Um Mecanismo para Isolamento Seguro de Redes Virtuais Usando a Abordagem Híbrida Xen e OpenFlow*

Diogo Menezes Ferrazani Mattos, Lyno Henrique Gonçalves 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

{menezes,lyno,otto}@gta.ufrj.br

Resumo. *Redes virtuais seguras devem prover a privacidade e evitar a negação de serviço. Este artigo propõe um mecanismo de isolamento seguro baseado no paradigma de separação do plano de controle do de dados e na técnica de encaminhamento de pacotes XenFlow, uma abordagem híbrida usando Xen e OpenFlow. O mecanismo proposto emprega uma aplicação de controle de redes OpenFlow para marcar com etiquetas cada rede formada por máquinas virtuais Xen. No paradigma de separação de planos o encaminhamento dos pacotes de cada rede virtual é realizado diretamente no plano de dados e o mecanismo proposto garante o isolamento do tráfego com reserva de recursos para cada rede virtual. O protótipo desenvolvido é avaliado perante dois modelos de atacante: o primeiro tenta exaurir os recursos das redes virtuais e o segundo tenta bisbilhotar a comunicação de redes virtuais. Os resultados dos experimentos revelam a eficácia do mecanismo proposto em isolar o tráfego até em situações adversas como redes virtuais com o mesmo endereço IP ou com pacotes em difusão.*

Abstract. *Secure virtual networks must provide privacy and avoid denial of service attacks. In this paper, we propose a mechanism which securely isolates virtual networks based on the paradigm of data and control plane separation. The proposal is deployed over XenFlow, a hybrid network virtualization tool that uses both Xen and OpenFlow. The proposed mechanism associates an OpenFlow control application with a labeling scheme for each virtual network of Xen virtual machines. In the plane separation paradigm, packets of each virtual network are forwarded directly in data plane, and the proposal ensures traffic isolation with resource reservation for each virtual network. The experiments evaluate two attackers' models: one that tries to exhaust the resources of virtual networks and other that attempts to eavesdrop on communications from other virtual networks. The results show that the proposal completely blocks the action of both attackers' model. Results reveal the effectiveness for traffic isolation, even on adverse situations, where virtual networks share the same IP address space or broadcast packets.*

1. Introdução

A virtualização de rede é uma tecnologia essencial para prover um ambiente de experimentação 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

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

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 cada rede virtual deve ser 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 composto por duas vertentes importantes: o isolamento de recursos [Mattos and Duarte 2012, Fernandes and Duarte 2010] e o isolamento da comunicação entre nós de uma rede virtual [Barabash et al. 2011]. O isolamento no compartilhamento de recursos garante que um elemento virtual não interfira no desempenho dos demais elementos virtuais. O isolamento de recursos é importante porque evita a negação de serviços entre redes virtuais, já que uma rede virtual não consegue exaurir os recursos de outras. O isolamento da comunicação das redes virtuais, por sua vez, garante que a comunicação de uma rede virtual só alcance os nós que de fato pertençam a essa rede virtual, mesmo que duas redes virtuais distintas usem o mesmo espaço de endereçamento IP. O isolamento da comunicação é essencial em meios de comunicação em difusão, como o Ethernet [Perlman et al. 2011]. Um pacote em difusão (*broadcast*), ou com múltiplas destinações (*multicast*), deve ser acessível somente pelos nós que pertençam a uma mesma rede virtual, enquanto que os nós de outras redes virtuais 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 pacotes de outras redes virtuais. Outro ponto importante do isolamento da comunicação é liberdade de uso de espaços de endereçamento e limitação do escopo de endereços somente para o alcance dos nós de uma rede virtual. Assim, 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 um mecanismo de isolamento da comunicação e também de isolamento de recursos entre redes virtuais. A proposta se baseia no paradigma da separação de planos, 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 sejam encaminhados diretamente no plano de dados, sejam acessíveis somente pelos nós dessa rede e não interfiram no funcionamento das demais. Para tanto, o mecanismo de isolamento proposto estende o 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, proposto no sistema QFlow [Mattos and Duarte 2012], garantindo a virtualização de redes em um ambiente em que haja garantia da reserva de recursos para redes virtuais e, ao mesmo tempo, garantia de que os pacotes de uma dada rede sejam somente entregues aos nós de destino.

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 visam o o isolamento da comunicação das redes virtuais, baseando-se no encapsulamento dos pacotes das redes virtuais [Barabash et al. 2011, Perlman et al. 2011, Nakagawa et al. 2012, Sridharan et al. 2013]. O mecanismo proposto, no entanto, usa o paradigma da separação de planos para garantir o máximo desempenho do encaminhamento de pacotes no plano de dados 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 que são definidas pela aplicação OpenFlow e traduz as regras do plano de controle para o plano de dados. A proposta deste artigo 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 Seção 4 detalha o mecanismo de isolamento seguro de redes virtuais proposto. O modelo de atacante considerado é apresentado na Seção 5, seguido pela avaliação do mecanismo proposto que é discutida na Seção 6. A Seção 7 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 propostas de virtualização de redes [Mattos and Duarte 2012, Pisa et al. 2010, Fernandes and Duarte 2011, Fernandes et al. 2010]. Uma forma de baixo custo de se alcançar a virtualização de redes é usar uma plataforma de virtualização de *hardware* para computadores pessoais em que as máquinas virtuais executam funções de elementos de rede [Wang et al. 2008, Egi et al. 2007].

Uma abordagem comum de virtualização em centros de dados é o uso ferramentas de virtualização de servidores, como o Xen, e o agrupamento de máquinas virtuais em uma mesma VLAN isolando o tráfego de cada inquilino do centro de dados em uma rede virtual [Bari et al. 2013]. Contudo, outras propostas defendem o uso de mecanismos mais complexos para alcançar o isolamento de recursos e de comunicação na rede de centros de dados [Mudigonda et al. 2011, Greenberg et al. 2009, Hao et al. 2010].

A proposta NetLord [Mudigonda et al. 2011] introduz um agente em *software* em servidor físico de cada servidor físico que encapsula o quadro de uma máquina virtual em um novo pacote IP cuja semântica dos endereços das Camadas 2 e 3 é sobrecarregada para indicar à qual rede virtual os quadros pertencem. Na proposta NetLord, o marcador de VLAN é usado para identificar um dos múltiplos caminhos que o pacote pdoe seguir na rede do centro de dados. De forma semelhante, a proposta VL2 [Greenberg et al. 2009] encapsula os pacotes IP de uma rede virtual em outro pacote IP. Nesse caso, a semântica dos endereços IP dos pacotes é sobrecarregada. O IP mais externo é usado para multiplexar a qual rede virtual o pacote pertence. A proposta SEC2 [Hao et al. 2010] divide a rede de centro de dados em domínio de um núcleo e múltiplos domínios de borda. Nos domínios de borda, uma rede virtual é identificada com uma etiqueta de VLAN. A comunicação entre dois domínios de borda distintos é realizada através do domínio de núcleo por meio de um elemento de encaminhamento que adiciona um cabeçalho extra de Camada 2 nos quadros, entre domínios, para realizar a multiplexação das redes virtuais.

Distributed Overlay Virtual Ethernet (DOVE) é uma proposta de virtualização de redes que provê isolamento [Barabash et al. 2011], permitindo a criação de redes virtuais dinâmicas, com espaços de endereçamentos isolados sobre uma infraestrutura física comum. O funcionamento do DOVE baseia-se no encapsulamento de pacotes e, portanto,

na criação de uma rede de sobrecamada para permitir a separação entre rede virtual e a rede física subjacente. O isolamento de redes virtuais é alcançado pelo uso de um identificador de rede que é adicionado ao cabeçalho do envelope da rede sobrecamada DOVE. Assim, os pacotes com um dado identificador só são entregues às máquinas virtuais que compartilhem o mesmo identificador de rede virtual.

O isolamento na comunicação também pode ser alcançado usando o encapsulamento VXLAN [Nakagawa et al. 2012]. Para tanto, são adicionados um cabeçalho mais externo Ethernet, seguido de um cabeçalho externo IP, UDP e VXLAN. O cabeçalho IP designa as extremidades do túnel VXLAN. O cabeçalho VXLAN inclui 24 bits para identificar a rede à qual o quadro Ethernet, encapsulado, pertence. No caso de um pacote de multidestinação ou em difusão, o cabeçalho IP mais externo tem como endereço de destino um IP *multicast* e cada rede virtual é representada por um grupo *multicast*. O padrão *Network Virtualization Generic Routing Encapsulation* (NVGRE) [Sridharan et al. 2013] também se serve da técnica de encapsulamento para prover uma virtualização de redes controlada por *software* para permitir múltiplos inquilinos (*multitenancy*) em nuvens públicas ou privadas. Os quadros Ethernet das redes virtuais são encapsulados em um túnel GRE que conecta duas estações. O cabeçalho GRE apresenta um campo para a marcação de a qual inquilino tal quadro pertence. Esse campo é chamado TNI (*Tenant Network Identifier*), composto por 24 bits para a identificação da rede virtual.

O isolamento da comunicação de redes virtuais pode ser alcançado 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 pessoal 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. Dessa forma, um pacote só é entregue à máquina virtual se sua interface pertencer à VLAN que está marcada no pacote. Contudo, o Open vSwitch não realiza o roteamento entre VLANs com separação de planos e, também, depende de uma lógica de marcação de VLANs definida pelo administrador da rede. O mecanismo proposto, por sua vez, permite a configuração dinâmica da rede com separação de planos.

As propostas XNetMon [Fernandes and Duarte 2010] e XNetMan [Fernandes and Duarte 2011] usam 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 IP da rede virtual a que se destinam. Assim, o isolamento se restringe a redes virtuais com espaços de endereçamento disjuntos. O sistema de virtualização de redes XenFlow [Mattos et al. 2011] 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 provê somente o isolamento do compartilhamento 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.

Todas as propostas acima apresentadas, para prover o isolamento do espaço de endereçamento e da comunicação das redes virtuais, realizam o encapsulamento dos pacotes das redes virtuais e, assim, criam uma rede sobrecamada que interconecta os nós da rede virtual. Assim, essas propostas falham ao se considerar o paradigma da separação de planos, em que o encaminhamento é realizado no plano de dados, na camada Ether-

net, enquanto que os protocolos de roteamento são executados no plano de controle, na camada de rede, e são responsáveis por gerenciar as rotas do plano de dados. O mecanismo proposto neste artigo isola redes virtuais ainda na camada Ethernet mesmo no cenário de separação de planos. A proposta insere uma etiqueta de VLAN nos pacotes de cada rede virtual, eliminando a necessidade de criar uma rede sobrecamada. A proposta estende o sistema XenFlow [Mattos et al. 2011] e aplica o isolamento de recursos do sistema QFlow [Mattos and Duarte 2012]. A principal contribuição do artigo é realizar o isolamento da comunicação de redes virtuais através de primitivas do plano de dados, como a marcação da VLAN. Vale ressaltar que o mecanismo proposto mantém todo o encaminhamento dos pacotes na camada de enlace, Ethernet, já que não há o encapsulamento dos quadros por protocolos de camada de rede, deixando todos os campos do pacote original da rede virtual visíveis para o controlador OpenFlow, enquanto propostas como NVGRE ou VXLAN escondem os campos do pacote original em um pacote encapsulado cuja destinação são as extremidades dos túneis criados.

3. O Encaminhamento de Pacotes em Redes Virtuais

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 com destino às diferentes máquinas virtuais [Egi et al. 2007, Figueiredo et al. 2013].

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 de planos, o plano de controle é responsável por calcular as rotas, já o plano de dados é responsável por encaminhar os pacotes de acordo com essas rotas. A separação de planos depende de informações da camada de rede e, portanto, o modo *bridge* não atende. Por outro lado, no modo *router*, a separação 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 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: 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. A identificação trivial falha se duas redes virtuais distintas usarem o mesmo espaço de endereçamento IP. Dessa forma, não há isolamento de comunicação entre as redes virtuais.

O segundo problema consiste no encaminhamento de pacotes com endereços multidestinatários ou de difusão (*multicast/broadcast*), pois neste caso um único endereço de IP se refere a um grupo de hospedeiros, ou seja, pacote multidestinatário ou de difusão possuem 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 rotas 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 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 de comutação de fluxos. A definição de fluxo OpenFlow considera campos das camadas de enlace, de rede e superiores para definir as regras de encaminhamento dos pacotes. O Open vSwitch 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]. 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 do XenFlow, com a proposta de isolamento seguro desse artigo.

4. O Mecanismo Proposto

O mecanismo proposto marca os pacotes de cada rede virtual com uma etiqueta de VLAN para executar o isolamento da comunicação e de recursos. O desempenho do mecanismo é garantido executando as operações de marcação e roteamento diretamente no plano de dados, de acordo com as informações calculadas pelo plano de controle nos roteadores virtuais.

Na máquina virtual, executa o `Client` que é um aplicativo que verifica se há atualização na tabela de rotas e 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 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 de comutadores OpenFlow que permite o desenvolvimento de aplicações escritas 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 vSwitch 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 1.

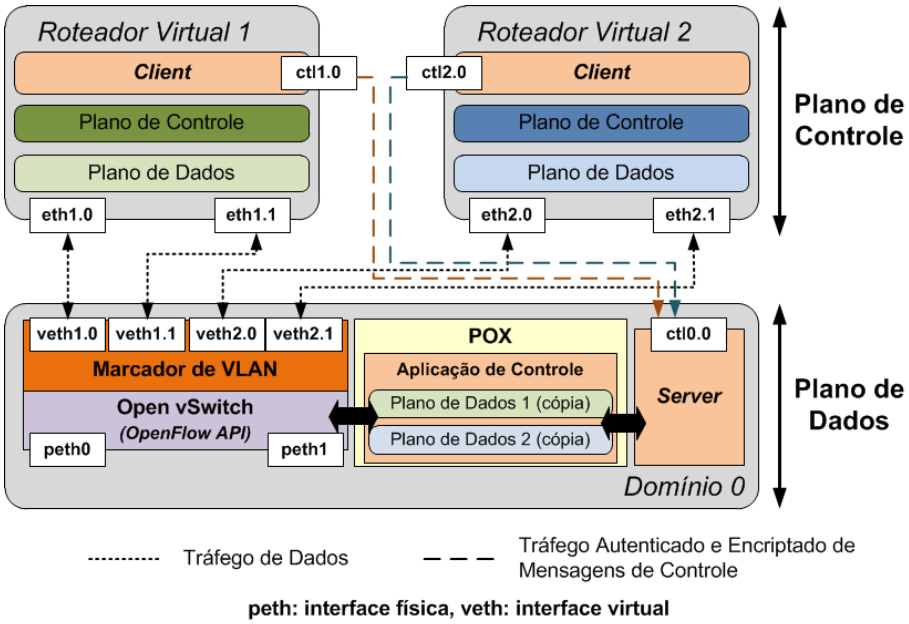


Figura 1. A nova arquitetura XenFlow proposta com o marcador de VLANs, o comutador Open vSwitch, a aplicação POX para a construção das tabelas e o `Server` para a comunicação das tabelas de rotas do roteador virtual para o Domínio 0.Toda comunicação de controle é encriptada e autenticada através do protocolo SSL.

A ideia chave da proposta se baseia no uso do Open vSwitch como um comutador 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 o comutador OpenFlow. Assim, o pacote que entra ou sai de uma máquina virtual obrigatoriamente deve ser marcado, ou desmarcado, com a etiqueta da VLAN a qual ele pertence. O marcador 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

[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/.

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 que 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 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 e verifica 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. 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 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 aquele pacote chegou e um temporizador associado 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 o mecanismo proposto 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 ainda não conhece como alcançar um endereço MAC de destino.

Considerando o mecanismo proposto, os responsáveis por garantirem o isolamento da comunicação, mesmo em um 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. Já o isolamento de recursos é obtido mapeando-se os fluxos de cada rede virtual em filas com garantia de banda. O mapeamento de fluxos em filas é realizado considerando o identificador da etiqueta de VLAN como o critério de classificação de fluxos adotado pelo sistem QFlow [Mattos and Duarte 2012].

5. O Modelo de Atacante

As redes virtuais são administradas por entidades diferentes e todas compartilham os mesmos recursos físicos. Assim, não é possível assumir que todas as redes virtuais são confiáveis e bem comportadas, ou que não tentariam comprometer as demais, seja intencionalmente ou não. Dessa forma, dois modelos de rede virtual atacante são considerados nesse artigo. O primeiro consiste em um nó atacante, pertencente a uma dada rede virtual, que gera pacotes a uma taxa superior à reservada para a sua rede virtual, para outro nó de sua mesma rede virtual. Assim, o atacante aumenta a taxa de geração de pacotes na tentativa de degradar o desempenho do encaminhamento de pacotes nas demais redes virtuais que compartilham os mesmos nós físicos com o nó malicioso. O segundo modelo é baseado na quebra da privacidade da rede virtual. O nó atacante assume o endereço IP de um nó de outra rede virtual e tenta escutar os pacotes dessa outra rede virtual. Os comportamentos de degradar o desempenho de outras redes virtuais que compartilhem recursos e bisbilhotar o tráfego de outra rede virtual são definidos como comportamentos prejudiciais e o nó, ou a rede, que adotarem esses comportamentos são maliciosos.

6. Resultados

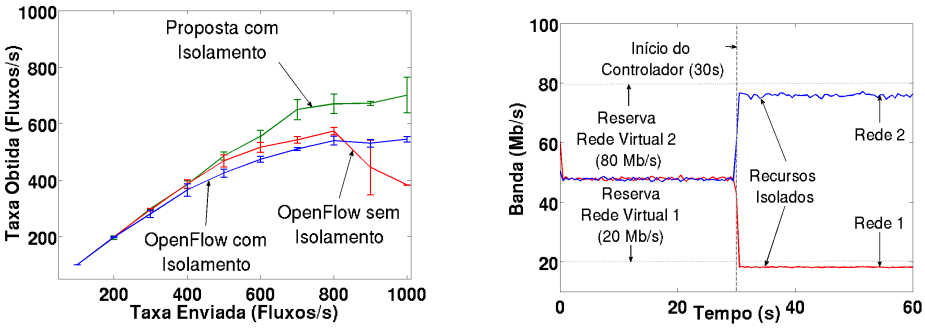
O protótipo desenvolvido utiliza o hipervisor Xen 4.0 para prover os domínios virtuais que hospedam os planos de controle dos elementos de rede virtuais e utiliza o comutador programável Open vSwitch 1.2.2 para prover a função de encaminhamento no plano de dados do sistema. O Open vSwitch [Pfaff et al. 2009] é configurado para ser usado pelo controlador POX do OpenFlow. As ferramentas `Iperf`[3], `Tcpdump`[4] e `httperf`[5] foram usadas para realizar as medidas de avaliação de desempenho do protótipo. Dois computadores pessoais compõem o cenário dos experimentos. Ambos executam o protótipo da arquitetura proposta. Nos computadores pessoais foram instanciadas três máquinas 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. Durante os testes, foram configuradas rotas estáticas entre as máquinas virtuais. Os resultados apresentados são médias, com intervalo de confiança de 95%.

O primeiro experimento avalia a negação de serviço devido a baixo desempenho de processamento na construção de tabelas de fluxos. Assim, avalia-se a capacidade do

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

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

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



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

(b) Isolamento de recursos. Atacante tenta exaurir recursos de rede.

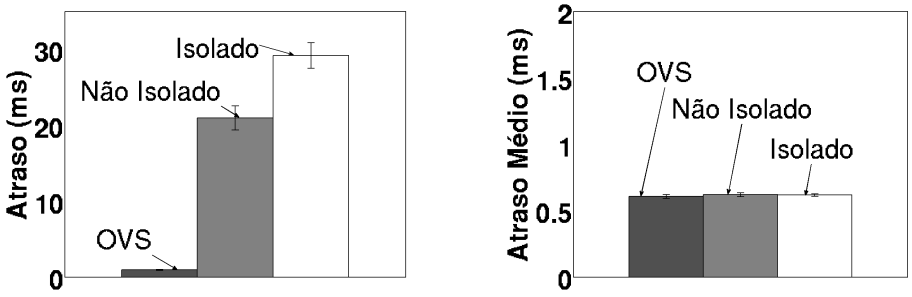
Figura 2. Os experimentos foram realizados entre duas máquinas virtuais hospedadas no nó físico 1 se comunicando com duas outras máquinas virtuais hospedadas no nó físico 2. Cada par, geradora/receptora, pertence a uma rede virtual.

mecanismo proposto em reagir à definição de fluxos em rajadas, uma vez que optou-se pelo uso do OpenFlow. Deve ser ressaltado que o OpenFlow envia o cabeçalho do primeiro pacote de todo fluxo para seja inserido este novo fluxo na tabela de fluxos. Esse procedimento é um gargalo do OpenFlow e, portanto, é importante avaliar se o desempenho do mecanismo proposto nessas condições é pior que o do OpenFlow. 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 físico 2, através de um roteador virtual também hospedado no nó físico 2. A geração de novos fluxos foi realizada pelo aplicativo `httperf` para gerar uma taxa fixa de conexões HTTP por segundo. O resultado do teste avalia a taxa de conexões atendidas. Cada conexão bem sucedida resulta na instanciação de dois novos fluxos. A Figura 2(a) evidencia que a proposta de isolamento seguro 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 do mecanismo proposto. A melhora introduzida pela proposta deve-se à implementação da separação entre planos de controle e dados que processa todos os pacotes, inclusive o primeiro pacote, no Domínio 0, enquanto no OpenFlow age como um comutador somente, enviando os pacotes para o roteador virtual.

O segundo experimento evidencia o isolamento dos recursos de rede provido pelo mecanismo proposto. O experimento avalia o comportamento do mecanismo sob o ataque de negação de serviço de um nó virtual que tenta exaurir os recursos de rede do nó físico. Para tanto, o experimento mostra a diferenciação de qualidade de serviço entre redes virtuais. Esse experimento consiste em criar duas redes virtuais, `Rede 1` e `Rede 2`, cada uma com, respectivamente, banda passante garantida de 20 Mb/s e 80 Mb/s. Para a realização do experimento, cada rede virtual é composta por duas máquinas virtuais, uma geradora e uma receptora para cada rede virtual, sendo que as duas máquinas virtuais geradoras 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 para cada rede virtual[6]. Ambas as redes têm capacidade

[6]A carga útil dos pacotes gerados é de 1472 B, que somados aos cabeçalhos do UDP e do IP gera um

de exaurir os recursos do nó físico. A banda total requerida pelas duas redes virtuais é então de 200 Mb/s e, portanto, superior a capacidade dos enlaces de 100 Mb/s. A Figura 2(b) mostra a proteção do mecanismo proposto ao ataque de negação de serviço entre redes virtuais. Até 30 s de teste não há o isolamento de recursos, assim, as duas redes virtuais disputam 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 2(b) evidencia que o isolamento de banda é alcançado pelo mecanismo proposto.



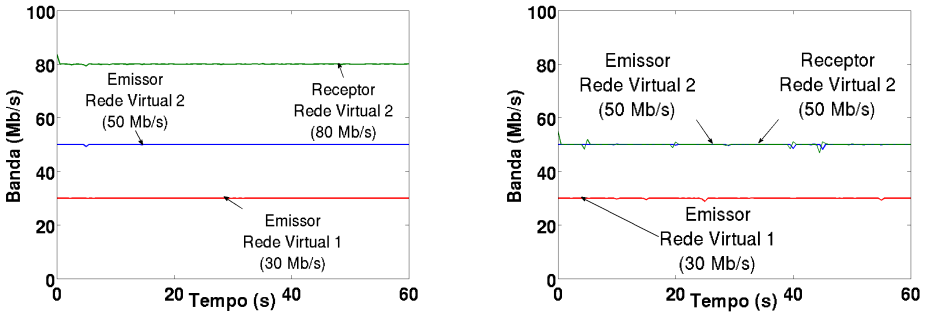
(a) Primeiro pacote do fluxo. (b) Demais pacotes do fluxo.

Figura 3. Comparação dos atrasos introduzidos pelo encaminhamento a) do primeiro pacote de um fluxo e b) dos demais pacotes de um fluxo.

O terceiro experimento tem como objetivo avaliar o atraso introduzido pelo mecanismo proposto 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 3 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`. Na figura, a curva do Open vSwitch se refere a configuração para somente comutar os pacotes entre as interfaces do nó encaminhador, agindo semelhante a uma *bridge*; enquanto o mecanismo de isolamento seguro proposto também usa o mecanismo Open vSwitch, mas realiza também a separação de planos, isolando a 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 3(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 chega a aproximadamente 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 introduzido é o mesmo para as três abordagens, o que evidencia que esse atraso é devido somente ao encaminhamento do Open vSwitch.

O próximo experimento verifica a eficácia do mecanismo isolamento da comunicação de redes virtuais, quando um atacante tenta bisbilhotar a comunicação de outra rede virtual (*eavesdropping*) ou injetar tráfego em outra rede. A ideia central desse experimento é definir duas redes virtuais e, então, uma rede tentar injetar tráfego na outra. A Rede Virtual 1 é composta por uma máquina virtual hospedada no Roteador Físico 1. A

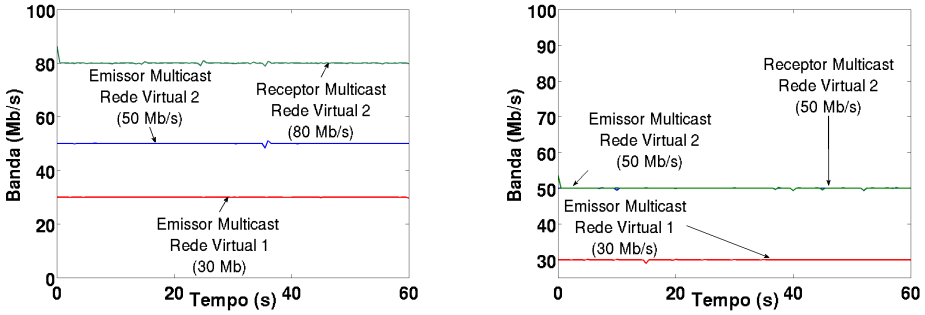
tamanho total de 1500 B, que corresponde ao tamanho máximo de conteúdo de um quadro Ethernet.



(a) Sem isolamento de comunicação. (b) Com isolamento da comunicação.

Figura 4. 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.

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 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 4(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, comprovando a inexistência de isolamento da comunicação de redes virtuais e sendo capaz de bisbilhotar os fluxos da Rede Virtual 1. Já a Figura 4(b) demonstra que o mecanismo proposto isola o espaço 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 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.

A Figura 5 mostra o isolamento da comunicação entre redes virtuais no cenário de uma comunicação *multicast*. Nesse caso, o cenário é semelhante ao do experimento ante-

rior. 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. A eliminação desta vulnerabilidade é significativa, pois os 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, nem seja possível que outras redes injetem tráfego *multicast* em uma dada rede virtual. A Figura 5(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. A Figura 5(b) comprova que, ao usar o mecanismo de 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.

7. Conclusão

O isolamento seguro do espaço de endereçamento e o tratamento apropriado dos pacotes multidestinatários é fundamental para garantir que uma rede virtual maliciosa não seja capaz de bisbilhotar (*eavesdropping*) ou injetar pacotes em outras redes virtuais. Este artigo propôs um mecanismo de isolamento para a 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. O mecanismo proposto 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. A principal contribuição desse artigo é a garantia de isolamento de redes virtuais empregando o paradigma de separação de planos que resulta também em um alto desempenho do encaminhamento de pacotes. Assim, um dos desafios da proposta é rotear os 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 proposta se baseia no uso do Open vSwitch, como um comutador OpenFlow, para obter ao mesmo tempo informações da Camada Ethernet, de VLAN e da Camada IP através dos campos da especificação OpenFlow, e também na inserção do marcador de VLAN entre as interfaces virtuais e o 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. O mecanismo proposto 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 em *multicast* ou compartilham o mesmo espaço de endereçamento IP. Os experimentos demonstram a segurança do mecanismo proposto a dois modelos de atacante: (i) capaz de bisbilhotar redes virtuais que compartilham o mesmo substrato; (ii) capaz de exaurir recursos de rede compartilhados.

Como trabalhos futuros a etiqueta de VLAN será substituída por uma marcação MPLS (*Multiprotocol Label Switching*), cuja semântica do identificador de circuitos virtuais será sobrecarregada. Nessa abordagem, é possível usar até 23 bits para a marcação da rede virtual mantendo as vantagens do sistema proposto. Assim, o mecanismo proposto poderia identificar até 8 milhões de redes virtuais em vez das atuais 4 mil possíveis com VLAN, o que é um requisito para centros de dados de grande porte.

8. 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.

[Bari et al. 2013] Bari, M., Boutaba, R., Esteves, R., Granville, L., Podlesny, M., Rabbani, M., Zhang, Q., and Zhani, M. (2013). Data center network virtualization: A survey. *Communications Surveys Tutorials, IEEE*, 15(2):909–928.

[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*.

[Figueiredo et al. 2013] Figueiredo, U., Lobato, A., Mattos, D. M. F., Ferraz, L. H. G., and Duarte, O. C. M. B. (2013). Análise de desempenho de mecanismos de encaminhamento de pacotes em redes virtuais. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos 2013 - XVIII Workshop de Gerência e Operação de Redes e Serviços (WGRS)*.

[Greenberg et al. 2009] Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. (2009). Vl2: a scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009*, SIGCOMM '09, pages 51–62, New York, NY, USA. ACM.

[Hao et al. 2010] Hao, F., Lakshman, T. V., Mukherjee, S., and Song, H. (2010). Secure cloud computing with a virtualized network infrastructure. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, HotCloud'10, Berkeley, CA, USA. USENIX Association.

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

[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.

[Mudigonda et al. 2011] Mudigonda, J., Yalagandula, P., Mogul, J., Stiekes, B., and Pouffary, Y. (2011). Netlord: a scalable multi-tenant network architecture for virtualized datacenters. In *Proceedings of the ACM SIGCOMM 2011*, SIGCOMM '11, pages 62–73, Toronto, Ontario, Canada. ACM.

[Nakagawa et al. 2012] Nakagawa, Y., Hyoudou, K., and Shimizu, T. (2012). A management method of ip multicast in overlay networks using openflow. In *Proceedings of the first workshop on Hot topics in software defined networks*, HotSDN '12, pages 91–96, Helsinki, Finland. ACM.

[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.

[Sridharan et al. 2013] Sridharan, M., Duda, K., Ganga, I., Greenberg, A., Lin, G., Pearson, M., and Thaler, P. (2013). NVGRE: Network Virtualization using Generic Routing Encapsulation. NVGRE.

[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.