, conforme o funcionamento normal do OpenFlow. No POX, o pacote é processado pela aplicação que realiza a separação de planos e verifica a qual máquina virtual o pacote se destina de acordo com o seu endereço MAC de destino. Uma vez identificada a máquina virtual, o endereço IP de destino do pacote é verificado. Se o endereço IP de destino do pacote se adequar a alguma rota daquela máquina virtual, a aplicação extrai o IP do próximo salto através da rota na tabela de rotas da máquina virtual identificada que melhor se adeque ao endereço do pacote, algoritmo de *best match*. Após extrair o IP do próximo salto do pacote na rede, a aplicação do POX verifica a cópia da tabela ARP da máquina virtual, para a qual o pacote se destina, se a máquina virtual já conhece o mapeamento do endereço IP no endereço MAC do próximo salto. Conhecendo o endereço MAC, a aplicação POX define um novo fluxo no plano de dados OpenFlow. Contudo, o roteamento ocorre entre redes distintas e, assim, entre VLANs diferentes. Assim, ao identificar a interface de saída do roteador virtual em que o pacote deve ser encaminhado, a aplicação POX identifica também a VLAN de saída do pacote. Para tanto, a aplicação POX consulta qual o marcador de VLAN está associado à interface virtual de rede, pela qual o pacote seria encaminhado caso realmente fosse encaminhado pela máquina virtual, recupera o novo identificador de VLAN do pacote e adiciona o novo fluxo no plano de dados OpenFlow. O novo fluxo é introduzido de acordo com os campos do pacote que dispararam o seu cálculo e as ações associadas a esse novo fluxo são trocar os endereços MAC de origem e destino do pacote, a troca do identificador de VLAN do pacote e, por fim, encaminhar o pacote na porta de saída adequada.

As ações configuradas para cada fluxo no comutador OpenFlow correspondem ao roteamento do pacote. A troca dos endereços MAC marca a troca do enlace pelo qual o pacote está sendo encaminhado. Nesse sentido, as ações são colocar o endereço MAC de origem do pacote como sendo o endereço da interface pelo qual o pacote seria encaminhado pela máquina virtual e colocar como MAC de destino do pacote como o endereço MAC consultado na tabela ARP da máquina virtual, este endereço é a tradução do IP do próximo salto para o MAC do próximo salto do pacote na rede. A troca do identificador de VLAN corresponde à troca do segmento Ethernet em que o pacote é difundido. A descoberta de em qual porta física do comutador OpenFlow o pacote modificado deve ser encaminhado é feita pelo mecanismo de aprendizagem do comutador. Assim, quando um pacote chega ao comutador, este armazena o endereço MAC de origem do pacote, por qual porta o aquele chegou e um temporizador associada a essa entrada. Desde que o temporizador esteja válido, o comutador sempre sabe em qual porta um endereço MAC está disponível. Se qualquer uma dessas fases falhar, a aplicação POX encaminha o pacote como se fosse um comutador comum, mantendo o mesmo identificador de VLAN. Esse comportamento faz com que a arquitetura proposta se comporte tanto como um roteador, quando conhece uma rota para o pacote, ou como um comutador, quando alguma das etapas de processamento do pacote falha. No caso de o nó físico da rede não conhecer nenhuma rota para o pacote em processamento e, também, não conhecer em que porta o MAC de destino é acessível, o pacote é inundado na rede. O comportamento de inundação é o comportamento padrão de um comutador que não conhece ainda como alcançar o endereço MAC de destino.

O isolamento do espaço de endereçamento de redes virtuais depende da garantia de que os pacotes de uma rede virtual não sejam encaminhados a outras redes virtuais e que os pacotes enviados em difusão, *broadcast*, alcancem somente as estações pertencentes a essa rede virtual. Assim, os responsáveis por garantirem o isolamento mesmo no cenário de separação de planos são o marcador de VLAN e a aplicação de controle do POX, pois todo pacote encaminhado recebe uma etiqueta de VLAN e a troca da etiqueta ocorre sempre que houver o roteamento do pacote no plano de dados.

5. Resultados

O protótipo desenvolvido se serve do Xen 4.0 para prover os planos de controle e do Open vSwitch 1.2.2 para prover a função de encaminhamento no plano de dados do sistema. O Open vSwitch [Pfaff et al. 2009] é configurado para ser usado pelo controlador POX do OpenFlow. A aplicação que realiza a separação de planos e o direcionamento dos pacotes para as VLANs adequadas no plano de encaminhamento de dados foi escrita em Python e executa sobre o POX. As ferramentas `Iperf`[3] e `Tcpdump`[4] foram usadas para realizar as medidas de avaliação de desempenho do sistema.

Dois computadores pessoais compõem o cenário dos experimentos. Ambos executam o protótipo da arquitetura proposta. Nos computadores pessoais foram instanciadas três máquinas que virtuais que apresentam a função de emissor, encaminhador e receptor de pacotes. Todos os computadores possuem processadores Intel Core 2 Quad 2.4 GHz e 4 GB de memória RAM. Cada computador possui, no mínimo, 2 interfaces de rede sendo que todas são configuradas para funcionarem a 100 Mb/s, uma vez que havia também interfaces de 1 Gb/s. As três máquinas virtuais que realizam a função de roteador são configuradas com uma CPU virtual, 128 MB de memória RAM e executa o Debian Linux 2.6-32-5. As máquinas virtuais executam os protocolos de roteamento através da plataforma XORP [Handley et al. 2003], contudo, durante os testes, foram configuradas rotas estáticas. Os resultados apresentados são médias, com intervalo de confiança de 95%.

O primeiro experimento avalia a capacidade da arquitetura proposta em reagir à definição de fluxos em rajadas. Para tanto, a rede de testes foi configurada para uma máquina virtual hospedada no nó físico 1 se comunicar com outra máquina virtual hospedada no nó físco 2, através de um roteador virtual também hospedado no nó físico 2. A geração de novos fluxos foi realizada pelo aplicativo httperf[5] para gera uma taxa fixa conexões HTTP por segundo. O resultado do teste avalia a taxa de conexões atendidas. A Figura 3(a) evidencia que a proposta de isolamento de redes virtuais alcança, nesse cenário, uma taxa de aproximadamente 700 fluxos/s, que é superior ao cenário em que o encaminhamento em Camada 2 do OpenFlow padrão, executado pela aplicação de comutação `forwarding.l2_learning` do POX. O cenário OpenFlow foi testado sob duas hipóteses: usando a marcação de VLANs, referenciado na figura como `OpenFlow com Isolamento`; e sem a marcação de VLANs, `OpenFlow sem Isolamento`. Em ambos os casos o desempenho do OpenFlow foi inferior ao da arquitetura proposta. A melhora introduzida pela proposta deve-se à implementação da separação entre planos de controle e dados, enquanto no OpenFlow age como um comutador somente, enviando os pacotes ao roteador virtual.

O segundo experimento evidencia o isolamento dos recursos de rede implementado pela arquitetura proposta. O experimento avalia o comportamento da proposta para a diferenciação entre redes virtuais. Esse experimento consiste em criar duas redes virtuais, `Rede 1` e `Rede 2`, cada uma com, respectivamente, banda garantida de 20 Mb/s e 80 Mb/s de banda. Para a realização do experimento, cada rede virtual é composta duas máquinas virtuais, uma geradora e uma receptora para cada rede virtual, sendo que as duas máquinas virtuais estão no nó físico 1, enquanto as duas máquinas virtuais receptoras estão no nó físco 2. O gerador de cada uma das redes transmite, a uma taxa constante, 100 Mb/s de tráfego UDP de pacotes de 1472 B[6] para cada rede virtual. A banda total requerida pelas três redes virtuais é então de 200 Mb/s e, portanto, superior a capacidade

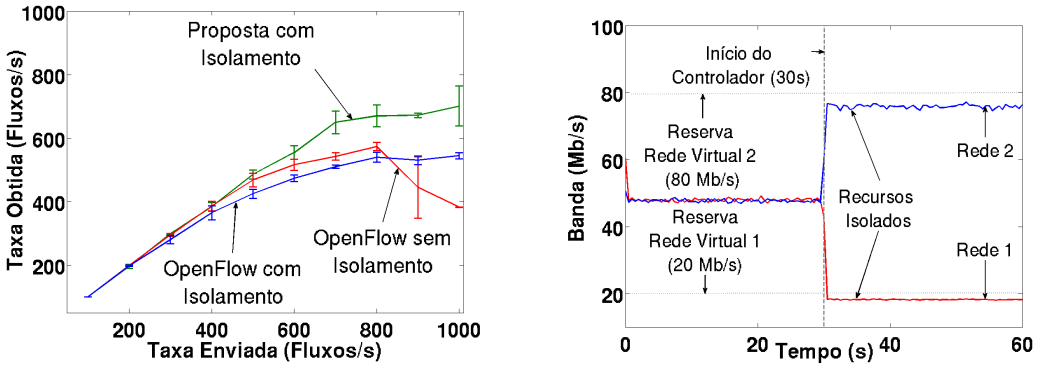
[3]http://iperf.sourceforge.com.

[4]http://www.tcpdump.org.

[5]http://www.hpl.hp.com/research/linux/httperf/

[6]A carga útil dos pacotes gerados é de 1472 B, que somados aos cabeçalhos do UDP e do IP gera um tamanho total de 1500 B, que corresponde ao tamanho máximo de conteúdo de um quadro Ethernet.

dos enlaces de 100 Mb/s. A Figura 3(b) mostra como os recursos são garantidos na arquitetura proposta. Até 30 s de teste não há o isolamento de recursos, assim, as duas redes virtuais compartilham igualmente os recursos do enlace. Após 30 s, o controlador de recursos é ativado, redirecionando os pacotes de cada rede virtual para uma fila e configurando os limites de banda de cada fila. A Figura 3(b) evidencia que o isolamento de banda é alcançado pela arquitetura proposta.



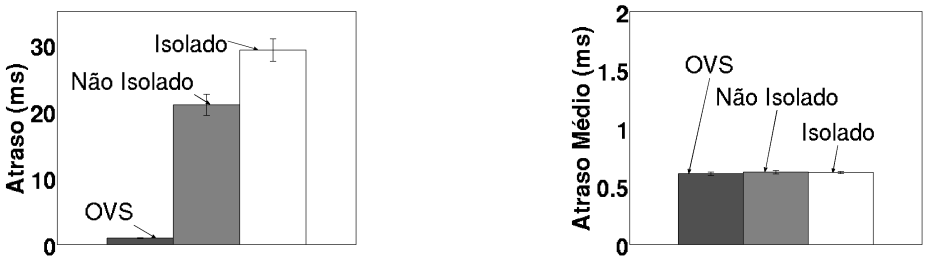
(a) Taxa de fluxos configurados com sucesso em relação a taxa de novos fluxos submetidos ao sistema.

(b) Isolamento de recursos.

Figura 3. Os experimentos foram realizados entre uma máquina virtual hospedada no nó físico 1 se comunicando com duas outra máquinas virtuais hospedada no nó físico 2.

O terceiro experimento tem como objetivo avaliar o atraso introduzido pela arquitetura proposta ao encaminhar os pacotes. A avaliação do atraso baseia-se no tempo de ida e volta (RTT - *Round Trip Time*) de pacotes ICMP *Echo Request* e *Echo Reply*. A Figura 4 compara o atraso introduzido pelo roteamento proposto, referenciado como `Isolado`, com o encaminhamento sem a marcação de VLANs, referenciado como `Não Isolado`, e o atraso do Open vSwitch atuando como comutador, referenciado como `OVS`. Neste experimento, o Open vSwitch foi configurado para somente comutar os pacotes entre as interfaces do nó encaminhador, agindo semelhante a uma *bridge*; e a arquitetura proposta realiza a separação de planos, o isolamento da comunicação de redes virtuais com base na aplicação POX desenvolvida para controlar o plano de dados OpenFlow. O experimento do atraso do primeiro pacote considera somente o atraso do pacote que gera a instanciação do fluxo no plano de dados OpenFlow. Os resultados de atraso mostrados na Figura 4(a) revelam que o atraso introduzido pelo controle da aplicação POX, mesmo sem isolamento, é da ordem de 20 ms. Quando o tratamento do pacote envolve a definição da VLAN em que o pacote deve ser encaminhado, o atraso do primeiro pacote fica em torno de 30 ms. Esse atraso é referente ao encaminhamento do primeiro pacote de cada fluxo para o POX, para que esse processe as informações do pacote e instale um novo fluxo no comutador OpenFlow do plano de dados. No encaminhamento dos demais pacotes, o atraso no encaminhamento é o mesmo para as três abordagens, o que evidencia que esse atraso é devido ao encaminhamento do Open vSwitch.

O próximo experimento verifica a eficácia do mecanismo isolamento da comunicação de redes virtuais. A ideia central desse experimento é definir duas redes virtuais. A Rede Virtual 1 é composta por uma máquina virtual hospedada no Roteador Físico 1. A Rede Virtual 2 é composta por uma máquina virtual hospedada no Roteador Físico 1 e outra hospedada no Roteador Físico 2. Ambos os nós do Roteador Físico 1 são emissores de dados UDP de 1472 B. Todas as máquinas virtuais foram configuradas para pertencerem ao mesmo espaço de endereçamento IP e enviam dados para um

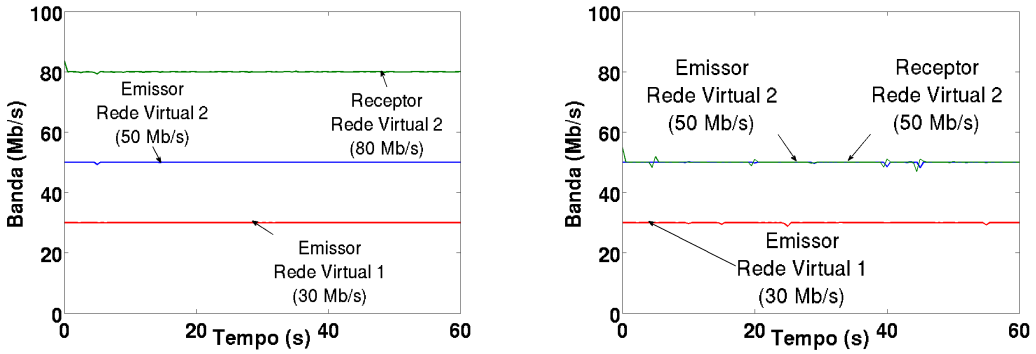


(a) Primeiro pacote do fluxo.

(b) Demais pacotes do fluxo.

Figura 4. Comparação dos atrasos introduzidos pelo encaminhamento a) do primeiro pacote de um fluxo e b) dos demais pacotes de um fluxo.

mesmo IP de destino. No entanto, como as redes virtuais são isoladas, espera-se que o fluxo da Rede Virtual 1 não interfira no fluxo da Rede Virtual 2. A Figura 5(a) mostra que no cenário sem isolamento, o nó receptor da Rede Virtual 2 recebe tanto os pacotes da Rede Virtual 2, como os pacotes da Rede Virtual 1, rompendo com o isolamento da comunicação de redes virtuais. Já a Figura 5(b) demonstra que a proposta isola o espaço de endereçamento de endereçamento de cada rede virtual, pois o tráfego da Rede Virtual 1 não interfere no tráfego da Rede Virtual 2, já que o receptor da Rede Virtual 2 só recebe os pacotes pertencentes à sua rede.



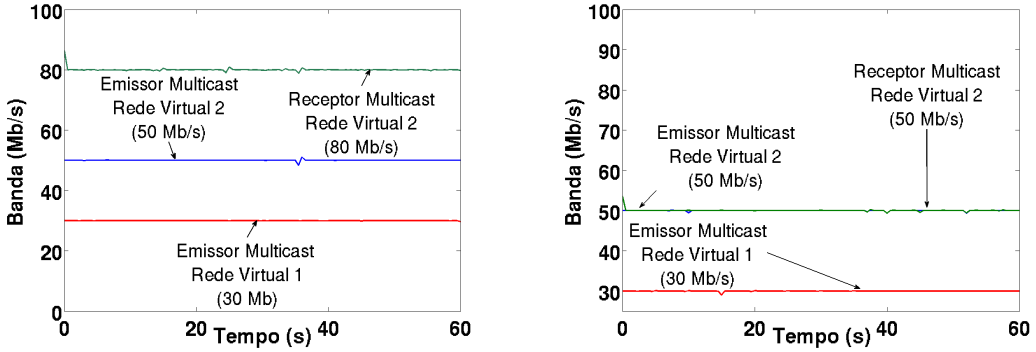
(a) Sem isolamento de comunicação.

(b) Com isolamento da comunicação.

Figura 5. Encaminhamento de pacotes para a Rede Virtual 1 e para Rede Virtual 2 que possuem o mesmo endereço IP. a) redes não isoladas - as duas redes recebem os pacotes encaminhados para o IP comum, pois a Rede 2 recebe a soma (80 Mb/s) da taxa relativa aos pacotes da Rede 1 (30 Mb/s) mais a taxa de pacotes da Rede 2 (50 Mb/s). b) redes isoladas - cada rede virtual só recebe os pacotes que lhe são destinados, pois a Rede 2 só recebe a taxa de 50 Mb/s.

A Figura 6 mostra o isolamento da comunicação entre redes virtuais no cenário de uma comunicação *multicast*. Nesse caso, o cenário é semelhante ao do experimento anterior. No entanto, ao invés de os nós emissores enviarem pacotes para um dado endereço IP *unicast*, os nós agem como fontes de dados *multicast* para um dado grupo. O nó receptor age como sorvedouro desse grupo. Esse teste é importante, pois protocolos de roteamento, como o OSPF (*Open Shortest Path First*), usam comunicação *multicast* para descoberta de topologia, logo, é importante que os pacotes *multicast* de uma dada rede virtual não sejam encaminhados para outras redes. A Figura 6(a) mostra que no caso sem isolamento de comunicação, o nó receptor da Rede Virtual 2 recebe tanto os pacotes de sua rede, quanto os da Rede Virtual 1. Contudo, ao usar o isolamento de comunicação

de redes proposto, o nó receptor da Rede Virtual 2 só recebe os pacotes enviados pelo nó emissor de sua mesma rede, como mostrado na Figura 6(b).



(a) Sem isolamento de comunicação.

(b) Com isolamento da comunicação.

Figura 6. Encaminhamento de pacotes para um endereço de um grupo IP *multicast*. a) redes não isoladas - as duas redes recebem os pacotes encaminhados para o grupo *multicast* comum, pois a Rede 2 recebe a soma (80 Mb/s) da taxa relativa aos pacotes da Rede 1 (30 Mb/s) mais a taxa de pacotes da Rede 2 (50 Mb/s). b) redes isoladas - cada rede virtual só recebe os pacotes que lhe são destinados, pois a Rede 2 só recebe a taxa de 50 Mb/s.

6. Conclusão

O isolamento completo do espaço de endereçamento e o tratamento apropriado dos pacotes multidestinatáios é fundamental para a virtualização seja, em toda a sua plenitude, redes virtuais podendo assumir qualquer valor de espaço de endereçamento e, também, para a segurança, uma vez que impede uma rede virtual de bisbilhotar (*eavesdropping*) os pacotes de outras redes virtuais. Este artigo propôs uma arquitetura de virtualização de redes que provê isolamento completo entre redes virtuais, isolando tanto a comunicação de redes virtuais quanto o consumo de recursos de cada rede virtual. A arquitetura proposta baseia-se no sistema híbrido de virtualização de redes XenFlow, em que o plano de controle executa em uma máquina virtual Xen e o plano de dados é realizado em um comutador por *software* programável OpenFlow. Para garantir o desempenho da arquitetura de virtualização de redes proposta, adota-se o paradigma de separação de planos. Assim, um dos desafios da proposta é realizar o roteamento dos pacotes entre as VLANs, sem que o pacote deixe o plano de dados. A proposta, então, adiciona a identificação da VLAN de destino, associando um identificador de VLAN (*Virtual Local Area Network*) a cada segmento de rede virtual, às ações tomadas no plano de dados ao rotear pacotes. Logo, a ideia chave da arquitetura proposta se baseia no uso do Open vSwitch no modo OpenFlow, para ter ao mesmo tempo informações da Camada Ethernet, de VLAN e da Camada IP através dos 12 campos da especificação OpenFlow, e também na inserção do marcador de VLAN entre as interfaces virtuais e comutador OpenFlow.

Um protótipo da arquitetura foi implementado e avaliado. Os resultados demonstram que o mapeamento dos pacotes em VLANs introduz um atraso no encaminhamento no primeiro pacote. Contudo, o atraso não afeta os demais pacotes do fluxo. A arquitetura proposta alcançou uma taxa de definição de fluxos por segundo superior à alcançada pela aplicação padrão de comutação do OpenFlow. Por fim, os resultados mostram que o isolamento da comunicação entre redes virtuais é alcançado mesmo no cenário em que as redes virtuais executam aplicações de *multicast* ou compartilham o mesmo espaço de endereçamento IP.

Como trabalhos futuros, pretende-se desenvolver mecanismos de controle de admissão de novas redes virtuais, assim como desenvolver métodos de autenticação de roteadores virtuais em roteadores físicos.

7. Referências

[Barabash et al. 2011] Barabash, K., Cohen, R., Hadas, D., Jain, V., Recio, R., and Rochwerger, B. (2011). A case for overlays in dcn virtualization. In *Proceedings of the 3rd Workshop on Data Center-Converged and Virtual Ethernet Switching*, pages 30–37. ITCP.

[Bavier et al. 2006] Bavier, A., Feamster, N., Huang, M., Peterson, L., and Rexford, J. (2006). In vini veritas: realistic and controlled network experimentation. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 3–14, New York, NY, USA. ACM.

[Chun et al. 2003] Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., and Bowman, M. (2003). Planetlab: An overlay testbed for broad-coverage services. *ACM Computer Communication Review*, 33(3):3–12.

[Egi et al. 2007] Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Mathy, L., and Schooley, T. (2007). Evaluating Xen for router virtualization. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 1256–1261. IEEE.

[Feamster et al. 2007] Feamster, N., Gao, L., and Rexford, J. (2007). How to lease the Internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64.

[Fernandes and Duarte 2010] Fernandes, N. and Duarte, O. (2010). XNetMon: Uma arquitetura com segurança para redes virtuais. *Anais do X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 339–352.

[Fernandes et al. 2010] Fernandes, N., Moreira, M., Moraes, I., Ferraz, L., Couto, R., Carvalho, H., Campista, M., Costa, L., and Duarte, O. (2010). Virtual networks: Isolation, performance, and trends. *Annals of Telecommunications*, pages 1–17.

[Fernandes and Duarte 2011] Fernandes, N. C. and Duarte, O. C. M. B. (2011). Provendo isolamento e qualidade de serviço em redes virtuais. In *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*.

[Handley et al. 2003] Handley, M., Hodson, O., and Kohler, E. (2003). XORP: An open platform for network research. *ACM SIGCOMM Computer Communication Review*, 33(1):53–57.

[Huang 2005] Huang, M. (2005). Vnet: Planetlab virtualized network access. Technical report, Tech. Rep. PDN–05–029, PlanetLab Consortium.

[Jiang and Xu 2005] Jiang, X. and Xu, D. (2005). Violin: Virtual internetworking on overlay infrastructure. In Cao, J., Yang, L., Guo, M., and Lau, F., editors, *Parallel and Distributed Processing and Applications*, volume 3358 of *Lecture Notes in Computer Science*, pages 937–946. Springer Berlin Heidelberg.

[Mattos and Duarte 2012] Mattos, D. M. F. and Duarte, O. C. M. B. (2012). QFlow: Um sistema com garantia de isolamento e oferta de qualidade de serviço para redes virtualizadas. In *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2012*.

[Mattos et al. 2011] Mattos, D. M. F., Fernandes, N. C., and Duarte, O. C. M. B. (2011). XenFlow: Um sistema de processamento de fluxos robusto e eficiente para migração em redes virtuais. In *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

[Perlman et al. 2011] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and Ghanwani, A. (2011). Routing Bridges (RBridges): Base Protocol Specification. RFC 6325 (Proposed Standard). Updated by RFCs 6327, 6439.

[Pfaff et al. 2009] Pfaff, B., Pettit, J., Koponen, T., Amidon, K., Casado, M., and Shenker, S. (2009). Extending networking into the virtualization layer. *Proc. HotNets*.

[Pisa et al. 2010] Pisa, P., Fernandes, N., Carvalho, H., Moreira, M., Campista, M., Costa, L., and Duarte, O. (2010). Openflow and xen-based virtual network migration. In Pont, A., Pujolle, G., and Raghavan, S., editors, *Communications: Wireless in Developing Countries and Networks of the Future*, volume 327 of *IFIP Advances in Information and Communication Technology*, pages 170–181. Springer Boston.

[Wang et al. 2008] Wang, Y., Keller, E., Biskeborn, B., van der Merwe, J., and Rexford, J. (2008). Virtual routers on the move: live router migration as a network-management primitive. *ACM SIGCOMM Computer Communication Review*, 38(4):231–242.

[Wood et al. 2009] Wood, T., Gerber, A., Ramakrishnan, K., Shenoy, P., and Van der Merwe, J. (2009). The case for enterprise-ready virtual private clouds. *Usenix Workshop HotCloud'09*.