

XenFlow: Um Sistema de Processamento de Fluxos Robusto e Eficiente para Migração em Redes Virtuais*

Diogo Menezes Ferrazani Mattos, Natalia Castro Fernandes e
Otto Carlos Muniz Bandeira Duarte

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

Resumo. *O remapeamento das topologias das redes virtuais sobre o substrato físico é uma importante funcionalidade que oferece uma flexibilidade significativa para um ambiente de redes virtuais. Na plataforma OpenFlow, o remapeamento é simples porque equivale a reconfigurações de fluxos. Contudo, redes OpenFlow apresentam um controle centralizado que representa um ponto único de falhas. A plataforma de virtualização Xen permite criar ambientes virtuais com controle distribuído, mas o remapeamento das topologias é bem mais restrito e complexo. Este artigo propõe um sistema de virtualização híbrido baseado nas plataformas Xen e OpenFlow. O principal objetivo da proposta é criar redes virtuais com controle distribuído nas quais o remapeamento de topologias virtuais seja flexível, simples e robusto. Foi desenvolvido um protótipo e os resultados obtidos mostram que a proposta apresenta um desempenho superior, quando comparada com a migração convencional de uma máquina virtual Xen, além de permitir a migração de topologias virtuais entre diferentes redes locais sem necessidade de túneis.*

Abstract. *Remapping virtual network topologies over the physical substrate is an important feature and provides flexibility to virtual network environments. Topology remapping, over the OpenFlow platform, is simple because it is equivalent to reset flow definitions. The control of an OpenFlow network, however, is centralized, which represents a single point of failure. On the other hand, the Xen virtualization platform allows instantiating virtual environments, with distributed control, but the topology remapping is restrictive and complex. In this paper, we propose a hybrid virtualization system, based on Xen and OpenFlow platforms. The proposal main goal is creating virtual networks with distributed control, in which the topology remapping is flexible, simple and robust. We developed a prototype and our results show that the proposed system has performed better than the Xen virtual machine migration, and also allows migrating virtual topologies among local area networks, without creating tunnels.*

1. Introdução

A implantação de novos protocolos e serviços no núcleo da Internet sofre a rejeição de grande parte dos provedores de serviços devido ao grande risco que essas mudanças representam para o bom funcionamento da rede. Uma das propostas para conciliar o desenvolvimento de inovações sem prejudicar o tráfego de produção é a

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

virtualização de redes [Feamster et al. 2007, Ratnasamy et al. 2005]. Essa técnica baseia-se no compartilhamento do substrato físico por diferentes redes virtuais. Como as redes virtuais são isoladas, o tráfego experimental não influencia o tráfego de produção. Além disso, o modelo de redes virtuais paralelas é apontado como uma das principais abordagens para a Internet do Futuro [Moreira et al. 2009], na qual cada rede virtual possui a sua própria pilha de protocolos e arcabouço de gerenciamento. Nesse contexto, novas funções de gerenciamento são necessárias, como o mapeamento das topologias das redes virtuais sobre a topologia física.

Uma primitiva de controle fundamental para remapear as topologias das redes virtuais, também chamadas de topologias lógicas, sobre o substrato físico é a migração de roteadores virtuais [Wang et al. 2008]. A primitiva de migração pode ser usada em diversos contextos, desde a realocação da topologia lógica para a manutenção de nós da rede, até o provimento de uma rede verde, que busca minimizar os gastos energéticos. Operações de manutenção de um nó na rede frequentemente requerem o seu desligamento. Isso gera a quebra de conexões e, no caso de os nós serem roteadores, o desligamento pode gerar a perda das adjacências dos protocolos de roteamento, introduzindo um atraso de convergência até que as rotas sejam reorganizadas em todos os nós. Como a primitiva de migração garante que a topologia lógica não é alterada, o serviço prestado por um roteador que vai entrar em manutenção pode ser migrado para outro roteador e as rotas continuam válidas. A migração também é usada em cenários de redes verdes, para reduzir o número de nós físicos ativos na rede. Quando há nós físicos subutilizados, a migração é chamada para remapear a rede virtual sobre a rede física, de forma que alguns nós possam ser desligados sem comprometer os serviços prestados [Bolla et al. 2009]. A migração pode, ainda, ser usada para evitar danos no caso de ataques de negação de serviço (DoS – *Denial of Service*). Nessa situação, as redes virtuais, que compartilham o mesmo substrato com a rede sob ataque, são migradas para outros nós físicos fora da região do ataque. Contudo, a migração ainda apresenta grandes desafios, como a migração dos enlaces virtuais, ou seja, como mapear um enlace lógico em um ou mais enlaces físicos. Outro desafio é a redução da perda de pacotes causada pelo período de suspensão dos serviços do roteador virtual durante a migração.

Duas importantes técnicas de virtualização de redes com suporte a migração são oferecidas pelas plataformas Xen [Egi et al. 2008] e OpenFlow [McKeown et al. 2008]. O Xen é uma plataforma de virtualização de recursos de computadores pessoais que é utilizada para prover redes virtuais através da virtualização de roteadores [Egi et al. 2008, Fernandes et al. 2010, Pisa et al. 2010]. No entanto, o suporte à migração no Xen não é adequado ao cenário de roteadores virtuais, pois apresenta perda de pacotes e está restrito a migrações no interior de uma rede local. O OpenFlow é uma plataforma baseada em comutadores programáveis que dá suporte a virtualização de redes quando utilizada em conjunto com a aplicação FlowVisor [Sherwood et al. 2009]. O encaminhamento no OpenFlow é feito com base na definição de regras de encaminhamento para cada fluxo de pacotes. Contudo, todos os mecanismos de controle da rede OpenFlow são realizados por um nó central, chamado de controlador, o que limita a escalabilidade dessas redes. Além disso, o controle centralizado dificulta a utilização de mecanismos de controle bem conhecidos na rede, como os protocolos de roteamento, os quais são implementados com base em um controle distribuído.

Este artigo propõe o XenFlow, um sistema de virtualização híbrido baseado nas

plataformas Xen e OpenFlow. O XenFlow garante um amplo suporte à primitiva de migração em um ambiente que facilita o desenvolvimento de inovações e que apresenta a opção por controle distribuído ou centralizado da rede. Assim, o XenFlow disponibiliza ferramentas para um remapeamento das topologias virtuais, realizando tanto migração de roteadores virtuais, quanto a migração de enlaces virtuais. No XenFlow, 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, é responsável por todas as tarefas de controle, como a atualização da tabela de roteamento, enquanto que o plano de dados, criado utilizando-se a plataforma OpenFlow, é responsável pelo encaminhamento dos pacotes. A vantagem da utilização do Xen é a criação de uma plataforma com controle distribuído e com amplo suporte a inovação, pois as máquinas virtuais são isoladas umas das outras e o administrador da rede virtual pode desenvolver seus mecanismos de controle sem restrições. A vantagem da utilização do OpenFlow é que, por ser um comutador programável, a migração dos enlaces virtuais torna-se uma atividade trivial, permitindo um remapeamento simples de topologias lógicas sobre a topologia física. No XenFlow, cada máquina física possui um comutador OpenFlow para ligar as máquinas virtuais Xen ao resto da rede e cada máquina virtual Xen funciona como um controlador do comutador. O remapeamento das topologias é orquestrado por uma terceira entidade capaz de atuar sobre os comutadores OpenFlow e disparar a migração das máquinas virtuais em qualquer nó da rede.

Um protótipo do sistema XenFlow foi construído para a validação da proposta. Os resultados experimentais mostram que a migração é feita sem perda de pacotes e sem a interrupção do serviço de encaminhamento. O sistema também é eficiente, pois faz a migração sem que haja a perda de conexões ou atrasos no encaminhamento dos pacotes. Quando comparado à migração de máquina virtual nativa do Xen, o sistema XenFlow apresentou perda zero de pacotes, enquanto a migração nativa perdeu uma quantidade significativa de pacotes, além de apresentar um período maior de interrupção do plano de controle.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta as tecnologias usadas na definição da plataforma híbrida de virtualização de redes. A definição do sistema proposto, sua arquitetura e principais componentes são apresentados na Seção 3. A Seção 4 apresenta a análise dos resultados experimentais. A Seção 5 discute os trabalhos relacionados. As conclusões e trabalhos futuros são apresentados na Seção 6.

2. Plataformas de Virtualização Xen e OpenFlow

XenFlow é um sistema de virtualização de redes, baseado na plataformas Xen e OpenFlow, que provê as funcionalidades de amplo suporte à inovação, de remapeamento de topologias virtuais de forma simples, com baixos atrasos e sem perda de pacotes, e com controle distribuído da rede. Essas características são obtidas por utilizar uma plataforma para suprir as deficiências da outra, enquanto que as vantagens são somadas.

O OpenFlow é uma plataforma de comutadores programáveis que permite configurar o encaminhamento dos pacotes com base em definições de fluxos. Um fluxo é definido como um conjunto de doze características da comunicação entre dois nós na rede, definidas pelos cabeçalhos das camadas de enlace, de rede e de transporte de cada pacote [McKeown et al. 2008]. Como em um comutador padrão, o comutador OpenFlow

também possui uma tabela de encaminhamento, que é chamada de tabela de fluxo. As entradas nessa tabela relacionam um fluxo com um conjunto de ações definidas pelo controlador sobre como o comutador deve tratar cada pacote que tenha as características definidas pelo fluxo. Um dos principais controladores para OpenFlow é o Nox [Gude et al. 2008], que age como uma interface entre as aplicações de controle e a rede OpenFlow. Quando um pacote chega ao comutador OpenFlow, o comutador verifica se o pacote se adequa a algum fluxo já existente. Em caso positivo, as ações definidas para aquele fluxo são aplicadas ao pacote. Em caso negativo, o primeiro pacote do fluxo é enviado ao controlador, que define um novo fluxo e determina as ações a serem tomadas.

No OpenFlow, o remapeamento de uma topologia virtual sobre a topologia física corresponde a uma sequência de migrações de fluxos e, portanto, é bastante simples. Para migrar um fluxo, basta reconfigurar as tabelas de fluxo de todos os comutadores envolvidos nas topologias lógicas de antes, removendo configurações de fluxo, e de depois da migração, adicionando e/ou modificando configurações de fluxo [Pisa et al. 2010], como mostrado na Figura 1. Isso é possível, pois o controlador centralizado tem acesso e controla a configuração das tabelas de todos os comutadores na rede virtual.

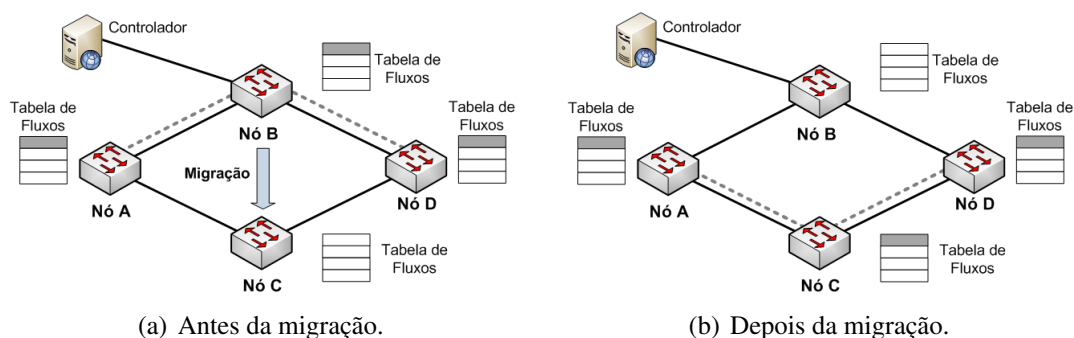


Figura 1. Modelo de rede OpenFlow, na qual o fluxo que passa pelo caminho A-B-D é migrado para o caminho A-C-D.

O Xen é uma plataforma de virtualização de computadores pessoais, bastante empregada na consolidação de servidores, cuja arquitetura é baseada em uma camada de virtualização, denominada Monitor de Máquina Virtual (VMM) ou hipervisor, responsável por compartilhar o *hardware* entre diversos ambientes virtuais. O ambiente virtual Xen é chamado de máquina virtual, ou domínio, e apresenta ao seu sistema operacional uma abstração de *hardware* (CPU, memória, disco rígido e interface de rede). Cada ambiente virtual está isolado dos demais. Há, ainda, um ambiente virtual privilegiado, denominado Domínio 0, que detém o acesso aos dispositivos físicos e provê acesso às operações de Entrada/Saída aos demais domínios, além de executar operações de gerência do hipervisor. A Figura 2 mostra uma rede virtualizada baseada em Xen. A rede é composta por máquinas virtuais agindo como roteadores [Fernandes et al. 2010, Egi et al. 2008]. Nesse cenário, migrar um roteador virtual equivale a migrar uma máquina virtual. Como um roteador executa um serviço em tempo real, a migração de um roteador virtual demanda que o tempo de interrupção de seu serviço seja o menor possível.

A migração de máquinas virtuais nativa do Xen [Clark et al. 2005] consiste em instanciar uma nova máquina virtual na máquina física de destino que tenha os mesmos dados de disco e memória que tinha a máquina a ser migrada no momento que a

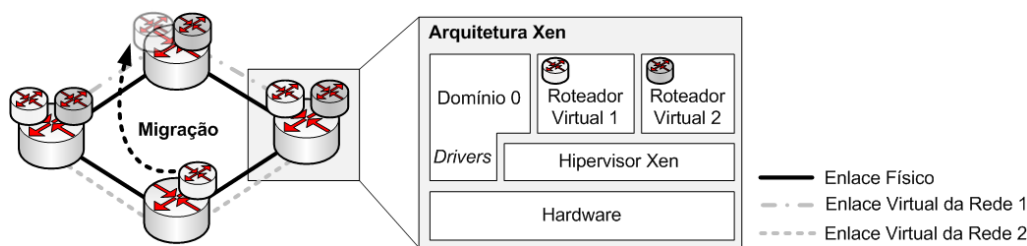


Figura 2. Virtualização de redes usando Xen, na qual cada roteador físico hospeda um conjunto de roteadores virtuais.

migração foi realizada. Como o estado da memória na nova máquina virtual é igual ao da máquina virtual de origem, então a nova máquina funciona a partir do mesmo estado em que se encontrava a máquina antiga. Contudo, esse processo possui uma fase na qual a máquina virtual fica indisponível tanto na máquina física de origem quanto na de destino da migração, pois a máquina virtual precisa ser suspensa para que os seus dados de memória sejam atualizados corretamente na máquina de destino. Após o término da cópia, a máquina virtual é restaurada na máquina física de destino. Portanto, uma desvantagem dessa proposta para a migração de roteadores virtuais é a perda de pacotes durante o período entre a suspensão e a restauração da máquina virtual. Outra desvantagem é que esse mecanismo está limitado a migrações entre máquinas físicas em uma mesma rede local, pois a migração dos enlaces da máquina virtual é realizada através do envio de pacotes de *ARP Reply*. Esse pacote anuncia no enlace virtual qual é o novo endereço físico da placa de rede para que os pacotes da rede virtual possam ser encaminhados corretamente na camada de enlace.

3. Sistema XenFlow

O sistema proposto combina as vantagens da virtualização de redes com controle distribuído e ambiente flexível para o desenvolvimento de novas aplicações de controle, que é provida pela plataforma Xen, e processamento por fluxo e programabilidade do plano de dados, providas pela plataforma OpenFlow. A arquitetura de um elemento de rede XenFlow é mostrada na Figura 3. A arquitetura proposta é baseada no paradigma de separação de planos, segundo o qual as funções de controle e de encaminhamento de um roteador são separadas em dois planos. A base da arquitetura é a plataforma de virtualização Xen. Dessa forma, a virtualização do plano de controle é alcançada através da instanciação de máquinas virtuais, que executam as funções de controle da rede. Os planos de dados dos roteadores virtuais são criados através da instanciação de um comutador OpenFlow no Domínio 0 do Xen. O comutador OpenFlow faz a ligação entre todas as interfaces de rede das máquinas virtuais e todas as interfaces físicas. Para configurar o comutador OpenFlow, é instanciado um controlador NOX também no Domínio 0. Esse controlador se comunica com cada um dos planos de controle, localizados nas máquinas virtuais, para obter as tabelas de encaminhamento geradas pelos protocolos de roteamento em execução em cada máquina virtual. Com base nesses dados, o controlador é capaz de instanciar as regras de encaminhamento corretamente para cada novo fluxo da rede. Dessa forma, o controlador NOX no Domínio 0 tem o papel de mapear a tabela de encaminhamento na tabela de fluxos sob demanda. Uma outra possibilidade para o desenvolvimento da proposta seria instalar um FlowVisor no Domínio 0 e um controlador NOX em cada uma das máquinas virtuais. Isso, no entanto, iria implicar em uma perda de generalidade,

pois os *softwares* de controle de rede amplamente utilizados teriam que ser adaptados para se tornarem uma aplicação do NOX. Com o sistema proposto, cada máquina virtual funciona como um controlador para o comutador OpenFlow, mas usando o *software* de controle legado ao invés de reimplementar todas as funções na plataforma de desenvolvimento do NOX. Devido a todas essas características, embora uma rede OpenFlow pura apresente um controle centralizado, o controle de cada rede virtual no XenFlow é distribuído.

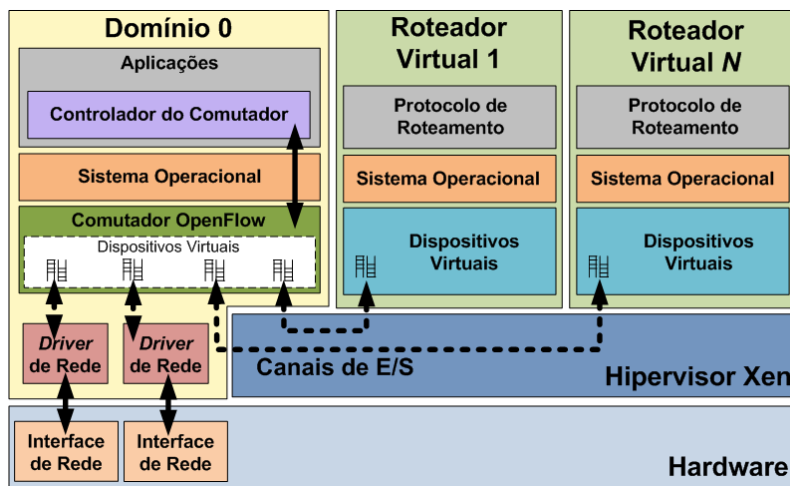


Figura 3. Arquitetura de um elemento de rede do sistema XenFlow.

De acordo com a arquitetura proposta, todos os nós da rede devem ter um comutador OpenFlow, embora nem todos os nós precisem executar o Xen, como mostrado na Figura 4. No exemplo da Figura, é apresentada uma infraestrutura física com duas redes virtuais. Cada elemento de rede virtual pode ser um comutador (nível 2), roteador (nível 3) ou um elemento de processamento de pacotes (nível maior que 3), também chamado de *middle box*. Os roteadores implementam o controle da rede, os *middle boxes* implementam qualquer tipo de processamento por pacote, enquanto que os comutadores tem por objetivo apenas ligar máquinas que pertencem a uma mesma rede local.

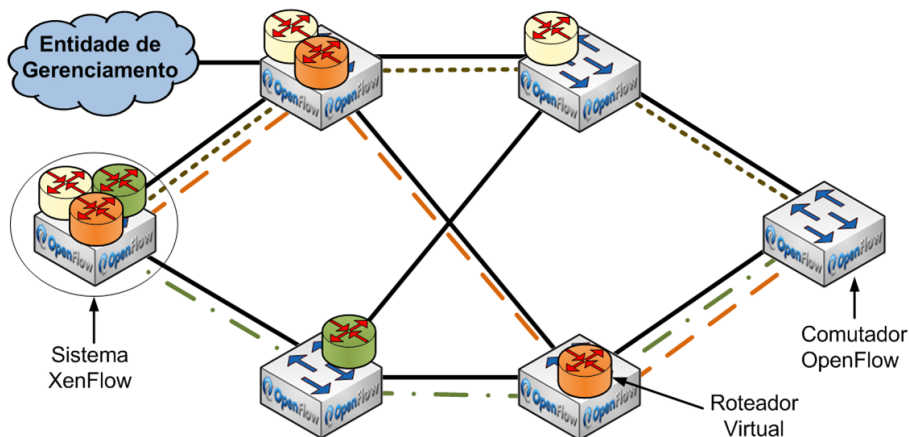


Figura 4. Rede XenFlow, composta por comutadores OpenFlow puros e nós XenFlow, formados pela união de máquinas virtuais Xen e um comutador OpenFlow..

Na Figura 4, há ainda uma entidade de gerenciamento. Essa entidade tem por ob-

jetivo orquestrar as migrações na rede e, para tanto, tem consciência da topologia física da rede. A topologia pode ser obtida através de informações obtidas a partir de protocolos de roteamento de estado de enlace, como o OSPF (*Open Shortest Path First*). A entidade pode ser centralizada, tendo conhecimento de toda a rede, ou distribuída, na qual cada nó da entidade conhece uma área da rede e se comunica com os demais para obter informações além da sua própria área. Assim, a migração é iniciada a partir da entidade de gerenciamento e pode ser disparada em qualquer nó da rede.

3.1. Separação de Planos e Tradução de Rotas em Fluxos

A migração sem perda de pacotes é alcançada através da técnica de separação de planos. As informações de rotas estão no plano de controle e as regras de encaminhamento, no plano de dados. Com a separação de planos há necessidade de se ter uma cópia do plano de dados de cada roteador. No XenFlow, o plano de dados no Domínio 0 é composto pelo comutador OpenFlow e pelo Controlador Nox, como ilustrado na Figura 5. O controlador NOX mantém uma Tabela de Regras que corresponde a cópias atualizadas do conteúdo do plano de dados de cada roteador virtual. O *Daemon* de Atualização da Tabela de Regras, executado em cada roteador virtual, é responsável por manter a cópia do plano de dados atualizada, através da monitoração constante de mudanças na tabela de roteamento. A aplicação Tabela de Regras, que executa sobre o controlador NOX, passa, então, a ser responsável por traduzir a tabela de regras, contendo todas as tabelas de encaminhamento, em entradas na tabela de fluxo do OpenFlow. Cabe observar que o OpenFlow possui apenas uma tabela de fluxo que é compartilhada por todas as redes. Uma vez que os fluxos do OpenFlow são definidos pelo controlador NOX com base em todos os 12 campos do cabeçalho do pacote e assume-se que as características que definem cada uma das redes são bem definidas e não-superpostas, então isso é suficiente para garantir que a regra definida para um fluxo de uma rede não irá interferir nos fluxos das outras redes. Com isso, não existe necessidade para isolar as tabelas de fluxo de cada rede no Domínio 0.

A tabela de fluxos do OpenFlow é criada sob demanda. Todo pacote que chega ao comutador OpenFlow que não possui algum fluxo definido é enviado para o controlador, que tem por função primária instanciar os fluxos. Assim, ao chegar ao comutador OpenFlow, o pacote tem dois tratamentos possíveis: ser encaminhado diretamente para a interface de saída, caso coincida com um fluxo existente na tabela de fluxos, e, caso não coincida, ser encaminhado para o controlador, para que este defina o seu caminho. Se o pacote encaminhado para o controlador for de controle, o controlador classifica para qual roteador virtual o pacote deve ser encaminhado e adiciona um fluxo de curta duração no comutador OpenFlow. O pacote é então enviado para o plano de controle no roteador virtual. Ao receber o pacote de controle, o roteador virtual atualiza a sua tabela de rotas. Quando o *daemon* de atualização de tabela de regras verifica que houve uma atualização na tabela de rotas, ele envia uma mensagem de atualização para o controlador NOX.

Caso o pacote encaminhado pelo comutador para o controlador seja de dados, o o controlador NOX extrai os doze campos necessários para a definição do fluxo, consulta a tabela de regras para definir qual é rota o pacote deve seguir e insere um fluxo na tabela de fluxos do comutador. É importante ressaltar que um roteador convencional ao encaminhar um pacote incrementa o campo TTL e troca o campo de endereço MAC de destino. Tal comportamento deve ser realizado também no roteador virtual do XenFlow. Assim, o controlador, ao consultar a tabela de regras, obtém a interface de saída do pacote e o

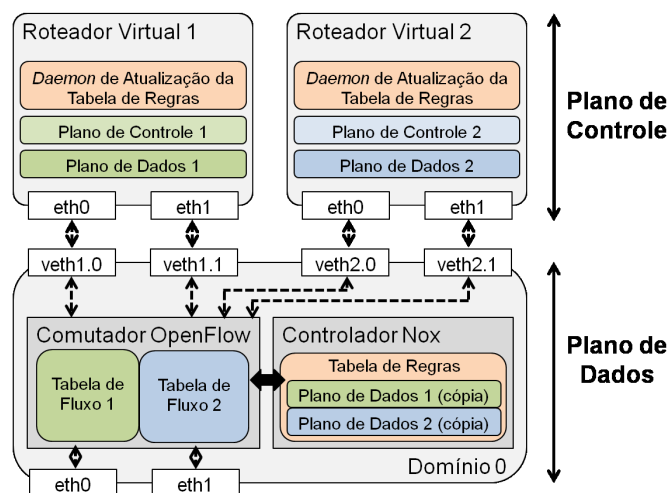


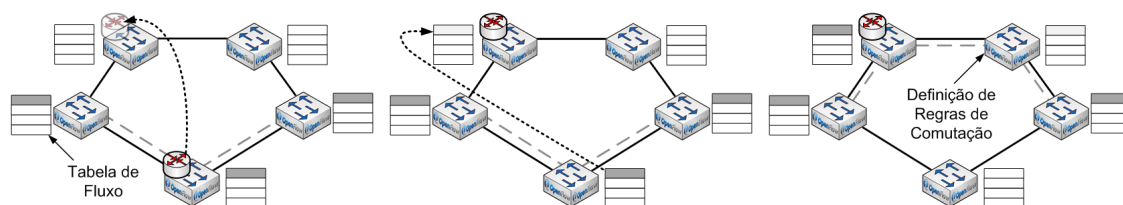
Figura 5. Roteamento no sistema XenFlow, que é baseado no paradigma da separação de planos.

MAC de destino e usa esses dados para configurar as ações a serem tomadas para todos os pacotes daquele fluxo, que incluem trocar o MAC e encaminhar para a porta de saída correta no comutador OpenFlow. A consulta na tabela de regras é realizada com base no endereço MAC de destino do pacote, que identifica a qual roteador virtual o pacote se destina, e o endereço IP de destino do pacote, que é usado para verificar, entre as rotas definidas por aquele roteador virtual, a qual rota o pacote melhor se adequa. Os dados retornados por essa consulta são qual é o endereço MAC do próximo salto na rede e por qual porta, interface física conectada ao comutador OpenFlow do Domínio 0, o próximo salto é alcançável. Dessa forma, as informações da Tabela de Regras são mapeadas sob demanda na tabela de fluxos do comutador OpenFlow e os enlaces virtual são mapeados nos enlaces físicos.

3.2. Migração de Topologias Virtuais no XenFlow

Em uma rede com o sistema XenFlow, um enlace virtual pode ser mapeado em um ou mais enlaces físicos. A migração de uma topologia virtual no XenFlow, mostrada na Figura 6, se dá em três etapas: migração do plano de controle, migração do plano de dados e migração de enlaces. O plano de controle é migrado entre dois nós físicos da rede, através do mecanismo de migração ao vivo de máquinas virtuais convencional do Xen [Clark et al. 2005]. Em seguida, a migração do plano de dados é realizada. Os fluxos referentes ao roteador virtual migrado são selecionados e migrados para o roteador físico de destino. A migração do fluxo segue o algoritmo de migração do OpenFlow, engloba o mapeamento das portas de entrada e saída, dos dois roteadores físicos envolvidos, e a migração dos enlaces. No destino, os fluxos migrados são mapeados para atual configuração do roteador físico e do roteador virtual. As interfaces referidas nos fluxos migrado são substituídas, na definição dos fluxos no destino, por interfaces do roteador de destino. A interface que melhor substitui a original é aquela que alcança os mesmos vizinhos da interface no roteador de origem, com o menor número de saltos. Em seguida, os fluxos traduzidos são adicionados à tabela de fluxos do comutador OpenFlow do Domínio 0 de destino. Após, ocorre a migração dos enlaces. A migração de enlaces ocorre nos comutadores OpenFlow dos Domínios 0 e nos outros comutadores da rede. A

migração de enlaces ocorre de forma a criar um caminho comutado entre os vizinhos, a um salto lógico do roteador virtual, até o roteador físico de destino da migração. Para tanto, são criados fluxos nos roteadores físicos que estão no caminho entre o roteador físico de destino e os roteadores físicos que hospedam os roteadores virtuais vizinhos do migrado. No entanto, adicionar somente os fluxos já existentes às tabelas de fluxo dos nós físicos no caminho não é suficiente. É necessário que haja um mecanismo de criação automática de novos fluxos sob demanda. Esse mecanismo se dá através da introdução de novas regras nas tabelas de regras dos controladores dos nós no caminho. Essas novas regras, nos nós intermediários, definem que todos os pacotes que cheguem ao nó intermediário, com destino ao roteador virtual migrado ou vindos dele, devem ser encaminhados diretamente na interface que liga o nó intermediário ao próximo nó que esteja do caminho previamente definido. A verificação de se um pacote destina-se ao roteador virtual migrado ou foi encaminhado por ele é realizada através da comparação, respectivamente, dos endereços MAC de destino e de origem do pacote com os endereços MAC das interfaces do roteador migrado.



(a) Migração do Plano de Controle. (b) Migração do Plano de Dados. (c) Migração dos Enlaces.

Figura 6. As três etapas da migração de topologia virtual em uma rede XenFlow.

4. Resultados Experimentais

Foi desenvolvido um protótipo para realizar a prova de conceito da migração de roteadores virtuais sem perda de pacotes. O protótipo foi implementado em Python e utiliza a técnica de separação de planos, oferecendo uma interface de migração de roteadores virtuais e uma interface de migração de enlaces para aplicações de controle. O plano de dados foi implementado como uma tabela de fluxos e uma aplicação do controlador NOX, do comutador OpenFlow. A aplicação do NOX define os fluxos diretamente entre interface de entrada e de saída, na tabela de fluxo do módulo comutador OpenFlow local, alterando o endereço MAC de destino dos pacotes encaminhados para o endereço MAC do próximo salto. Para avaliar o desempenho, foi adotada a ferramenta *Iperf*¹ como gerador de pacotes e a ferramenta *tcpdump*² para medir quantidade de pacotes gerados, recebidos e perdidos. A perda de pacotes foi medida a partir da comparação das informações coletadas pelo *tcpdump* nas interfaces de rede responsáveis geração e de recepção dos pacotes.

O cenário de testes foi composto por quatro máquinas. Duas máquinas executam a função de encaminhamento de pacotes e nelas foi instalado o protótipo. Essas máquinas são equipadas com processador Intel Core 2 Quad e três interfaces de rede Ethernet de 1Gb/s, executando o hipervisor Xen 4.0-amd64. Em uma dessas máquinas, foi instanciada

¹<http://iperf.sourceforge.net/>

²<http://www.tcpdump.org/>

uma máquina virtual, com um CPU virtual, 128 MB de memória, duas interfaces de rede e executando o sistema operacional Debian 2.6-32-5. A máquina virtual realiza a função de roteador. Os testes usam ainda duas outras máquinas, equipadas com processador Intel Core 2 Duo, que geram ou recebem pacotes, cada uma com uma placa de rede Ethernet de 1Gb/s, ligadas a uma rede de controle, e duas placas de redes Ethernet de 100Mb/s, para se comunicarem simultaneamente com os dois roteadores físicos. Os testes foram realizados com o roteador virtual encaminhando pacotes UDP de 64 e 1500 *bytes*, que são, respectivamente, o tamanho mínimo e máximo, respectivamente do conteúdo de um quadro Ethernet.

Durante a migração o plano de controle sofre um período de suspensão e este período não pode ser muito grande para que mensagens de controle de roteamento sejam perdidas. Assim, o primeiro teste objetiva medir o tempo de suspensão do plano de controle durante a migração. O teste foi realizado enviando-se pacotes de controle que passem pelo roteador virtual durante a migração. Durante o período de suspensão do plano de controle, para a cópia das últimas páginas de memória, verifica-se uma interrupção no encaminhamento desses pacotes de controle. O tempo de perda da conexão com o plano de controle é dado pela diferença do tempo do pacote de controle recebido imediatamente antes da migração com o tempo do pacote de controle recebido imediatamente após a migração. A Figura 7 mostra o tempo de suspensão do plano de controle para o sistema XenFlow e para a migração nativa do Xen, em função da taxa de pacotes enviada. Os resultados mostram que o tempo de suspensão do roteador virtual é próximo de zero no sistema XenFlow, independente do tamanho dos pacotes. Já na migração nativa do Xen, o tempo de suspensão do roteador virtual variou entre 12 e 35 segundos. Essa diferença se justifica por dois motivos principais. O primeiro é que, na migração usando o XenFlow, não há escrita de memória na máquina virtual, pois os pacotes são encaminhados diretamente pelo Domínio 0, ao passo que na migração do Xen, todos pacotes são encaminhados pela máquina virtual, gerando escritas e leituras de memória enquanto a máquina virtual é migrada. O maior uso da memória acarreta em mais páginas sujas e, portanto, no momento da cópia das últimas páginas, maior tempo de suspensão da máquina virtual. O segundo motivo é que, na migração do XenFlow, há uma etapa de migração de enlaces, realocando os fluxos nas máquinas geradora e receptora para as interfaces corretas. Na migração nativa do Xen, tal tarefa é realizada através do envio de pacotes de *ARP Reply*, para indicar em qual nova interface uma máquina virtual migrada está disponível. No entanto, o funcionamento do mecanismo de *ARP Reply* está condicionado ao vencimento da entrada ARP nas tabelas do sistema. Isso pode adicionar um atraso na atualização da interface que a máquina deve utilizar para se comunicar com a máquina migrada.

Não se pode iniciar uma nova migração enquanto uma migração anterior não estiver completada. Logo, o tempo total de uma migração estabelece o período mínimo de tempo entre duas migrações consecutivas de um mesmo elemento virtual.

O segundo teste realizado avalia o tempo total da migração. O tempo total da migração considera o tempo de execução de todas as operações referentes ao processo de migração. A Figura 8 apresenta os resultados do tempo total de migração em função da taxa de pacotes enviada. Os resultados demonstram que a migração no sistema XenFlow apresenta um acréscimo no tempo total que pode chegar até quatro vezes mais do que o tempo de migração de uma máquina virtual no Xen nativo, porque envolve mais etapas, sendo uma das etapas é a própria migração nativa do Xen. A Figura 8(b) mostra que para

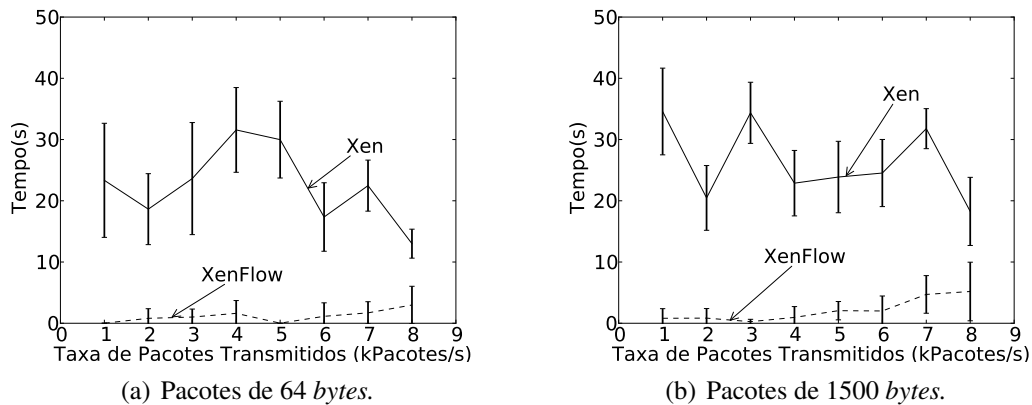


Figura 7. Tempo de suspensão do plano de controle durante a migração.

pacotes de 1500 bytes há um aumento no tempo total da migração no XenFlow, a medida em que a taxa de pacotes aumenta. O aumento, para pacotes de 1500 bytes, é porque há a saturação dos enlaces de 100 Mb/s a taxas de aproximadamente 8.000 pacotes/segundo.

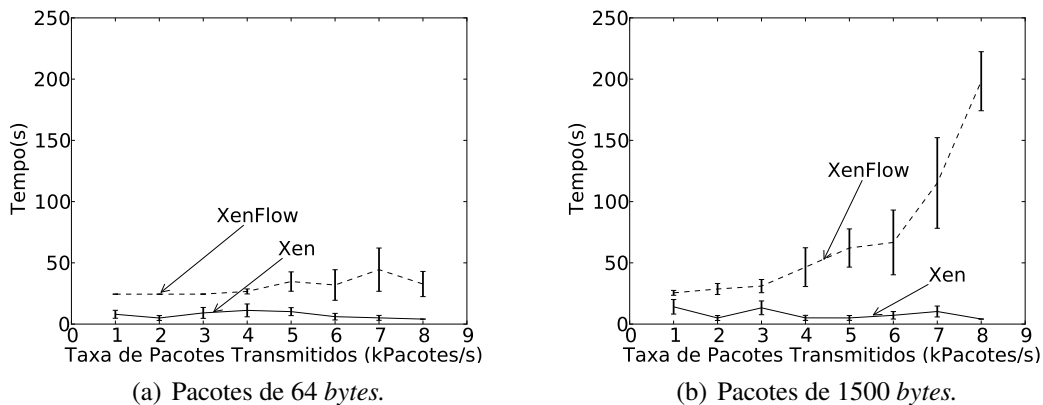


Figura 8. Tempo de total de duração do processo de migração.

Em migração de roteadores o fato de não haver perdas de pacotes é fundamental. O terceiro experimento apresenta o número de pacotes perdidos durante a migração convencional e a comprovação de perda zero no XenFlow. A Figura 9 revela que durante a migração no XenFlow não há perdas de pacotes. No entanto, uma característica intrínseca à migração é que pode ocorrer de os pacotes chegarem desordenados, pois os caminhos novo e antigo, respectivamente, o que passa pelo roteador físico de origem e o que passa de destino, podem possuir condições de atrasos diferentes. A Figura 9 mostra ainda que a perda zero de pacotes do sistema XenFlow é independente da taxa de pacotes encaminhados. Já a migração nativa do Xen apresenta perdas maiores para taxas maiores de pacotes enviados. Isso é reflexo do tempo de interrupção do serviço de encaminhamento, como visto na Figura 7. Como o tempo de interrupção do encaminhamento no Xen nativo é aproximadamente constante, a quantidade de pacotes perdidos nesse intervalo de tempo tende a aumentar proporcionalmente à taxa de pacotes encaminhados.

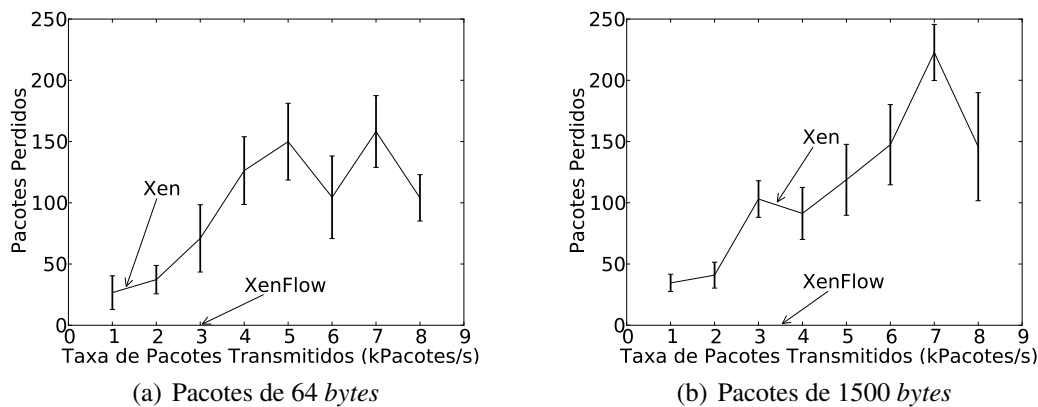


Figura 9. Número de pacotes perdidos em função da taxa de pacotes transmitida.

5. Trabalhos Relacionados

Uma proposta de arquitetura capaz de fornecer flexibilidade a redes, associada a baixos custos, é a Flowstream [Greenhalgh et al. 2009]. A arquitetura Flowstream baseia-se em módulos, implementados em máquinas virtuais, que processam os fluxos encaminhados por um plano de dados programável. Os módulos de processamento implementam aplicações de tratamento de pacotes, como *proxies* e *firewalls*, e estão interligados através do plano de dados programável. Os módulos de processamento, para os quais os fluxos são redirecionados, são selecionados dinamicamente. A tecnologia sugerida por Greenholgh *et al.* para a implementação do plano de dados programável é o OpenFlow. A arquitetura Flowstream aplica o conceito de processadores de fluxos. O XenFlow se diferencia do Flowstream porque, ao invés de executar aplicações genéricas de processamento de pacotes, nas máquinas virtuais, o XenFlow permite que cada administrador execute o seu próprio conjunto de ferramentas de controle, como protocolos de roteamento, de forma isolada, garantindo maior flexibilidade e segurança para as redes virtuais.

Existem propostas [Wang et al. 2007, Pisa et al. 2010] de realizar a migração da topologia lógica de forma transparente para as extremidades que utilizam a rede, sem que haja perda de pacotes ou quebra de conexão. No entanto, os cenários em que essas propostas são válidas são limitados. Em [Wang et al. 2007], supõe-se a existência de um mecanismo para a migração de enlaces que é externo ao mecanismo de migração de topologia. Supõe-se ainda que um roteador virtual só é migrado de uma máquina física para outra que tenha as mesmas adjacências ou que seja possível criar túneis entre elas. A proposta de Pisa *et al.* para a migração de roteadores virtuais baseados em Xen supõe a existência de uma rede local de controle sobre os elementos migrados e supõe o uso de túneis para a migração de enlaces ou que os roteadores físicos tenham conjuntos de adjacências compatíveis. Já a migração de fluxos na plataforma OpenFlow é fácil. Pisa *et al.* apresentam, também, um algoritmo que se baseia na redefinição de um caminho para fluxos na rede OpenFlow [Pisa et al. 2010]. Tal proposta apresenta perda zero de pacotes e baixa sobrecarga na rede. Entretanto, a proposta de migração de Pisa *et al.* para redes OpenFlow não é aplicável à virtualização de roteadores, pois é baseada em um controle de rede centralizado.

Existem, ainda, propostas para realizar o roteamento de pacotes sobre redes OpenFlow, como a arquitetura QuagFlow [Nascimento et al. 2010]. A ideia chave do Quag-

Flow é permitir que protocolos de roteamento convencionais, que funcionam de forma distribuída, controlem o encaminhamento de pacotes em uma rede OpenFlow. No Quag-Flow, o controlador emula a topologia física e cada um dos nós emulados executa o protocolo de roteamento convencional. Todas as mensagens de controle recebidas pelos comutadores reais são replicadas na rede emulada. As tabelas de roteamento geradas na rede emulada são, então, utilizadas pelo controlador para estabelecer as regras dos fluxos. O problema dessa abordagem é que o enlace do controlador com a rede passa a ser sobrecarregado com todas as mensagens de controle de roteamento de todos os nós da rede. O XenFlow evita esse problema porque a rede OpenFlow, ao invés de ter um controlador centralizado, possui um esquema de controle distribuído, evitando atrasos para configurar fluxos e também que o controlador seja um ponto único de falhas.

6. Conclusão

Este artigo propõe o XenFlow, um sistema de processamento de fluxos que provê uma forma robusta e eficiente de migração de topologias virtuais. O principal objetivo do XenFlow é realizar migrações de roteadores virtuais, com perda zero de pacotes e eliminar a necessidade de túneis ou mecanismos externos para a migração de enlaces. A proposta se baseia na introdução de um comutador OpenFlow na arquitetura da plataforma de virtualização Xen. O comutador passa a ser o elemento responsável por fazer o mapeamento das interfaces de rede dos roteadores virtuais em interfaces reais. Isso permite que a migração de roteadores virtuais seja feita para fora dos limites de uma rede local e sem perda de pacotes. O protótipo foi avaliado e os resultados mostram que o tempo de interrupção do plano de controle no sistema proposto chega a ser 30 vezes menor do que o tempo de interrupção na abordagem nativa do Xen. Os resultados também mostram que o tempo total de migração aumenta quando comparamos o sistema XenFlow com a migração nativa do Xen. Esse último resultado deve-se ao fato de o XenFlow introduzir novas etapas em relação à migração nativa do Xen. Contudo, o aumento do tempo total não é um fator significativo para a migração de roteadores virtuais e apenas estabelece o tempo mínimo entre duas migrações consecutivas. Os resultados obtidos confirmam que a migração de um roteador virtual sobre o sistema XenFlow ocorre sem perda de pacotes, o que torna o sistema adequado ao cenário de redes virtuais, ao contrário da migração nativa da plataforma de virtualização Xen.

Como trabalhos futuros, pretende-se desenvolver novas aplicações além de roteamento. Nesse novo contexto, as máquinas virtuais passam a prestarem serviços antes destinados à *middle boxes*, tais como balanceador de carga e *firewall*. Assim, o sistema XenFlow torna-se uma plataforma para a implementação de nós especializados, que podem ser migrados para entre diferentes pontos da rede.

7. Referências

- [Bolla et al. 2009] Bolla, R., Bruschi, R., Davoli, F., and Ranieri, A. (2009). Energy-aware performance optimization for next-generation green network equipment. In *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*, pages 49–54. ACM.
- [Clark et al. 2005] Clark, C., Fraser, K., Hand, S., Hansen, J., Jul, E., Limpach, C., Pratt, I., and Warfield, A. (2005). Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 273–286. USENIX Association.

- [Egi et al. 2008] Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Huici, F., and Mathy, L. (2008). Towards high performance virtual routers on commodity hardware. In *Proceedings of the 2008 ACM CoNEXT Conference*, pages 1–12. ACM.
- [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 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.
- [Greenhalgh et al. 2009] Greenhalgh, A., Huici, F., Hoerdt, M., Papadimitriou, P., Handley, M., and Mathy, L. (2009). Flow processing and the rise of commodity network hardware. *ACM SIGCOMM Computer Communication Review*, 39(2):20–26.
- [Gude et al. 2008] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. (2008). NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110.
- [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.
- [Moreira et al. 2009] Moreira, M., Fernandes, N., Costa, L., and Duarte, O. (2009). Internet do futuro: Um novo horizonte. *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC 2009*, pages 1–59.
- [Nascimento et al. 2010] Nascimento, M., Rothenberg, C., Salvador, M., and Magalhães, M. (2010). QuagFlow: partnering Quagga with OpenFlow. In *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, pages 441–442. ACM.
- [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.
- [Ratnasamy et al. 2005] Ratnasamy, S., Shenker, S., and McCanne, S. (2005). Towards an evolvable Internet architecture. *ACM SIGCOMM Computer Communication Review*, 35(4):313–324.
- [Sherwood et al. 2009] Sherwood, R., Gibb, G., Yap, K., Appenzeller, G., Casado, M., McKeown, N., and Parulkar, G. (2009). Flowvisor: A network virtualization layer. Technical report, Tech. Rep. OPENFLOW-TR-2009-01, OpenFlow Consortium.
- [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.
- [Wang et al. 2007] Wang, Y., van der Merwe, J., and Rexford, J. (2007). VROOM: Virtual routers on the move. In *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking*. Citeseer.