Projeto ReVir

Redes Virtualizadas
Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais
para Informação e Comunicação - CTIC

Relatório R_2

Desenvolvimento de uma rede de testes baseada em
técnicas de virtualização de redes
Tarefa T_3: Redes Xen, OpenFlow e Híbridas

Instituições

Coordenação
Universidade Federal do Rio de Janeiro - UFRJ
Universidade Estadual de Campinas - Unicamp
Universidade Federal de Pernambuco - UFPE
Universidade Federal do Rio Grande do Sul - UFRGS
Universidade Federal de São Carlos – UFSCar

Parcerias
Universidade Estadual do Rio de Janeiro - UERJ
Universidade de São Paulo – USP
Instituto Federal de Educação Tecnológica de Alagoas - IFET- AL
Universidade Federal do Paraná - UFPR
Universidade Federal Fluminense – UFF
Centro de Pesquisa e Desenvolvimento em Telecomunicações – CPqD
Universidade Federal do Espírito Santo – UFES

Índice

Resumo

O principal objetivo do projeto ReVir - Redes Virtualizadas - é prover redes virtualizadas com garantias de programabilidade, segurança, isolamento, controle e gerência de recursos, desempenho e qualidade de serviço. O foco reside no desenvolvimento e na avaliação de mecanismos para prover qualidade de serviço e gerenciar recursos das redes virtuais considerando esse modelo em camadas. O projeto aborda duas plataformas principais de virtualização, Xen e OpenFlow, além de disponibilizar um novo modelo de virtualização híbridopara tornar o núcleo da rede mais flexível.

Este relatório[1] apresenta os resultados obtidos na execução da Tarefa T_3 que visa a criação de um modelo híbrido de virtualização de redes baseado em Xen e em OpenFlow, chamado XenFlow, com suporte à migração de roteadores, comutadores e enlaces virtuais. A plataforma de virtualização Xen apresenta uma ferramenta nativa de migração de máquinas virtuais. No entanto, essa ferramenta é inadequada para o cenário de virtualização de roteadores, pois, durante a migração, ocorrem perdas de pacotes que reduzem a eficiência do roteamento. A migração de redes virtuais agrega flexibilidade às redes virtualizadas, mas é um grande desafio. O sistema XenFlow permite uma migração sem perda de pacotes tanto de roteadores quanto de enlaces virtuais. O sistema utiliza um mecanismo de encaminhamento de pacotes baseado em fluxos, que torna a funcionalidade de migração flexível, simples e robusta. O sistema permite a migração de roteadores virtuais, assim como o mapeamento simplificado de um enlace lógico sobre um ou mais enlaces físicos sem a utilização de túneis. A ideia chave baseia-se em um mecanismo híbrido que combina as técnicas de virtualização das plataformas Xen e OpenFlow, disponibilizando, ao mesmo tempo, um plano de dados programável e um controle distribuído da rede. Os resultados obtidos no protótipo P_1 mostram um desempenho superior do sistema, quando comparado com a migração convencional do Xen, além de permitir a migração de redes virtuais entre diferentes redes locais.

[1]Esse relatório se baseia na produção científica GTA-11-28 "XenFlow: Um Sistema de Processamento de Fluxos Robusto e Eficiente para Redes Virtuais".

Capítulo 1

Introdução

Atualmente a Internet conta com quase 2 bilhões de usuários em todo o mundo. O crescimento do número de usuários e a diversificação do uso da rede vêm desde o seu surgimento na década de 70. No início, os requisitos levantados para a Internet restringiam-se a uma rede universitária em que os usuários detinham conhecimentos técnicos e eram confiáveis. Hoje, o cenário é diferente, pois usuários espalhados por todo o mundo, com variados tipos de formação, têm acesso à Internet, criando um ambiente completamente heterogêneo e distinto do ambiente para o qual foram levantados os requisitos do projeto inicial da Internet.

O sucesso da Internet é baseado em dois pilares, o serviço de transferência fim-a-fim e a pilha de protocolos TCP/IP [1, 2]. Na arquitetura atual da Internet, a inteligência da rede está localizada nos sistemas de extremidades da rede, enquanto o núcleo é simples e transparente. Embora, essas sejam as razões do sucesso da Internet, paradoxalmente, também são as razões para o seu engessamento. Os nós do núcleo da rede são simples e não fornecem informações sobre o funcionamento da rede. Isso implica que o usuário fique frustrado quando algo não funciona, pois ele não obtém da rede informação de onde se encontra o erro [2]. Outras consequências do núcleo simples e transparente é que há uma grande sobrecarga de configurações manuais, depuração de erros e projeto de novas aplicações. Outra limitação da Internet atual é que o modelo TCP/IP apresenta alguns problemas estruturais que são difíceis de serem resolvidos, tais como escalabilidade, mobilidade, gerenciamento e segurança [3].

1.1 Evolução da Internet

A estrutura da Internet vem se modificando, ao longo dos anos, através de "remendos". Essas modificações foram introduzidas na rede para atender novas demandas e requisitos que não estavam previstos no projeto original. Assim, a Internet foi sendo "remendada" com a criação das sub-redes, dos sistemas autônomos, do serviço de nome de domínios (DNS - *Domain Name Service*), do *Classless Inter-Domain Routing* (CIDR) [2], entre outros. Esses "remendos" foram criados para fornecer escalabilidade à rede. Já o TCP sofreu modificações para introdução do controle de congestionamento, pois com o crescimento acelerado da rede, a Internet chegou a passar por uma série de colapsos devido a congestionamentos. O protocolo IP também sofreu "remendos". Alguns "remendos" no IP foram a criação do IP *multcasting*, que permite que uma estação envie dados para um grupo de estações; o IPv6, que tinha como objetivo aumentar o número de endereços disponíveis, simplificar o cabeçalho IP, melhorar o suporte para opções, permitir a identificação de fluxos, inserir mecanismos de autenticação e de privacidade na camada IP; a criação do *Networking Address Translation* (NAT), que quebra o princípio do endereçamento global único, pois um conjunto de estações passam a ser endereçado por um único endereço válido; o IPsec, que introduz princípios de segurança na camada IP; e o IP móvel, que opera através de túneis para fornecer mobilidade às estações na camada IP. Embora esses remendos tenham sido adotados para atender novas demandas para a Internet, a introdução de novas mudanças encontra dificuldades e a rejeição dos provedores de serviços. Os provedores de serviços não se arriscam a implementar novos serviços que possam indisponibilizar, mesmo que temporariamente, o uso da rede, ou que não sejam garantidamente seguros e robustos. Sendo assim, a evolução através de "remendos" já se mostra precária para alguns cenários.

A necessidade de desenvolver novos "remendos" para a Internet indica que o projeto original da Internet não se adéqua mais ao cenário atual. Dessa forma, já existem propostas para uma nova arquitetura para a Internet que promovem a flexibilidade e o suporte à inovação no núcleo da rede. Os modelos de Internet do Futuro são, então, divididos em duas abordagens, a purista e a pluralista [1, 4, 5].

1.2 Arquiteturas para a Internet do Futuro

A abordagem purista, Figura 1.1, modela a Internet através de uma arquitetura monolítica. Essa abordagem define a Internet do Futuro como uma

única rede, flexível o suficiente para atender todas as demandas e requisitos presentes e futuros. Essa arquitetura é semelhante à atual arquitetura da Internet. No entanto, a diferença entre a arquitetura purista e a atual são os protocolos que executam. Os protocolos da nova arquitetura devem ser mais flexíveis e adaptáveis a novos requisitos e demandas do que a pilha de protocolos TCP/IP. Já a abordagem pluralista, Figura 1.2, baseia-se na ideia de que a Internet deve ser capaz de dar suporte a múltiplas pilhas de protocolo executando simultaneamente. Comparando as duas abordagens, a purista apresenta-se como mais complexa do que a pluralista. A complexidade da abordagem purista reside no fato de que essa proposta para a Internet do Futuro prevê o projeto de uma nova arquitetura de rede e novos protocolos de comunicação que sejam capazes de resolver todos os problemas atualmente conhecidos e outros problemas que ainda nem são conhecidos. A abordagem pluralista, por sua vez, é mais simples, pois ao permitir que diversas redes possam executar em paralelo, permite que diferentes redes sejam estabelecidas para atender aos requisitos de cada nova aplicação. Outra vantagem da abordagem pluralista é que a sua implementação pode se dar de forma gradual, pois tal abordagem é intrinsecamente compatível com a Internet atual. A compatibilidade é alcançada executando-se a pilha de protocolos TCP/IP em paralelo com as demais redes em uma arquitetura pluralista. Dessa forma, a arquitetura pluralista é mais fácil de ser projetada do que uma arquitetura purista, monolítica, em que os protocolos previamente definidos têm que dar suporte a aplicações cujos requisitos ainda nem são conhecidos.

Todas as abordagens pluralistas baseiam-se na mesma ideia de executar múltiplas redes virtuais sobre um substrato físico compartilhado [1]. No entanto, as propostas de redes pluralistas diferem no formato dos pacotes, no esquema de endereçamento e nos protocolos que executam, mas em todas, embora as redes virtuais compartilhem o mesmo substrato, cada pilha de protocolo executada é independente das demais.

1.3 Modelo Econômico da Internet do Futuro

Os dois maiores atores na Internet atual são os provedores de serviços e os provedores de serviço de Internet (*Internet Service Provider* – ISP) [6]. Nesse sentido, os ISPs fornecem acesso à Internet aos seus clientes, através de infraestrutura própria, através do aluguel de infraestrutra de outros ISPs ou através da combinação dessas duas situações [7, 6]. Já os provedores de serviço oferecem apenas serviços na Internet, como por exemplo, o Google. Sendo assim, os ISPs fornecem basicamente o serviço de conectividade, enquanto os provedores de serviço fornecem serviços na Internet.

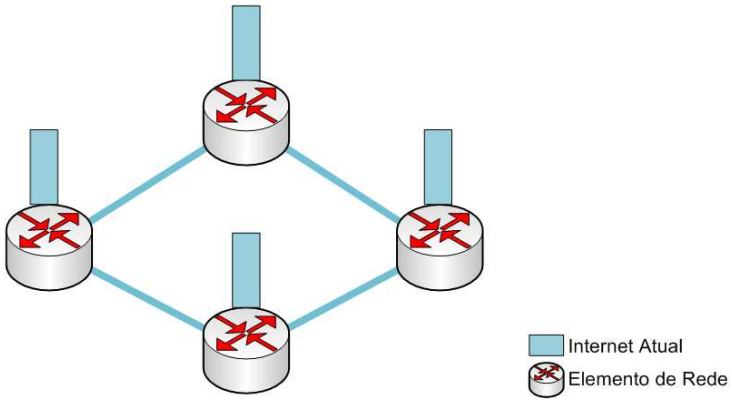


Figure 1.1: Arquitetura Purista. Apenas uma pilha de protocolos executa sobre a infraestrutura física. Arquitetura atual.

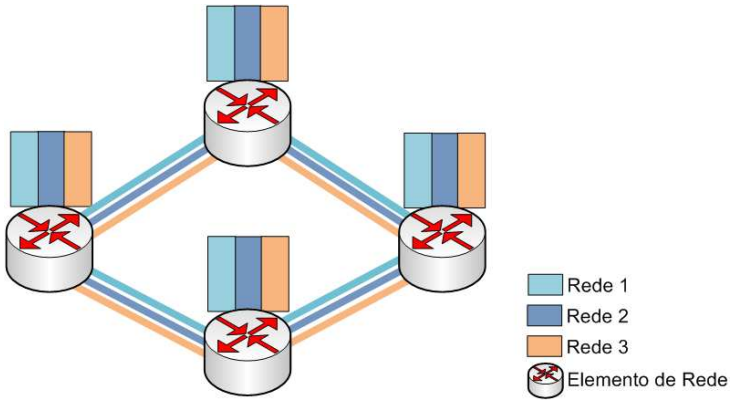


Figure 1.2: Arquiteura pluralista. Diferentes pilhas de procolo executam sobre a mesma infraestrutra física.

Os provedores de serviço de Internet estão sob pressão para fornecerem serviços com cada vez mais valor agregado, em resposta às demandas dos seus clientes e a popularização do acesso à Internet [7]. No entanto, para que os serviços providos por um ISP alcancem toda a rede, é necessário que o ISP confie na conectividade de outros ISPs, já que um único ISP raramente controla todo o caminho entre os clientes e os serviços. Dessa forma, novos serviços são implementados em pequenas ilhas dentro de um único ISP ou são fornecidos para fora do ISP completamente degradados. Alguns ISPs chegam ao extremo de degradarem alguns serviços, enquanto fornecem melhor qualidade a outros serviços mais lucrativos.

Concorrentemente a atividade dos ISPs, pesquisadores desenvolvem "remendos" para a Internet, de maneira isolada, testando cada um de forma

independente dos demais. Dessa forma, o desenvolvimento da Internet fica voltado para mudanças incrementais que não exijam compromisso de atualizações de todos os ISPs de uma única vez. Assim, propostas que não são possíveis de serem implementadas de forma incremental, frequentemente são deixadas de lado. No entanto, esse modelo de evolução da rede não necessariamente leva à melhora da arquitetura da Internet atual, pois, embora cada mudança isolada faça sentido, o estado final, após a implementação de todas, pode ser inconsistente. Sendo assim, este fracasso de não conseguir avanços na tecnologia da Internet pela falta de interesse dos ISPs em promove-los é também um aspecto que leva a necessidade de mudanças radicais na arquitetura da Internet.

Além dos ISPs e dos provedores de serviços, o cenário atual da Internet ainda apresenta outros dois participantes principais no modelo de negócios da Internet: o provedor de infraestrutura física (*Physical Infrastructure Provider - PIP*), o qual controla e gerencia a infraestrutura da camada física, chamada de substrato; e o provedor de conectividade, que provê conectividade fim-a-fim para os usuários finais. No entanto, a distinção entre esses atores não fica clara, pois todos são representados geralmente por uma única empresa. Mesmo dentro de um ISP, essa divisão não é tão clara, pois, por exemplo, a equipe que é responsável pela operação diária da infraestrutura física, no geral, é a mesma que planeja e especifica a evolução da rede [6]. Sendo assim, a partir da identificação dessas diversas funções, surgem propostas de novos modelos de negócio para a Internet do Futuro, que separam cada função em um ator diferente.

Uma primeira proposta para o modelo de negócios da Internet do Futuro é desacoplar os provedores de infraestrutura dos provedores de serviço [7]. Assim, os provedores de infraestrutura seriam os responsáveis por implementar e manter os equipamentos de rede, fornecendo a conectividade dos clientes à rede, enquanto os provedores de serviço seriam os responsáveis por implementar protocolos e oferecer os serviços fim-a-fim. A proposta CABO (*"Concurrent Architectures are Better than One"*) explora a virtualização para permitir que um provedor de serviços execute múltiplos serviços fim-a-fim sobre equipamentos de rede de diferentes provedores de infraestrutura [7]. Essa proposta alinha-se com a abordagem pluralista para a Internet do Futuro e reconsidera o modelo de negócios da Internet atual, para que os provedores de serviço possam contratar um ou mais provedores de infraestrutura para construir as suas redes virtuais que fornecem os serviços fim-a-fim [7].

Outra proposta de modelo de negócios para a Internet do Futuro baseia-se no tipo de virtualização de redes usado para fornecer uma arquitetura pluralista [6]. Assim, a virtualização permite a existência de vários serviços em paralelo e, portanto, introduz uma nova camada de abstração, a camada de

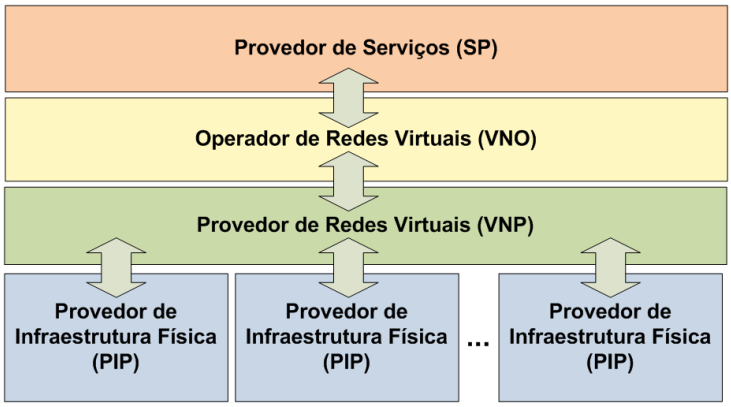


Figure 1.3: Esquema de um novo modelo de negócios com as camadas de provisão e operação de redes virtuais representando novos atores econômicos.

definição de redes virtuais. A redefinição das funções de cada ator e a criação de novas funções nesse modelo de negócios leva às funções [6] mostradas na Figura 1.3 e descritas como:

- **Provedor de Infraestrutura Física** (*Physical Infrastructure Provider* - PIP): Detém a propriedade e gerencia o substrato físico. Vende serviço de processamento de dados, sob a forma de fatias(*slices*) de rede, para os provedores de rede virtual;
- **Provedor de Rede Virtual** (*Virtual Network Provider* - VNP): É responsável por agregar os recursos virtuais de um ou vários provedores de infraestrutura física, formando uma topologia virtual, de acordo com os requisitos dos operadores de rede virtual;
- **Operador de Rede Virtual** (*Virtual Network Operator* - VNO): É responsável pela instalação e operação das redes virtuais, sobre a topologia virtual, de acordo com os requisitos dos provedores de serviço. Oferecem serviços de conectividade fim-a-fim em sua rede ou oferecem as suas redes virtuais para provedores de serviços;
- **Provedor de Serviço** (*Service Provider* - SP): Usa a rede virtual para oferecer o seu serviço. Dependendo do serviço oferecido, o SP pode agir como um servidor de aplicação, caso o serviço apresente grande valor agregado, ou como um provedor de serviço de rede, caso o serviço oferecido seja um serviço de transporte de dados.

A princípio, todas as funções podem ser exercidas por uma mesma empresa. No entanto, cada função depende de uma equipe específica para a

sua execução. Nesse novo modelo de negócios, um servidor de infraestrutura pode ter em seu substrato físico diversas aplicações sendo fornecidas por diferentes SPs, de forma completamente isolada e independente.

Schaffrath e outros [6] descrevem três oportunidades de negócio que já existem, mas que só são viabilizadas pela separação das funções de cada ator no novo modelo de negócios da Internet do Futuro. Considerando três atores nesse cenário, Ator A, Ator B e Ator C, esses podem se organizar de diversas formas para fornecer um serviço para o usuário final. Alguns exemplos são os seguintes:

- O Ator A opera somente como provedor de infraestrutura, então ele só opera fazendo o tratamento dos dados na camada física. Já o Ator C decide focar-se no serviço de prover uma aplicação, terceirizando todos os aspectos operacionais. Nesse caso, surge a oportunidade de negócio para o Ator B, que pode oferecer o serviço de terceirização dos aspectos operacionais para o Ator C. O serviço provido pelo Ator B a ser oferecido ao Ator C compreende os serviços de operação e provimento de redes virtuais e, portanto, o Ator B compra as fatias de rede do provedor de infraestrutura que é o Ator A;

- O Ator A opera o seu próprio provedor de infraestrutura, mas também age como provedor de rede virtual. Assim, o Ator A monta uma rede virtual consistente através do uso de seus próprios recursos de infraestrutura e contratando serviços de infraestrutura de outros provedores de infraestrutura física (PIPs). O Ator B atua como operador de rede virtual, enquanto o Ator C opera como provedor de serviços, dedicando-se ao desenvolvimento de aplicações;

- Outro cenário é o Ator C operando como provedor de serviços, operador de rede virtual e provedor de rede virtual. Nesse caso, o Ator C só não detém a infraestrutra física, que por sua vez, pode ser contratada dos provedores de infraestrura, atores A e B.

1.4 Inovação na Internet do Futuro

A implantação de novos protocolos e serviços no núcleo da Internet atual sofre a rejeição de parte dos provedores de serviços devido ao grande risco que essas mudanças podem representar para o bom funcionamento da rede. Uma das propostas para conciliar o desenvolvimento de inovações e tráfego de produção é a virtualização de redes [7, 8], pois, nesse ambiente, o substrato físico é compartilhado por diferentes redes virtuais, isoladas entre si. Como

as redes virtuais são isoladas, o tráfego experimental não influencia o tráfego de produção, como mostrado na Figura 1.4. Além disso, o modelo de redes virtuais paralelas é apontado como uma das principais abordagens para a Internet do Futuro [2], na qual cada rede virtual possui a sua própria pilha de protocolos e arcabouço de gerenciamento. As redes virtuais devem ser totalmente isoladas e, portanto, o funcionamento de uma rede não deve afetar o funcionamento das outras redes. No entanto, no contexto de redes virtuais, novas primitivas de gerenciamento são necessárias para mapear a topologia lógica sobre a topologia física da rede.

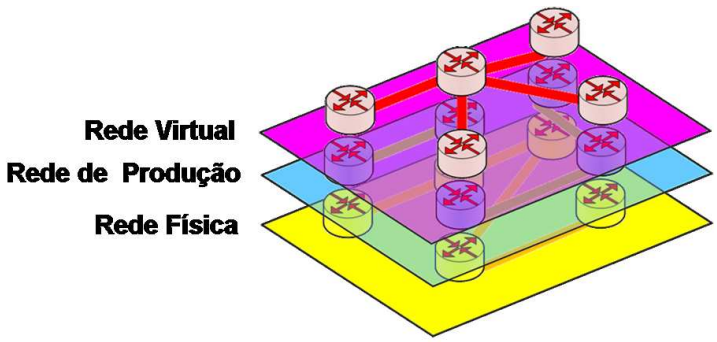


Figure 1.4: Exemplo de virtualização de redes, no qual três redes virtuais isoladas compartilham o mesmo substrato físico.

Uma funcionalidade de controle fundamental de redes virtuais é a migração de redes virtuais, a qual permite a movimentação de nós de redes virtuais sobre os nós da rede física [9]. A funcionalidade de migração de redes virtuais pode ser usada em diversos contextos, desde a realocação da rede lógica sobre a física, como por exemplo, para a manutenção de nós da rede, até no provimento de uma rede verde que procura diminuir o consumo de energia reduzindo o número de nós físicos ativos em uma rede. Operações de manutenção de nós de rede frequentemente requerem o seu desligamento, o que modifica a topologia lógica da rede e, consequentemente, gera a quebra de conexão e, no caso da manutenção de roteadores, gera a perda das adjacências dos protocolos de roteamento. Dessa forma, introduz-se um atraso para que as rotas sejam reorganizadas em todos os nós. A primitiva de migração de redes virtuais garante que a topologia lógica não é alterada e, assim, as rotas continuam válidas. 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 [10]. A migração de redes virtuais pode ainda ser usada para evitar danos no caso de ataques de negação de serviço (DoS – *Denial of Service*), no qual as demais redes virtuais, que compartilham o mesmo substrato com a rede sob ataque, são migradas para outros

nós físicos fora da região do ataque. Outro cenário, em que a migração de redes pode ser utilizada, é na implementação de processadores de fluxo [11], que utilizam a técnica de virtualização de redes para fornecer *middle boxes*[1] Neste caso, a migração fornece aos processadores de fluxo a possibilidade de remanejar *middle boxes*, dinamicamente, sem a interrupção do serviço.

No entanto, a migração de redes virtuais ainda apresenta grandes desafios, como o remapeamento dos enlaces virtuais sobre um ou mais enlaces físicos, a redução do tempo em que os nós virtuais migrados ficam indisponíveis durante o processo de migração e a possibilidade de perda de pacotes no núcleo da rede durante o processo de migração.

Este relatório apresenta o XenFlow, um sistema de processamento de fluxos que permite a criação de redes virtuais programáveis que suportam migração de redes virtuais sem perdas de pacotes e sem interrupção de serviços na rede. O XenFlow realiza tanto a migração de nós virtuais, quanto a migração de enlaces virtuais. No caso de o nó virtual agir como um roteador, chamado de roteador virtual, as funções de roteamento são divididas em dois planos, o plano de controle e o plano de dados. O plano de controle é responsável por funções de controle da rede, como por exemplo, a execução do protocolo de roteamento para descobrir as rotas. O plano de dados é responsável pelo encaminhamento dos pacotes, de acordo com as políticas de controle, como as definições de rotas, as políticas de filtragem de pacotes e as políticas de prioridade. A principal contribuição do sistema XenFlow consiste na proposta de um sistema de virtualização de redes que se baseia em uma abordagem híbrida de virtualização, que combina as plataformas de Xen[2][12] e OpenFlow[3][13]. O sistema proposto provê a programabilidade em um ambiente de rede virtual distribuído e permite a realização da migração de redes virtuais para fora dos limites de uma rede local. A proposta não limita a rede virtual a uma rede local, pois a interconexão dos elementos de rede virtual é programável e pode ser definida dinamicamente. Assim, o plano de controle é implementado de forma distribuída em máquinas virtuais Xen, nas quais executam os protocolos de controle da rede, por exemplo, os de roteamento. As máquinas virtuais Xen comunicam-se com a rede física através de um comutador OpenFlow, que realiza o encaminhamento dos pacotes e implementa o plano de dados compartilhado em cada nó físico.

As principais vantagens da técnica de virtualização de redes proposta são

[1] *Middle boxes* são dispositivos na rede que provêem a execução de políticas de transporte de forma transparente. São exemplos de *middle boxes*: *firewalls*, *proxies* e dispositivos NAT.

[2] Xen é uma plataforma de virtualização para computadores pessoais.

[3] OpenFlow é uma plataforma de virtualização de redes baseado na definição e comutação de fluxos.

duas. De um lado, a proposta elimina a deficiência da plataforma de virtualização Xen, que, na migração de um roteador virtual, limita o local de destino do roteador virtual a roteadores físicos conectados na mesma rede local que o roteador físico de origem. De outro lado, a proposta não fica restrita a centralização do controlador das redes comutadas OpenFlow e é possível programar os elementos de encaminhamento de forma distribuída. No XenFlow, ao realizar a migração de redes virtuais, pode-se agora mapear um enlace lógico sobre um ou mais enlaces físicos como é mostrado no Seção 1.6. Essa facilidade permite que a migração de um roteador virtual ocorra entre diferentes redes físicas, sem ser limitada a migrações no interior de uma mesma rede local como em outras propostas existentes [9, 14, 15]. No sistema XenFlow, a migração dos elementos de rede virtual ocorre sem perda de pacotes.

Um protótipo do sistema foi construído para a validação da proposta. Os resultados experimentais mostram que o sistema proposto é robusto, pois durante a migração não há perda de pacotes nem a interrupção do serviço de encaminhamento de pacotes. O sistema também é eficiente, já que permite a execução da migração de roteador e enlace virtuais sem que haja a perda da conexão ou atraso 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 cerca de 100 pacotes durante cada migração e apresentou um período de interrupção na atualização do plano de controle de até 40 vezes maior do que o XenFlow.

1.5 Sistema XenFlow na Internet do Futuro

O sistema XenFlow enquadra-se na camada do Provedor de Infraestrutura, pois visa à criação de um modelo híbrido de virtualização de redes baseado em Xen e em OpenFlow. O principal objetivo desse modelo híbrido é o desenvolvimento de um núcleo de rede com controle distribuído e amplo suporte à migração de elementos de rede virtuais. Assim, propõe-se uma arquitetura de virtualização flexível com controle distribuído, na qual é possível a migração eficiente dos enlaces e dos elementos de rede virtuais. A arquitetura proposta prevê que o controle de cada rede virtual seja feito de forma descentralizada e que o controle da migração seja feito pelo provedor de infraestrutura de forma centralizada, possibilitando inclusive a migração de elementos de rede virtuais entre máquinas físicas que não pertençam ao mesmo provedor de infraestrutura.

1.6 Redes Virtuais e Migração

O conceito de virtualização é definido, nesse trabalho, como a criação de uma camada de abstração, que permite que recursos sejam compartilhados por ambientes virtuais, também chamados de "fatias" dos recursos físicos compartilhados. A camada de virtualização é geralmente implementada em *software* e permite que os ambientes virtuais tenham acesso a interfaces similares às reais [1]. A técnica de virtualização permite, assim, que o uso de um recurso, por uma camada superior, seja desacoplado do recurso real. Essa técnica tem sido largamente utilizada para a virtualização de computadores. Em uma das principais abordagens para a virtualização de computadores, o recurso a ser compartilhado é o *hardware* [12, ?, 16, 17]. Nesse caso, os ambientes virtuais, chamados de máquinas virtuais, acessam interfaces similares às de *hardware* convencional e, assim, têm a impressão de executarem diretamente sobre o *hardware* físico. Para ter a impressão de executar diretamente sobre o *hardware* físico as máquinas virtuais devem ser ambientes isolados, ou seja, a execução de uma máquina virtual não interfere na de outra, e cada máquina virtual deve ter acesso aos recursos físicos come se fosse a única a ter acesso a tais recursos. Analogamente, a virtualização de redes também faz a abstração de um recurso, que é compartilhado por diversas fatias virtuais. No entanto, na virtualização de redes, o recurso compartilhado é a infraestrutura física de rede.

Em um cenário de rede virtual, múltiplas redes coexistem, compartilhando o mesmo substrato físico, através da técnica da virtualização. Dessa forma, uma rede virtual é uma rede composta pela interconexão de um conjunto de roteadores virtuais, os quais são "fatias" de roteadores físicos compartilhados. A maneira mais imediata de visualizar uma rede virtual é uma rede lógica, composta de roteadores virtuais, sobreposta a uma rede física, ou seja, a topologia da rede lógica corresponde à topologia da rede física. No entanto, a topologia da rede lógica não precisa ser idêntica à topologia da rede física, embora seja essencial uma rede física pra veicular os dados. Nesta abordagem mais abrangente, uma rede virtual é composta por roteadores lógicos conectados em uma determinada topologia, através de enlaces virtuais [18]. Os enlaces virtuais são criados através do particionamento dos enlaces físicos. O particionamento é realizado através da divisão da banda disponível em cada enlace entre os enlaces virtuais. Os enlaces virtuais também podem ser estabelecidos através de túneis, quando roteadores virtuais, que são adjacentes na topologia lógica, não estão hospedados em roteadores físicos adjacentes. Assim, a topologia lógica da rede virtual pode não corresponder exatamente a topologia física e roteadores lógicos adjacentes são conectados por túneis. Esta característica de poder mapear a rede

lógica na rede física permite que as redes virtuais possuam uma grande flexibilidade e que a funcionalidade de migração de redes virtuais assuma um papel preponderante neste tipo de redes.

A virtualização de redes agrega flexibilidade às redes reais, pois permite a instanciação, a remoção e a configuração de recursos de redes virtuais sob demanda, assim como, também, permite que as redes virtuais sejam monitoradas e migradas, enquanto estão ativas. Como consequência, a tecnologia de virtualização de redes tem sido amplamente usada para o desenvolvimento de propostas para a Internet do Futuro [2] e para o desenvolvimento de redes experimentais [19]. Nesse sentido, existem diversas iniciativas na comunidade científica voltadas para o desenvolvimento de redes virtuais capazes de fornecerem ambientes realísticos, programáveis e controlados para implementarem e testarem novos protocolos, serviços e arquiteturas de rede [20].

Uma das principais iniciativas de se desenvolver uma rede virtual para a experimentação é o PlanetLab [19]. O PlanetLab é uma rede que interconecta nós de diversas instituições por todo o mundo, através da Internet. O objetivo do PlanetLab é prover uma rede sobreposta a rede IP em que os pesquisadores possam desenvolver testes em escala real. Dessa forma, o compartilhamento da rede de testes é provido pela virtualização da rede. Nesse contexto, uma fatia da rede de testes corresponde a um conjunto de máquinas virtuais que, por sua vez, são hospedadas pelos nós físicos. Sendo assim, a virtualização de redes no PlanetLab é alcançada através do uso da técnica de virtualização, associada ao uso de uma rede sobreposta, de forma que cada máquina virtual execute a função de um nó na rede, em uma rede de larga escala. Sobre a plataforma de experimentação do PlanetLab, desenvolveu-se o VINI [21], uma infraestrutura de rede virtual que permite aos pesquisadores testarem protocolos e serviços em ambientes realistas, provendo alto grau de controle sobre as condições da rede. O VINI permite que os pesquisadores implementem e avaliem suas ideias em *softwares* de roteamento, com tráfego real e com eventos de rede reais, como falhas e congestionamento nos enlaces.

Embora a técnica de virtualização de redes seja amplamente utilizada para prover a conectividade, esse modelo também traz algumas desvantagens [18]. Os principais problemas do modelo estão no fato de o gerenciamento de uma rede virtual ser muito similar ao de uma rede real e a banda dos enlaces virtuais ser limitada. Outra limitação do uso de redes virtuais é o mapeamento de um roteador virtual para um roteador físico, pois a falha de um componente físico é refletida na rede virtual que usa esse componente.

1.6.1 Programabilidade em Redes

As principais propostas para a Internet do Futuro baseiam-se em redes capazes de serem programadas sob demanda, chamadas redes programáveis, o que agrega flexibilidade para os requisitos atuais e futuros da Internet. Uma das formas de se prover programabilidade às redes é através da implementação de redes definidas por *software*. Redes definidas por *software* são redes cujo substrato físico é composto por equipamentos de propósito geral e a função de cada equipamento, ou conjunto de equipamentos, é realizada por um *software* especializado. O conceito de redes definidas por *software* [11, 13, 12] agrega programabilidade às redes, com baixo custo, pois esse conceito combina *hardware* de propósito geral, como os de computadores pessoais, a *softwares* especializados de redes [11]. Na abordagem de redes definidas por *software*, os elementos de rede são programáveis e maior controle é oferecido à gerência da rede. Paralelamente, dispositivos de *hardware* especializados são substituídos por *hardwares* comuns, assistidos por *softwares* especializados. Essa substituição só é possível, devido aos desenvolvimentos recentes do *hardware* padrão de mercado, como o desenvolvimento de arquiteturas de múltiplas CPUs, com múltiplos núcleos, além do desenvolvimento de tecnologias de interconexão de sistemas de altas velocidades [12].

Outra tendência no desenvolvimento de redes programáveis é a necessidade de se definir múltiplas redes isoladas, compartilhando o mesmo substrato físico. Essa tendência é representada pela virtualização de redes. A existência de múltiplas instâncias virtuais compartilhando o mesmo substrato físico é possível, pois a virtualização de redes provê a separação entre o recurso virtualizado e a camada que usa tal recurso [19, 1]. Dessa forma, há propostas para prover a programabilidade em redes, de acordo com as propostas de redes definidas por *software*, associadas à garantia de isolamento da virtualização de redes. Nesse sentido, uma proposta que agrega a programabilidade, provida pelas redes definidas por *software*, com o isolamento entre redes, provido pela virtualização de rede, é o OpenFlow [13]. O OpenFlow é uma plataforma de virtualização de redes baseada na comutação de fluxos. Um comutador OpenFlow encaminha os pacotes de acordo com regras definidas por um controlador centralizado. No entanto, há outras propostas para a implementação de redes definidas por *software* e para a virtualização de redes. Em outros trabalhos [19, 1, 12, 9], os autores defendem uma arquitetura de rede virtual em que os elementos de rede virtual são roteadores, que por sua vez executam sobre uma plataforma de virtualização de computadores pessoais. Nesse cenário, os roteadores virtuais são máquinas virtuais que executam funções de roteamento. Sendo assim, existem duas correntes

para prover a programabilidade em redes na Internet do Futuro. A primeira é representada pelo OpenFlow, uma tecnologia promissora, que provê alto desempenho e controle da rede, associados às desvantagens de não permitir o processamento por pacote e ter o controle centralizado. A outra corrente é representada pelas propostas de arquiteturas de rede baseadas em roteadores virtuais, sobre tecnologias de virtualização de computadores pessoais [1, 12, 9]. Roteadores virtuais agem de maneira similar aos roteadores físicos convencionais, mas o desempenho dos roteadores virtuais é inferior ao dos roteadores físicos convencionais e, além disso, os roteadores virtuais apresentam problemas de isolamento nas operações de entrada e saída [1].

Uma proposta para a virtualização do plano de controle da rede OpenFlow é o FlowVisor [22]. O FlowVisor executa a virtualização do plano de controle da rede, ou seja, permite que mais de um controlador controle uma mesma rede OpenFlow, de forma que cada controlador acredite ser o único exercer o controle sobre a rede. A segmentação da rede em áreas de atuação de cada plano de controle virtual, chamada de fatiamento da rede, é baseada em políticas estáticas e permite a definição de um controlador para cada segmento, ou fatia, da rede. Dessa forma, o controle de uma rede física é realizado por diversos controladores, embora o controle de cada rede virtual é centralizado em um único controlador daquela fatia.

Quanto a distribuição do controle na rede OpenFlow, existem duas propostas principais a DIFANE [23] e o HyperFlow [24]. A arquitetura DIFANE [23] consiste de um controlador que gera regras coringas e as aloca em comutadores especiais, chamados de autorização. Assim, no momento em que um pacote entra em um comutador da rede, esse comutador verifica se o pacote se adéqua a alguma regra já estabelecida. Em caso positivo, o pacote é encaminhado por essa regra. Em caso negativo, o pacote, ou sequência de pacotes, é encaminhado para o comutador de autorização mais próximo. A paritr das regras coringas definidas pelo controlador, nos comutadores de autorização, são geradas regras para o encaminhamento dos pacotes que foram enviados aos comutadores de autorização e essas regras são instaladas nos comutadores de entrada do pacote na rede. Já na arquitetura HyperFlow [24], a distribuição do controle é realizada por uma aplicação que executa sobre o controlador. Cada instância do controlador é responsável por um conjunto de comutadores da rede OpenFlow. Cada comutador é controlador por apenas um dos controladores. Os controladores, por sua vez, executam o mesmo conjunto de aplicações de controle. A aplicação HyperFlow é responsável por capturar os eventos gerados na rede, controlada por uma dada instância do controlador, e os divulga para os demais controladores da rede OpenFlow através do paradigma de *publish/subscribe*. Para realizar a divulgação, a aplicação HiperFlow usa um sistema de arquivos distribuído, no

qual o HiperFlow define um diretório como canal de divulgação de eventos e as mensagens de divulgação como arquivos. A partir de então, toda a tarefa de divulgação dos eventos é realizada pelo sistema de arquivos, enquanto a o HyperFlow trata do recebimento e entrega dos eventos para as aplicações afetadas por cada evento.

Uma proposta de arquitetura capaz de fornecer flexibilidade a redes, associada a baixos custos, é a Flowstream [11]. 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 e outros para a implementação do plano de dados programável é o OpenFlow. Assim, a arquitetura Flowstream aplica o conceito de processadores de fluxos.

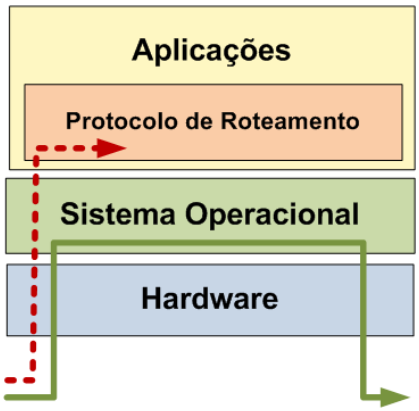
1.6.2 Migração de Redes Virtuais

A virtualização de redes introduz uma nova primitiva de gerenciamento, a migração de redes virtuais [9]. Existem propostas [25, 14] de realizar a migração de nós virtuais de forma transparente para as extremidades que utilizam a rede, sem alterar a topologia lógica e sem que haja perda de pacotes ou quebra de conexões. No entanto, os cenários, em que essas propostas são válidas, são limitados. Wang e outros [25] assumem a existência de um mecanismo para a migração de enlaces que seja externo ao mecanismo de migração de nós virtuais. Assume-se ainda que um roteador virtual só pode ser migrado de uma máquina física de origem para outra máquina física de destino que tenha as mesmas adjacências, ou seja, que esteja no mesmo barramento da máquina física de origem, ou que seja possível criar túneis entre elas. A proposta de Pisa e outros [14] apresenta outras limitações. Por ser baseada na migração nativa do Xen [15], essa proposta está sujeita à limitação de os nós virtuais só poderem ser migrados para máquinas físicas dentro de uma mesma rede local. A migração assume que exista uma rede local de controle sobre os elementos migrados e assume-se 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 menos complexa. Pisa e outros apresentam, também, um algoritmo que se baseia na redefinição de um caminho para fluxos em uma rede OpenFlow [14]. Tal proposta apresenta perda zero de pacotes e baixa sobrecarga na rede [26]. No entanto, esta proposta de migração para redes OpenFlow não é aplicável à virtualização de roteadores ou a sistemas que implementem

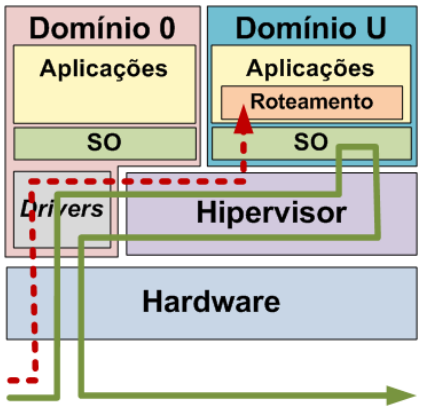
o conceito de processadores de fluxos, pois, nesses casos, o encaminhamento, ou processamento, dos pacotes não se baseia em regras estáticas definidas no ambiente virtual, como no caso da migração de fluxos. A migração de roteadores virtuais, ou de elementos processadores de fluxos, depende da migração de processos que tratam os pacotes encaminhados. No caso de um roteador virtual especificamente, a migração da rede virtual ocorre quando a máquina virtual que implementa o serviço de roteamento virtual é migrada entre duas máquinas físicas.

1.6.3 Comparação de Propostas para o Roteamento em Computadores Pessoais

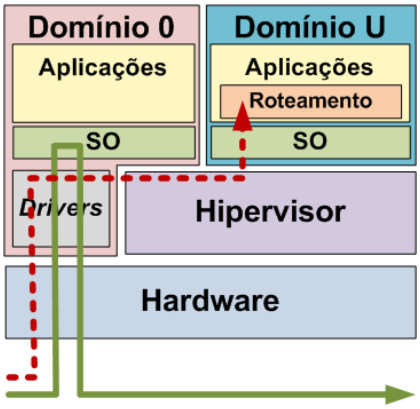
A Figura 1.5 apresenta uma comparação entre quatro formas distintas de se implementar um roteador virtual em uma plataforma de computadores pessoais. A Figura 1.5(a) apresenta o funcionamento básico de um computador pessoal agindo como roteador. Nesse caso, o protocolo de roteamento é implementado como uma aplicação que executa sobre o sistema operacional. A aplicação de roteamento configura políticas no *kernel*, núcleo, do sistema operacional para realizar o encaminhamento dos pacotes diretamente pelo núcleo. Quando consideramos um cenário virtualizado, Figura 1.5(b), esse procedimento ocorre dentro de uma máquina virtual. No entanto, a virtualização das operações de entrada e saída (E/S) adiciona a sobrecarga de que todos os pacotes devem passar pelo Domínio 0, que detém o acesso exclusivo aos dispositivos de E/S, e pelo hipervisor, a camada responsável pela virtualização. Uma alternativa para evitar a sobrecarga gerada pela virtualização de dispositivos de E/S é a técnica de separação de planos. Essa técnica prevê a execução normal do protocolo de roteamento na máquina virtual, mas o processo de encaminhamento dos pacotes se dá no Domínio 0, como mostrado na Figura 1.5(c). Nesse caso, o encaminhamento é feito pelo sistema operacional do Domínio 0, que possui uma cópia das regras de encaminhamento da máquina virtual. Já no sistema XenFlow, o roteamento dos pacotes se dá de outra maneira, como mostrado na Figura 1.5(d). No XenFlow, tanto o encaminhamento dos pacotes de dados para a porta de saída correta, quanto o encaminhamento dos pacotes de controle para as máquinas virtuais de destino, são realizados por um comutador OpenFlow instanciado no Domínio 0. O comutador OpenFlow, por ser programável, agrega maior flexibilidade ao roteamento do que as demais propostas de roteamento apresentadas.



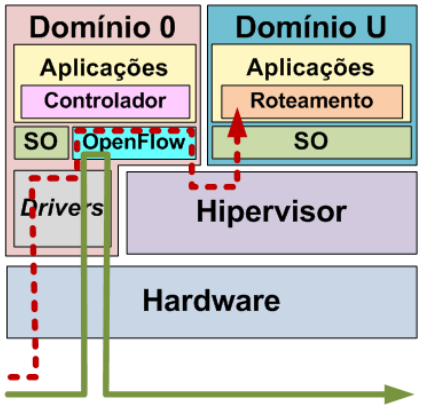
(a) Roteamento usando um computador pessoal sem virtualização.



(b) Roteamento através de uma máquina virtual Xen sem separação de planos. Os pacotes de controle e de dados são encaminhados para a máquina virtual.



(c) Roteamento através de uma máquina virtual Xen com separação de planos. Os pacotes de controle são encaminhados para a máquina virtual. Os pacotes de dados são encaminhados diretamente pelo Domínio 0.



(d) Roteamento através do sistema de virtualização XenFlow. Os pacotes de dados e de controle são encaminhados por um comutador OpenFlow no Domínio 0. Os pacotes de controle são encaminhados para o roteador virtual de destino e os pacotes de dados são encaminhados diretamente para a interface de saída.

Figure 1.5: Comparação entre quatro propostas de roteamento usando computadores pessoais. As setas tracejadas indicam os fluxos dos pacotes de controle dos protocolos de roteamento. As setas contínuas indicam os fluxos de pacotes encaminhados por cada configuração de roteador.

1.6.4 Roteamento em Redes OpenFlow

Uma rede OpenFlow é naturalmente comutada, ou seja, todos os nós, a princípio estão em um mesmo domínio de difusão. Contudo, existem propostas para realizar o roteamento de pacotes em redes OpenFlow. Um exemplo dessas propostas é a arquitetura QuagFlow [27]. A ideia chave da arquitetura QuagFlow é permitir que protocolos de roteamento convencionais, que funcionam de forma distribuída, controlem o encaminhamento de pacotes em uma rede OpenFlow. No QuagFlow, 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 de encaminhado dos fluxos no plano de dados OpenFlow. 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. Outro problema é que essa arquitetura emula o roteamento sobre uma rede OpenFlow, mas não separa o plano de dados do plano de controle de cada comutador. Assim, não é possível mover o plano de controle de um comutador para outro, ou seja, a topologia lógica da rede no QuagFlow é estática.

Esse trabalho apresenta uma arquitetura de processamento de fluxos, na qual as máquinas virtuais agem como roteadores, e desenvolve a primitiva de migração para esse cenário. Com o intuito de atingir tal objetivo, implementa-se uma plataforma de virtualização híbrida. A plataforma proposta permite o compartilhamento de um plano de dados flexível e a distribuição do controle da rede. O plano de controle executa em máquinas virtuais. A arquitetura proposta emprega um modelo de roteador virtual capaz de desacoplar a topologia virtual da topologia física. O roteador proposto possui um plano de dados programável, capaz de se comportar tanto como um elemento de comutação, quanto como um roteador. Nesse contexto, ao realizar a migração de uma rede virtual, quebra-se o mapeamento entre roteador virtual e roteador físico e, consequentemente, há a necessidade de se refazer o mapeamento de enlaces virtuais sobre enlaces físicos. Dessa forma, a rede virtual não fica mais restringida pelos limites da rede física, pois a rede virtual pode mover-se livremente sobre a topologia física, sem que haja mudanças em sua topologia virtual. A proposta também emprega uma nova técnica de migração de enlaces, que torna a migração de enlaces mais simples, no contexto de redes virtuais, sem a necessidade de criação de túneis e com perda zero de pacotes. Essa nova técnica torna-se viável, pois o plano de dados é programável.

1.7 Organização do Relatório

O restante do relatório está organizado da seguinte forma. O Capítulo 2 apresenta a ferramenta de virtualização de computadores Xen. A ferramenta de virtualização de redes OpenFlow é discutida no Capítulo 3. A definição do sistema proposto, sua arquitetura e principais componentes são apresentados no Capítulo 4. O Capítulo 5 apresenta a análise dos resultados experimentais, as conclusões e os trabalhos futuros.

Capítulo 2

A Plataforma de Virtualização Xen

O Xen é uma plataforma de virtualização de computadores pessoais, bastante empregada na consolidação de servidores[1]. A arquitetura do Xen é baseada em uma camada de virtualização, localizada sobre o *hardware*, denominada Monitor de Máquina Virtual (VMM – *Virtual Machine Monitor*) ou hipervisor, como pode ser visto na Figura 2.1. Sobre o hipervisor executam os ambientes virtuais, chamados de máquinas virtuais, ou domínios desprivilegiados (Domínio U), que acessam recursos de forma independente, como CPU, memória, acesso a disco e à rede. Cada ambiente virtual está isolado dos demais, isto é, a execução de uma máquina virtual não afeta a execução de outra máquina virtual as quais, inclusive, podem ter sistemas operacionais distintos. Há, ainda, um ambiente virtual privilegiado, denominado Domínio 0, que detém a exclusividade do acesso aos dispositivos físicos e, portanto, provê o acesso às operações de Entrada/Saída dos demais domínios e também executa operações de gerência do hipervisor. Já os demais domínios, referenciados como Domínio U ou domínios desprivilegiados, não possuem acesso direto ao *hardware*. Sendo assim, os domínios desprivilegiados possuem dispositivos (*drivers*) virtuais, que se comunicam com o Domínio 0 para acessarem os dispositivos físicos.

2.1 A Virtualização do Processador

A virtualização do processador físico (*Central Processing Unit* - CPU) é realizada através da atribuição de CPUs virtuais (vCPU) às máquinas

[1]A consolidação de servidores consiste em instalar diferentes servidores em máquinas virtuais isoladas hospedadas por uma mesma máquina física.

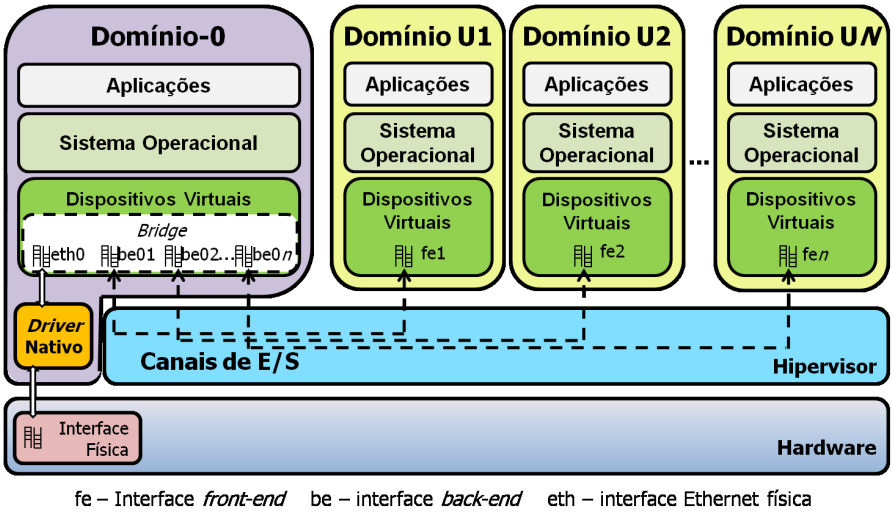


Figure 2.1: Arquitetura da plataforma de virtualização Xen.

virtuais. A vCPU é a CPU que os processos, que estão sendo executados em uma máquina virtual, podem acessar. Os processos da máquina virtual são escalonados, pelo sistema operacional da máquina virtual, para serem executados nas vCPUs atribuídas àquela máquina virtual. O mapeamento de vCPU em uma CPU real é realizado pelo hipervisor do Xen. O hipervisor do Xen implementa um mecanismo de escalonamento dinâmico de vCPUs sobre as CPUs reais. O algoritmo padrão de escalonamento de CPU no Xen é o *Credit Scheduler*, ou escalonador por crédito, que gera um compartilhamento proporcional de CPU entre as vCPUs. Nesse esquema de escalonamento, os recursos de CPU são alocados para as vCPUs de acordo com pesos definidos para cada máquina virtual.

2.2 A Virtualização da Memória

A virtualização do recurso de memória no Xen é estática. A memória RAM da máquina física é dividida entre as máquinas virtuais. Cada máquina virtual recebe uma quantidade fixa de memória no momento de sua instanciação. Atualmente é possível manipular a quantidade de memória que uma máquina virtual tem direito durante a execução da máquina virtual. No entanto, a quantidade de memória a ser alocada de forma dedicada à máquina virtual deve respeitar valores de máximo e mínimo definidos em sua configuração inicial e devem, ainda, respeitar a disponibilidade de memória na máquina física. A virtualização da memória exige o mínimo de envolvimento

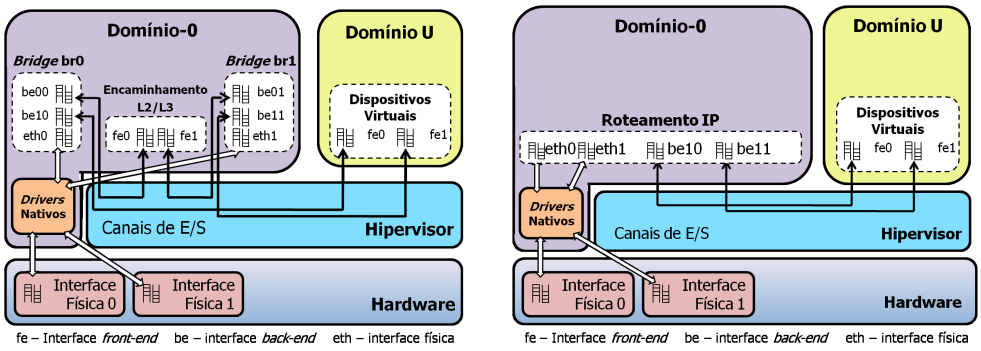
do hipervisor. As máquinas virtuais são responsáveis pelas suas tabelas de páginas de memória. Todas as vezes que uma máquina virtual requisita uma nova tabela de páginas, ela a aloca e a inicializa em seu próprio espaço de memória e a registra no hipervisor Xen, o qual é responsável por garantir o isolamento. O isolamento é realizado pelo hipervisor de modo que cada máquina virtual, ou domínio, acesse apenas a área de memória reservada a ela.

2.3 A Virtualização da Interface de Rede

A virtualização da interface de rede no Xen é feita demultiplexando os pacotes que entram pela interface física para os domínios desprivilegiados e, de forma similar, multiplexando os pacotes que saem desses domínios para as interfaces físicas de rede. A virtualização das operações de entrada e saída nas interfaces de rede se dá da seguinte forma. Os Domínios Us possuem acesso a dispositivos virtuais de entrada e saída, que são controlados por dispositivos (*drivers*) virtuais que fazem requisições ao Domínio 0 para acessarem os dispositivos físicos. Ao contrário dos Domínios Us, o Domínio 0 tem acesso direto aos dispositivos de entrada e saída, através dos controladores de dispositivos (*drivers*) nativos. Dessa forma, ao receber uma requisição de um Domínio U, o Domínio 0 executa a requisição diretamente sobre o controlador de dispositivo (*driver*) nativo. A comunicação entre os dispositivos virtuais dos domínios desprivilegiados e o Domínio 0 é realizada através de uma dupla de interfaces: interface *back-end* e interface *front-end* [1]. Cada domínio desprivilegiado tem interfaces virtuais, chamadas *front-end*, que são utilizadas para todas as comunicações de rede. Essas interfaces virtuais são tratadas pelos sistemas operacionais dos Domínios Us como se fossem interfaces físicas reais. Para cada interface *front-end* criada nos domínios desprivilegiados, é criada uma interface *back-end* no Domínio 0. As interfaces *back-end* atuam como representantes das interfaces dos domínios desprivilegiados no Domínio 0. As interfaces *back-end* e *front-end* se comunicam através de um canal de E/S (*I/O Channel*). A troca de pacotes entre as interfaces dos domínios Us e o Domínio 0 é realizada de forma eficiente e, portanto, sem cópia de memória. Um mecanismo empregado pelo canal de E/S remapeia a página física contendo o pacote no domínio de destino.

Por padrão no Xen, a conexão das interfaces *back-end* e as interfaces físicas de rede podem ser realizadas de duas maneiras. A primeira, e padrão do Xen, é através do modo comutado (*Bridge*), mostrado na Figura 2.2(a). Nesse modo, são instanciadas pontes (*bridges*) no Domínio 0 e as interfaces *back-end* e as interfaces reais são associadas a elas. Uma ponte (*bridge*)

é um comutador por *software*. Assim, o encaminhamento do pacote para interface *back-end* correta é realizado através do encaminhamento do pacote pela interface que responde ao endereço MAC , de destino do pacote. Vale ressaltar que são necessárias tantas pontes (*Bridges*) quantas são o número de interfaces física. O segundo modo de interconexão das interfaces reais e as *back-end* é o modo roteado (*Router*), mostrado na Figura 2.2(b). No modo roteado, o Domínio 0 passa a se comportar como um roteador. Dessa forma, a cada pacote que chega, o Domínio 0 verifica o endereço IP de destino e encaminha o pacote de acordo com as rotas definidas em suas tabelas de roteamento. Assim, o encaminhamento do pacote para a interface *back-end*, ou física, correta, depende somente da definição correta da rota no Domínio 0.



(a) Modo comutado (*Bridge*).

(b) Modo roteado (*Router*).

Figure 2.2: Virtualização do recurso de rede no Xen. Adaptado de [1].

A virtualização da rede é alcançada no Xen através da instanciação de diversas máquinas virtuais, que correspondem aos elementos virtuais de rede, sobre um mesmo *hardware* físico, pois o Xen permite a execução de múltiplas máquinas virtuais simultaneamente sobre a mesma máquina física. Um exemplo de virtualização de redes usando Xen é o caso em que os elementos de rede virtuais instanciados são roteadores virtuais. Nesse caso, como a camada de virtualização do Xen está abaixo dos sistemas operacionais, cada roteador virtual pode ter o seu próprio sistema operacional e cada um detém os seus próprios planos de dados e controle isolados dos demais roteadores. Nessa arquitetura de rede virtual, um roteador virtual pode ser instanciado, configurado, monitorado e desativado sob demanda. O roteador virtual pode, ainda, ser migrado, em funcionamento, usando o mecanismo de migração ao vivo do Xen [15].

2.4 A Virtualização e Migração de Redes

A Figura 2.3 mostra uma rede virtualizada com base no Xen. A rede é composta por máquinas virtuais Xen executando a função de roteador [1]. 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 do serviço de encaminhamento de pacotes seja o mínimo possível.

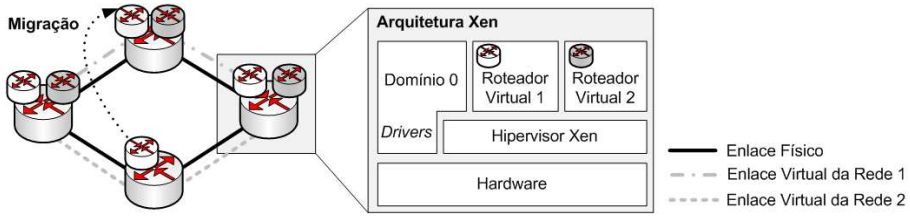


Figure 2.3: Migração do roteador virtual 1 que se encontra no roteador físico de baixo para o roteador físico de cima.

A migração nativa do Xen [15] baseia-se na migração ao vivo de máquinas virtuais, que consiste em copiar a memória da máquina virtual de uma máquina física de origem para a de destino. Como as páginas de memória da máquina virtual, na origem, são alteradas durante o procedimento de migração, utiliza-se um mecanismo de cópias iterativas de páginas de memória, no qual as páginas alteradas, denominadas páginas sujas, são marcadas e recopiadas na iteração seguinte. Isso se repete até que o número de páginas de memória alteradas desde a última rodada seja suficientemente pequeno. Nesse momento, a execução da máquina virtual é suspensa na máquina física de origem, as últimas páginas de memória alteradas são copiadas para o destino, e a máquina virtual é, então, restaurada na máquina física de destino. Uma desvantagem dessa proposta para a migração de roteadores virtuais é a perda de pacotes durante o tempo em que a máquina virtual fica indisponível, entre a suspensão e a restauração. Este mecanismo está limitado a migrações entre máquinas físicas conectadas a um mesmo barramento, pois a migração dos enlaces da máquina virtual é realizada através do envio de pacotes de *ARP Reply*. Considerando a abordagem em que as interfaces de rede das máquinas virtuais estão na mesma rede local que as interfaces da máquina física, o *ARP Reply* é enviado pela máquina migrante para informar aos comutadores da rede local que os endereços MAC das interfaces virtuais estão, agora, disponíveis através de outros caminhos, já que a máquina física de destino apresenta outras conexões físicas com a rede em relação à máquina de origem. A migração de enlaces no Xen se dá após o término da migração

da máquina virtual. Assim, quando a máquina virtual já está instalada e pronta para funcionar na máquina física de destino, a máquina física de destino envia pacotes de *ARP Reply*, como se fosse a máquina virtual migrada, em todas as interfaces que a máquina virtual tem acesso. Esse procedimento faz com que os comutadores em uma rede local saibam que o endereço MAC referente à máquina virtual migrada, agora, está acessível através de um novo caminho.

Para evitar a perda de pacotes na migração, propôs-se a migração com separação de planos. A separação de planos consiste em dividir a tarefa de roteamento em dois planos distintos, um de controle e outro de dados. O processo de controle, definido no plano de controle, é responsável pela execução do protocolo de roteamento e pela atualização da tabela de rotas. Já o processo de encaminhamento é responsável por encaminhar os pacotes para as interfaces de saída corretas, de acordo com as rotas definidas nas tabelas de encaminhamento do sistema. Para a plataforma de virtualização Xen, Pisa *et al.* propõem a mudança do plano de dados de todos os roteadores virtuais para o Domínio 0 dos roteadores físicos [14]. Assim, os pacotes são encaminhados pelo Domínio 0, antes mesmo de serem repassados para o roteador virtual ao qual estavam endereçados. No entanto, o plano de controle continua no roteador virtual, domínio U. Por conseguinte, as informações de controle do roteamento são processadas no roteador virtual e este atualiza as regras de encaminhamento no Domínio 0. Dessa forma, a migração do plano de controle ocorre sem afetar o encaminhamento dos pacotes, já que a suspensão do roteador virtual, durante a migração, não interfere nas regras de encaminhamento definidas no Domínio 0. No entanto, a solução é restritiva, no sentido de que um roteador virtual só pode ser migrado para outro roteador físico que apresente os mesmos vizinhos que o de origem e que esteja conectado ao mesmo barramento que o de origem. Essa solução limita o alcance da migração a um salto a partir do roteador de origem, pois é o processo de migração utilizado é o mesmo da migração nativa do Xen.

Capítulo 3

OpenFlow

O OpenFlow [13] permite que a infraestrutura física de redes seja compartilhada pela rede de produção e por redes experimentais. O OpenFlow é um projeto desenvolvido na Universidade de *Stanford* que tem por objetivo implementar uma tecnologia capaz de promover a inovação no núcleo da rede, através da execução de redes de testes em paralelo com a rede de produção. A ideia chave do OpenFlow é promover a inovação em redes universitárias. A tecnologia OpenFlow promove a criação de redes definidas por *software*, usando elementos comuns de rede, tais como comutadores, roteadores, pontos de acesso ou, até mesmo, computadores pessoais[1] [28].

O OpenFlow implementa a virtualização do plano de dados. A arquitetura do OpenFlow é baseada na separação física das funções de encaminhamento e de controle da rede. A função de encaminhamento, desempenhada pelo plano de dados, é executada por elementos especializados da rede que apresentam uma tabela de fluxos compartilhada. É através dessa tabela de fluxos compartilhada que o plano de dados é virtualizado. Já a função de controle, exercida pelo plano de controle, é centralizada em outro elemento da rede, o chamado controlador. O controlador executa funções de controle para a rede virtual.

3.1 Protocolo OpenFlow

A comunicação entre os elementos de rede e o controlador é definida pelo protocolo OpenFlow. A comunicação é estabelecida através de um canal seguro definido entre o controlador e cada elemento OpenFlow. O protocolo

[1]Uma das propostas para a implementação do comutador OpenFlow em computadores pessoais é o Open vSwitch [28], que funciona como um módulo do Linux que implementa o encaminhamento programável de pacotes diretamente no *kernel* do sistema.

OpenFlow define funções para configurar e monitorar os elementos. O encaminhamento é definido com base em fluxos. Um fluxo é uma sequência de pacotes com um conjunto de características comuns. Quando um pacote chega ao elemento encaminhador, o elemento verifica se o pacote se adéqua a algum fluxo já definido. Em caso positivo, as ações definidas para aquele fluxo são aplicadas ao pacote. Em caso negativo, o pacote é encaminhado para o controlador, que extrai as características do fluxo, a partir do pacote, e cria um novo fluxo, introduzindo-o na tabela de fluxos do elemento OpenFlow. Uma das possíveis ações que o controlador pode definir para um fluxo é que ele siga o processamento normal, seja de comutação na camada 2, seja roteamento na camada 3, como se não existisse o protocolo OpenFlow. Essa funcionalidade garante que a rede de produção execute em paralelo com redes experimentais, sem que estas afetem o funcionamento daquela. A Figura 3.1 mostra a organização de uma rede OpenFlow e a comunicação dos comutadores com o controlador.

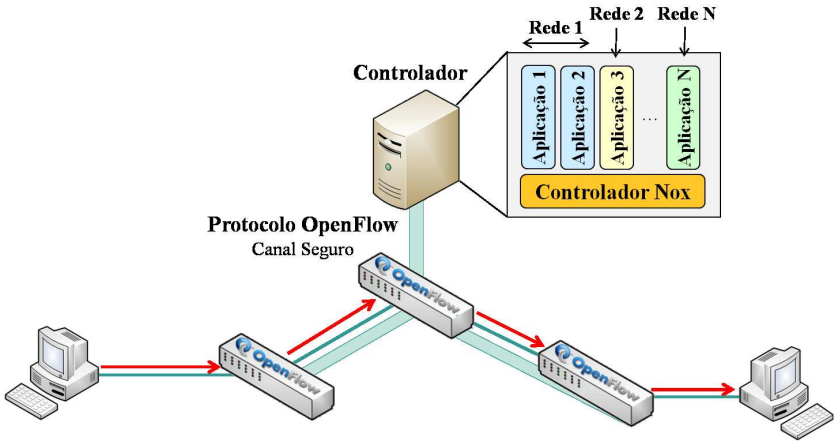


Figure 3.1: Arquitetura de uma rede OpenFlow. Os comutadores OpenFlow comunicam-se com o controlador através do protocolo OpenFlow em um canal seguro. O controlador executa as aplicações de controle de cada rede virtual.

As entradas na tabela de fluxo do OpenFlow são compostas por campos do cabeçalho dos pacotes, contadores e ações. Os campos do cabeçalho são a descrição do fluxo, ou seja, descrevem quais pacotes combinam com aquele fluxo. Esses campos formam uma tupla de doze elementos, que reúne características dos protocolos de várias camadas do pacote, como pode ser visto na Figura 3.2. No OpenFlow, a regra de encaminhamento de um pacote não se restringe ao endereço de rede, endereço IP , ou endereço físico, endereço MAC , do pacote. Os elementos dessa tupla podem conter valores

exatos ou valores coringas, que combinam com qualquer valor que o pacote comparado apresente para o campo. O encaminhamento pode se dar por outras características do pacote, como por exemplo, as portas de origem e destino do protocolo de transporte. Um objetivo futuro do projeto OpenFlow é permitir a criação de campos definidos pelo usuário como um critério de encaminhamento. Dessa forma, será possível definir regras de encaminhamento com base em protocolos experimentais, como protocolos de uma arquitetura pós-IP.

Figure 3.2: Definição de um fluxo em um comutador OpenFlow. Os campos que compõem a definição do fluxo são extraídos do cabeçalho do primeiro pacote que é enviado ao controlador.

A entrada na tabela de fluxos apresenta a descrição do fluxo, composta pelos campos extraídos do cabeçalho, e contadores para monitorar o fluxo. Esses contadores computam os dados referentes ao fluxo descrito, tais como, quantidade de *bytes* transmitidos, duração do fluxo e quantidade de pacotes transmitidos, em cada elemento de encaminhamento. Seguindo os contadores, a entrada na tabela de fluxos ainda tem as ações definidas para cada fluxo. O conjunto de ações relacionadas a cada fluxo é definido pelo controlador para cada elemento de encaminhamento. Esse conjunto define o tratamento dado a todos os pacotes que chegam ao elemento de encaminhamento e combinam com a descrição do fluxo. Essas ações incluem o encaminhamento do pacote em uma determinada porta de saída, mas, também, incluem outras ações como a modificação de campos no cabeçalho dos pacotes.

3.2 Controlador

O controlador é um elemento centralizado que executa aplicações de controle sobre a rede OpenFlow, configurando as tabelas de fluxo dos elementos encaminhadores. O controlador implementa o protocolo OpenFlow para se comunicar com os elementos encaminhadores e, através desse protocolo, manda os comandos para a rede. Um dos controladores OpenFlow mais usados é o Nox [29]. O Nox age como um sistema operacional de rede. O Nox provê as funções básicas de configuração e monitoramento da rede para as aplicações que controlam a rede. Dessa forma, o controlador age somente

como uma interface entre a rede e as aplicações. O plano de controle é exercido pelas aplicações que executam sobre o Nox. Sendo assim, uma rede virtual no OpenFlow é representada pelo seu conjunto de fluxos, plano de dados, e pelas suas aplicações de controle, plano de controle.

3.3 FlowVisor: Virtualização do Plano de Controle

A virtualização do plano de controle em redes OpenFlow é feita pelo FlowVisor [22]. O FlowVisor é um controlador especial do OpenFlow, que funciona de forma transparente entre os dispositivos em uma rede OpenFlow e os outros controladores, como por exemplo Nox. O FlowVisor permite que mais de um controlador controle a rede OpenFlow, para tanto, o FlowVisor introduz o conceito de fatia. Cada fatia de rede é controlado por um controlador. As fatia são definidas por políticas no FlowVisor. As mensagens de controle trocadas entre os dispositivos e os controladores são intermediadas pelo FlowVisor, que verifica para cada mensagem quais políticas são aplicáveis e verifica, também, se um determinado controlador pode controlar um dado fluxo em um dispositivo.

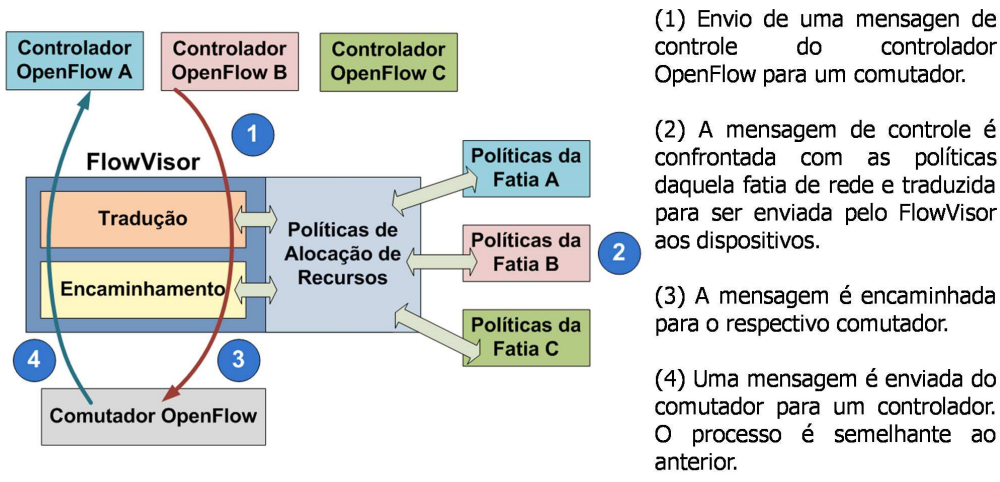


Figure 3.3: Arquitetura do FlowVisor, um controlador especial do OpenFlow, que permite a definição de redes virtuais.

A Figura 3.3 mostra que o FlowVisor intercepta as mensagens OpenFlow mandadas pelos controladores visitantes (1), compara-as com as políticas

de fatiamento da rede (2) e, após, reescreve a mensagem. Esse procedimento propicia que o controle da fatia fique limitado ao controlador que está definido nas políticas do FlowVisor. O procedimento de envio de mensagens dos dispositivos para os controladores visitantes também é semelhante (4). O FlowVisor apenas encaminha as mensagens dos dispositivos para os controladores visitantes se as mensagens dos dispositivos forem compatíveis com as políticas de cada fatia de rede que o controlada por um dado controlador. O fatiamento do plano de controle da rede é realizado de forma a manter cada fatia isolada das demais.

O FlowVisor executa a virtualização do plano de controle da rede, isolando o controle, mas compartilhando o plano de dados dos comutadores da rede OpenFlow. O fatiamento da rede realizado pelo FlowVisor é focado no compartilhamento de cinco recursos primitivos da rede [22]: isolamento de banda, descoberta de topologia, engenharia de tráfego, monitoramento de CPU e controle das tabelas de encaminhamento. O FlowVisor executa apenas a virtualização do plano de controle, permitindo que mais de um controlador troque mensagens de controle com um dispositivo OpenFlow. No entanto, cada controlador exerce o controle em uma fatia da rede e, nessa fatia, só um controlador exerce o controle. Dessa forma, o FlowVisor cria ilhas de controle em uma rede OpenFlow, de forma que o controle global da rede fica distribuído, enquanto o controle de cada fatia de rede fica centralizado em um controlador por fatia.

3.4 Migração de Fluxos

A infraestrutura provida pelo OpenFlow permite que a rede física seja particionada em múltiplas redes virtuais. No OpenFlow, a instanciação de uma rede virtual é a criação dos fluxos referentes a essa rede. Esses fluxos são instanciados sob demanda, de acordo com o conjunto de aplicações que executam no controlador. Nesse cenário, a realocação de recursos de rede é flexível. A realocação dos recursos significa a reprogramação das tabelas de fluxo de forma que os fluxos encaminhados por um elemento passem a serem encaminhados por outro elemento da rede. Essa função é facilmente exercida pelo controlador, pois este possui uma visão global da rede.

No OpenFlow a migração de redes virtuais corresponde à migração de fluxos. A migração de fluxos é bastante simples, basta reconfigurar as tabelas de fluxo, como mostrado na Figura 3.4. Isso é possível, pois o controlador centralizado tem acesso à configuração das tabelas de todos os comutadores na rede virtual. O algoritmo de migração consiste em adicionar uma nova entrada para o fluxo na tabela de fluxos de cada um dos comutadores no novo

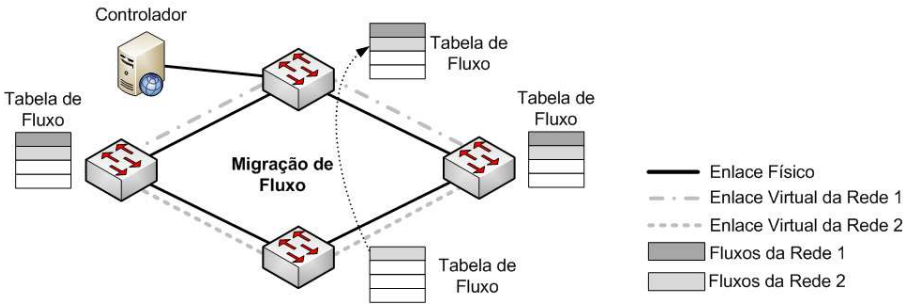


Figure 3.4: Virtualização de redes usando OpenFlow. Cada rede virtual é representada por um conjunto de fluxos. A migração da rede virtual é a redefinição dos fluxos referentes àquela rede em outro comutador.

caminho, com exceção do primeiro comutador de origem, o qual é comum ao caminho antigo e ao novo. Após essa primeira etapa, o controlador modifica o fluxo definido no primeiro comutador, de modo que a antiga porta de saída seja modificada para uma nova porta, que redireciona o fluxo ao restante do caminho previamente definido. Assim, um fluxo é migrado no OpenFlow, sem perda de pacotes e sem a interrupção do serviço de encaminhamento de pacotes, pois a todo momento existe um caminho fim a fim entre origem e destino.

Capítulo 4

Sistema Proposto: XenFlow

O sistema proposto combina as vantagens da virtualização de redes com o controle distribuído e a programabilidade, da plataforma Xen, e com o processamento por fluxo provido pela plataforma OpenFlow. A arquitetura de um elemento de rede XenFlow é mostrada na Figura 4.1. A base da arquitetura da máquina física é a plataforma de virtualização Xen. Dessa forma, a virtualização de redes é alcançada através da instanciação de máquinas virtuais, que executam funções referentes a um elemento de rede virtual, e pelo conjunto de fluxos definidos por essas máquinas. Devido ao processo de encaminhamento de fluxo ser baseado no OpenFlow, cada elemento de rede virtual pode ser um comutador (nível 2), roteador (nível 3) ou *middle box* (nível maior que 3). A função que cada elemento de rede executa depende da pilha de protocolos que ele implementa e do seu conjunto de aplicações. A comunicação entre dois elementos de rede virtuais, de uma mesma máquina física na qual estão hospedados, e de um elemento de rede com a rede externa à máquina física, é intermediada pelo Domínio 0. O sistema XenFlow, para prover maior flexibilidade na definição dos caminhos dos fluxos no Domínio 0, comuta os fluxos por uma matriz de comutação programável que implementa o protocolo OpenFlow. Assim, a funcionalidade de migração de rede virtual no sistema XenFlow apresenta perda zero de pacotes, como na plataforma OpenFlow, e torna possível reorganizar a rede lógica sem que sejam necessárias mudanças na topologia física.

No sistema XenFlow, assim como na plataforma Xen, os *drivers* dos dispositivos físicos ficam do Domínio 0 e, portanto, toda comunicação entre máquinas virtuais e dispositivos físicos deve passar pelo Domínio 0. Assim, o Domínio 0 realiza a multiplexação dos pacotes que estão saindo dos elementos de rede virtuais para os dispositivos físicos de rede e a demultiplexação dos pacotes que estão chegando aos dispositivos físicos de rede e vão para os elementos de rede virtuais [1]. O XenFlow age como um terceiro modo ma-

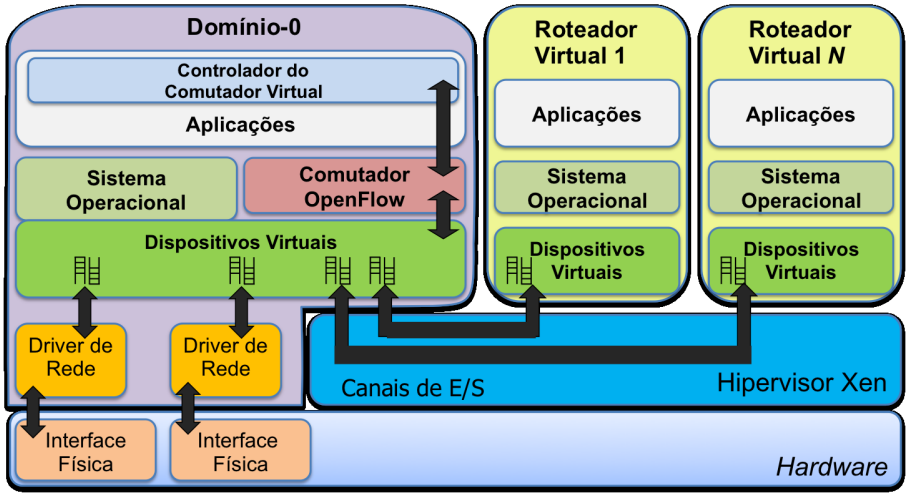


Figure 4.1: Arquitetura de um elemento de rede do sistema XenFlow.

peamento entre interfaces virtuais e interfaces físicas, similar ao modo *bridge* e ao modo *router* mostrados no Capítulo 2. No caso do sistema XenFlow, o processo de multiplexação e demultiplexação é realizado por um comutador OpenFlow. O OpenFlow executa o encaminhamento entre as interfaces *back-end* e as interfaces físicas no Domínio 0 de acordo com as regras definidas por um controlador. Nesse modo de encaminhamento entre interfaces *back-end* e interfaces físicas, é como se as interfaces das máquinas virtuais e as interfaces físicas do Domínio 0 estivessem conectadas a um comutador físico que implementa o protocolo OpenFlow. Uma das vantagens de realizar a multiplexação dos pacotes no Domínio 0 em um comutador OpenFlow é que o encaminhamento dos pacotes pode ser programado de acordo com as regras definidas no controlador do comutador OpenFlow, que, por sua vez, é uma aplicação que executa no Domínio 0 e define as regras de encaminhamento dos pacotes. Outra vantagem de se estabelecer o mapeamento de interfaces físicas com interfaces virtuais a partir de um comutador OpenFlow é que se pode estabelecer regras no controlador para garantir uma banda mínima para cada fluxo ou conjunto de fluxos [13]. O comutador OpenFlow agrega as interfaces virtuais, que realizam a comunicação ponto-a-ponto entre domínios virtuais e Domínio 0, as chamadas interfaces *back-end*, e as interfaces físicas. Sendo assim, os fluxos de entrada e saída da máquina física também podem ser programados segundo as políticas do controlador. Vale ainda lembrar, que ao estabelecer o mapeamento entre as interfaces através de um comutador OpenFlow, o encaminhamento dos pacotes pode ser realizado com base nos doze campos que compõem a tupla de definição de fluxo do OpenFlow, não ficando mais restrito aos modos de encaminhamento do Xen nativo.

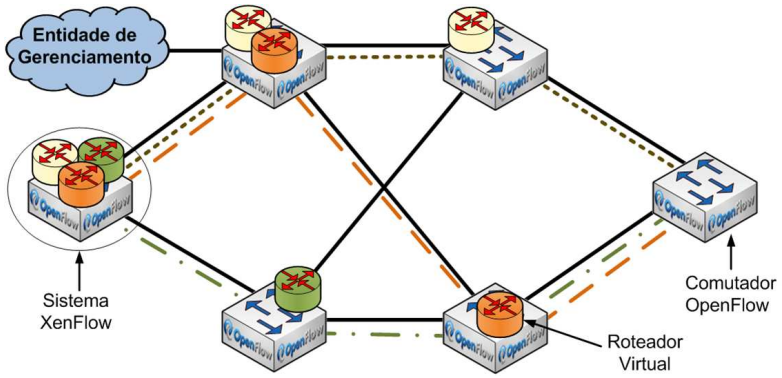


Figure 4.2: A rede XenFlow é composta por nós que implementam um plano de dados OpenFlow e um plano de controle Xen e por comutadores OpenFlow.

O controle da rede XenFlow é distribuído em fatias de roteadores físicos que participam da rede virtual. Esta, por sua vez, é representada por um conjunto de roteadores virtuais que implementam uma pilha de protocolos e pelo conjunto de fluxos instanciados para aquela rede. A Figura 4.2 mostra o sistema XenFlow com três redes virtuais instanciadas. Nessa rede, há nós da rede física que são comutadores e outros que são roteadores. Os comutadores físicos executam o protocolo OpenFlow, o que os permite implementarem comutadores virtuais. Isso permite que o plano de dados dos comutadores da rede sejam tão programáveis quanto o dos roteadores XenFlow. Os roteadores físicos implementam o sistema, como descrito na Figura 4.1. Na Figura 4.2, há ainda uma entidade de gerenciamento. Essa entidade tem consciência da topologia física da rede, que pode ser obtida através da topologia calculada por protocolos de roteamento de estado de enlace, como o OSPF (*Open Shortest Path First*) , que calculam as melhores rotas com base na topologia da rede armazenada localmente. É a partir da entidade de gerência que as migrações são iniciadas. Essa entidade é responsável por desempenhar a migração de redes virtuais, verificando a necessidade da migração, definindo a origem e o destino da migração e, por fim, realizando a migração propriamente dita. Essas funções podem ser desempenhadas a partir de qualquer nó da rede. Quando uma migração ocorre, a topologia física da rede é consultada para que o sistema possa mapear os enlaces lógicos sobre os enlaces físicos. O controlador OpenFlow, no Domínio 0, também implementa uma interface de gerenciamento dinâmico da rede. Através dessa interface, pode-se inserir novas regras na Tabela de Regras do controlador ou atualizar regras antigas. A interface de gerenciamento dinâmico permite a reconfiguração dinâmica das tabelas de fluxos e das regras de definição de fluxos no Nox. A interface de

gerenciamento dinâmico é a interface que permite que a Entidade de Gerenciamento execute os comandos de gerenciamento nos roteadores XenFlow.

4.1 Separação de Planos e Tradução de Rotas em Fluxos

A migração sem perda de pacotes é fundamental na migração de roteadores e este objetivo pode ser alcançado com a 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 elemento de rede virtual, no Domínio 0. No XenFlow a cópia do plano de dados no Domínio 0 é composta pelo comutador OpenFlow e pelo Controlador Nox, como ilustrado na Figura 4.3. As informações contidas na tabela de rotas do roteador virtual referem-se às redes que o roteador virtual consegue alcançar. Já as informações contidas nas tabelas de fluxo de um comutador OpenFlow referem-se às regras de encaminhamento de fluxo fluxos específicos. Sendo assim, as informações de rotas são muito mais genéricas do que as informações de encaminhamento de fluxos. Logo, há a necessidade de um elemento que armazene as rotas no controlador do comutador OpenFlow e faça a tradução das rotas em fluxos. Assim, o plano de dados de cada elemento de rede virtual é equivalente ao conjunto formado pelas tabelas de fluxos do OpenFlow e pela tabela de regras do Nox, responsável por armazenar as rotas e traduzir rotas em fluxos. Logo, quando o plano de controle uma atualização de rotas, o módulo de *daemon* de atualização de tabela de regras as envia para a Tabela de Regras, no controlador Nox. As rotas são então traduzidas em fluxos pelo módulo Tabela de Regras, executado no Nox, que, por sua vez, também atualiza a tabela de fluxos do comutador OpenFlow, sempre que uma atualização de rotas interfira em fluxos já estabelecidos.

Os pacotes de dados são encaminhados pelo XenFlow da seguinte forma. Um pacote, ao chegar ao comutador OpenFlow do roteador físico, tem dois tratamentos possíveis, ser encaminhado diretamente, caso coincida com um fluxo existente na tabela de fluxos, ou, caso contrário, ser encaminhado para o controlador, para que este defina o seu caminho. O controlador Nox, então, extrai do pacote os doze campos necessários para a definição do fluxo no OpenFlow, consulta a tabela de regras para definir qual é o destino do pacote e insere um fluxo na tabela de fluxos do comutador. É importante ressaltar que o pacote chega com o endereço MAC de destino do roteador virtual e este endereço deve ser trocado pelo endereço MAC do próximo salto, que é obtido

da tabela de regras. O procedimento de definir o caminho dos pacotes com base nas regras do Nox é que viabiliza o mapeamento de um enlace virtual em um ou mais enlaces físicos. O mapeamento é realizado por uma aplicação do controlador que define regras para o encaminhamento dos pacotes. Uma das regras é fazer com que o roteador físico XenFlow comporte-se como um comutador e apenas encaminhe determinado fluxo para o próximo nó na rede, sem modificar o pacote ou submetê-lo às regras de roteamento.

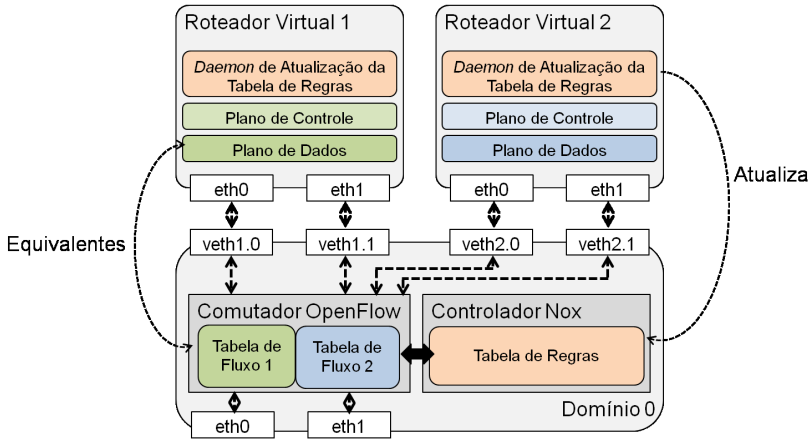


Figure 4.3: Roteamento no sistema XenFlow com separação de planos. Os pacotes são encaminhados diretamente pelo Domínio 0.

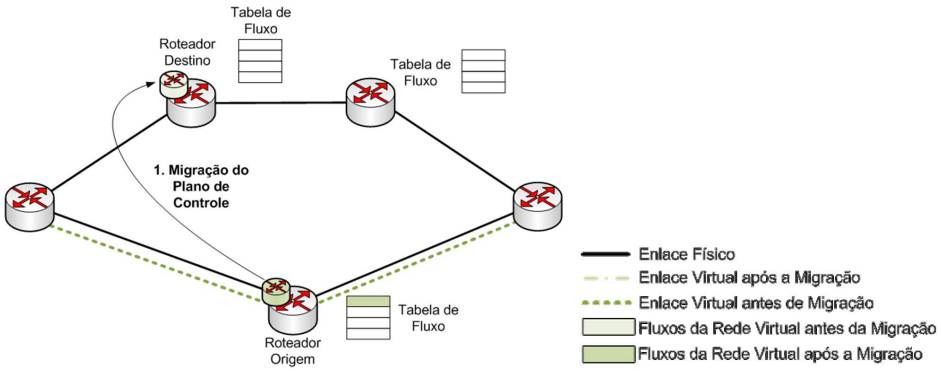
A tabela de regras é atualizada pelo módulo de *daemon*, o qual repassa as mudanças de rota ocorridas no plano de controle para o módulo de tabela de regras, no plano de dados. Assim, o controlador Nox avalia quais fluxos são afetados pela atualização e os adéqua às novas regras de roteamento.

4.2 Migração de Topologias Virtuais no XenFlow

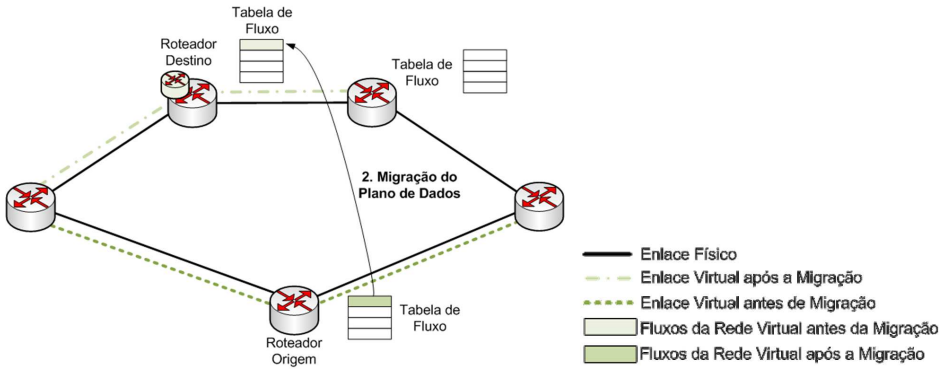
Em uma rede com o sistema XenFlow, um enlace virtual pode ser mapeado em um ou mais enlaces físicos. O encaminhamento é feito por uma tabela de fluxos programada dinamicamente pelo Controlador Nox. Logo, a topologia lógica fica desassociada da topologia física. A migração de nós virtuais em uma rede XenFlow, mostrada na Figura 4.4, se dá em três etapas: migração do plano de controle, migração do plano de dados e migração dos enlaces. O plano de controle é migrado entre dois nós físicos da rede, Figura 4.4(a), através do mecanismo de migração ao vivo de máquinas virtuais convencional do Xen [15]. Em seguida, a migração do plano, mostrada na

Figura 4.4(b), de dados é realizada da seguinte forma: os fluxos referentes ao roteador virtual migrado são selecionados e enviados para o roteador físico de destino; no destino, a definição dos fluxos é mapeada para atual configuração do roteador físico e do roteador virtual. Dessa forma, mantém-se a correspondência das portas de entrada e de saída do fluxo em relação ao comutador virtual do Domínio 0, de origem, no destino. Em seguida, os fluxos traduzidos são adicionados à tabela de fluxos do comutador OpenFlow do Domínio 0 de destino. Depois da migração do plano de dados e do plano de controle, ocorre a migração dos enlaces, como na Figura 4.4(c). A migração de enlaces envolve operações nos comutadores OpenFlow dos Domínios 0, de origem e destino, e em outros comutadores da rede. A migração de enlaces ocorre de forma a criar um caminho comutado entre os roteadores físicos, que hospedam vizinhos de um salto lógico do roteador virtual, até o roteador físico de destino da migração. Para tanto, os fluxos referentes ao roteador virtual migrante são selecionados no plano de dados do roteador físico de origem. A descrição desses fluxos é adicionada aos planos de dados dos roteadores físicos no caminho entre o roteador físico de destino e os roteadores físicos que são vizinhos do roteador físico de origem e não são vizinhos do roteador físico de destino. O objetivo de adicionar esses fluxos à tabela de fluxos, dos roteadores intermediários, é criar um caminho comutado entre os vizinhos do roteador físico de origem e o roteador físico de destino, tornando a migração nativa do Xen viável. No entanto, adicionar somente os fluxos já existentes às tabelas de fluxo dos nós físicos nesse 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 desse caminho. A Tabela de Regra é a aplicação do controlador que define as quais regras devem ser aplicadas a cada fluxo no momento de sua instanciação. Assim, quando um pacote, que não combine com nenhum fluxo previamente definido e tenha como destino o roteador virtual migrado, passar por um roteador intermediário, a regra introduzida na tabela de regras gerará um fluxo na tabela de fluxo do roteador intermediário para que o encaminhamento desse pacote siga o processo de comutação, ao invés do processo de roteamento. O controle do processo de migração de uma rede virtual no sistema XenFlow é desempenhado pelo nó que iniciou o processo de migração.

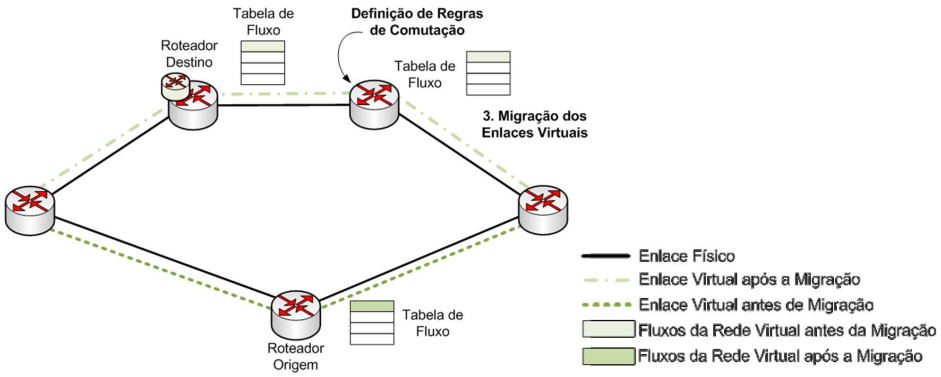
O sistema XenFlow agrega a primitiva de migração às redes virtuais, de forma simples e garantindo a perda zero de pacotes, como será mostrado no Capítulo 5. Nesse sistema, os elementos de redes virtuais podem assumir diversas funções, como por exemplo, a função de *middle boxes*. No entanto, o cenário em que a migração se torna mais crítica é na virtualização de roteadores, pois esses são elementos do núcleo da rede, responsáveis pelo



(a) Migração do Plano de Controle



(b) Migração do Plano de Dados.



(c) Migração de Enlaces.

Figure 4.4: As três etapas da migração de topologia virtual em uma rede XenFlow.

encaminhamento dos pacotes, e devem realizar o serviço de roteamento de forma transparente para as extremidades da rede.

Capítulo 5

Resultados Experimentais

Um protótipo do sistema foi desenvolvido para realizar a prova de conceito sobre como desenvolver a separação de planos de dados e controle, como realizar a migração de roteadores virtuais sem perda de pacotes e medir a quantidade de pacotes perdidos durante a migração do roteador virtual e quanto tempo dura a perda de comunicação do controle com a máquina virtual. O protótipo foi implementado em *Python* e fornece a separação de planos, uma interface de migração de roteadores virtuais e uma interface de migração de enlaces. O plano de dados foi implementado como uma aplicação do controlador Nox do comutador OpenFlow. A aplicação define os fluxos diretamente entre interface de entrada e de saída, na tabela de fluxos do comutador OpenFlow local, alterando o endereço MAC de destino dos pacotes encaminhados para o endereço MAC do próximo salto. Para a geração de pacotes, foi adotada a ferramenta *Iperf* [1]. Para medir o quanto de pacotes foi gerado, recebido e perdido, foi usada a ferramenta *tcpdump* [2], que captura os pacotes que são transmitidos em uma dada interface de rede. A perda de pacotes foi medida a partir da comparação das informações coletadas pelo *tcpdump* nas interfaces de rede responsáveis pela geração e pela recepção dos pacotes.

5.1 Cenário de Testes

O cenário de testes foi composto por quatro máquinas, como mostrado na Figura 5.1. 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 com

[1] http://iperf.sourceforge.net/
[2] http://www.tcpdump.org/

banda de 1Gb/s, executando o hipervisor Xen 4.0-amd64. Em uma dessas máquinas, foi instanciada 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 do conteúdo de um quadro Ethernet e o tamanho mais comum de MTU (*Maximum Transmission Unit*). Os resultados a seguir mostram a média dos resultados após 10 rodadas de cada experimento, em um intervalo de confiança de 95 %.

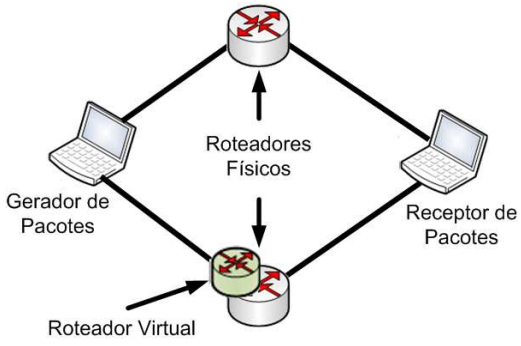
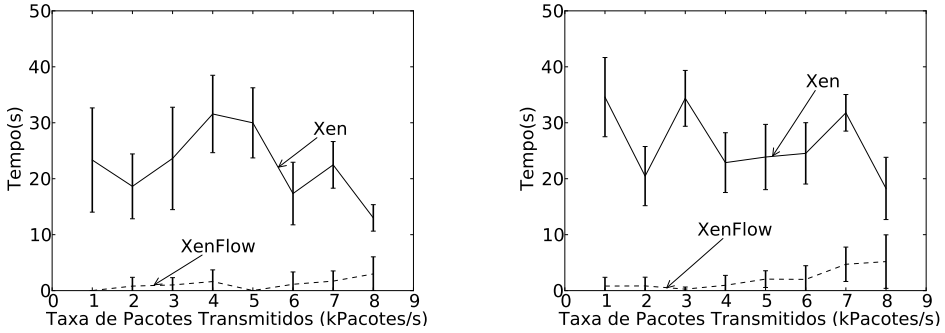


Figure 5.1: Cenário de avaliação do XenFlow. O cenário é composto por quatro máquinas físicas. Uma máquina age como geradora, outra como como receptora de pacote e outras duas, intermediárias, como roteadores.

5.2 Experimentos

O primeiro teste objetiva medir o tempo de suspensão do plano de controle durante a migração. O teste consiste no envio de pacotes de controle que são obrigados a passarem 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 5.2 mostra o tempo de suspensão do plano



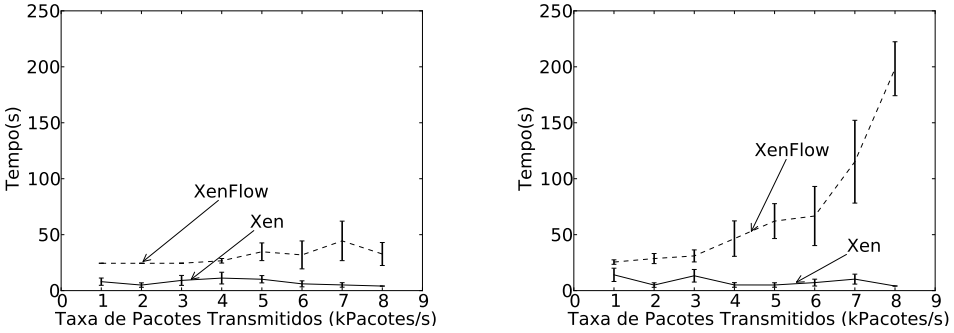
(a) Pacotes de 64 *bytes*.

(b) Pacotes de 1500 *bytes*.

Figure 5.2: Tempo de suspensão do plano de controle durante a migração.

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 plano de controle 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 chegou próximo de 40 segundos. Essa diferença se dá 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.

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 5.3 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



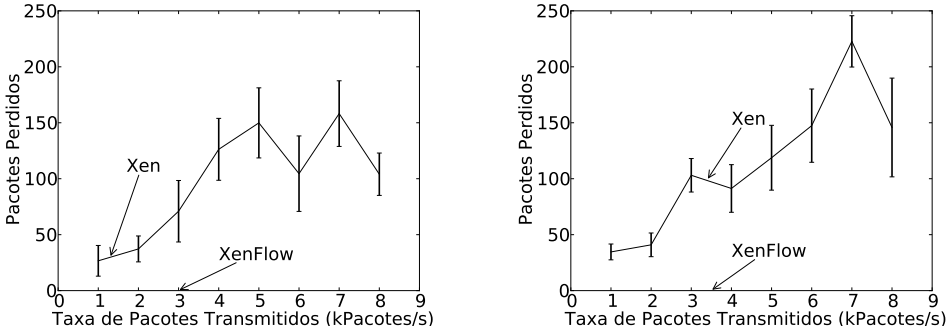
(a) Pacotes de 64 *bytes*.

(b) Pacotes de 1500 *bytes*.

Figure 5.3: Tempo de total de duração do processo de migração.

no tempo total que pode chegar até quatro vezes mais do que o tempo de migração de uma máquina virtual no Xen nativo. Esse resultado se dá pelo fato de que a migração do XenFlow envolve mais etapas do que a migração nativa do Xen, sendo que uma das etapas é a própria migração nativa do Xen. Parte do aumento no tempo da migração deve-se à migração dos fluxos, para a reconstrução do plano de dados no roteador físico de destino, e também à migração de enlaces, que é responsável por configurar a nova topologia da rede virtual sobre a rede física. A Figura 5.3(b) mostra que para pacotes de 1500 *bytes* há um aumento no tempo total da migração no XenFlow, a medida em que a taxa de pacotes aumenta. Para esse aumento é o fato de que para pacotes grandes, há a saturação dos enlaces de 100Mb/s a taxas de aproximadamente 8.000 pacotes/s UDP. No entanto, os enlaces que se encontram próximos a situação de saturação são os mesmos utilizados para a transferência das páginas de memória da máquina virtual de um roteador físico para outro. Sendo assim, como o tráfego das páginas de memória é realizado com base no TCP , que apresenta controle de fluxo, e o tráfego concorrente que está ocupando o enlace é sobre UDP , o comportamento esperado para o tráfego TCP é reduzir a sua vazão e, dessa forma, aumentar o tempo para a transferência das páginas de memória, consequentemente, há o aumento do tempo total de migração no sistema XenFlow.

O terceiro experimento apresenta o número de pacotes perdidos durante a migração. A Figura 5.4 revela que durante a migração no XenFlow não há perdas de pacotes. Durante a migração, há um momento em que o caminho antigo e o novo caminho, respectivamente, o que passa pelo roteador físico de origem e o que passa de destino, estão ativos. Assim, pode ocorrer de a máquina receptora receber pacotes desordenados e até mesmo duplica-



(a) Pacotes de 64 *bytes*.

(b) Pacotes de 1500 *bytes*.

Figure 5.4: Número de pacotes perdidos em função da taxa de pacotes transmitida.

dos. Embora gere um maior custo de processamento, esse procedimento é necessário para garantir que não haverá pacotes perdidos entre a interrupção do plano de dados antigos e a ativação do novo. A Figura 5.4 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 5.2. 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 com o aumento da taxa de pacotes encaminhados.

Capítulo 6

Conclusão

A Internet é um grande sucesso. No entanto, a sua arquitetura baseia-se em dois pilares, o serviço de transferência fim-a-fim e a pilha de protocolos TCP/IP , que apresentam limitações quanto a escalabilidade, mobilidade, gerenciamento e segurança. Nesse sentido, durante a evolução da rede, foram sendo desenvolvidos "remendos", funcionalidades que não estavam previstas no projeto original da Internet, para atender demandas pontuais na rede. Atualmente, a quantidade de "remendos" dificulta a criação de novas aplicações em seu núcleo e inibe o desenvolvimento de inovação no núcleo da rede, pois para desenvolver uma nova aplicação, é necessário que a nova aplicação seja compatível com os demais "remendos" e não afete o bom funcionamento da rede. Sendo assim, estudos argumentam pela necessidade de se desenvolver uma nova arquitetura para a Internet, a Internet do Futuro [2, 3].

As projeções para a Internet do Futuro apontam para um modelo pluralista, no qual a infraestrutura da rede deve ser capaz de dar suporte a diversas redes em paralelo, cada uma com sua pilha de protocolo e primitivas de gerenciamento próprias. Esse modelo visa garantir uma alta flexibilidade e permitir a inovação no núcleo da rede. Uma tecnologia que permite o desenvolvimento desse modelo é a virtualização de redes. A virtualização de redes consiste no desenvolvimento de uma camada de abstração sobre a infraestrutura física da rede, gerando ambientes virtuais que compartilham a infraestrutura física, isolados entre si, e que desenvolvem funções de elementos de rede, como roteadores ou comutadores. Nesse cenário de redes virtuais, surge uma nova primitiva de gerenciamento, a migração de redes virtuais. A migração de redes virtuais é uma primitiva que permite o remapeamento sob demanda dos elementos de rede virtual sobre a infraestrutura física da rede.

Este trabalho propõe o XenFlow, um sistema de processamento de fluxos para realizar a migração de redes virtuais, de forma robusta e eficiente. O

objetivo do sistema é realizar migrações de elementos de rede virtuais, com perda zero de pacotes e sem a necessidade de criar túneis ou usar mecanismos externos para a migração de enlaces. A proposta foi aplicada a roteadores virtuais e permite a separação de planos. O funcionamento do sistema baseia-se na existência de uma aplicação que controla o plano de dados baseada em regras de encaminhamento provinda de roteadores virtuais. As regras de encaminhamento são reproduzidas no plano de dados e são mantidas atualizadas por um *daemon* que executa em cada roteador virtual.

Os resultados mostram que a migração de um roteador virtual sobre o sistema XenFlow ocorre sem perda de pacotes. A migração sem perdas torna o sistema mais adequado ao cenário de redes virtuais do que a abordagem baseada na virtualização provida nativamente pela plataforma Xen. Os resultados mostram ainda que o tempo de interrupção do plano de controle no sistema proposto chega a ser até 40 vezes menor do que o tempo de interrupção na abordagem nativa do Xen. Os resultados também demonstram que o tempo total de migração aumenta quando comparamos o sistema XenFlow com a migração nativa do Xen. No entanto, esse último resultado deve-se ao fato de o XenFlow introduzir novas etapas em relação à migração do Xen, durante o processo de migração da topologia virtual. Contudo, o aumento do tempo total de migração não é um fator significativo para a migração de roteadores virtuais, quando considerado que a proposta apresenta perda zero de pacotes encaminhados e o tempo de suspensão do plano de controle é reduzido. O aumento no tempo total de migração apenas define o intervalo mínimo entre duas migrações consecutivas de um mesmo elemento virtual.

Como trabalhos futuros, pretende-se desenvolver novas aplicações de processamento de fluxo. Assim, as máquinas virtuais podem executar outras tarefas, além do roteamento. Dessa forma, as máquinas virtuais passam a executar serviços antes destinados à *middle boxes* e o sistema XenFlow torna-se um repositório de nós especializados, que podem ser migrados para qualquer nó físico na rede. Outra proposta de trabalho futuro é integrar o XenFlow com propostas de migração automatizada na rede, para gerenciá-la e otimizar a alocação de recursos físicos. Nesse sentido, pretende-se desenvolver algoritmos e heurísticas para a migração autônoma de elementos de redes virtuais, além do desenvolvimento de algoritmos para realizar a migração de enlaces automaticamente, otimizando o uso dos enlaces da rede e garantindo a conectividade entre os nós virtuais.

Os resultados obtidos nesse trabalho foram aceitos para publicação no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2011, realizado em maio de 2011. Os resultados forma publicados no artigo XenFlow: Um Sistema de Processamento de Fluxos Robusto e Eficiente para Migração em Redes Virtuais [30].

Bibliografia

[1] N. Fernandes, M. Moreira, I. Moraes, L. Ferraz, R. Couto, H. Carvalho, M. Campista, L. Costa, and O. Duarte, "Virtual Networks: Isolation, Performance, and Trends," *Annals of Telecommunications*, pp. 1–17, 2010.

[2] M. Moreira, N. Fernandes, L. Costa, and O. Duarte, "Internet do futuro: Um Novo Horizonte," *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC 2009*, pp. 1–59, 2009.

[3] D. Clark, R. Braden, K. Sollins, J. Wroclawski, D. Katabi, and M. I. O. T. C. L. F. C. SCIENCE, "New Arch: Future Generation Internet Architecture." 2004.

[4] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield, "Plutarch: An Argument for Network Pluralism," in *ACM SIGCOMM workshop on Future directions in network architecture*, pp. 258–266, Sept. 2003.

[5] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet," in *CoNEXT*, ACM, Dec. 2008.

[6] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, and L. Mathy, "Network Virtualization Architecture: Proposal and Initial Prototype," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, VISA '09, (New York, NY, USA), pp. 63–72, ACM, 2009.

[7] N. Feamster, L. Gao, and J. Rexford, "How to Lease the Internet in your Spare Time," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 61–64, Jan. 2007.

[8] S. Ratnasamy, S. Shenker, and S. McCanne, "Towards an Evolvable Internet Architecture," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, no. 4, pp. 313–324, 2005.

[9] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual Routers on the Move: Live Router Migration as a Network-Management Primitive," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, no. 4, pp. 231–242, 2008.

[10] R. Bolla, R. Bruschi, F. Davoli, and A. Ranieri, "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*, pp. 49–54, ACM, 2009.

[11] A. Greenhalgh, F. Huici, M. Hoerdt, P. Papadimitriou, M. Handley, and L. Mathy, "Flow Processing and the Rise of Commodity Network Hardware," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 2, no. 2, pp. 20–26, 2009.

[12] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Towards High Performance Virtual Routers on Commodity Hardware," in *Proceedings of the 2008 ACM CoNEXT Conference*, pp. 1–12, ACM, 2008.

[13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, no. 2, pp. 69–74, 2008.

[14] P. Pisa, N. Fernandes, H. Carvalho, M. Moreira, M. Campista, L. Costa, and O. Duarte, "OpenFlow and Xen-Based Virtual Network Migration," in *Communications: Wireless in Developing Countries and Networks of the Future* (A. Pont, G. Pujolle, and S. Raghavan, eds.), vol. 327 of *IFIP Advances in Information and Communication Technology*, pp. 170–181, Springer Boston, 2010.

[15] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pp. 273–286, USENIX Association, 2005.

[16] F. Bellard, "QEMU, a Fast and Portable Dynamic Translator," in *Proceedings of the USENIX Annual Technical Conference, FREENIX Track*, pp. 41–46, 2005.

[17] K. Kolyshkin, "Virtualization in Linux," *OpenVZ*, 2006.

[18] E. Keller and J. Rexford, "The Platform as a Service Model for Networking," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, p. 4, USENIX Association, 2010.

[19] L. Peterson, S. Muir, T. Roscoe, and A. Klingaman, "PlanetLab Architecture: An overview," *PlanetLab Consortium May*, 2006.

[20] G. Kontesidou and K. Zarifis, "Openflow Virtual Networking: A Flow-Based Network VirtualizationArchitecture," 2009.

[21] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI Veritas: Realistic and Controlled Network Experimentation," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 3–14, ACM, 2006.

[22] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A Network Virtualization Layer." 2009.

[23] M. Yu, J. Rexford, M. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," in *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, pp. 351–362, ACM, 2010.

[24] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, pp. 3–3, USENIX Association, 2010.

[25] Y. Wang, J. van der Merwe, and J. Rexford, "VROOM: Virtual Routers on the Move," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking*, Citeseer, 2007.

[26] D. M. F. Mattos, N. C. Fernandes, L. P. Cardoso, V. T. da Costa, L. H. Mauricio, F. P. B. M. Barretto, A. Y. Portella, I. M. Moraes, M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "OMNI: Uma Ferramenta para Gerenciamento Autônomo de Redes OpenFlow," in *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011 - Salão de Ferramentas*, (Campo Grande, MS), May 2011.

[27] M. Nascimento, C. Rothenberg, M. Salvador, and M. Magalhães, "QuagFlow: Partnering Quagga with OpenFlow," in *Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, pp. 441–442, ACM, 2010.

[28] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, and S. Shenker, "Extending Networking into the Virtualization Layer," (New York City, NY), Oct. 2009.

[29] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an Operating System for Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, no. 3, pp. 105–110, 2008.

[30] D. M. F. Mattos, N. C. Fernandes, and O. C. M. B. Duarte, "XenFlow: Um Sistema de Processamento de Fluxos Robusto e Eficiente para Migração em Redes Virtuais," in *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*, (Campo Grande, MS), May 2011.