

Provendo Isolamento e Qualidade de Serviço em Redes Virtuais *

Natalia Castro Fernandes e Otto Carlos Muniz Bandeira Duarte

¹Grupo de Teleinformática e Automação (GTA/PEE/UFRJ)
Universidade Federal do Rio de Janeiro (UFRJ) - Rio de Janeiro - Brasil

Resumo. A virtualização de redes é uma nova tecnologia que permite o compartilhamento de recursos de infraestrutura da rede. No entanto, prover qualidade de serviço (QoS) e isolamento em redes virtuais constitui um desafio. Nesse trabalho, é proposto um sistema para isolar, prover QoS e gerenciar redes virtuais baseadas na plataforma de virtualização Xen. O elemento principal da proposta é o controlador dos recursos físicos que isola as redes virtuais e disponibiliza diferentes parâmetros para caracterizar os acordos de nível de serviço de cada rede. Além disso, o sistema de controle e gerenciamento proposto disponibiliza dois níveis de controle de QoS, sendo um para a rede virtual e outro para o provedor de infraestrutura. Foi desenvolvido um protótipo cujos resultados comprovam que o novo mecanismo garante a conformidade com os acordos contratados e ainda usa de forma mais eficiente os recursos físicos quando comparado a outras propostas.

Abstract. Network virtualization is a new technology that enables resource sharing in the network infrastructure. Providing quality of service (QoS) and isolation for virtual networks, however, is a challenge. In this work, we propose a system to isolate, provide QoS, and manage virtual networks based on the Xen virtualization platform. The main module of the proposal is the physical resource controller, which isolates virtual networks and offers different parameters to describe the service level agreements of each virtual network. Furthermore, the proposed control and management system provides two levels of QoS control, one for the virtual network operator and the other for the infrastructure provider. We developed a prototype whose results indicate that the new mechanism ensures compliance with contracted agreements and also is more efficient in the use of physical resources when compared to other proposals.

1. Introdução

A virtualização de redes é uma nova tecnologia apontada como uma solução promissora para o problema do ‘engessamento’ da Internet [Feamster et al., 2007, Moreira et al., 2009]. Com a virtualização, a rede física é subdividida em fatias chamadas de redes virtuais, cada uma com suas próprias características, como pilha de protocolos e sistema de endereçamento. Com isso, o núcleo da rede é flexibilizado e passa a dar suporte à inovação. Alguns aspectos principais que devem ser providos para as redes virtuais são o isolamento, de forma que uma rede virtual não interfira nas demais, o desempenho no encaminhamento de pacotes e o provimento de qualidade de serviço [Fernandes et al., 2010, Carapinha e Jiménez, 2009].

As redes virtuais podem ser criadas utilizando-se *softwares* para virtualização de servidores. Nesse caso, roteadores virtuais são criados sobre a plataforma de computadores pessoais, como é feito com o Xen [Egi et al., 2007]. O Xen permite a criação de

*Este trabalho foi apoiado por recursos da FAPERJ, FINEP, CAPES, CNPq, FUJB e FUNTEL.

máquinas virtuais que simulam uma máquina física com *hardware* e sistema operacional próprios. Com isso, cada máquina virtual Xen pode ser configurada como um roteador por *software* diferente, todas compartilhando tanto a máquina física quando os enlaces físicos. Embora o Xen seja apontado como uma boa alternativa para a criação de redes virtuais, ele apresenta deficiências no isolamento e no desempenho para encaminhamento de pacotes, além de não disponibilizar nenhum tipo de suporte para provimento de qualidade de serviço (*Quality of Service* - QoS) nas máquinas virtuais [Egi et al., 2007, Egi et al., 2008, Fernandes et al., 2010].

Nesse artigo é apresentado um sistema para controle e gerenciamento de recursos de redes virtuais criadas sobre a plataforma Xen, chamado de Xen Network Manager (XNetMan). O XNetMan disponibiliza um controle diferenciado dos recursos compartilhados entre as redes virtuais, tais como CPU, memória e banda, garantindo o isolamento entre as redes virtuais e o cumprimento dos acordos de nível de serviço (*Service Level Agreement* - SLA) de cada rede virtual. Além disso, o XNetMan provê uma garantia de QoS dentro de ambientes virtualizados com o Xen, permitindo uma diferenciação dos parâmetros de QoS pelo provedor de infraestrutura e pelo operador da rede virtual.

O XNetMan monitora o tráfego de cada rede virtual e, com base nesses dados, o controlador reajusta os parâmetros de cada rede para garantir as reservas de recursos, aplicando punições sobre as redes virtuais que violam os acordos de nível de serviço (SLAs). O XNetMan disponibiliza ainda um módulo para controle de acesso de novas redes virtuais, que verifica se é possível atender aos requisitos da nova rede na máquina física. Um protótipo foi implementado e os resultados mostram que o XNetMan garante uma alta conformidade entre as características do tráfego encaminhado e as características definidas nos SLAs, ao mesmo tempo em que utiliza de forma mais eficiente o enlace quando comparado a outras propostas. Os testes para avaliar características como isolamento e atendimento dos SLAs mostram que o sistema proposto apresenta reduções de até 18 vezes no atraso de tráfegos com requisitos de prioridade, quando comparado a sistemas sem suporte a QoS no provedor de infraestrutura. Além disso, o sistema se mostrou até cinco vezes mais eficiente que outras ferramentas no atendimento dos SLAs.

O restante do artigo está organizado da seguinte forma. Na Seção 2, são apresentados os trabalhos relacionados. Nas Seções 3 e 4, são descritos o modelo para virtualização de rede e a arquitetura do Xen. Na Seção 5, é apresentado o sistema proposto e, na Seção 6, o sistema é analisado. Por fim, na Seção 7, são apresentadas as conclusões.

2. Trabalhos Relacionados

O controle e o gerenciamento das redes virtuais são discutidos para as principais plataformas de virtualização existentes e incluem o controle global e local da rede. O controle global inclui operações como a instanciação de nós e enlaces virtuais. Já o controle local monitora os recursos de cada nó físico que são atribuídos a cada rede virtual, tratando o isolamento entre redes.

Schaffrath *et al.* propuseram uma arquitetura para controle global das redes virtualizadas [Schaffrath et al., 2009]. A proposta foi implementada com o Xen e assume uso de um controle centralizado responsável por criar fatias da rede. Outras abordagens semelhantes para o controle global são encontradas nos *testbeds* baseados em virtualização, como o GENI [GENI, 2009]. O controle de acesso dos pesquisadores à rede de teste é feito através de uma entidade central chamada de câmara de compensação (*Clearing House*). A câmara de compensação monitora quais nós físicos e serviços estão disponíveis em cada um dos *testbeds* federados, quem está autorizado a usá-los e quais fatias estão agendadas para cada pesquisador. As entidades com controle global também podem

realizar outras funções como a migração. Houidi *et al.* propuseram um sistema de controle global baseado em multi-agentes para alocação dinâmica de recursos para as redes virtuais através do uso de migração [Houidi et al., 2010]. Ao notar que os recursos estão escassos, o agente no nó físico busca um nó físico semelhante para receber uma ou mais de suas máquinas virtuais.

Os sistemas de controle global não lidam com a divisão dos recursos dentro da máquina física, assumindo que as fatias construídas já apresentam um bom isolamento criado por algum mecanismo de controle local. Egi *et al.* investigam a construção de uma plataforma de roteadores virtuais usando o Xen com controle local, avaliando o provimento de isolamento e justiça entre as redes [Egi et al., 2007]. Os autores investigam a utilização de diferentes planos de dados, assumindo o roteamento por um domínio privilegiado e pela máquina virtual, e avaliam a capacidade de se compartilhar os recursos entre as redes virtuais. O trabalho, no entanto, não apresenta mecanismos de gerenciamento que permitam alocar recursos para redes específicas e diferenciar ou priorizar tráfegos para garantir o QoS das redes virtuais. McIlroy e Sventek propuseram um controle local baseado em Xen para a Internet atual no qual cada fluxo que precisa de QoS é alocado para uma máquina virtual, chamada de *QoS routelet* [McIlroy e Sventek, 2006]. Cada máquina virtual aplica, então, as suas políticas de QoS em todo o tráfego recebido. Os autores observam que não é possível garantir QoS no sentido mais estrito com esse modelo, uma vez que o escalonador do Xen não é apropriado para essa tarefa. Outros problemas da proposta são a escalabilidade, pois é preciso uma máquina virtual por fluxo com QoS, e o baixo desempenho no encaminhamento de pacotes das máquinas virtuais prejudicaria o atendimento dos fluxos com QoS.

Fernandes e Duarte propuseram o *Xen Network Monitor* (XNetMon), o qual é um monitor de redes para o Xen que garante o isolamento entre redes virtuais e impede que ações maliciosas de uma rede prejudiquem as demais [Fernandes e Duarte, 2010]. Embora esse sistema de controle local permita a diferenciação dos recursos para cada rede, essa diferenciação é restrita. O XNetMon tem como objetivo principal impedir que redes maliciosas prejudiquem outras redes e não se foca no desenvolvimento de uma interface de gerenciamento e nas garantias de QoS para redes virtuais.

Outras plataformas de virtualização também demandam mecanismos para garantir o isolamento entre redes. O Trellis [Bhatia et al., 2008] é um sistema para garantir o isolamento em sistemas baseados na plataforma VINI [VINI, 2010], a qual é baseada na virtualização a nível de sistema operacional. Devido a essa opção de virtualização, o VINI apresenta uma plataforma menos flexível que o Xen para a inovação, além de apresentar menor desempenho no encaminhamento de pacotes que o Xen com separação de planos [Egi et al., 2008]. A plataforma OpenFlow [McKeown et al., 2008], que também permite criar o ambiente de redes pluralistas, se baseia na premissa de uma rede com comutadores programáveis e um plano de controle centralizado. Para atribuir os recursos compartilhados e isolar as redes virtuais, a plataforma OpenFlow utiliza a ferramenta FlowVisor [Sherwood et al., 2010]. O FlowVisor controla o uso de memória e CPU dos comutadores programáveis e controla a divisão dos espaços de rede, que determina quais características definem cada rede.

O XNetMan, proposto nesse artigo, é um sistema para controle local no Xen que além de apresentar as funcionalidades do XNetMon, também provê um controle mais preciso e eficiente do compartilhamento dos recursos, com suporte a mais definições dentro dos SLAs. Além disso, o XNetMan provê um controle de QoS tanto pelo provedor de infra-estrutura quanto pelo operador da rede virtual, obtendo resultados satisfatórios na redução dos atrasos em redes com prioridade. O XNetMan também possui uma interface

de gerenciamento para controle dos recursos de cada rede virtual, que permite não apenas definir redes virtuais, mas monitorar violações de SLAs e avaliar a possibilidade de hospedagem de novas redes virtuais sobre o mesmo substrato físico.

3. Modelo de Redes Virtualizadas

No cenário atual da Internet, existem duas entidades principais: os provedores de serviço (*Internet Service Providers - ISP*), que oferecem o acesso a rede através de sua infraestrutura para usuários finais ou para outros ISPs, e os provedores de aplicações, que oferecem serviços através da Internet. Ao considerar o uso de redes virtuais, onde diversos provedores de infraestrutura podem co-existir e competir, uma nova camada de abstração deve ser acrescentada, adicionando duas novas entidades ao modelo da Internet do Futuro: os provedores de redes virtuais e os operadores de redes virtuais [Schaffrath et al., 2009, Achemlal e et al., 2010]. O esquema desse novo modelo está na Figura 1(a).

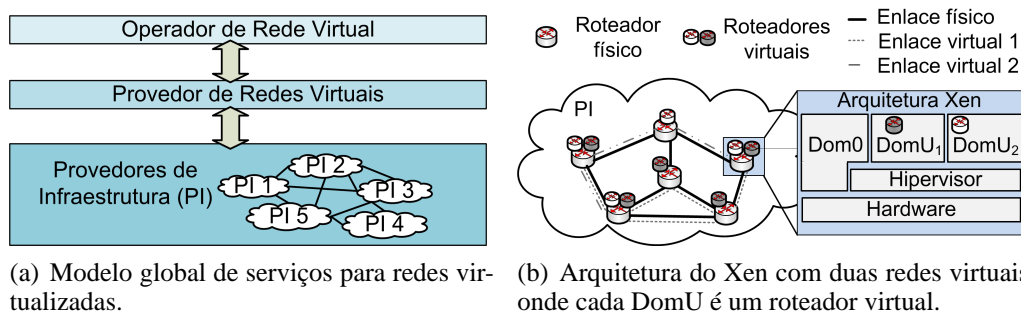


Figura 1. Modelo global e modelo local para virtualização de redes.

Nesse cenário, o provedor de infraestrutura possui a infraestrutura física, gerenciando os recursos físicos e vendendo fatias de rede para o provedor de redes virtuais. O provedor de redes virtuais é responsável por unir diversos provedores de infraestrutura para formar uma topologia virtual de acordo com os requisitos do operador de rede virtual. O operador de rede virtual é responsável por instalar e operar a rede virtual, oferecendo serviços na rede ou oferecendo a sua rede virtual para servidores de aplicação.

Nesse artigo, assume-se que é necessário o provimento de QoS pelo provedor de infraestrutura, pois as regras de QoS internas à rede virtual não são suficientes para garantir o QoS dessa rede. De fato, as garantias de QoS só são possíveis quando existir a diferenciação entre as redes, estabelecendo, por exemplo, qual rede possui maior prioridade no encaminhamento de pacotes. Assim, o XNetMan funciona como parte do provedor de infraestrutura, oferecendo uma interface de controle de requisitos da rede e de parâmetros de QoS para o provedor de redes virtuais e para o operador de redes virtuais.

4. O Xen e as Redes Virtualizadas

Assume-se o uso do Xen como tecnologia de virtualização no provedor de infraestrutura. Nesse modelo, cada máquina virtual atua como um roteador. Assim, uma rede virtual é uma fatia da rede física, definida como um conjunto de roteadores virtuais e os enlaces virtuais correspondentes. A arquitetura do Xen com o modelo de redes virtuais está representada na Figura 1(b).

O Xen apresenta problemas para garantir o isolamento entre redes virtuais, assim como para garantir um alto desempenho no encaminhamento de pacotes [Egi et al., 2007, Fernandes et al., 2010, Fernandes e Duarte, 2010]. De fato, esses problemas existem porque o compartilhamento dos recursos físicos utilizados para operações de entrada e saída (E/S) é efetuado em uma máquina virtual privilegiada chamada de Domínio 0 (Dom0). O

Dom0 é uma máquina virtual especial que serve como interface de gerenciamento entre o usuário e o hipervisor e que escalona o acesso das máquinas virtuais ao *hardware* físico. Por ter acesso direto ao *hardware*, o Dom0 também funciona com um domínio de *drivers*, mapeando os *drivers* virtuais das máquinas virtuais sem privilégio (*Unprivileged Domain- DomU*) nos *drivers* físicos. Uma vez que o hipervisor não controla o uso dos recursos do Dom0 por cada DomU, um DomU pode afetar a operação de E/S de outros DomUs através da exaustão dos recursos do Dom0. Isto é uma importante vulnerabilidade do Xen, pois o isolamento entre as redes fica prejudicado.

Outro problema do Xen é o desempenho no encaminhamento de pacotes. Uma vez que os DomUs não têm acesso direto ao *hardware*, a taxa de encaminhamento através dos roteadores virtuais é muito inferior à taxa alcançada com um Linux nativo [Fernandes et al., 2010]. Para solucionar esse problema, foi proposto o paradigma da separação de planos. Nesse paradigma, o encaminhamento de pacotes das redes virtuais é feito pelo Dom0, que tem acesso direto ao *hardware*, enquanto que o controle da rede virtual continua sendo feito pelo DomU [Egi et al., 2007]. Para tanto, é preciso criar e manter uma réplica no Dom0 do plano de dados criado dentro da máquina virtual (DomU). Portanto, com a separação de planos, cada rede virtual possui o seu próprio plano de controle sem perder desempenho de encaminhamento. Por outro lado, perde-se flexibilidade no encaminhamento de dados, pois todas as redes passam a compartilhar um mesmo plano de dados.

5. O Sistema Proposto

O XNetMan é um sistema para controlar e gerenciar redes virtuais baseadas em Xen. Os principais objetivos do XNetMan são o isolamento das redes virtuais, através do controle do uso dos recursos compartilhados do Dom0, como CPU, memória e banda passante, e o provimento de QoS. O XNetMan garante um isolamento robusto entre as máquinas virtuais além de permitir uma diferenciação dos recursos físicos atribuídos a cada rede virtual de acordo com os SLAs. Outras funcionalidades do XNetMan incluem o monitoramento de violações de SLA, o controle de acesso de novas redes virtuais à máquina física e a modificação dinâmica dos parâmetros da rede.¹

A arquitetura do XNetMan está descrita na Figura 2. O principal módulo do XNetMan é o controlador, o qual é responsável pelo monitoramento e punição de redes virtuais para garantir o isolamento entre as redes. Os demais módulos são funções de gerenciamento, dentre os quais se destacam os módulos de QoS para operador e provedor, que possibilitam a diferenciação de tráfego entre redes, definindo quais redes tem prioridade sobre as outras, e dentro das redes, definindo quais pacotes tem prioridade dentro da rede.

5.1. O Controlador de Redes Virtuais

O controlador de redes virtuais é responsável pelo monitoramento do uso dos recursos físicos por cada rede virtual e pela punição das redes virtuais que excedem o uso dos recursos contratados no SLA. O controle é feito através de observações a curto e a longo prazo da utilização da CPU, da memória e da banda no Dom0.

O controlador assume a existência de um conjunto de características para especificar o uso de cada recurso físico por cada rede de acordo com o SLA contratado, sendo

¹Os módulos de comunicação segura e separação de planos são comuns ao XNetMon [Fernandes e Duarte, 2010]. Logo, o XNetMan garante uma comunicação segura entre os DomU e o Dom0 para troca de dados de configuração e controle. São oferecidas duas opções: separação de planos, caso a rede necessite de alto desempenho, ou por roteamento pela máquina virtual, caso a rede precise de alta flexibilidade no plano de dados.

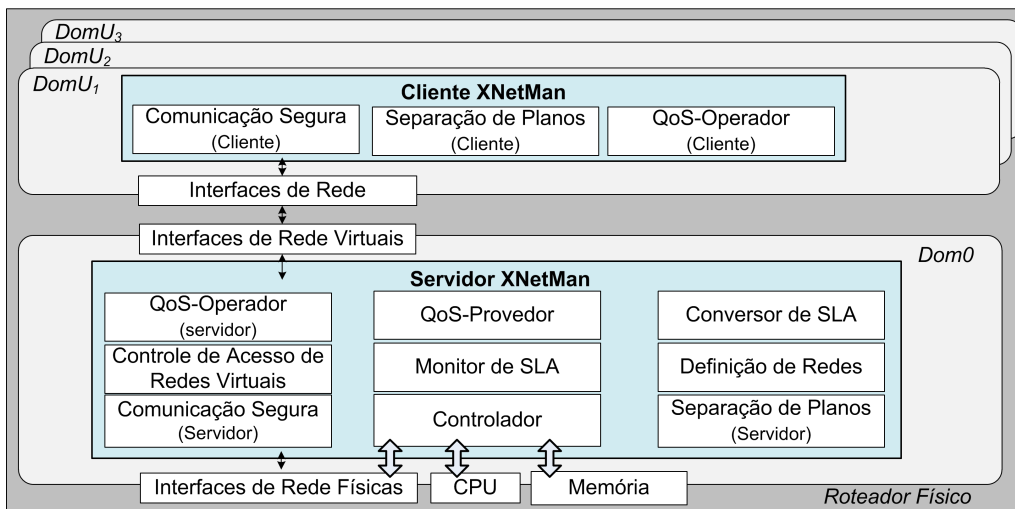


Figura 2. Arquitetura do XNetMan com clientes nos DomUs e o servidor no Dom0.

eles: os recursos reservados a curto prazo (R_{c_i}), que são os recursos que devem ser garantidos para a rede i sempre que houver demanda em um intervalo de tempo curto I_c ; os recursos reservados a longo prazo (R_{l_i}), que são recursos que devem ser garantidos apenas em um intervalo longo de tempo I_l , caso exista demanda; a reserva a curto prazo exclusiva (R_{e_i}), que é um tipo de reserva a curto prazo que não deve ser disponibilizada a outras redes, mesmo que não exista demanda da rede i ; e a reserva máxima (R_{max_i}), que é o uso máximo do recurso que a rede i pode usar dentro de I_l . A rede pode optar também por uma reserva restrita à R_{c_i} , o que significa que o tráfego da rede i não deve ultrapassar nunca o uso da reserva mínima. Parâmetros para a construção de limitadores, conformadores e prioridades de tráfego também fazem parte do SLA, mas são tratados pelos módulos de QoS do operador e QoS do provedor. Com base nesse conjunto de características, é possível especificar diferentes tipos de perfis de uso de recursos físicos. A escolha dos parâmetros de cada rede deve variar de acordo com a necessidade do operador da rede virtual e do preço que ele está disposto a pagar por sua fatia de rede.

O controlador proposto se baseia nas características especificadas para cada rede virtual i para determinar se a demanda da rede virtual i , D_i , está dentro ou fora dos SLAs determinados. O Algoritmo 1 é uma versão simplificada de como o XNetMan calcula as punições de cada rede². O Algoritmo 1 deve ser executado a cada intervalo curto I_c para cada um dos recursos monitorados do Dom0, ou seja, a banda de saída de cada enlace físico, a CPU e a memória. Esse algoritmo assume que a D_i da rede i no último I_c é conhecida e esse valor é utilizado como uma estimativa da demanda da rede no próximo I_c . Com base na estimativa de demanda de todas as redes, os recursos são divididos entre as redes virtuais. O Algoritmo 1 assume ainda que são conhecidos o número de redes virtuais hospedadas (N), o total de recursos físicos disponibilizados para as redes virtuais (R_t), o total de recursos utilizados por cada rede até o momento dentro do intervalo longo ($utilizado_i$) e o peso atual de cada rede virtual i ($peso_i$). O peso é um parâmetro gerado pelo controlador a cada I_c para controlar o uso da reserva a longo prazo de cada rede, de forma que quanto maior o peso, maior o acesso da rede aos recursos não-reservados a curto prazo.

Primeiramente, o algoritmo zera todas as punições, calcula o total de recursos não

²A versão estendida considera os recursos como a reserva a curto prazo garantida, a reserva máxima e o uso restrito à reserva a curto prazo.

Algoritmo 1: Cálculo da punição para cada rede virtual.

Entrada: $R_t, N, I_c, I_l, R_c[], R_l[], peso[], utilizado[], D[]$
Saída: $punicao[]$

```
1 Zerar( $punicao[]$ );  $total\_disponivel = R_t - \sum_{i=1}^N (R_c[i]); peso\_total = \sum_{i=1}^N peso[i];$ 
2 para  $rede = 1$  até  $N$  faça
3   se ( $utilizado[rede] < R_l[rede] \cdot I_l$ ) então
4      $proximo\_liberado[rede] = R_c[rede] + total\_disponivel \cdot peso[rede] / peso\_total;$ 
5   senão  $proximo\_liberado = R_c[rede];$ 
6   se ( $D[rede] > proximo\_liberado[rede]$ ) então
7      $punicao[rede] = 1 - proximo\_liberado[rede] / D[rede];$ 
8 fim
9 enquanto ( $verifica(proximo\_liberado[], D[], N) == 1$ ) faça
10   $peso\_total = 0;$ 
11  para  $rede = 1$  até  $N$  faça
12    se ( $proximo\_liberado > D[rede]$ ) então  $proximo\_liberado = D[rede];$ 
13    se ( $(D[rede] > proximo\_liberado[rede]) \& (utilizado[rede] < R_l[rede] \cdot I_l)$ )
14      então  $peso\_total + = peso[rede];$ 
15 fim
16  $sobra = R_t - \sum_{i=1}^N (proximo\_liberado[i]);$ 
17 para  $rede = 1$  até  $N$  faça
18   se ( $(D[rede] > proximo\_liberado[rede]) \& (utilizado[rede] < R_l[rede] \cdot I_l)$ )
19   então  $proximo\_liberado[rede] + = sobra \cdot peso[rede] / peso\_total;$ 
20   se ( $proximo\_liberado[rede] < D[rede]$ ) então
21      $punicao[rede] = 1 - proximo\_liberado[rede] / D[rede];$ 
22   senão  $punicao[rede] = 0;$ 
23 fim
24 fim
```

reservados e o somatório dos pesos de todas as redes. Esses valores são utilizados para calcular o volume de recursos que devem ser liberados para cada rede no próximo I_c , o que é representado pela variável $proximo_liberado$. Assim, caso a rede virtual ainda não tenha gastado toda a sua reserva a longo prazo, ela recebe como $proximo_liberado$ a sua reserva a curto prazo mais uma fatia proporcional ao seu peso dos recursos não-reservados. Caso contrário, a política do XNetMan define que a vazão da rede deve ser limitada pela reserva a curto prazo desta rede. Com isso, todas as redes que possuem uma demanda superior ao $proximo_liberado$ recebem uma punição proporcional ao uso excedente. Para aperfeiçoar a distribuição dos recursos, o XNetMan verifica, após esses cálculos, se alguma rede recebeu uma fatia dos recursos superior a demanda estimada, o que é feito pela função $verifica()$ na linha 7. Caso isso ocorra, as redes que tiveram uma demanda inferior ao $proximo_liberado$ tem essa variável reduzida até a sua demanda e a diferença é disponibilizada para as demais redes. O $peso_total$ também é recalculado, considerando apenas as redes que devem participar da divisão dos recursos disponibilizados. Em seguida, o $proximo_liberado$ das redes que tem uma demanda excedente e que ainda não usaram toda a reserva a longo prazo é recalculado com base nos recursos disponibilizados proporcionalmente ao peso da rede. Por fim, as punições de todas as redes são atualizadas. Esse processo é repetido até que nenhuma rede possua um $proximo_liberado$ superior a sua demanda. Com isso, os recursos físicos são distribuídos de forma proporcional ao peso e à demanda de cada rede.

Outro ponto importante do módulo controlador é o cálculo do valor do peso de cada rede em cada recurso em cada intervalo curto. O peso é uma variável adaptativa, calculada com base no que a rede já utilizou dos recursos ($utilizado$) e na reserva a longo prazo. A base do algoritmo para o ajuste do peso é dar a cada rede um peso proporcional a razão entre o volume de dados ainda não gasto da reserva a longo prazo da rede e o volume ainda não gasto da reserva a longo prazo de todas as redes. Com isso, as redes que pos-

suem maior reserva a longo prazo disponível ganham prioridade na alocação dos recursos disponíveis, recebendo uma fatia maior dos recursos caso haja demanda. Consequentemente, aumenta-se a probabilidade de que a reserva a longo prazo seja integralmente provida a todas as redes virtuais.

5.1.1. Aplicação da Punição

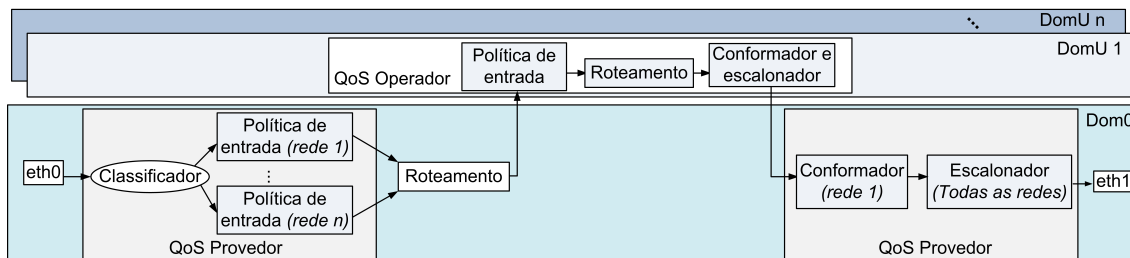
A punição é calculada no XNetMan com base na demanda estimada de cada rede virtual para o próximo intervalo curto e no volume de recursos liberado para cada rede virtual, como mostrado no Algoritmo 1. Embora o valor percentual da punição seja calculado com base no mesmo algoritmo para todos os recursos monitorados, a forma de como é efetuada a punição difere e depende de qual recurso está sendo violado. Quando existe uma sobrecarga dos recursos de processamento por uma determinada rede virtual, a punição dessa rede é aplicada em forma de descarte de um percentual pacotes em todas as interfaces de entrada. Logo, um percentual de pacotes correspondente à punição relativos à máquina virtual que violou as SLAs deixará de chegar e ser processado no Dom0, reduzindo os gastos de recursos de processamento que seriam usados com o encaminhamento de pacotes. No caso de uma sobrecarga de banda em uma interface de saída, os pacotes destinados àquela interface são então descartados. Cabe observar que a punição de rede reduz a vazão na interface de saída, mas os gastos de processamento para o Dom0 se mantêm e são contabilizados no uso de CPU da rede virtual. Embora a filtragem e o encaminhamento de pacotes impliquem um gasto de memória no Dom0, esse é pequeno e pode ser desconsiderado. Portanto, o consumo de memória é estimado apenas pelo volume de regras de filtragem e de encaminhamento de pacotes armazenadas no Dom0 em cópia do que existe no DomU para cada rede virtual. Assim, uma rede virtual que opte por fazer o encaminhamento de pacotes pela máquina virtual tem custo de memória zero para o XNetMan, pois essas duas operações são feitas dentro da máquina virtual. Já as redes que optarem pela separação de planos deverão observar também um volume máximo de regras de filtragem e de tamanho de tabela de encaminhamento.

A punição devido ao gasto excessivo de memória de uma determinada rede virtual que viola a SLA implica a eliminação percentual de rotas da tabela de encaminhamento desta rede virtual que consta no Dom0. Para impedir a perda de pacotes, uma rota padrão é criada encaminhando o pacote para a máquina virtual correspondente, na qual o a tabela de roteamento completa está presente com todas as rotas. Assim, como o encaminhamento pelo DomU é bem mais lento, se uma rede que opta pela separação de planos não deseja perder desempenho com o encaminhamento de pacotes, ela deve controlar o número de rotas instaladas no Dom0.

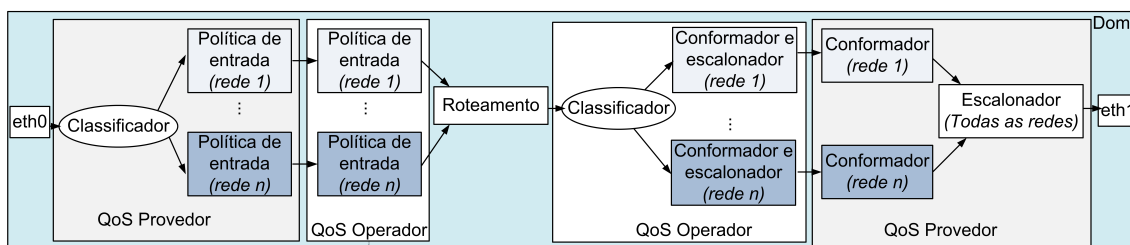
5.2. Provisão de QoS

Uma das principais vantagens do XNetMan com relação a outras propostas é o suporte para provisão de QoS na infraestrutura de redes virtuais. Embora um roteador virtual que esteja contido dentro de uma máquina virtual possa configurar um controle de tráfego para os pacotes que chegam até a máquina virtual, um controle neste nível decididamente não é suficiente para garantir a qualidade de serviço. Uma máquina virtual não possui controle sobre o que ocorre com o seu tráfego no Dom0 até a sua chegada na máquina virtual e após deixar a máquina virtual, o que pode incluir atrasos e descarte de pacotes. Assim, é necessário incluir algumas regras dentro do Dom0 para diferenciação da qualidade de serviço ofertada para cada rede virtual na chegada e na saída do pacote, como mostra a Figura 3(a). Nessa figura, são apresentados os pontos aonde o provedor de infraestrutura e o operador de rede virtual podem inserir suas configurações de QoS. De

acordo com esse esquema, o provedor de infraestrutura poderia inserir prioridades para o acesso a rede física pelas redes virtuais, assim como poderia limitar o tráfego de uma rede virtual. Da mesma forma, o operador da rede virtual poderia escolher quais fluxos precisam ter a sua vazão limitada e quais devem ser priorizados.



(a) Provimento de QoS no XNetMan assumindo roteamento pela máquina virtual.



(b) Provimento de QoS no XNetMan assumindo o uso de separação de planos, o que implica na transferência das regras de QoS do operador do DomU para o Dom0.

Figura 3. Provimento de QoS dentro da rede virtual (QoS operador) e entre redes virtuais (QoS provedor) no XNetMan.

Com a técnica de separação de planos, os planos de dados de cada rede virtual que antes estavam nos DomU passam para o Dom0 e encaminhamento de pacotes é completamente efetuado no Dom0 sem passar pelo DomU. Assim, a separação de planos modifica o esquema para provimento de QoS para o operador de redes virtuais e o provedor de infraestrutura, pois a separação de planos implica na inserção do módulo de provimento de QoS do operador dentro do Dom0, como mostrado na Figura 3(b). Assim, o XNetMan precisa dar suporte para que cada máquina virtual possa configurar o seu controle de tráfego no Dom0, garantindo que uma rede virtual não poderá interferir no controle de tráfego das demais redes virtuais.

5.3. Criação e Gerenciamento de Recursos para Redes Virtuais

O módulo de controle de acesso de redes virtuais avalia se a máquina física pode receber ou não uma nova rede virtual, dados os requisitos dessa rede virtual. Assim, esse módulo verifica informações como o tamanho do disco da máquina virtual, o número de interfaces, as vazões máximas por interface e o gasto de CPU e memória no Dom0 que a nova rede necessita. Para verificar se o sistema pode dar suporte à vazão de saída requisitada, são observadas as reservas a curto e longo prazo da nova rede e o total já reservado para curto e longo prazo para as redes virtuais que já estão hospedadas.

O gerenciamento dos recursos atribuídos a cada rede virtual após a admissão da rede é feito pelos módulos conversor de SLA, definição de rede e monitor de SLA. O conversor de SLA é responsável pela tradução dos SLAs contratados em parâmetros do XNetMan. Dessa forma, se uma rede contrata um uso variável dos recursos físicos ao longo do tempo, esse módulo deve atualizar as definições dessa rede virtual de forma que o controlador possa atender a essas mudanças sazonais de demanda. O módulo definição de redes armazena as definições atuais de cada rede virtual e as disponibiliza para o

controlador e para o módulo de QoS do provedor. Por fim, o módulo monitor de SLAs observa os padrões de punições gerados pelo controlador e envia relatórios periódicos ao operador de rede virtual sobre as quebras de SLA e as punições aplicadas.

6. Descrição do Protótipo e Resultados Obtidos

Foi desenvolvido um protótipo do XNetMan em C++ no Xen-4.0 configurado no modo roteador. O monitoramento do uso de CPU, que é estimado com base no volume de pacotes encaminhados, e do uso da banda foi feito utilizando-se a ferramenta Iptables. A mesma ferramenta é utilizada na aplicação das punições. Para fazer o controle dos parâmetros de QoS, utilizou-se a ferramenta *Traffic Control* (TC). O protótipo foi implementado em uma máquina física com processador Intel Core 2 Quad, 4 GB de memória RAM, 5 interfaces de rede gigabit *ethernet*, hospedando 3 máquinas virtuais com sistema operacional Debian 2.6.32-5. O Dom0 é configurado com quatro CPUs lógicas, enquanto que cada DomU possui uma CPU lógica. Com base nesse protótipo foram desenvolvidos testes utilizando-se três máquinas externas conectadas a máquina física que executa o Xen com o XNetMan. O tráfego é gerado através do módulo do kernel do Linux ‘pktgen’, sendo caracterizado por pacotes UDP com 1472 B de carga útil.

Primeiramente, analisou-se o funcionamento do XNetMan controlando três redes virtuais com configurações diferentes, mas demandas iguais dadas por $D_i = 300 \text{ Mb/s}$, $0 < i < 3$, e sem requisitos de QoS no provedor de infraestrutura. A Rede 1 possui uma reserva a curto prazo $R_{c1} = 50 \text{ Mb/s}$ e uma reserva a longo prazo $R_{l1} = 350 \text{ Mb/s}$, de forma que sua demanda está de acordo com os SLAs ($D_1 < R_{l1}$) e deve ser provida integralmente. A Rede 2 possui $R_{c2} = R_{l2} = 100 \text{ Mb/s}$, apresentando uma demanda superior ao contratado no SLA. Já a Rede 3 simula uma rede com grande volume de tráfego, mas sem nenhuma prioridade ou requisito de atraso, de forma que a $R_{c3} = 0$ e $R_{l3} = 250 \text{ Mb/s}$. Portanto, a Rede 3 também apresenta demanda superior ao seu SLA. Nenhuma das redes virtuais possui reserva exclusiva e, portanto, os recursos de CPU e memória foram divididos igualmente entre as três redes. Os tráfegos da Rede 1 e da Rede 2 são transmitidos de uma máquina externa para outra através de roteamento pelo Dom0, pois se assume que essas redes utilizam a separação de planos. Já o tráfego da Rede 3 vem do roteador virtual para a máquina externa, simulando um tráfego gerado ou encaminhado pelo roteador virtual. O enlace de saída é compartilhado pelas três redes e é o objeto dessa análise³. A duração total do teste é de 200 s.

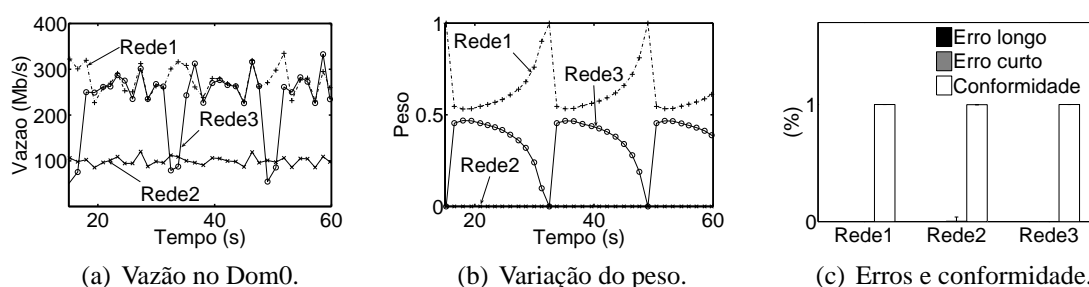


Figura 4. Operação do XNetMan com diferentes padrões de rede virtual, assumindo demanda de 300Mb/s para cada rede.

O comportamento da vazão (V_i) obtida com as três redes ao longo do tempo está apresentada na Figura 4(a). A Rede 2 possui apenas a reserva a curto prazo, o XNetMan

³Neste e nos demais experimentos, o XNetMan está configurado com o intervalo curto $I_c = 1 \text{ s}$ e o intervalo longo $I_l = 15 \text{ s}$, os quais foram obtidos através de testes com o controlador do XNetMan para se obter o melhor desempenho.

limitou então o seu uso de forma constante em 100 Mb/s . Já a Rede 1 teve a sua demanda atendida em todos os momentos, enquanto que a Rede 3 teve sua demanda bloqueada ao atingir o valor de total de dados permitido pela reserva longa em cada intervalo longo. O uso da reserva longa pelas redes é regido de acordo com o peso adaptativo da rede, como descrito na Seção 5.1. A Figura 4(b) mostra a evolução do valor do peso. O peso da Rede 2 é sempre zero, porque essa rede não tem direito a usar nada além da reserva a curto prazo. O peso da Rede 1 aumenta enquanto que o peso da Rede 3 diminui, porque a Rede 3 possui uma demanda média superior a sua reserva a longo prazo e a Rede 1 possui uma demanda média inferior a sua reserva a longo prazo. De fato, o peso se adapta proporcionalmente entre as redes de acordo com a fatia da reserva longa que cada rede ainda pode gastar. Com isso, o peso das redes que possuem uma demanda inferior à sua reserva a longo prazo tende a aumentar enquanto que o peso das redes que possuem uma demanda superior à sua reserva a longo prazo tende a diminuir. A Figura 4(c) mostra o erro a curto prazo, o erro a longo prazo e a conformidade, os quais são parâmetros definidos para verificar o atendimento dos SLAs para cada uma das redes virtuais. O erro a curto prazo da rede i , E_{c_i} , é calculado a cada intervalo curto, I_c , e demonstra o quanto a reserva a curto prazo não foi atendida. Analogamente, o erro a longo prazo da rede i , E_{l_i} , é calculado no final de cada intervalo longo, I_l , e demonstra o quanto a reserva longa não foi atendida. Assim,

$$E_{c_i} = \max(0, 1 - V_i/\min(R_{c_i}, D_i)) \quad e \quad E_{l_i} = \max(0, 1 - V_i/\min(R_{l_i}, D_i)). \quad (1)$$

A conformidade da rede i , C_i , é calculada também ao final de I_l e demonstra um valor médio de atendimento ao SLA. Assim,

$$C_i = 1 - \frac{\text{media}(E_{c_i})}{2} - \frac{E_{l_i}}{2}. \quad (2)$$

De acordo com a Figura 4(c), todas as redes tiveram sua conformidade atendida ao longo de todos os intervalos longos do teste, pois o XNetMan oferece um controle eficiente dos recursos compartilhados na máquina física. A Rede 2 apresenta um pequeno erro com relação ao erro curto apenas devido ao primeiro intervalo longo, no qual o XNetMan ainda estava se ajustando a demanda.

O segundo teste compara o XNetMan com outras ferramentas da literatura. Para tanto, montou-se um cenário no qual existem duas redes virtuais, uma utilizando separação de planos (Rede 1) e outra utilizando roteamento pela máquina virtual (Rede 2). As redes possuem $R_{c_1} = R_{c_2} = 100 \text{ M}$, $R_{l_1} = 600 \text{ M}$ e $R_{l_2} = 200 \text{ M}$. Ambas as redes possuem uma demanda superior aos parâmetros de seus SLAs, $D_1 = D_2 = 1 \text{ Gb/s}$, em tráfegos entre duas máquinas externas. Uma vez que a máquina externa receptora é comum às duas redes, então existe o compartilhamento do enlace de saída.

O XNetMan (XMan) foi comparado ao XNetMon (XMon), pois esse também realiza o controle de recursos entre redes virtuais, embora o XNetMon não possua a opção pela especificação de uma reserva longa ou o controle adaptativo dos recursos utilizados. Além disso, comparou-se o XNetMan com três perfis diferentes de controle de tráfego utilizando-se a ferramenta TC com o *Hierarchical Token Bucket* (HTB), tentando simular comportamentos próximos ao do XNetMan. O perfil TC1 possui reserva mínima $R_{m_i} = R_{l_i}$ e permite que ambos os tráfegos cheguem a 1 Gb/s caso haja demanda e disponibilidade de recursos. O perfil TC2 utiliza $R_{m_i} = R_{c_i}$, mas limita o uso do enlace à taxa média da reserva a longo prazo da rede, R_{l_i} , sendo o mais próximo do comportamento do XNetMan. O perfil TC3 utiliza $R_{m_i} = R_{c_i}$ e permite que os recursos sejam utilizados até o limite do enlace caso haja demanda e disponibilidade, sendo o mais próximo do

comportamento do XNetMon. O TC, embora seja uma ferramenta precisa para o controle de tráfego, também não possui controle adaptativo, além de não fazer o controle da divisão dos recursos compartilhados de memória e CPU como o fazem o XNetMan e o XNetMon. Dessa forma, para tornar a comparação do controle de recursos da rede mais justa, tanto no XNetMan quanto no XNetMon não foram feitas restrições quanto ao uso de CPU e memória para nenhuma das redes. Por fim, também se analisou o resultado quando nenhum controle era aplicado às redes virtuais (*S/*).

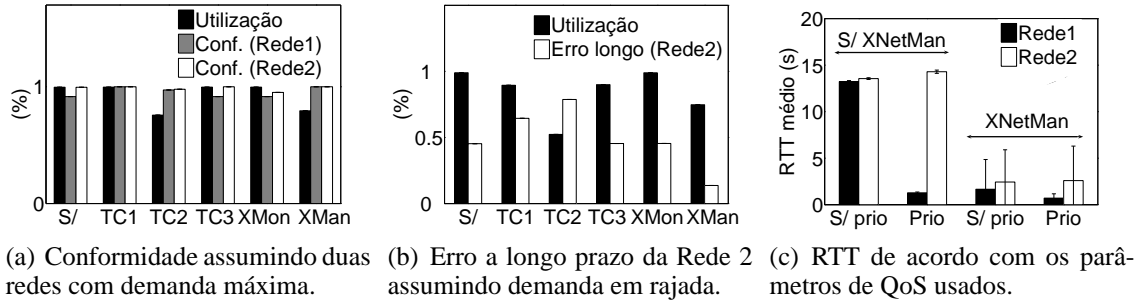


Figura 5. Comparação do XNetMan com outras ferramentas de controle.

A Figura 5(a) apresenta os resultados desse teste, mostrando a conformidade e a utilização com relação ao enlace de saída⁴. Apenas o XNetMan e os perfis TC1 e TC2 conseguiram prover conformidade máxima a ambas as redes. Com os demais perfis, a Rede 2 acabou sendo privilegiada devido à baixa capacidade de diferenciar o uso dos recursos além da reserva mínima de cada rede. Os perfis TC1 e TC2 apresentaram bons resultados de conformidade, porque são adaptados a diferenciar o tráfego com base na taxa de uso a longo prazo. Contudo, apenas o XNetMan e o TC2 apresentaram baixa utilização do enlace, o que é uma característica importante para as redes virtualizadas.

O terceiro teste realizado utiliza os mesmos perfis com as mesmas configurações para as redes, mudando apenas a demanda. Nesse caso, $D_1 = 1 \text{ Gb/s}$ durante todo o teste, enquanto que $D_2 = 0$ durante os primeiros dois terços de I_l e $D_2 = 1 \text{ Gb/s}$ durante 3 s no último terço de I_l . Este cenário simula o comportamento de uma rede com tráfego em rajadas, no qual a rajada equivale ao total de recursos contratados a longo prazo para aquela rede⁵. Os resultados na Figura 5(b) mostram que o erro a longo prazo do XNetMan foi o menor de todos, apresentando um ganho de mais de cinco vezes quando comparado ao perfil TC2, que apresentou um ganho semelhante ao do XNetMan no teste anterior. Além disso, o XNetMan apresentou a segunda menor utilização, mostrando que o controlador do XNetMan é positivo tanto para o operador da rede, que deseja o cumprimento do contrato, quanto para o provedor de infraestrutura, que deseja um baixo uso dos recursos físicos para que mais redes possam ser alocadas.

Por fim, foi avaliado o módulo de QoS do XNetMan. A Figura 5(c) mostra o impacto da utilização das premissas de QoS pelo provedor de redes virtuais. Nesse cenário, composto por duas redes virtuais, assume-se que a Rede 1 deseja um atraso pequeno para o seu tráfego. Assume-se também que a Rede 1 tem como parâmetros para o enlace de saída $R_{c_1} = 50 \text{ Mb/s}$ e $R_{l_1} = 400 \text{ Mb/s}$, enquanto que a Rede 2 possui $R_{c_2} = 100 \text{ Mb/s}$ e $R_{l_2} = 600 \text{ Mb/s}$. A CPU e a memória foram divididas igualmente entre as redes. A Rede 1 possui uma demanda máxima ($D_1 = 50 \text{ Mb/s}$) e a Rede 2 possui $D_2 = 1 \text{ Gb/s}$.

⁴A utilização é calculada como a relação entre o volume de dados utilizado por cada perfil e o volume de dados máximo transmitido no enlace de saída.

⁵Nesse cenário, o erro a longo prazo da Rede 2 não poderia ser nulo porque o atendimento total da demanda necessitaria do uso global da banda, mas a Rede 1 possui uma reserva a curto prazo.

Foi escolhido um valor alto para D_2 , porque um volume alto de tráfego dificulta o provimento de QoS para a Rede 1. O tráfego da Rede 2 passa entre duas máquinas externas com roteamento pelo Dom0, enquanto que o tráfego da Rede 1, que deve ser priorizado, é roteado pela máquina virtual também entre duas máquinas externas. O enlace de saída é compartilhado pelos tráfegos das Redes 1 e 2. A Figura 5(c) mostra os resultados para o *Round Trip Time* (RTT) com e sem o uso do XNetMan, comparando um cenário aonde os dois tráfegos possuem a mesma prioridade (S/ prio) a outro cenário onde a os pacotes da Rede 1 tem prioridade máxima (Prio). O uso do módulo de QoS do provedor mesmo sem a utilização do controle do XNetMan garantiu uma redução de mais de 10 vezes no atraso do tráfego privilegiado. Ao se utilizar o XNetMan com módulo de QoS e os demais módulos, o controlador garante que a Rede 2 não exceda o uso de rede ou CPU, reduzindo o volume de dados processados e, com isso, reduzindo ainda mais o atraso de transmissão. O ganho com relação à redução do atraso ao se utilizar o XNetMan com todos os módulos é de mais de 18 vezes quando comparado com o cenário sem XNetMan e sem prioridade, de mais de 1.8 vezes quando comparado ao cenário sem XNetMan com prioridade e de mais de 2.3 vezes quando comparado ao uso do XNetMan sem prioridade. Com isso, fica clara a necessidade de se prover QoS pelo provedor de infraestrutura e que o XNetMan é uma ferramenta eficiente para o provimento de QoS.

7. Conclusões

Os sistemas para virtualização de máquinas ainda apresentam vulnerabilidades graves quando utilizados para virtualização de rede, como é o caso do Xen, que não garante o isolamento quando existem operações de E/S frequentes. Este artigo propôs o XNetMan que é um sistema para controle e gerenciamento de redes virtuais no Xen. Com o XNetMan, é possível garantir um forte isolamento entre as redes e a conformidade com os SLAs contratados entre o provedor de infraestrutura e o operador de rede virtual. As principais contribuições do XNetMan são o controlador, que garante a divisão correta dos recursos entre as redes virtuais devido ao uso de um controle adaptativo e os módulos para prover QoS, que possibilitam a inserção de parâmetros de QoS já no nível do provedor de infraestrutura. Essas características tornam o XNetMan um sistema diferenciado com ganhos significativos em relação a outros tipos de controle de recursos existentes na literatura.

As características usadas para modelar os SLAs no XNetMan, que incluem a reserva a curto e a longo prazo, a reserva exclusiva e a definição de parâmetros de QoS, garantem um conjunto extenso de configurações de rede. Além disso, os demais módulos do XNetMan, como o módulo que controla o acesso de novas redes virtuais, fazem com que esse seja um sistema com alta gerenciabilidade e de grande utilidade para as redes virtualizadas. De fato, o XNetMan soluciona os problemas existentes na plataforma Xen para virtualização de redes disponibilizando um controle preciso dos recursos compartilhados e um gerenciamento simples dos recursos das redes virtuais.

Referências

- Achemlal, M. e et al. (2010). Virtualisation approach: Concept. Relatório Técnico D-3.1.1, 4WARD - Architecture and Design for the Future Internet.
- Bhatia, S., Motiwala, M., Muhlbauer, W., Valancius, V., Bavier, A., Feamster, N., Peterson, L. e Rexford, J. (2008). Hosting virtual networks on commodity hardware. Relatório Técnico GT-CS-07-10, Princeton University, Georgia Tech, and T-Labs/TU Berlin.

- Carapinha, J. e Jiménez, J. (2009). Network virtualization - a view from the bottom. Em *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, páginas 73–80, New York, NY, USA. ACM.
- Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Huici, F. e Mathy, L. (2008). Fairness issues in software virtual routers. Em *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, PRESTO '10, páginas 33–38.
- Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Mathy, L. e Schooley, T. (2007). Evaluating Xen for router virtualization. Em *International Conference on Computer Communications and Networks*, ICCCN'07, páginas 1256–1261.
- Feamster, N., Gao, L. e Rexford, J. (2007). How to lease the Internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64.
- Fernandes, N. C. e Duarte, O. C. M. B. (2010). XNetMon: Uma arquitetura com segurança para redes virtuais. Em *Anais do X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, SBSEG '10, páginas 339–352, Fortaleza, CE, Brazil.
- Fernandes, N. C., Moreira, M. D. D., Moraes, I. M., Ferraz, L. H. G., Couto, R. S., Carvalho, H. E. T., Campista, M. E. M., Costa, L. H. M. K. e Duarte, O. C. M. B. (2010). Virtual networks: Isolation, performance, and trends. *Annals of Telecommunication*, páginas 1–17.
- GENI (2009). Geni spiral 1 annual report 2009. Relatório técnico, National Science Foundation.
- Houidi, I., Louati, W., Zeghlache, D., Papadimitriou, P. e Mathy, L. (2010). Adaptive virtual network provisioning. Em *Proceedings of the 2nd ACM workshop on Virtualized Infrastructure Systems and Architectures*, VISA '10, páginas 41–48, New York, NY, USA. ACM.
- McIlory, R. e Sventek, J. (2006). Resource virtualisation of network routers. Em *Workshop on High Performance Switching and Routing*, páginas 1–6, New York, NY, USA. IEEE.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., S. e Turner, J. (2008). OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.
- Moreira, M. D. D., Fernandes, N. C., Costa, L. H. M. K. e Duarte, O. C. M. B. (2009). Internet do futuro: Um novo horizonte. Em *Minicursos do Simpósio Brasileiro de Redes de Computadores*, SBRC '09, páginas 1–59, Rio de Janeiro, Brazil.
- Schaffrath, G., Werle, C., Papadimitriou, P., Feldmann, A., Bless, R., Greenhalgh, A., Wundsam, A., Kind, M., Maennel, O. e Mathy, L. (2009). Network virtualization architecture: proposal and initial prototype. Em *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, páginas 63–72, New York, NY, USA. ACM.
- Sherwood, R., Chan, M., Covington, A., Gibb, G., Flajslik, M., Handigol, N., Huang, T.-Y., Kazemian, P., Kobayashi, M., Naous, J., Seetharaman, S., Underhill, D., Yabe, T., Yap, K.-K., Yiakoumis, Y., Zeng, H., Appenzeller, G., Johari, R., McKeown, N. e Parulkar, G. (2010). Carving research slices out of your production networks with OpenFlow. *ACM SIGCOMM Computer Communication Review*, 40(1):129–130.
- VINI (Acessado em agosto de 2010). *VINI - A Virtual Network Infrastructure*. <http://www.vini-veritas.net/>.